

A generalist approach to Array Signal Processing:

Beam forming explained

By Tobias Dahl

SINTEF Digital and University of Oslo

1



Amazon Echo dot

Google Home

Apple Homepod

Sonos one

Soon, we will all be owners of an array at home. There has never been a better time to learn about arrays!

2

Microphone Array Processing for Distant Speech Recognition: Towards Real-World Deployment

[2012]

When the signals from the individual sensors of a microphone array with a known geometry are suitably combined, the array functions as a spatial filter capable of suppressing noise, reverberation, and competing speech. Such *beamforming* techniques have received a great deal of attention within the acoustic array processing community in the recent past [2], [3], [4], [5], [6], [7], [8].

Despite this effort, however, such techniques have often been ignored within the mainstream community working on distant speech recognition. As pointed out in [7], [8], [9], this could be due to the fact that the disparate research communities for acoustic array processing and automatic speech recognition have failed to adopt each other's best practices. For instance, the array processing community ignores speaker adaptation techniques, which can compensate for mismatches between acoustic conditions during training and testing. Moreover, this community has largely preferred to work on controlled, synthetic recordings, obtained by convolving noise- and reverberation-free speech with measured, static room impulse responses, with subsequent artificial addition of noise, as in the recent PASCAL CHiME Speech Separation

The primary goal of this contribution is to provide a tutorial in the application of acoustic array processing to distant speech recognition that is intelligible to anyone with a general signal processing background, while still maintaining the interest of experts in the field. Our secondary goal is to bridge the gaps between the current acoustic array processing and speech recognition communities. A third and overarching goal is



3

Microphone Array Processing for Distant Speech Recognition: Towards Real-World Deployment

[2012]

When the signals from the individual sensors of a microphone array with a known geometry are suitably combined, the array functions as a spatial filter capable of suppressing noise, reverberation, and competing speech. Such *beamforming* techniques have received a great deal of attention within the acoustic array processing community in the recent past [2], [3], [4], [5], [6], [7], [8].

Despite this effort, however, such techniques have often been ignored within the mainstream community working on distant speech recognition. As pointed out in [7], [8], [9], this could be due to the fact that the disparate research communities for acoustic array processing and automatic speech recognition have failed to adopt each other's best practices. For instance, the array processing community ignores speaker adaptation techniques, which can compensate for mismatches between acoustic conditions during training and testing. Moreover, this community has largely preferred to work on controlled, synthetic recordings, obtained by convolving noise- and reverberation-free speech with measured, static room impulse responses, with subsequent artificial addition of noise, as in the recent PASCAL CHiME Speech Separation

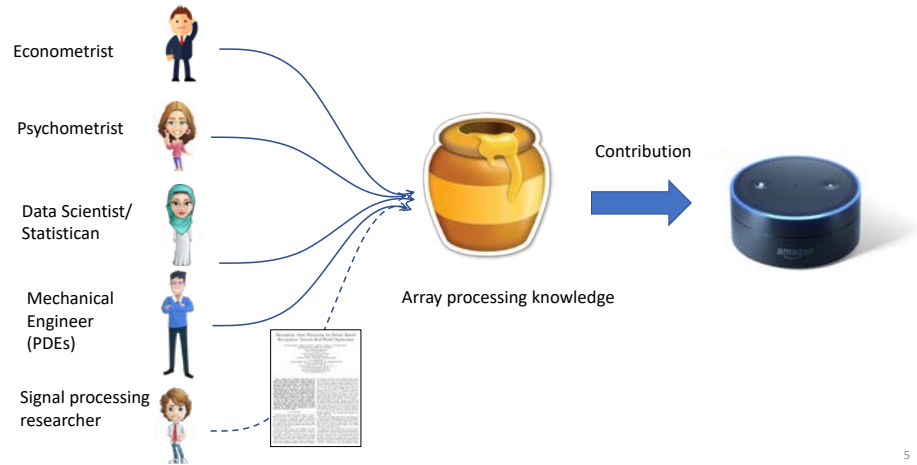
The primary goal of this contribution is to provide a tutorial in the application of acoustic array processing to distant speech recognition that is intelligible to anyone with a general signal processing background, while still maintaining the interest of experts in the field. Our secondary goal is to bridge the gaps between the current acoustic array processing and speech recognition communities. A third and overarching goal is

1. What if the gap between the fields is related to scientific jargon or language?

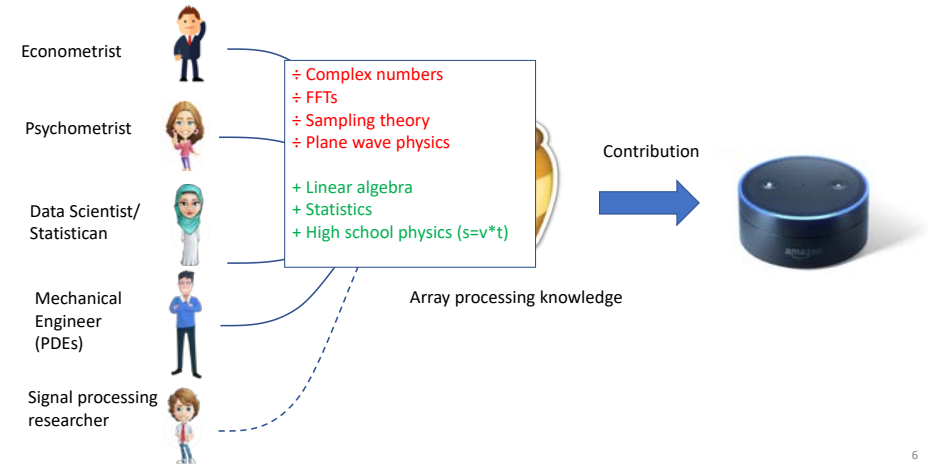
2. What if there are also many other researchers (not just signal processing folks) who could get access to array concepts and contribute to the field?

US Companies like Microsoft and Amazon now recruit their "beamformers" from Germany and France!

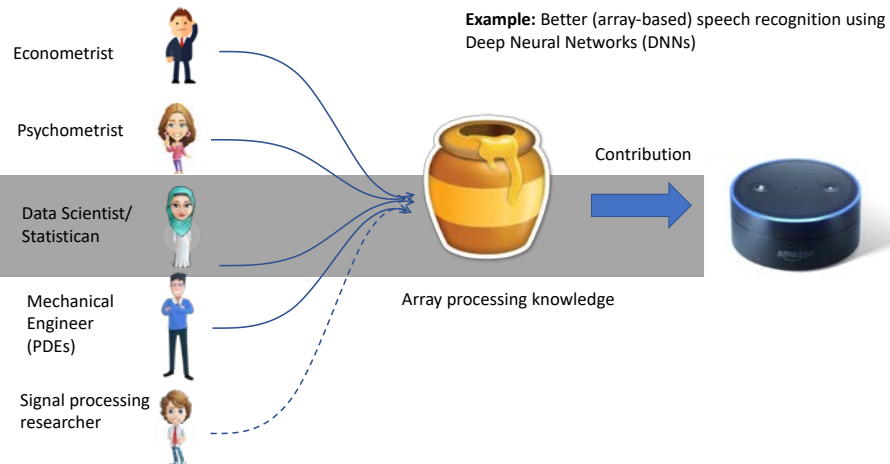
Ambition



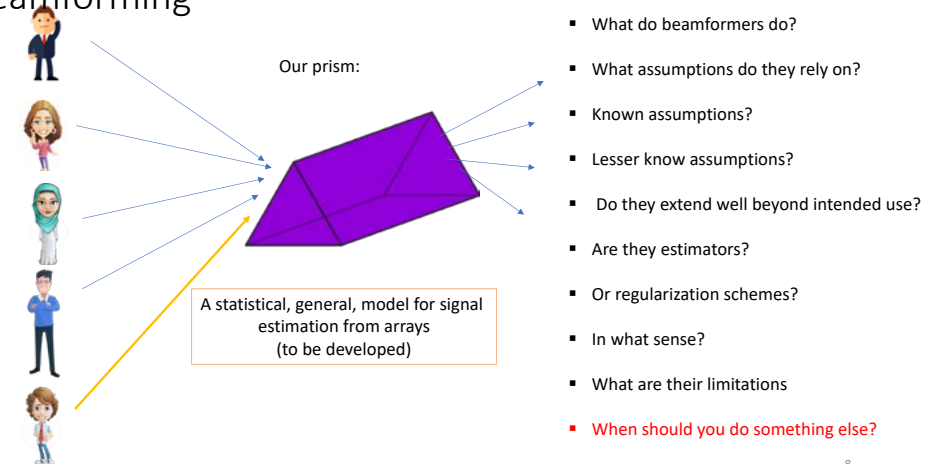
Ambition



Ambition



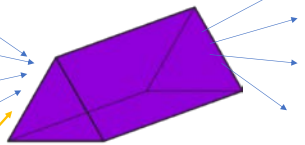
Using statistics as a prism for observing beamforming



Using statistics as a **prism** for observing beamforming



Our prism:



A statistical, general, model for signal estimation from arrays (to be developed)

- What do beamformers do?
- What assumptions do they rely on?
- Known assumptions?
- Lesser known assumptions?
- Do they extend well beyond intended use?
- Are they estimators?
- Or regularization schemes?
- In what sense?
- What are their limitations

Approaching arrays from **square one** we get to learn more about **beam forming** than **beam forming engineers** do!

Overview

Details

1. Array signal estimation (without beamforming)

- No sampling theory, complex numbers, plane waves, stationarity assumptions,...
- Based on matrix inversion

2. What goes on in a 'beam former'?

- Starting with single frequency (cosine/sine)
- What happens with "stationarity"
- The transient
- Fixed beams, minimum variance and other types of **regularization**

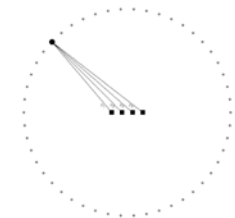
3. A statistical approach to signal recovery

- Deriving a statistical model
- Subspace "shaving" through filtering
- Sparsity and information

10

Array signal estimation (without beam forming)

A simplistic model using high school physics

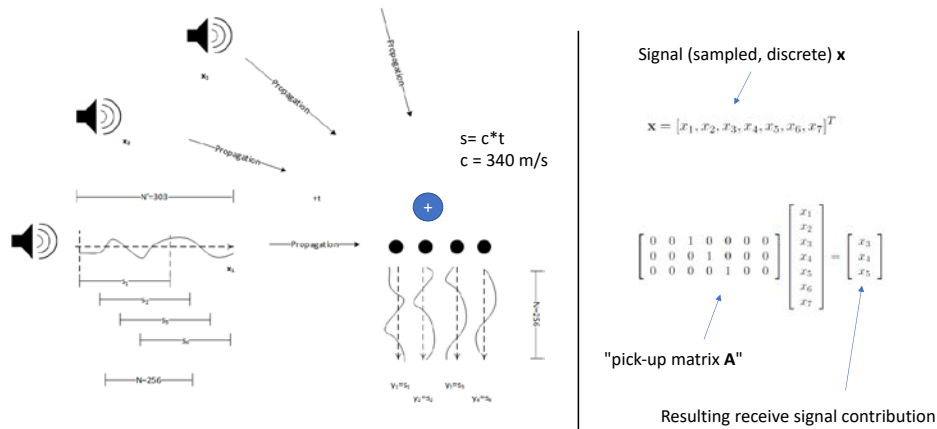


All sounds emerging from every direction can be modelled through an infinitesimal set of tiny speakers on a 3D ball playing back sounds

(the signals can add up to include "anything" – multiple sources, echoes, background noise..)

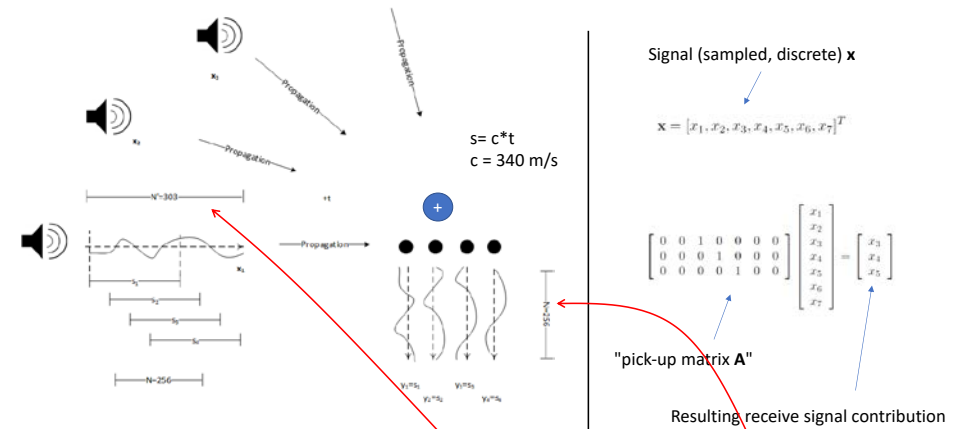
11

12



$$\mathbf{y}_i = \sum_{j=1}^M \mathbf{A}_{ij} \mathbf{x}_j \quad j=1, \dots, M \text{ (speaker index), } i=1, \dots, Q \text{ (microphone index)}$$

13



Observe that the transmit signals $\mathbf{x}_j$ are **longer than** the receive signals $\mathbf{y}_i$ (and each $\mathbf{x}_j$ may have different length)

j=1

14

Block matrix model

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_Q \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \dots & \mathbf{A}_{1M} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & & \mathbf{A}_{2M} \\ \vdots & & & \vdots \\ \mathbf{A}_{Q1} & & & \mathbf{A}_{QM} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_M \end{bmatrix}}_{\mathbf{x}} = \mathbf{A} \mathbf{x}$$

Signal (sampled, discrete) $\mathbf{x}$

$$\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]^T$$

Resulting receive signal contribution

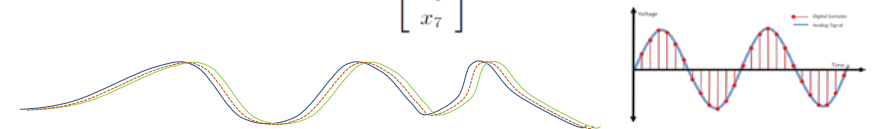
$$\mathbf{y}_i = \sum_{j=1}^M \mathbf{A}_{ij} \mathbf{x}_j \quad j=1, \dots, M \text{ (speaker index), } i=1, \dots, Q \text{ (microphone index)}$$

15

Fractional delays via interpolation

"Fractional pick-up matrix"

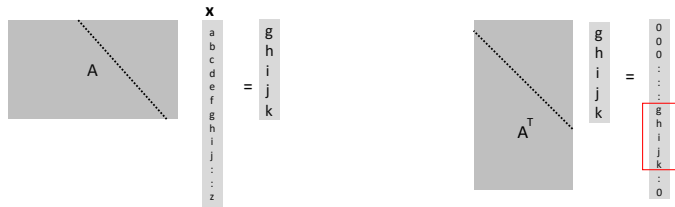
$$\begin{bmatrix} 0 & 0 & 0.6 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.6 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.6 & 0.4 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 0.6x_3 + 0.4x_4 \\ 0.6x_4 + 0.4x_5 \\ 0.6x_5 + 0.4x_6 \end{bmatrix}$$



(Linear) interpolation works – if the signals are "sufficiently smooth". We can assume they always are – by designing our system to sample signals really fast

16

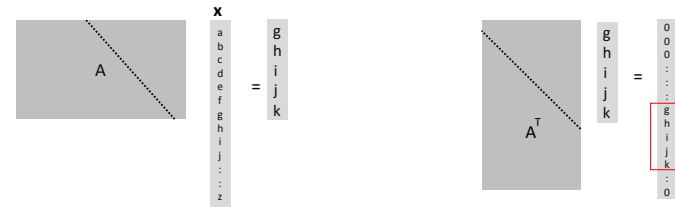
Other observations (for maths later)



A picks out a segment of its vector $\mathbf{x}$

$\mathbf{A}^T$ moves this segment back to its original position – and pads with 0's

17



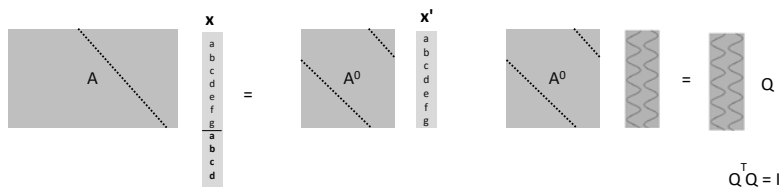
A picks out a segment of its vector $\mathbf{x}$

$\mathbf{A}^T$ moves this segment back to its original position – and pads with 0's

Note that $\mathbf{A}^T \mathbf{A}$ is not identity, i.e. we **don't** have $\mathbf{A}^T \mathbf{A} = \mathbf{I}$

18

Circulant matrices and self-repeating signals

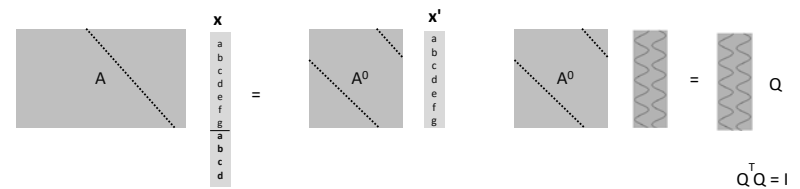


$\mathbf{A}^{0T} \mathbf{A}^0 = \mathbf{I}$ (to be used for self-repeating signals)

for sines and cosines (of same period),
we can replace **big left** matrix by **small** orthogonal one
on the **right** side

19

Circulant matrices and self-repeating signals



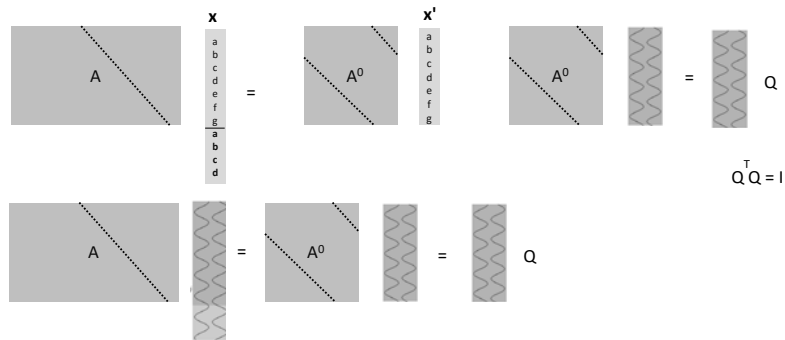
$\mathbf{A}^{0T} \mathbf{A}^0 = \mathbf{I}$ (to be used for self-repeating signals)

for sines and cosines (of same period),
we can replace **big left** matrix by **small** orthogonal one
on the **right** side

A world of self-repeating signals sounds wonderful? ☺

20

Circulant matrices and self-repeating signals



Bases of sines and cosine pairs share the same feature – replacing a big left matrix by a small right matrix..

21

A real speech example

4 sources x 4 microphones



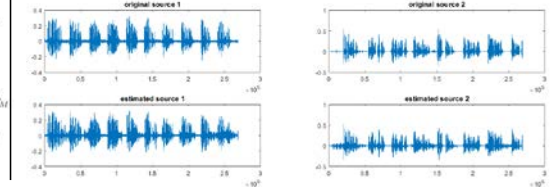
Linear equations, block matrix system



Algorithm

1. for each microphone, pull N next samples as vectors $\mathbf{y}_1, \dots, \mathbf{y}_Q$ from the microphone stack;
2. Arrange the vectors as a super-vector $\mathbf{y} = [\mathbf{y}_1^T \mathbf{y}_2^T \dots \mathbf{y}_Q^T]^T$.
3. Compute $\hat{\mathbf{x}} = \mathbf{A}^+ \mathbf{y}$
4. Subdivide the vector $\hat{\mathbf{x}}$ into M segments $\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_M$ of lengths $P_1, P_2, \dots, P_M$ respectively onto the source stack.
5. Push the N first samples of vectors $\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_M$ on the source estimate stack
6. Repeat from 1 until the microphone stack is empty.

Results



Summary on direct inversion of model

- Direct inversion of mixed signal segments works!
 - Perhaps no need to learn beam forming after all? ☺
- The example was very simplistic, straightforward. Used only 4 sources.
 - Not clear what will happen when number of *virtual sources* increases
 - Probably very ill-posed! Some though variance-bias choices ahead..
 - In real life, you have no control over number of or exact position of sources
 - So you just have to include "loads of virtual sources" to be sure to cover all options
- We didn't utilize knowledge of signal smoothness
 - => can be shown to make it very sensitive to noise

What we did didn't help us understand beamforming – yet!

23

Towards beam forming

$$\mathbf{y} = \mathbf{A} \mathbf{x} + \mathbf{n} \quad E[\mathbf{n} \mathbf{n}^T] = \sigma^2 \mathbf{I}$$

$$\mathbf{A}_\alpha^+ = [\mathbf{A}^T \mathbf{A} + \alpha \mathbf{I}]^{-1} \mathbf{A}^T$$

$$\hat{\mathbf{x}} = \mathbf{A}_\alpha^+ \mathbf{y}$$

for some choice of α which depends on the noise variance σ^2 .

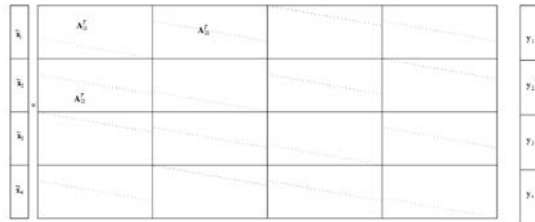
24

$$\hat{\mathbf{x}} = [\mathbf{A}^T \mathbf{A} + \alpha \mathbf{I}]^{-1} \mathbf{A}^T \mathbf{y} = \frac{1}{\alpha} \left[\frac{1}{\alpha} \mathbf{A}^T \mathbf{A} + \mathbf{I} \right]^{-1} \mathbf{A}^T \mathbf{y}$$

which means that for sufficiently large values of α , we get

$$\hat{\mathbf{x}} \approx \frac{1}{\alpha} \mathbf{A}^T \mathbf{y}$$

a new "estimator" $\tilde{\mathbf{x}} = \mathbf{A}^T \mathbf{y}$



Interpretation:

The (transposed) block matrices *pads the $\mathbf{y}_i$ vectors with zeros, shifts them up/down and adds them up!*

The shifts are the opposites of those introduced as the original sources were propagated to out the various array microphones

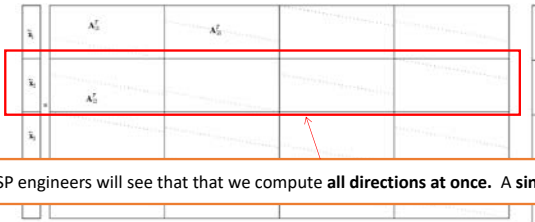
25

$$\hat{\mathbf{x}} = [\mathbf{A}^T \mathbf{A} + \alpha \mathbf{I}]^{-1} \mathbf{A}^T \mathbf{y} = \frac{1}{\alpha} \left[\frac{1}{\alpha} \mathbf{A}^T \mathbf{A} + \mathbf{I} \right]^{-1} \mathbf{A}^T \mathbf{y}$$

which means that for sufficiently large values of α , we get

$$\hat{\mathbf{x}} \approx \frac{1}{\alpha} \mathbf{A}^T \mathbf{y}$$

a new "estimator" $\tilde{\mathbf{x}} = \mathbf{A}^T \mathbf{y}$



Interpretation:

The (transposed) block matrices *pads the $\mathbf{y}_i$ vectors with zeros, shifts them up/down and adds them up!*

The shifts are the opposites of those introduced as the

DSP engineers will see that that we compute **all directions at once**. A **single direction** corresponds with **single block row**

array microphones

26

Delay-and-sum principle

The general principle at play here, is that if you have N copies of a signal $x(t)$, each with a some time-delay δ_i , so that one samples $y(t) = x(t - \delta_i) + z_i(t)$ on a set of sensors $i = 1, \dots, N$, where z_i is some other, residual signal such as noise or interfering signals, then a good estimator for $x(t)$ is

$$\hat{x}(t) = \frac{1}{N} \sum_{i=1}^N y(t + \delta_i) = \frac{1}{N} \sum_{i=1}^N x(t) + \frac{1}{N} \sum_{i=1}^N z_i(t + \delta_i) \quad (10)$$

as long as the second term vanishes as N increases, which is will be the case unless there are some very specific connections between the $z_i(t)$ functions.

This is what goes on in classic beam forming: the microphone signals are **artificially delayed and summed up** ("back into focus") so that the source of interest has coherent signals that are magnified relative to noise and interference when averaged (or summed up)

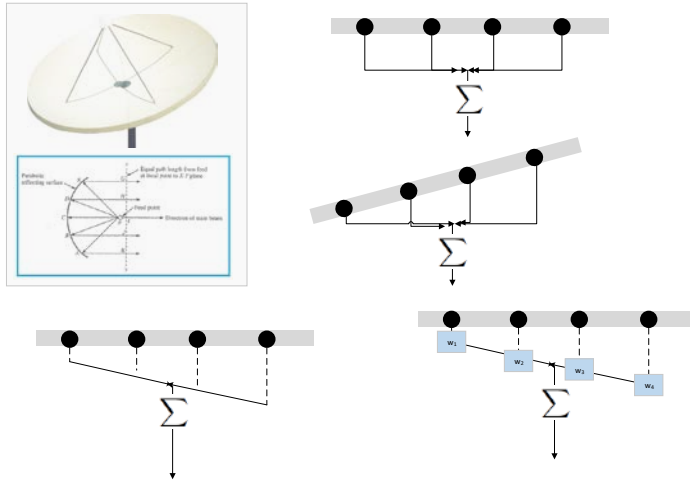
27

Summary on delay-and-sum

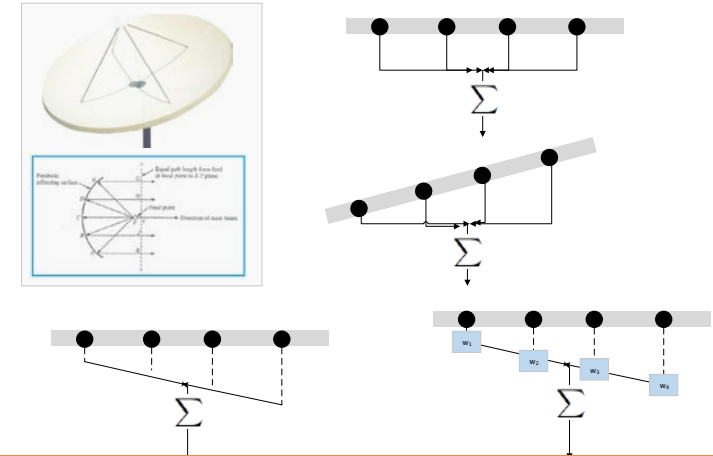
- If there is a lot of sensor noise, the resulting MMSE estimator (Least Squares with Noise) is close to delay-and-sum operator
- In some special cases ("critical sampling"), the signal noise $\mathbf{n}$ could also represent *interferents from the room*, i.e. not just sensor noise
 - Probably a more realistic assumption than "infinite sensor noise"

The real reasons for using delay-and-sum (and its variants) are tied to the history of **analog signal processing** and will be covered in greater detail in the next section

28



29



These delay-weight-and-sum operations are really easy to implement in analogue electronics!

30

What goes on inside a beam former?

(we know that it is not statistical estimation, really ;)

31

Workhorse 1: Least Squares & Bayes Estimation

$$y = Ax \Rightarrow \begin{aligned} \hat{x} &= (A^T A)^{-1} A^T y \\ \hat{x} &= V_0 S_0^{-1} U_0^T y, \quad A = U_0 S_0 V_0^T \end{aligned}$$

Least squares, no noise or errors

$$y = Ax + n \quad E(nn^T) = \sigma^2 I \Rightarrow \hat{x} = (A^T A + \sigma^2 I)^{-1} A^T y$$

Additive noise with known covariance (white)

$$y = Ax + n \quad \begin{aligned} E(nn^T) &= R \\ E(xx^T) &= C \end{aligned} \Rightarrow \begin{aligned} \hat{x} &= (A^T C A + R)^{-1} A^T C y \\ E(\hat{x} \hat{x}^T) &= C - C A (A^T C A + R)^{-1} A^T C \end{aligned}$$

$$\begin{aligned} E(\hat{x} - x)(\hat{x} - x)^T &= E \\ E(\hat{x}_i - x_i) &= E_{ii} \end{aligned}$$

Additive noise with known covariance AND known signal covariance

Workhorse 1: Least Squares & Bayes Estimation

$$y = Ax \Rightarrow \begin{aligned} \hat{x} &= (A^T A)^{-1} A^T y \\ \hat{x} &= V_0 S_0^{-1} U_0^T y, \quad A = U_0 S_0 V_0^T \end{aligned}$$

Least squares, no noise or errors

$$y = Ax + n \quad E(nn^T) = \sigma^2 I \Rightarrow \hat{x} = (A^T A + \sigma^2 I)^{-1} A^T y$$

Additive noise with known covariance (white)

$$y = Ax + n \quad \begin{aligned} E(nn^T) &= R \\ E(xx^T) &= C \end{aligned} \Rightarrow \begin{aligned} \hat{x} &= (A^T C A + R)^{-1} A^T C y \\ E(\hat{x}\hat{x}^T) &= C - C A (A^T C A + R)^{-1} A^T C \end{aligned}$$

$$E(\hat{x} - x)(\hat{x} - x)^T = E \dots$$

$$E(\hat{x}_i - x_i) = E_{ii}$$

Nice ☺ gives you an error estimate

Additive noise with known covariance AND known signal covariance

Workhorse 1: Least Squares & Bayes Estimation

$$y = Ax \Rightarrow \begin{aligned} \hat{x} &= (A^T A)^{-1} A^T y \\ \hat{x} &= V_0 S_0^{-1} U_0^T y, \quad A = U_0 S_0 V_0^T \end{aligned}$$

$$y = Ax + n \quad E(nn^T) = \sigma^2 I \Rightarrow \hat{x} = (A^T A + \sigma^2 I)^{-1} A^T y$$

$$y = Ax + n \quad \begin{aligned} E(nn^T) &= R \\ E(xx^T) &= C \end{aligned} \Rightarrow \begin{aligned} \hat{x} &= (A^T C A + R)^{-1} A^T C y \\ E(\hat{x}\hat{x}^T) &= C - C A (A^T C A + R)^{-1} A^T C \end{aligned}$$

The Bayes Estimator is easy to derive – and **very powerful** both for interpretation and practical estimation

34

Workhorse 2: Filters as matrix operations

$$y(t) = [x(t) * f(t)](t) = \sum_{k=0}^{K-1} x(t-k) f(k)$$

$$y = [y(0), y(1), \dots, y(N-1)]^T$$

$$x = [x(-K+1), x(-K+2), \dots, x(0), \dots, x(N)]^T$$

$$f = [f(0), f(1), \dots, f(K-1)]^T$$

$$y = Fx = \begin{pmatrix} f(K-1) & f(K-2) & \dots & f(0) & \dots & 0 \\ 0 & f(K-1) & f(K-2) & \dots & f(0) & \dots \\ \vdots & & & & & \vdots \end{pmatrix} x$$

35

Workhorse 2: Filters as matrix operations

$$y(t) = [x(t) * f(t)](t) = \sum_{k=0}^{K-1} x(t-k) f(k)$$

$$y = [y(0), y(1), \dots, y(N-1)]^T$$

$$x = [x(-K+1), x(-K+2), \dots, x(0), \dots, x(N)]^T$$

$$f = [f(0), f(1), \dots, f(K-1)]^T$$

$$y = Fx = \begin{pmatrix} f(K-1) & f(K-2) & \dots & f(0) & \dots & 0 \\ 0 & f(K-1) & f(K-2) & \dots & f(0) & \dots \\ \vdots & & & & & \vdots \end{pmatrix} x$$

Gives a the statistical distribution/model of a filtered "signal chunk"!

$$E(xx^T) = \gamma^2 I \Rightarrow E(yy^T) = E(Fxx^T F^T) = F E(xx^T) F^T = \gamma^2 F F^T$$

36

DSP people love cosines and sines!



Why? Because if a signal is a sine, a cosine or a delayed sine/cosine, the effect of a **signal time delay** can be expressed as a linear combination of a sine and cosine with the same frequency:

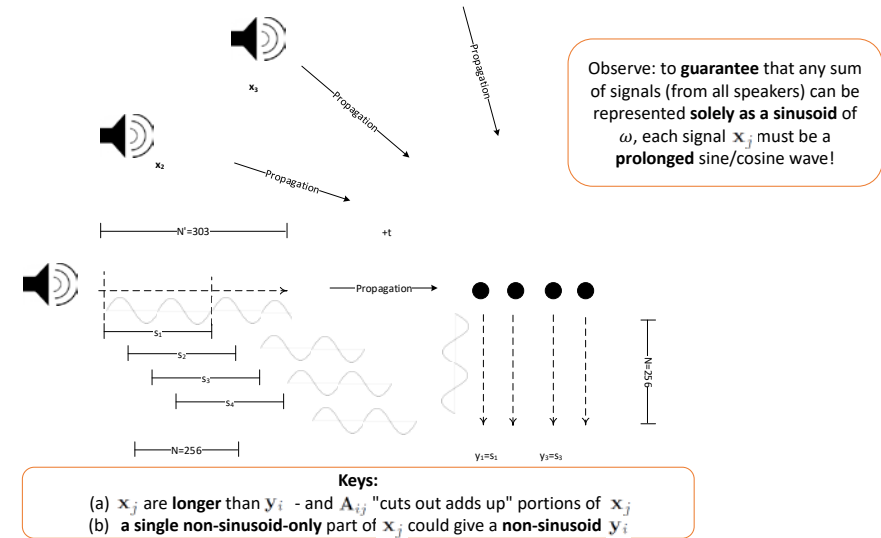
$$\begin{aligned}\cos(\omega t - \delta) &= a \cos(\omega t) + b \sin(\omega t) \\ \sin(\omega t - \delta) &= a \cos(\omega t) + b \sin(\omega t)\end{aligned}$$

Also sums (and differences) of sines and cosines of the same frequency becomes sum of sines/cosines of that same frequency

$$a_1 \cos(\omega t - \delta_1) + \dots + a_N \cos(\omega t - \delta_N) = K_1 \cos(\omega t) + K_2 \sin(\omega t)$$

Let's consider **any and all** microphone signals as linear combinations of a sine and cosine of a single frequency ω !

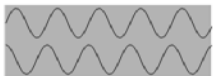
37



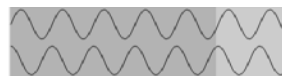
38

Why is this important?

- We want to **model** the receive signals y_i precisely
- We don't want any **errors** that can propagate upon inversion..
- **Orthogonal errors** may be ok (as we will see), but not other ones..



If we want y to definitely have this basis....



...all the x 's must have **this basis**

39

Let us define $s_\omega(t)$ as the *sine vector* of length t :

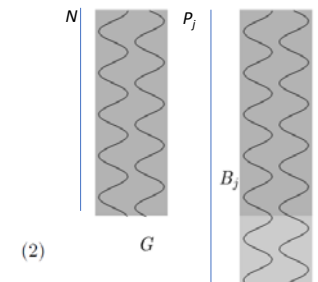
$$s_\omega(t) = [\sin(\omega), \sin(2\omega), \sin(3\omega), \dots, \sin(t\omega)]^T$$

pick $t = N$ and $\omega = 2k\pi/N$

Let $c_N(k)$ be the corresponding *cosine vector*.

$$G = [c_\omega(N) \mid s_\omega(N)]$$

It follows from the selection of ω that $G^T G = I$.



40

Let us define $\mathbf{s}_\omega(t)$ as the *sine vector* of length t :

$$\mathbf{s}_\omega(t) = [\sin(\omega), \sin(2\omega), \sin(3\omega), \dots, \sin(t\omega)]^T$$

pick $t = N$ and $\omega = 2k\pi/N$

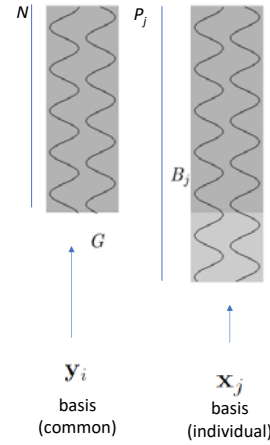
Let $\mathbf{c}_N(k)$ be the corresponding *cosine vector*.

$$\mathbf{G} = [\mathbf{c}_\omega(N) \mid \mathbf{s}_\omega(N)] \quad (2)$$

It follows from the selection of ω that $\mathbf{G}^T \mathbf{G} = \mathbf{I}$.

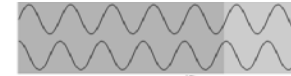
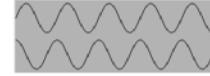
$$\mathbf{B}_j = [\mathbf{c}_\omega(P_j) \mid \mathbf{s}_\omega(P_j)] \quad (3)$$

We recall that $P_j \geq N$ for all sources $j = 1, \dots, M$. The matrix $\mathbf{B}_j$ is therefore a *periodic extension* of the matrix $\mathbf{G}$: the N first rows are identical, and then the next $P_j - N$ columns are repetitions of the first $P_j - N$ columns of $\mathbf{G}$.



41

Theorem 1 Let $\mathbf{y} = \sum_{j=1}^M \mathbf{A}_j \mathbf{x}_j$ where the matrices $\{\mathbf{A}_j\}$ are integer pick-up matrices. Let $\mathbf{B}_j = [\mathbf{c}_\omega(P_j) \mid \mathbf{s}_\omega(P_j)]$ and $\mathbf{G} = [\mathbf{c}_\omega(N) \mid \mathbf{s}_\omega(N)]$, where $P_j \geq N$ for all j , and where $\omega = k\pi/N$ for some integer k . Then, $\mathbf{x}_j \in R(\mathbf{B}_j)$ for all $j = 1, \dots, M \Leftrightarrow \mathbf{y} \in R(\mathbf{G})$, for every $\{\mathbf{x}_j\}$ as defined.



If we want $\mathbf{y}$ to definitely have this basis....

...all the $\mathbf{x}$'s must have **this basis**

This theorem says that if all the $\mathbf{x}$'s are in the extended cosine/sine space, then all the $\mathbf{y}$'s are in the corresponding cosine/sine space

42

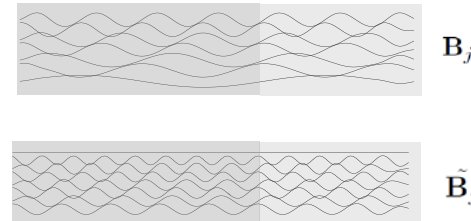
Theorem 1 Let $\mathbf{y} = \sum_{j=1}^M \mathbf{A}_j \mathbf{x}_j$ where the matrices $\{\mathbf{A}_j\}$ are integer pick-up matrices. Let $\mathbf{B}_j = [\mathbf{c}_\omega(P_j) \mid \mathbf{s}_\omega(P_j)]$ and $\mathbf{G} = [\mathbf{c}_\omega(N) \mid \mathbf{s}_\omega(N)]$, where $P_j \geq N$ for all j , and where $\omega = k\pi/N$ for some integer k . Then, $\mathbf{x}_j \in R(\mathbf{B}_j)$ for all $j = 1, \dots, M \Leftrightarrow \mathbf{y} \in R(\mathbf{G})$, for every $\{\mathbf{x}_j\}$ as defined.

$$\mathbf{x}_j \in R[\mathbf{B}_j(\omega_1)] \cup R[\mathbf{B}_j(\omega_2)] \cup \dots \cup R[\mathbf{B}_j(\omega_K)]$$

Theorem 2 Let $\mathbf{y} = \sum_{j=1}^M \mathbf{A}_j \mathbf{x}_j$ where the matrices $\{\mathbf{A}_j\}$ are integer pick-up matrices. Let $\mathbf{B}_{j,r} = [\mathbf{c}_{\omega_r}(P_j) \mid \mathbf{s}_{\omega_r}(P_j)]$ and $\mathbf{B}_j = [\mathbf{B}_{j1}, \dots, \mathbf{B}_{jK}]$ and $\mathbf{G}_r = [\mathbf{c}_{\omega_r}(N) \mid \mathbf{s}_{\omega_r}(N)]$ and $\mathbf{G} = [\mathbf{G}_1, \dots, \mathbf{G}_K]$, where $P_j \geq N$ for all j , and where $\omega_r = k\pi/N$ for some integer k . Then $\mathbf{x}_j \in R(\mathbf{B}_j)$ for all $j = 1, \dots, M \Leftrightarrow \mathbf{y} \in R(\mathbf{G})$, for every choice of $\{\mathbf{x}_j\}$ as defined.

Also works for multiple frequencies (as long as they are orthogonal), and also for *linear combinations* of pick-up matrices - including the fractional time-delay approach!

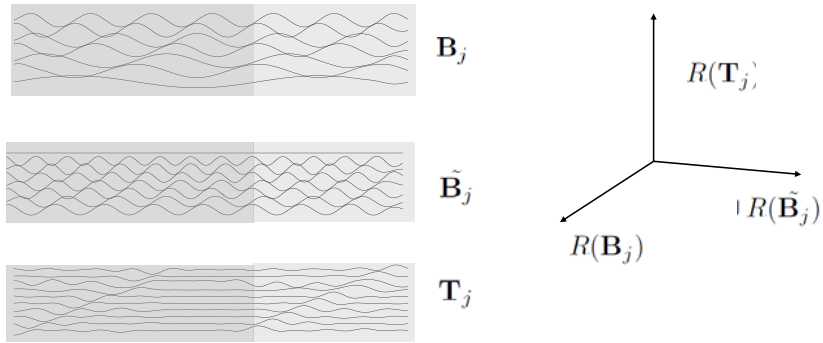
43



$$R(\mathbf{B}_j) \cup R(\tilde{\mathbf{B}}_j) = R^{P_j}?$$

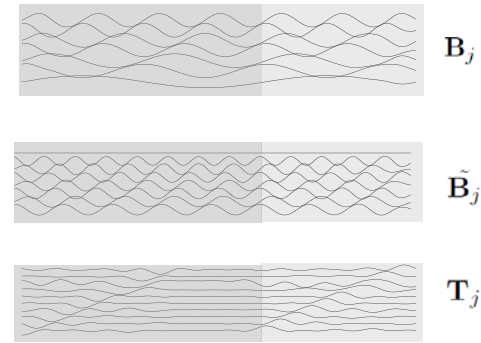
Does your (extended) selected basis + the other (extended) frequencies span all that the $\mathbf{x}$ can be?

44



$$R(\mathbf{B}_j) \cup R(\tilde{\mathbf{B}}_j) \cup R(\mathbf{T}_j) = R^{P_j}$$

No! To span the space, you need another part of it – the transient - the orthogonal complement to $R(\mathbf{B}_j) \cup R(\tilde{\mathbf{B}}_j)$



$$\mathbf{x}_j = \mathbf{B}_j \alpha_j + \tilde{\mathbf{B}}_j \gamma_j + \mathbf{T}_j \phi_j = [\mathbf{B}_j \ \tilde{\mathbf{B}}_j \ \mathbf{T}_j] \begin{bmatrix} \alpha_j \\ \gamma_j \\ \phi_j \end{bmatrix}$$

46

Now we use the full model.

$$\mathbf{y} = \mathbf{A}\mathbf{x} = \mathbf{A}[\mathbf{B}\alpha + \tilde{\mathbf{B}}\gamma + \mathbf{T}\phi] \quad (4)$$

$$= \mathbf{A}\mathbf{B}\alpha + \mathbf{A}\tilde{\mathbf{B}}\gamma + \mathbf{A}\mathbf{T}\phi \quad (5)$$

Let us assume that the supermatrix $\mathbf{B}$ is built up of matrices $\mathbf{B}_j$ that contain each a *single cosine/sine pair*. We then have a matching, block-diagonal $\tilde{\mathbf{G}}$ matrix that picks up this frequency by pre-multiplying it with $\mathbf{y}$ giving:

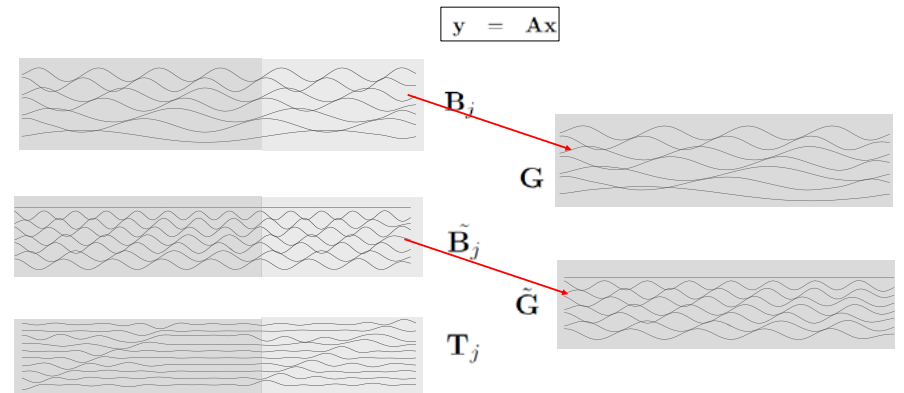
$$\tilde{\mathbf{G}}^T \mathbf{y} = \underbrace{\tilde{\mathbf{G}}^T \mathbf{A} \mathbf{B}}_{\tilde{\mathbf{G}} \hat{\alpha}} \alpha + \underbrace{\tilde{\mathbf{G}}^T \mathbf{A} \tilde{\mathbf{B}}}_{\mathbf{H} \tilde{\gamma}} \gamma + \tilde{\mathbf{G}}^T \mathbf{A} \mathbf{T} \phi \quad (6)$$

$$= \tilde{\mathbf{G}}^T \tilde{\mathbf{G}} \hat{\alpha} + \underbrace{\tilde{\mathbf{G}}^T \mathbf{H}}_{=0} \tilde{\gamma} + \underbrace{\tilde{\mathbf{G}}^T \mathbf{A} \mathbf{T}}_{=\epsilon} \phi \quad (7)$$

$$\boxed{\mathbf{A}^\dagger} = \hat{\alpha} + \epsilon \quad (8)$$

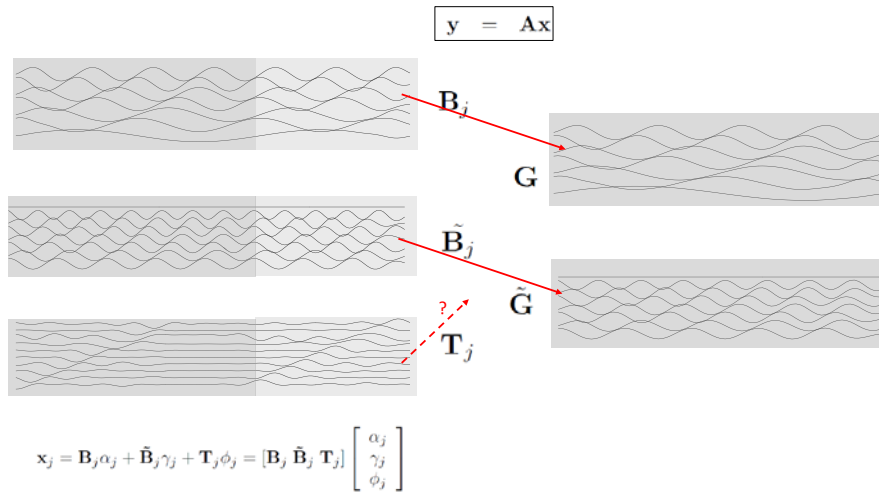
In the inversion step (to recover the signal of interest), the error will explode!

48

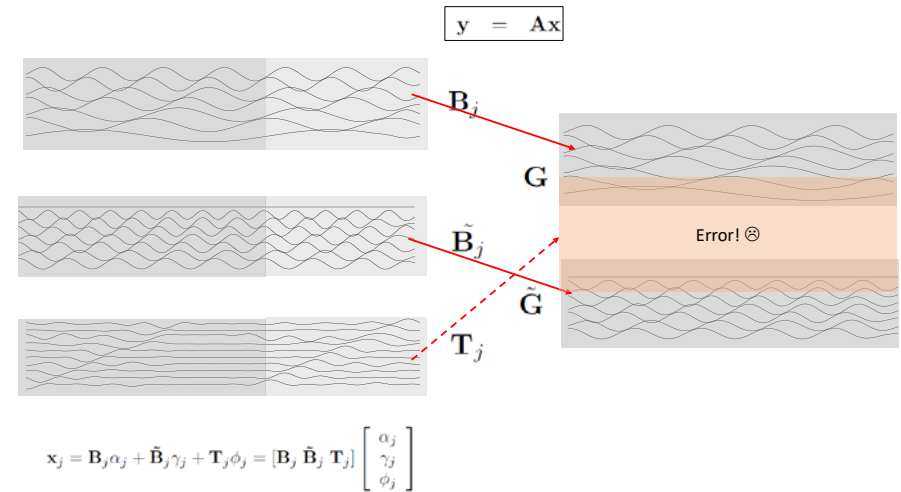


$$\mathbf{x}_j = \mathbf{B}_j \alpha_j + \tilde{\mathbf{B}}_j \gamma_j + \mathbf{T}_j \phi_j = [\mathbf{B}_j \ \tilde{\mathbf{B}}_j \ \mathbf{T}_j] \begin{bmatrix} \alpha_j \\ \gamma_j \\ \phi_j \end{bmatrix}$$

49



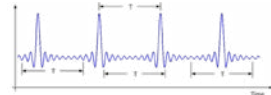
50



51

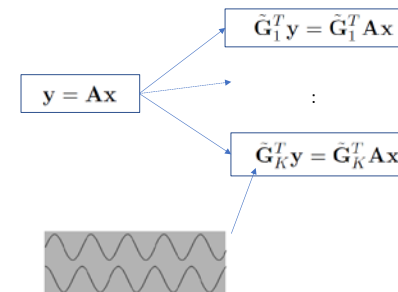
Summary on (multi-) narrow band and transients

- **Transient error impacts on each frequency: extending to multiple doesn't help!**
 - You can use the full spectrum if you like, but the transient will still plague you!
- **How badly the transient error impacts will depend on inverse matrix choice**
 - Transposes are more robust than pseudo-inverses
 - But also less accurate..
- **There would be no residual issue if our x's were sums of extended sinusoids**
 - Which is the same as saying they are **self-repeating signals**
 - It is conceivable that a single frequency signal would go on to repeat itself...
 - But a multi-frequency signal - ??? That's **weird!**
 - **Yet**, that is the assumption you implicitly make when doing fourier-based beam forming
 - If your signal is not periodic, you introduce a **real error** (not just a "conceptual one")

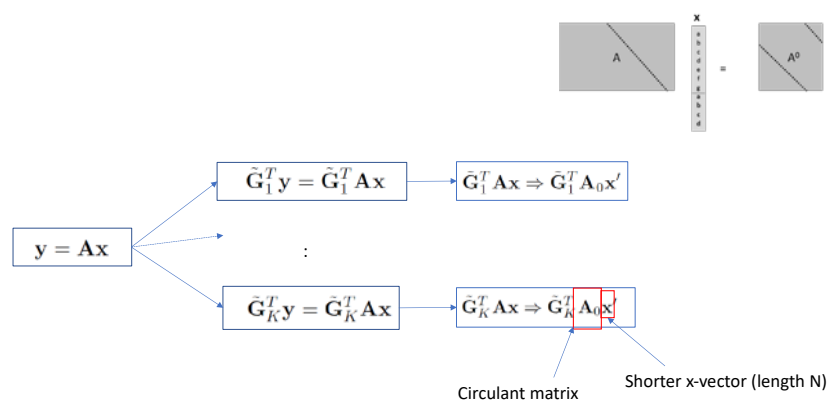


52

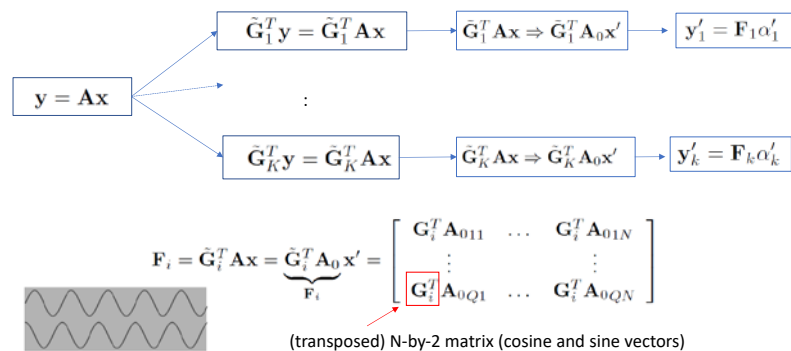
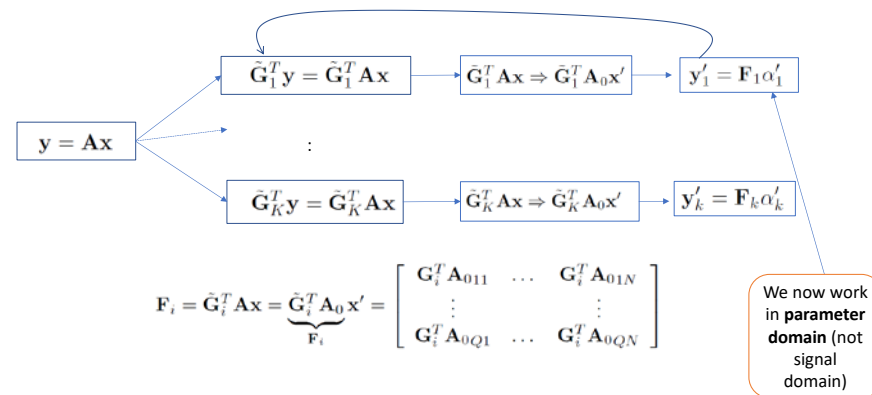
The Fourier Approach



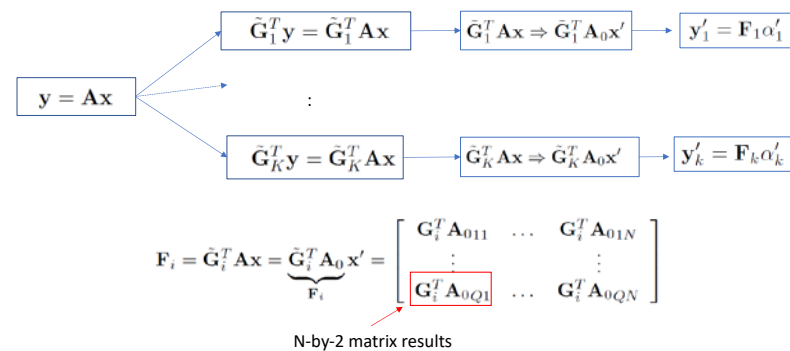
53



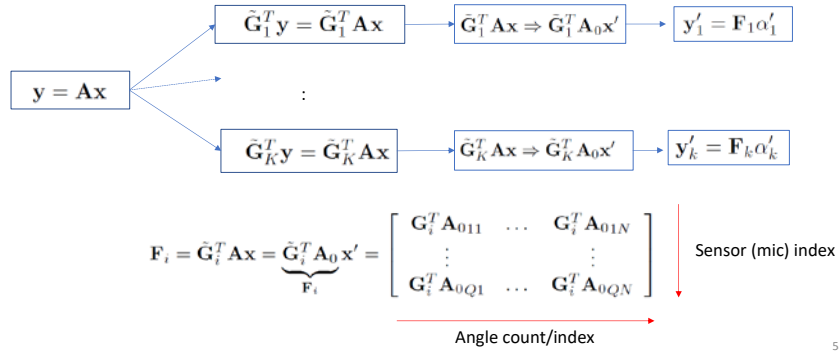
54



56

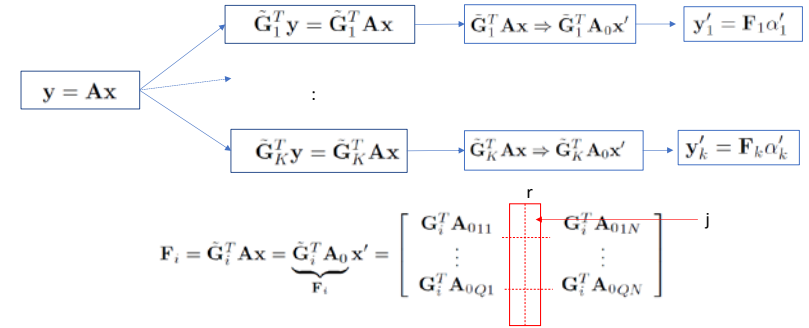


57



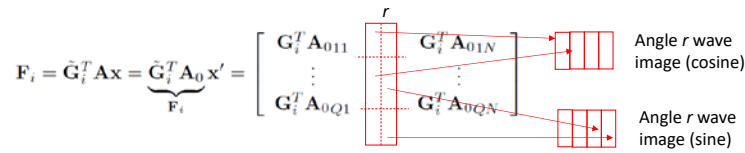
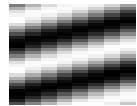
58

A block with 2 supervector columns. Each Nx1 sub-vector describes **potential contributions** from **r'th angle** to the signal of the **j'th microphone** (sine and cosine)

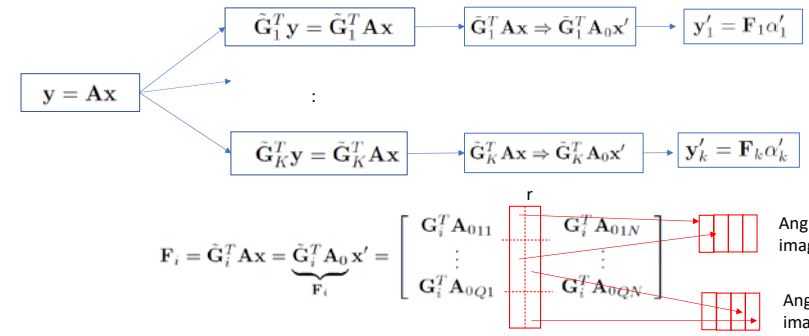


59

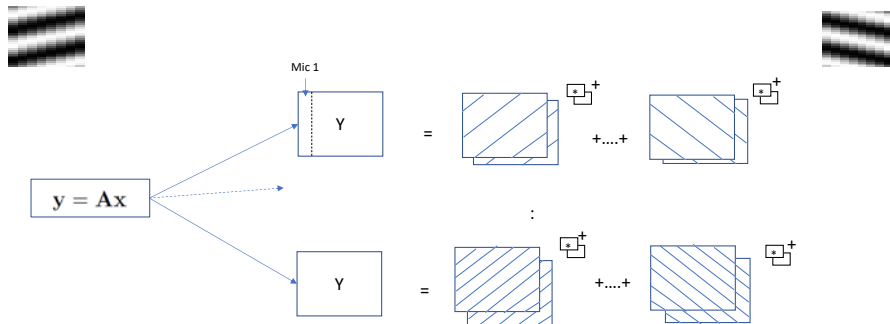
For each frequency, we "explain what we see" as linear combinations of signals that can possibly arrive from each of the N different angles



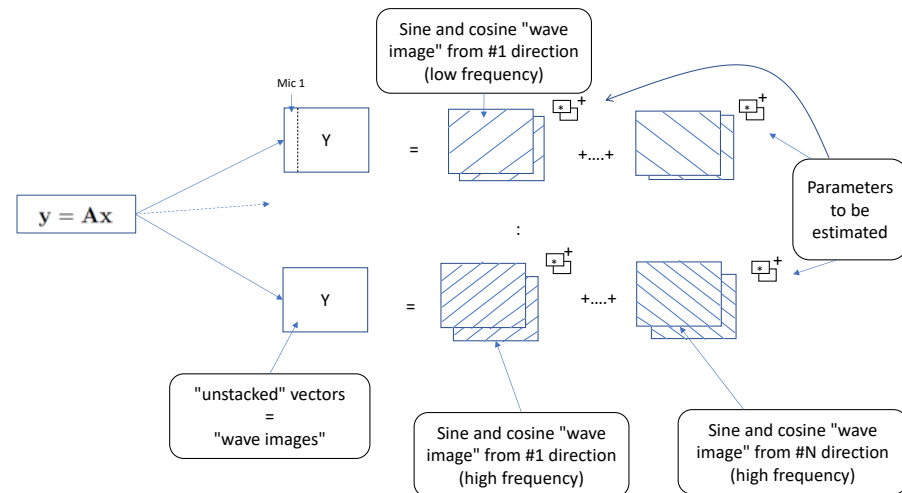
60



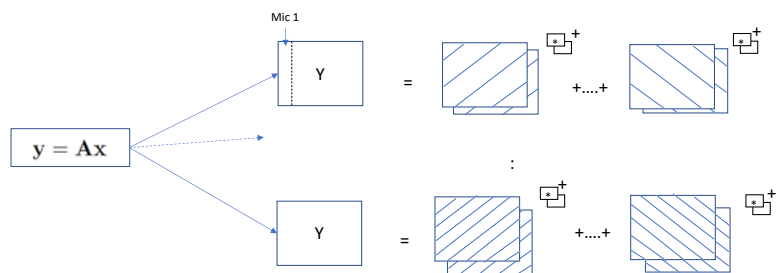
61



62

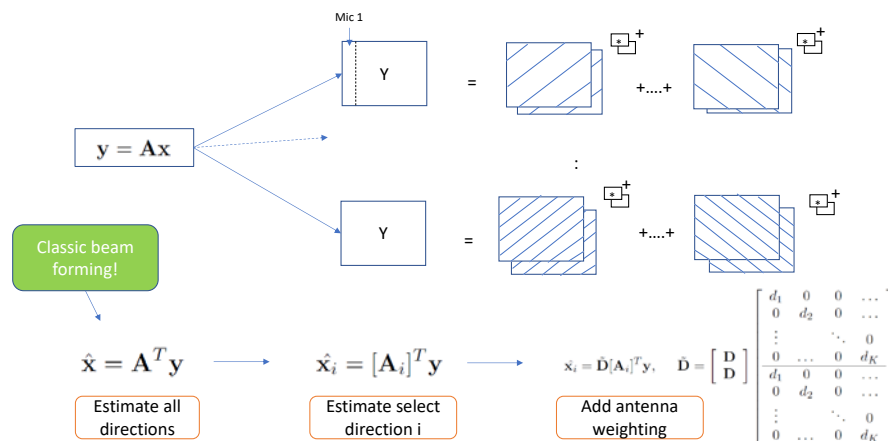


63



These "wave images" are not just important for visualization – they form the **basis of the regularization schemes** used in the beamforming "estimation" approach

64



$$\hat{x} = A^T y \rightarrow \hat{x}_i = [A_i]^T y \rightarrow \hat{x}_i = \hat{D} [A_i]^T y, \quad \hat{D} = \begin{bmatrix} \hat{D} & \hat{D} \\ \hat{D} & \hat{D} \end{bmatrix}$$

65

For all frequencies simultaneously:

$$\hat{\mathbf{x}} = \mathbf{A}^T \mathbf{y} \longrightarrow \hat{\mathbf{x}}_i = [\mathbf{A}_i]^T \mathbf{y} \longrightarrow \hat{x}_i = \hat{\mathbf{D}} [\mathbf{A}_i]^T \mathbf{y}', \quad \hat{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \mathbf{D} \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & \dots \\ 0 & d_2 & 0 & \dots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & d_K \end{bmatrix}$$

For any chosen frequency:

$$\hat{\alpha} = \mathbf{F}^T \mathbf{y}' \longrightarrow \hat{\alpha}_i = [\mathbf{F}]_i^T \mathbf{y}' = [\mathbf{F}_i]^T \mathbf{F} \alpha \longrightarrow \hat{\alpha}_i = \hat{\mathbf{D}} [\mathbf{F}_i]^T \mathbf{y}', \quad \hat{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \mathbf{D} \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & \dots \\ 0 & d_2 & 0 & \dots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & d_K \end{bmatrix}$$

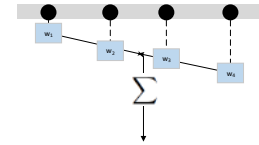
66

For all frequencies simultaneously:

$$\hat{\mathbf{x}} = \mathbf{A}^T \mathbf{y} \longrightarrow \hat{\mathbf{x}}_i = [\mathbf{A}_i]^T \mathbf{y} \longrightarrow \hat{x}_i = \hat{\mathbf{D}} [\mathbf{A}_i]^T \mathbf{y}', \quad \hat{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \mathbf{D} \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & \dots \\ 0 & d_2 & 0 & \dots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & d_K \end{bmatrix}$$

For any chosen frequency:

$$\hat{\alpha} = \mathbf{F}^T \mathbf{y}' \longrightarrow \hat{\alpha}_i = [\mathbf{F}]_i^T \mathbf{y}' = [\mathbf{F}_i]^T \mathbf{F} \alpha \longrightarrow \hat{\alpha}_i = \hat{\mathbf{D}} [\mathbf{F}_i]^T \mathbf{y}', \quad \hat{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \mathbf{D} \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & \dots \\ 0 & d_2 & 0 & \dots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & d_K \end{bmatrix}$$



Why $\mathbf{D}$? It enables us to design a our beam-pattern – to be defined.
"D is our flexibility in designing (or modifying) our inverse matrix - as enabled via analogue electronics"

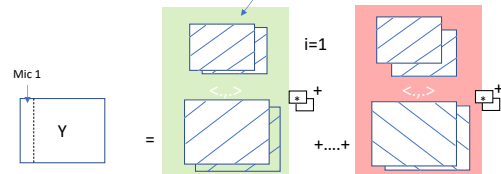
67

For all frequencies simultaneously:

$$\hat{\mathbf{x}} = \mathbf{A}^T \mathbf{y} \longrightarrow \hat{\mathbf{x}}_i = [\mathbf{A}_i]^T \mathbf{y} \longrightarrow \hat{x}_i = \hat{\mathbf{D}} [\mathbf{A}_i]^T \mathbf{y}', \quad \hat{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \mathbf{D} \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & \dots \\ 0 & d_2 & 0 & \dots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & d_K \end{bmatrix}$$

For any chosen frequency:

$$\hat{\alpha} = \mathbf{F}^T \mathbf{y}' \longrightarrow \hat{\alpha}_i = [\mathbf{F}]_i^T \mathbf{y}' = [\mathbf{F}_i]^T \mathbf{F} \alpha \longrightarrow \hat{\alpha}_i = \hat{\mathbf{D}} [\mathbf{F}_i]^T \mathbf{y}', \quad \hat{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \mathbf{D} \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & \dots \\ 0 & d_2 & 0 & \dots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & d_K \end{bmatrix}$$



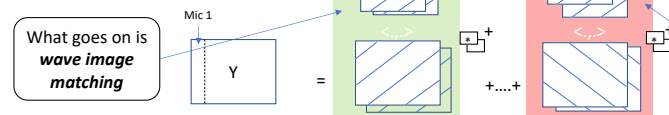
68

For all frequencies simultaneously:

$$\hat{\mathbf{x}} = \mathbf{A}^T \mathbf{y} \longrightarrow \hat{\mathbf{x}}_i = [\mathbf{A}_i]^T \mathbf{y} \longrightarrow \hat{x}_i = \hat{\mathbf{D}} [\mathbf{A}_i]^T \mathbf{y}', \quad \hat{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \mathbf{D} \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & \dots \\ 0 & d_2 & 0 & \dots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & d_K \end{bmatrix}$$

For any chosen frequency:

$$\hat{\alpha} = \mathbf{F}^T \mathbf{y}' \longrightarrow \hat{\alpha}_i = [\mathbf{F}]_i^T \mathbf{y}' = [\mathbf{F}_i]^T \mathbf{F} \alpha \longrightarrow \hat{\alpha}_i = \hat{\mathbf{D}} [\mathbf{F}_i]^T \mathbf{y}', \quad \hat{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \mathbf{D} \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & \dots \\ 0 & d_2 & 0 & \dots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & d_K \end{bmatrix}$$



Hope for mismatch
 = contribution
 close to 0!

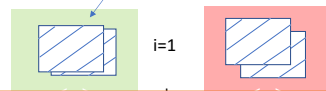
For all frequencies simultaneously:

$$\hat{\mathbf{x}} = \mathbf{A}^T \mathbf{y} \longrightarrow \hat{\mathbf{x}}_i = [\mathbf{A}_i]^T \mathbf{y} \longrightarrow \hat{\mathbf{x}}_i = \hat{\mathbf{D}} [\mathbf{A}_i]^T \mathbf{y}, \quad \hat{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \mathbf{D} \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & \dots \\ 0 & d_2 & 0 & \dots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \dots & 0 & d_K \end{bmatrix}$$

For any chosen frequency:

$$\hat{\alpha} = \mathbf{F}^T \mathbf{y}' \longrightarrow \hat{\alpha}_i = [\mathbf{F}_i]^T \mathbf{y}' = [\mathbf{F}_i]^T \mathbf{F} \alpha \longrightarrow \hat{\alpha}_i = \hat{\mathbf{D}} [\mathbf{F}_i]^T \mathbf{y}', \quad \hat{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \mathbf{D} \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & \dots \\ 0 & d_2 & 0 & \dots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \dots & 0 & d_K \end{bmatrix}$$

Mic 1



In essence, we have **constrained our estimator for a given component pair (direction)** to be equal to the **physical wave response** from that direction - plus possibly a weighting of the microphone signals.

70

For all frequencies simultaneously:

$$\hat{\mathbf{x}} = \mathbf{A}^T \mathbf{y} \longrightarrow \hat{\mathbf{x}}_i = [\mathbf{A}_i]^T \mathbf{y} \longrightarrow \hat{\mathbf{x}}_i = \hat{\mathbf{D}} [\mathbf{A}_i]^T \mathbf{y}, \quad \hat{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \mathbf{D} \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & \dots \\ 0 & d_2 & 0 & \dots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \dots & 0 & d_K \end{bmatrix}$$

For any chosen frequency:

$$\hat{\alpha} = \mathbf{F}^T \mathbf{y}' \longrightarrow \hat{\alpha}_i = [\mathbf{F}_i]^T \mathbf{y}' = [\mathbf{F}_i]^T \mathbf{F} \alpha \longrightarrow \hat{\alpha}_i = \hat{\mathbf{D}} [\mathbf{F}_i]^T \mathbf{y}', \quad \hat{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \mathbf{D} \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & \dots \\ 0 & d_2 & 0 & \dots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \dots & 0 & d_K \end{bmatrix}$$

Mic 1

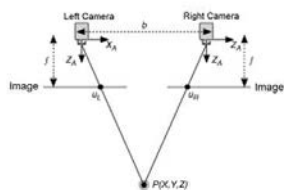


In essence, we have **constrained our estimate of a given component pair (direction)** to be equal to the **physical wave response**

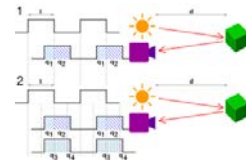
We could call this "physically inspired regularization"

71

Parallel: Bio-mimicry in 3D computer vision



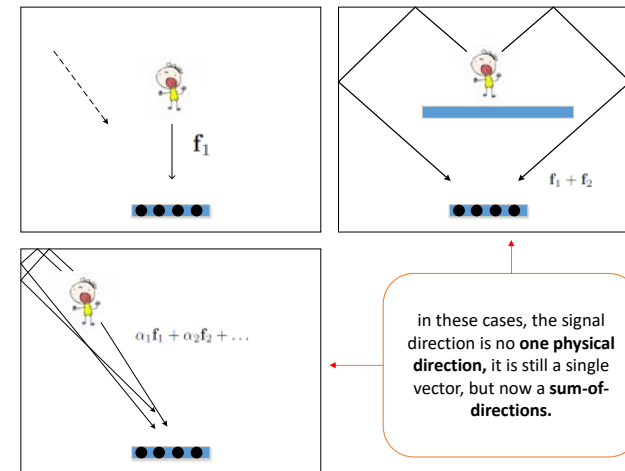
Passive stereo camera (correlation, matching) – "humanlike"



Pulsed laser – no human can do that ☺

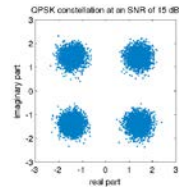
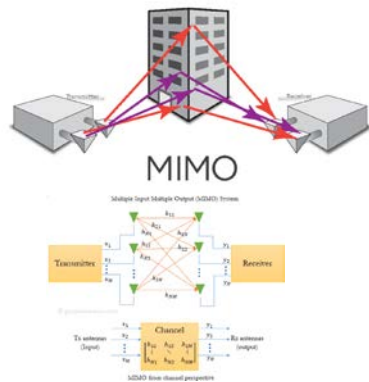
Lesson: You may **start out** by mimicing biology (or physics) but you shouldn't **constrain** yourself to doing so!

72



in these cases, the signal direction is no **one physical direction**, it is still a single vector, but now a **sum-of-directions**.

73



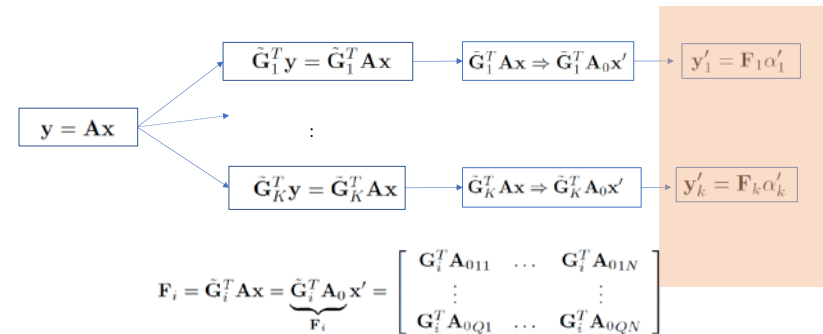
Multipath **increases** channel capacity
(echoes not a problem)

It's all about finding **directions** (physical
or not) with enough **information**

In MIMO communication systems, you cannot be picky as to whether your weights correspond to a **physical direction** – every % reduced bit error rates (BER) counts (\$ millions!)

74

Least Square Estimation: Part 1



75

$$\mathbf{y}_i = \mathbf{F}_i \alpha_i$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_Q \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 & \dots & 0 & 0 \\ 0 & \mathbf{F}_2 & 0 & \dots \\ \vdots & & \ddots & \\ 0 & \dots & 0 & \mathbf{F}_Q \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_N \end{bmatrix} = \mathbf{F} \alpha$$

and we have *frequency orthogonality*, i.e. $\mathbf{F}_i^T \mathbf{F}_j = 0$ for $i \neq j$. (Why do this on block matrix form?)

76

Clearly, the inverse estimation operation is also separable, as we already know:

$$\hat{\alpha} = [\mathbf{F}^T \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{y}$$

$$= \begin{bmatrix} \mathbf{F}_1^T \mathbf{F}_1 & \dots & 0 & 0 \\ 0 & \mathbf{F}_2^T \mathbf{F}_2 & 0 & \dots \\ \vdots & & \ddots & \\ 0 & \dots & 0 & \mathbf{F}_Q^T \mathbf{F}_Q \end{bmatrix}^{-1} \mathbf{F}^T \mathbf{y}$$

$$= \begin{bmatrix} [\mathbf{F}_1^T \mathbf{F}_1]^{-1} & \dots & 0 & 0 \\ 0 & [\mathbf{F}_2^T \mathbf{F}_2]^{-1} & 0 & \dots \\ \vdots & & \ddots & \\ 0 & \dots & 0 & [\mathbf{F}_Q^T \mathbf{F}_Q]^{-1} \end{bmatrix} \mathbf{F}^T \mathbf{y}$$

which means that (as we already could see from the first equation):

$$\hat{\alpha}_i = [\mathbf{F}_i^T \mathbf{F}_i]^{-1} \mathbf{F}_i^T \mathbf{y}$$

$$\mathbf{y}_i = \mathbf{F}_i \alpha_i$$

77

But, this solution whether from block matrix or from individual blocks, i.e.

$$\mathbf{y} = \mathbf{F}\alpha \Rightarrow \hat{\alpha} = [\mathbf{F}^T \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{y}$$

assumes that $E(\alpha\alpha^T) = \mathbf{I}$. We can see this from Bayes formula.

$$\begin{aligned} \mathbf{y} &= \mathbf{A}\mathbf{x} + \mathbf{n} & \begin{cases} E(\mathbf{n}\mathbf{n}^T) = \mathbf{R} \\ E(\mathbf{x}\mathbf{x}^T) = \mathbf{C} \end{cases} \\ \hat{\mathbf{x}} &= (\mathbf{A}^T \mathbf{C} \mathbf{A} + \mathbf{R})^{-1} \mathbf{A}^T \mathbf{C} \mathbf{y} \end{aligned}$$

78

But, this solution whether from block matrix or from individual blocks, i.e.

$$\mathbf{y} = \mathbf{F}\alpha \Rightarrow \hat{\alpha} = [\mathbf{F}^T \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{y}$$

assumes that $E(\alpha\alpha^T) = \mathbf{I}$. We can see this from Bayes formula.

$$\begin{aligned} \mathbf{y} &= \mathbf{A}\mathbf{x} + \mathbf{n} & \begin{cases} E(\mathbf{n}\mathbf{n}^T) = \mathbf{R} \\ E(\mathbf{x}\mathbf{x}^T) = \mathbf{C} \end{cases} \\ \hat{\mathbf{x}} &= (\mathbf{A}^T \mathbf{C} \mathbf{A} + \mathbf{R})^{-1} \mathbf{A}^T \mathbf{C} \mathbf{y} \end{aligned}$$

If $E(\alpha\alpha^T) = \mathbf{C}$ and $\mathbf{C} \neq \mathbf{I}$, and also $\mathbf{C}$ not *block-diagonal*, then we have a different solution:

$$\hat{\alpha} = [\mathbf{F}^T \mathbf{C} \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{C} \mathbf{y}$$

Which is typically *not* separable. The *hidden assumption* underlying the Fourier approach is therefore that the frequency bands are *independent*, and this becomes visible when seeing the problem in light of Bayes estimation.

79

But, this solution whether from block matrix or from individual blocks, i.e.

$$\mathbf{y} = \mathbf{F}\alpha \Rightarrow \hat{\alpha} = [\mathbf{F}^T \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{y}$$

assumes that $E(\alpha\alpha^T) = \mathbf{I}$. We can see this from Bayes formula.

$$\begin{aligned} \mathbf{y} &= \mathbf{A}\mathbf{x} + \mathbf{n} & \begin{cases} E(\mathbf{n}\mathbf{n}^T) = \mathbf{R} \\ E(\mathbf{x}\mathbf{x}^T) = \mathbf{C} \end{cases} \\ \hat{\mathbf{x}} &= (\mathbf{A}^T \mathbf{C} \mathbf{A} + \mathbf{R})^{-1} \mathbf{A}^T \mathbf{C} \mathbf{y} \end{aligned}$$

If $E(\alpha\alpha^T) = \mathbf{C}$ and $\mathbf{C} \neq \mathbf{I}$, and also $\mathbf{C}$ not *block-diagonal*, then we have a different solution:

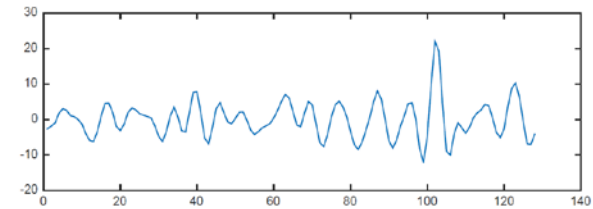
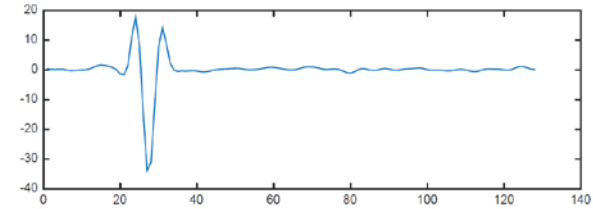
$$\hat{\alpha} = [\mathbf{F}^T \mathbf{C} \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{C} \mathbf{y}$$

Which is typically *not* separable. The *hidden assumption* underlying the Fourier approach is therefore that the frequency bands are *independent*, and this becomes visible when seeing the problem in light of Bayes estimation.

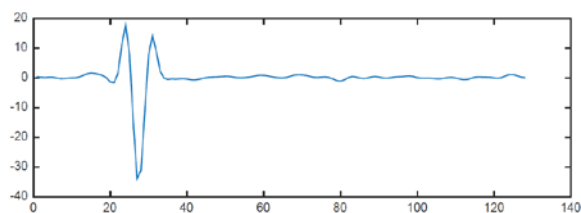
Frequency independency is typically not the case for many signals, such as speech. Think for instance of *harmonic overtones*.

We will also see that it is not the case for some signals with high information content.

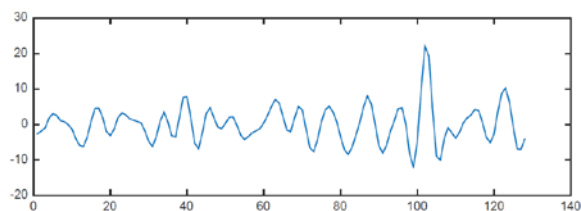
80



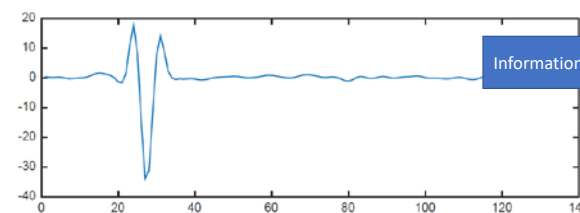
81



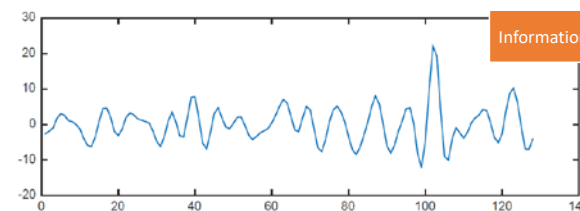
Signal
↓
Fourier representation
(sines, cosines)
↓
Phase "scrambling"
(per frequency)
↓
Modified Signal



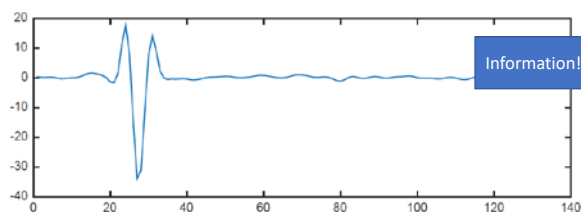
82



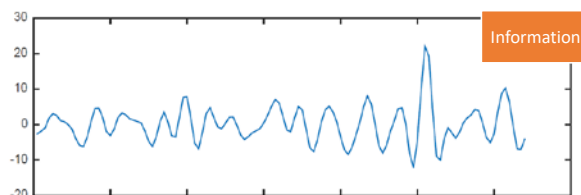
Signal
↓
Fourier representation
(sines, cosines)
↓
Phase "scrambling"
(per frequency)
↓
Modified Signal



83



Signal
↓
Fourier representation
(sines, cosines)
↓
Phase "scrambling"
(per frequency)
↓
Modified Signal



If we are hunting for **information**, then decoupling frequencies and working on them **individually** would be a really bad idea: The **information** lies in the relationship between the phases!

84

III. BEAMFORMING WITH HIGHER-ORDER STATISTICS

The conventional beamforming algorithms estimate their weights based on the covariance matrix of the snapshot vectors. In other words, the conventional beamformer's weights are determined solely by second-order statistics (SOS). Beamforming with higher-order statistics (HOS) has recently been

proposed in the literature [35]; it has been demonstrated that such HOS beamformers do *not* suffer from signal cancellation.

In this section, we introduce a concept of non-Gaussianity and describe how the fine structure of a non-Gaussian *probability density function* (pdf) can be characterized by measures such as kurtosis and negentropy. Moreover, we present empirical evidence that speech signals are highly non-Gaussian [37]; thereafter we discuss beamforming algorithms based on maximizing non-Gaussian optimization criteria.

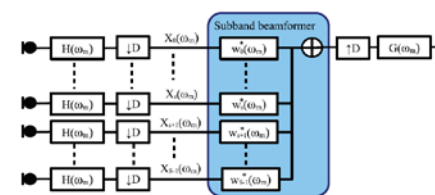


Fig. 6. Schematic view for beamforming in the subband domain.

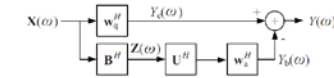


Fig. 8. Maximum kurtosis beamformer with the subspace filter.

From: "Microphone Array Processing for Distant Speech Recognition: Towards Real-world deployment", by Arakawa et al, 2012 [Disney Research, NEC, Toyohashi University of Tokyo, Carnegie Mellon University, Microsoft Research (Ivan Tashev)]

85

III. BEAMFORMING WITH HIGHER-ORDER STATISTICS

The conventional beamforming algorithms estimate their weights based on the covariance matrix of the snapshot vectors. In other words, the conventional beamformer's weights are determined solely by second-order statistics (SOS). Beamforming with higher-order statistics (HOS) has recently been

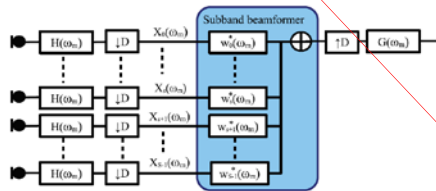


Fig. 6. Schematic view for beamforming in the subband domain.

proposed in the literature [35]; it has been demonstrated that such HOS beamformers do not suffer from signal cancellation.

In this section, we introduce a concept of non-Gaussianity and describe how the fine structure of a non-Gaussian probability density function (pdf) can be characterized by measures such as kurtosis and negentropy. Moreover, we present empirical evidence that speech signals are highly non-Gaussian [37]; thereafter we discuss beamforming algorithms based on maximizing non-Gaussian optimization criteria.

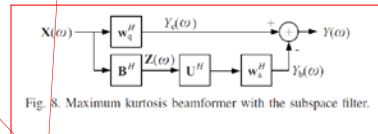


Fig. 8. Maximum kurtosis beamformer with the subspace filter.

Information-seeking + narrowband!

From: "Microphone Array Processing for Distant Speech Recognition: Towards Real-world deployment", by Arakawa et al, 2012 [Disney Research, NEC, Toyohashi University of Tokyo, Carnegie Mellon University, Microsoft Research (Ivan Tashev)]

THE THIRD "CHIME" SPEECH SEPARATION AND RECOGNITION CHALLENGE: DATASET, TASK AND BASELINES

Jon Barker & Richard Mørner
University of Sheffield, UK
[j.p.barker,r.morner]@sheffield.ac.uk

Emmanuel Vincent
Inria, France
emmanuel.vincent@inria.fr

Shinji Watanabe
MERI, USA
watanabe@meri.com

ABSTRACT
The CHIME challenge series aims to advance for field speech recognition technology by promoting research at the interface of signal processing and automatic speech recognition. This paper presents the design and outcomes of the 3rd CHIME Challenge, which targets the performance of automatic speech recognition in a real-world, commercially motivated scenario: a person talking to a tablet device that has been fitted with a six-channel microphone array. The paper describes the data collection, the task definition and the baseline systems for data simulation, enhancement and recognition. The paper then presents an overview of the 26 systems that were submitted to the challenge focusing on the strategies that proved to be most successful relative to the MVDR array processing and DNN acoustic modeling reference system. Challenge findings related to the role of simulated data in system training and evaluation are discussed.

Index Terms—Non-invasive ASR, microphone array, "CHIME" challenge

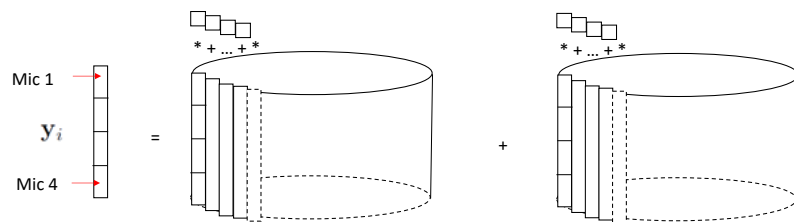
1. INTRODUCTION

System	Training	Sim. Ch. Enh.	Sim. Ch. Enh.	Feature Extract.	Feature Trans.	Acoustic Model	Lang. Model	System Comb.	Sim. Acc.	BUS	CAF	PED	STR	Ave.
Yoshida et al. [18]	✓	✓	✓	✓	✓	✓	✓	✓	4.5	7.3	4.5	6.9	5.2	5.8
Hori et al. [19]	✓	✓	✓	✓	✓	✓	✓	✓	5.6	13.5	7.2	7.1	8.1	9.1
Wu et al. [20]	✓	✓	✓	✓	✓	✓	✓	✓	7.0	13.5	11.1	9.3	7.5	10.6
Srivatsan et al. [21]	✓	✓	✓	✓	✓	✓	✓	✓	0.2	10.2	9.6	12.3	7.2	11.3
Monte et al. [22]	✓	✓	✓	✓	✓	✓	✓	✓	6.4	13.5	13.5	10.6	9.2	11.7
Nguyen et al. [23]	✓	✓	✓	✓	✓	✓	✓	✓	9.8	16.6	11.8	10.0	8.8	11.8
Zhao et al. [24]	✓	✓	✓	✓	✓	✓	✓	✓	8.2	14.5	11.7	11.5	10.6	11.9
Viet et al. [25]	✓	✓	✓	✓	✓	✓	✓	✓	8.5	17.6	12.1	8.5	9.6	11.9
Yuan et al. [26]	✓	✓	✓	✓	✓	✓	✓	✓	5.6	15.6	10.7	9.7	9.6	12.1
Heymann et al. [27]	✓	✓	✓	✓	✓	✓	✓	✓	0.0	17.5	10.5	11.9	10.0	12.3
Wang et al. [28]	✓	✓	✓	✓	✓	✓	✓	✓	9.7	17.7	11.8	13.4	11.0	13.2
Nguyen et al. [29]	✓	✓	✓	✓	✓	✓	✓	✓	7.1	17.7	14.1	13.0	9.2	13.5
Zhang et al. [30]	✓	✓	✓	✓	✓	✓	✓	✓	0.2	18.0	15.4	12.2	9.6	13.8
Tachikawa et al. [31]	✓	✓	✓	✓	✓	✓	✓	✓	8.4	23.2	13.9	11.1	8.4	14.2
Pang and Zhu [32]	✓	✓	✓	✓	✓	✓	✓	✓	6.1	19.2	13.4	17.0	10.5	14.5
Pruthi et al. [33]	✓	✓	✓	✓	✓	✓	✓	✓	13.8	19.4	11.5	16.0	10.5	14.3
Bugchi et al. [34]	✓	✓	✓	✓	✓	✓	✓	✓	21.0	21.7	14.0	13.7	12.9	16.3
Ma et al. [35]	✓	✓	✓	✓	✓	✓	✓	✓	20.0	24.6	18.4	16.0	14.1	18.6
Perrin et al. [36]	✓	✓	✓	✓	✓	✓	✓	✓	21.4	25.4	20.0	19.0	16.4	21.1
Castro Martinez et al. [37]	✓	✓	✓	✓	✓	✓	✓	✓	15.9	30.5	23.8	18.1	15.2	22.0
Pfisterberger et al. [38]	✓	✓	✓	✓	✓	✓	✓	✓	14.8	25.0	24.0	19.8	17.2	22.3
Baby et al. [39]	✓	✓	✓	✓	✓	✓	✓	✓	6.9	26.4	26.5	22.3	15.1	23.1
Monica et al. [40]	✓	✓	✓	✓	✓	✓	✓	✓	21.5	30.7	27.3	21.3	18.3	24.1
Burrows et al. [41]	✓	✓	✓	✓	✓	✓	✓	✓	15.2	35.6	32.7	26.6	19.0	28.7
Watanabe et al. [42]	✓	✓	✓	✓	✓	✓	✓	✓	15.9	44.0	29.2	23.9	13.1	29.3
DNN Baseline	✓	✓	✓	✓	✓	✓	✓	✓	21.6	51.8	34.7	27.2	20.1	33.1

MVDR + DNN = baseline – all competitors beat it!

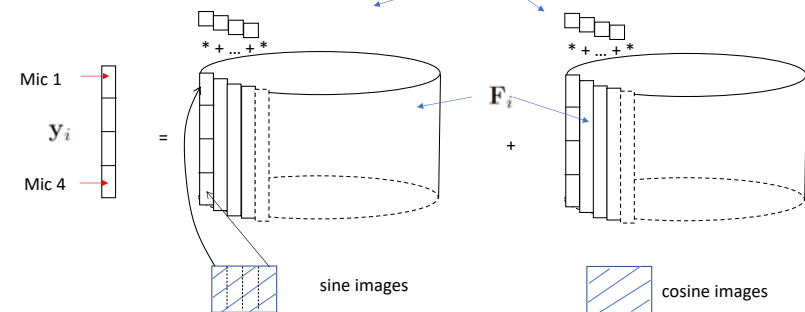
Least Squares Estimation – part 2

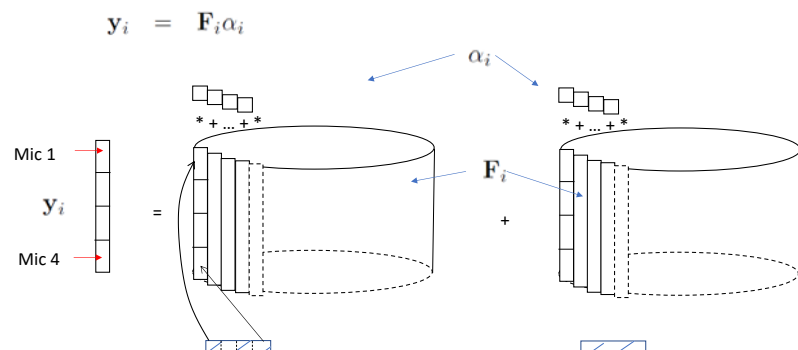
$$y_i = F_i \alpha_i$$



What kind of linear inverse problem is this – really?

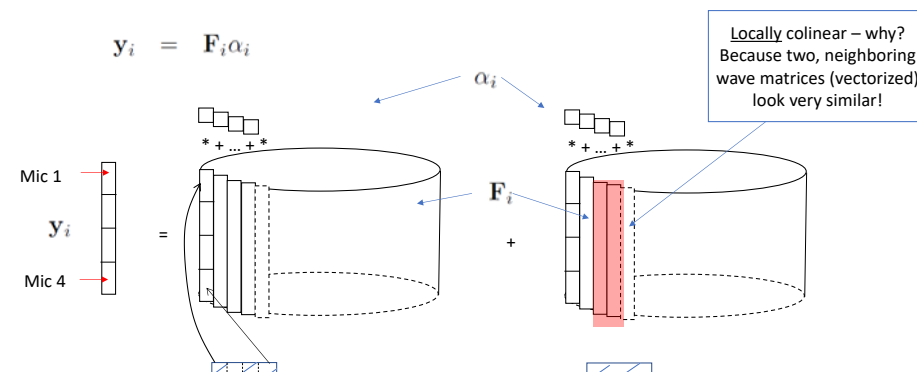
$$y_i = F_i \alpha_i$$





We have a 4 x 1000 (or 10,000?) linear **circulant** and highly **locally colinear** system!

90

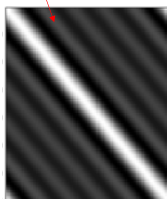


We have a 4 x 1000 (or 10,000?) linear **circulant** and highly **locally colinear** system!

91

$$y = F\alpha$$

$$\hat{\alpha} = F^\dagger y = F^\dagger F\alpha = \underbrace{[F^T F]^{-1} F^T}_{\tilde{I}} \alpha = \tilde{I} \alpha$$

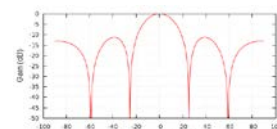


Applying the inverse matrix (or some estimate or variant of it) results in **smearing of parameters!**

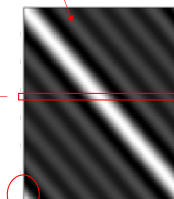
92

$$y = F\alpha$$

$$\hat{\alpha} = F^\dagger y = F^\dagger F\alpha = \underbrace{[F^T F]^{-1} F^T}_{\tilde{I}} \alpha = \tilde{I} \alpha$$



$20 \cdot \log_{10}(|\cdot|^2)$



The result of computing a single parameter (pair) in this context = **beam pattern**

93

Dealing with a (very) ill-posed problem

- The system has **really poor dimensionality**
- Co-linearity adds to the problem and results in smearing
- We can perhaps tweak the inverse matrix a little...?
- But better still – do we **know something** we could use?

94

Option 1: Least Squares with Known Covariance Matrix

$$\begin{aligned}
 \mathbf{y} &= \mathbf{F}\alpha \\
 E(\alpha\alpha^T) &= \mathbf{C} \\
 \hat{\alpha} &= [\mathbf{F}^T \mathbf{C} \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{C} \mathbf{y} = \underbrace{[\mathbf{F}^T \mathbf{C} \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{C} \mathbf{F}}_{\mathbf{I}} \alpha = \tilde{\mathbf{I}} \alpha
 \end{aligned}$$

The **spatial covariance matrix** could be **estimated** (or guessed or modelled) by studying the scene over time..

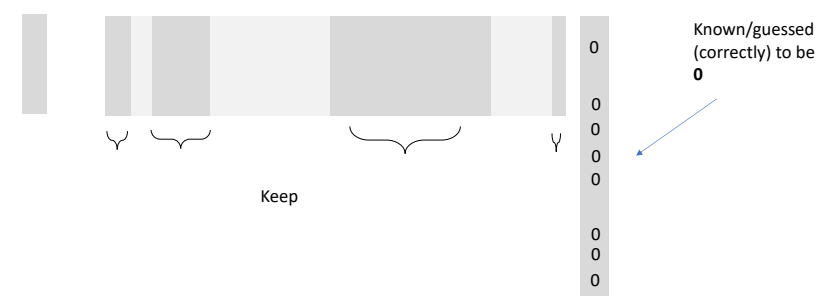
If for instance, many of the coefficients were 0 or close, we would get some useful **dimensionality reduction**

95

Option 2: Pruning of directions – by guess or by estimates

$$\mathbf{y} = \mathbf{F}\alpha$$

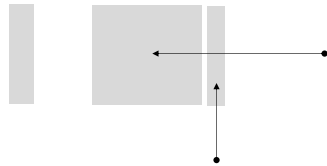

96

$$\mathbf{y} = \mathbf{F}\alpha$$


Known/guessed (correctly) to be 0

97

$$\mathbf{y} = \mathbf{F}\boldsymbol{\alpha}$$



Note: an L1 or L0-norm approach
Would get us close(r) to this – even if the 0-
or close-to-0 directions were not known!

We almost have to hope for something like this – to get down to something **managable!**
[Hidden assumption: our scene is **sparse** and/or can be modelled as something **sparse** + something **known**]

98

Option 3: A guess on a stronger signal from one direction

$$\hat{\boldsymbol{\alpha}} = [\mathbf{F}^T \mathbf{C} \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{C} \mathbf{y}$$

$$\hat{\alpha}_k = [\mathbf{F}^T \mathbf{C} \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{C} \mathbf{y}$$

Single pair
(=one direction)

Single vector pair
(="weights")

99

$$\hat{\boldsymbol{\alpha}} = [\mathbf{F}^T \mathbf{C} \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{C} \mathbf{y}$$

$$\hat{\alpha}_k = [\mathbf{F}^T \mathbf{C} \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{C} \mathbf{y}$$

Single pair
(=one direction)

Single vector pair
(="weights")

100

$$\hat{\boldsymbol{\alpha}} = [\mathbf{F}^T \mathbf{C} \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{C} \mathbf{y}$$

$$\hat{\alpha}_k = [\mathbf{F}^T \mathbf{C} \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{C} \mathbf{y}$$

Single pair
(=one direction)

Single vector pair
(="weights")

Minimum Variance Distortionless Receiver (MVDR)
or **Capon Beamformer**

$$\min_{\mathbf{w}_k} E(\|\mathbf{w}_k^T \mathbf{y} - \alpha_k\|_2^2) \text{ subject to } \mathbf{w}_k^T \mathbf{f}_k = 1$$

Sequential (per-direction) computation of
inverse matrix – subject to "unbiased
estimation" in the direction chosen

101

$$\hat{\alpha} = [F^T C F]^{-1} F^T C y$$

$$\hat{\alpha}_k = [F_k^T C F_k]^{-1} F_k^T C y$$

Single pair (=one direction)

Single vector pair (= "weights")

1) Bayesian interpretation (after looking at Lagrange multiplications): reflects a **strong belief** that there is a **"dominant"** signal of interest in the **chosen** direction

2) Prevents "damping" of target source estimate when there are many zero-like neighbors

3) This is **still** a physics-inspired regularization scheme

$\min_{w_k} E(\|w_k^T y - \alpha_k\|_2^2) \text{ subject to } w_k^T f_k = 1$

Sequential (per-direction) computation of inverse matrix – subject to "unbiased estimation" in the direction chosen

102

Method	Transient blow-up issue	Utilizes sparsity in space [if present in scene]	Utilizes learning (of covariance matrix) to improve estimates	Physics-mimicry: prefers physical direction	Facilitates connection modelling between frequencies	Facilitates utilization of sparse-time modelling
Direct inversion (A*) on original signal x	No	No	No	No	Yes	Yes
Inversion in [Fourier] subspace	Yes	No	No	No	Yes	Yes
Delay-and-sum (A*) on original signal x	No	No	No	Yes	Yes	No
Least squares per frequency	Yes	No	No	No	No	No
Least squares per frequency with knowledge of E(xx')	Yes(?)	Yes	Yes	No (perhaps)	No	No
Minimum Variance Estimation (MVDR) per frequency	?	Yes	Yes	Somewhat	No	No
L0/L1-norm minimization per frequency	?	Yes	[How?]	probably moderately	No	No

103

Method	Transient blow-up issue	Utilizes sparsity in space [if present in scene]	Utilizes learning (of covariance matrix) to improve estimates	Physics-mimicry: prefers physical direction	Facilitates connection modelling between frequencies	Facilitates utilization of sparse-time modelling
Direct inversion (A*) on original signal x	No	No	No	No	Yes	Yes
Inversion in [Fourier] subspace with E(xx')	Yes	Yes	Yes	No	Yes	Yes
Delay-and-sum (A*) on original signal x	No	No	No	Yes	Yes	No
Least squares per frequency	Yes	No	No	No	No	No
Least squares per frequency with knowledge of E(xx')	Yes(?)	Yes	Yes	No (perhaps)	No	No
Minimum Variance Estimation (MVDR) per frequency	?	Yes	Yes	Somewhat	No	No
L0/L1-norm minimization per frequency	?	Yes	[How?]	probably moderately	No	No
L0/L1-norm on subspace or original	?	Yes	[How?]	probably moderately	Yes	Yes

Method	Transient blow-up issue	Utilizes sparsity in space [if present in scene]	Utilizes learning (of covariance matrix) to improve estimates	Physics-mimicry: prefers physical direction	Facilitates connection modelling between frequencies	Facilitates utilization of sparse-time modelling
Direct inversion (A*) on original signal x	No	No	No	No	Yes	Yes
Inversion in [Fourier] subspace with E(xx')	Yes	Yes	Yes	No	Yes	Yes
Delay-and-sum (A*) on original signal x	No	No	No	Yes	Yes	No
Least squares per frequency	Yes	No	No	No	No	No
Least squares per frequency with knowledge of E(xx')	Yes(?)	Yes	Yes	No (perhaps)	No	No
Minimum Variance Estimation (MVDR) per frequency	?	Yes	Yes	Somewhat	No	No
L0/L1-norm minimization per frequency	?	Yes	[How?]	probably moderately	No	No
L0/L1-norm on subspace or original	?	Yes	[How?]	probably moderately	Yes	Yes

Method	Transient blow-up issue	Utilizes sparsity in space [if present in scene]	Utilizes learning (of covariance matrix) to improve estimates	Physics-mimicry: prefers physical direction	Facilitates connection between frequencies	Facilitates utilization of sparse-time modelling
Direct inversion ($A^\dagger$) on original signal x	No	No	No	No	Yes	Yes
Inversion in [Fourier] subspace with $E(xx^T)$	Yes	Yes	Yes	No	Yes	Yes
Delay-and-sum ($A^\dagger$) on original signal x	No	No	Use a sparse wavelet-basis for x -signals to realise sparse modelling – or information-theoretic modifications		Yes	No
Least squares per frequency	Yes	No			No	No
Least squares per frequency with knowledge of $E(xx^T)$	Yes(?)	Yes			No	No
Minimum Variance Estimation (MVDRE) per frequency	?	Yes	Yes	Somewhat	No	No
L0/L1-norm minimization per frequency	?	Yes	[How?]	probably moderately	No	No
L0/L1-norm on subspace or original	?	Yes	[How?]	probably moderately	Yes	Yes

A statistical approach

Forget about fourier bases, transients and the likes

107

$$y = Ax \Rightarrow \hat{x} = A^\dagger y$$

Simple approach, no utilization of additional knowledge such as prior signal information or signal smoothness

108

$$y = Ax \Rightarrow \hat{x} = A^\dagger y$$

Simple approach, no utilization of additional knowledge such as prior signal information or signal smoothness

Next level: subspace approach – include knowledge about x , including (known) smoothness of signals:

$$\begin{aligned}
 y &= Ax \\
 x_i &= B_i \alpha_i \Rightarrow x = B\alpha, \quad B = \text{diag}\{B_1, B_2, \dots, B_Q\}, \quad \alpha = [\alpha_1^T \alpha_2^T, \dots, \alpha_Q^T]^T \\
 y &= AB\alpha \\
 \hat{\alpha} &= [AB]^\dagger y \\
 (\hat{x} &= B\hat{\alpha})
 \end{aligned}$$

109

What happens if there is a small error in the modelling of $\mathbf{x}$?:

$$\mathbf{x}_i = \mathbf{B}_i \alpha_i + \mathbf{z}_i \Rightarrow \mathbf{x} = \mathbf{B} \alpha + \mathbf{z}$$

$$\mathbf{y} = \mathbf{A} \mathbf{x} = \mathbf{A} [\mathbf{B} \alpha + \mathbf{z}]$$

where $\mathbf{z}^T \mathbf{B} = 0$ and then we progress naively as if there were no issue.

$$\begin{aligned} \hat{\alpha} &= [\mathbf{A} \mathbf{B}]^\dagger \mathbf{y} \\ \hat{\alpha} &= [\mathbf{A} \mathbf{B}]^\dagger \{ \mathbf{A} [\mathbf{B} \alpha + \mathbf{z}] \} = \underbrace{[\mathbf{A} \mathbf{B}]^\dagger [\mathbf{A} \mathbf{B}]}_{\approx \mathbf{I}} \alpha + [\mathbf{A} \mathbf{B}]^\dagger \mathbf{z} \end{aligned}$$

The term $[\mathbf{A} \mathbf{B}]^\dagger \mathbf{z}$ will explode. We are working with a (pseudo-)inverse matrix, and $\mathbf{z}$ may have components in the directions corresponding to small singular values...

110

What happens if there is a small error in the modelling of $\mathbf{x}$?:

$$\mathbf{x}_i = \mathbf{B}_i \alpha_i + \mathbf{z}_i \Rightarrow \mathbf{x} = \mathbf{B} \alpha + \mathbf{z}$$

$$\mathbf{y} = \mathbf{A} \mathbf{x} = \mathbf{A} [\mathbf{B} \alpha + \mathbf{z}]$$

where $\mathbf{z}^T \mathbf{B} = 0$ and then we progress naively as if there were no issue.

$$\hat{\alpha} = [\mathbf{A} \mathbf{B}]^\dagger \mathbf{y}$$

Note that the error does not really have to be in $\mathbf{x}$, it could equally well be on the sensor side, without us being able to notice:

$$\mathbf{y} = \mathbf{A} \mathbf{x} = \mathbf{A} [\underbrace{\mathbf{B} \alpha}_{\mathbf{x}_0} + \mathbf{z}] = \mathbf{A} \mathbf{x}_0 + \underbrace{\mathbf{A} \mathbf{z}}_{\mathbf{n}} = \mathbf{A} \mathbf{x}_0 + \mathbf{n}$$

Just a little bit of sensor noise can ruin (or help ruin) the subspace approach!

What happens if there is a small error in the modelling of $\mathbf{x}$?:

$$\mathbf{x}_i = \mathbf{B}_i \alpha_i + \mathbf{z}_i \Rightarrow \mathbf{x} = \mathbf{B} \alpha + \mathbf{z}$$

$$\mathbf{y} = \mathbf{A} \mathbf{x} = \mathbf{A} [\mathbf{B} \alpha + \mathbf{z}]$$

where $\mathbf{z}^T \mathbf{B} = 0$ and then we progress naively as if there were no issue.

Note also that using $\mathbf{A}^T$ is much more robust than using $[\mathbf{A} \mathbf{B}]^\dagger$:

$$\tilde{\alpha} = \mathbf{A}^T \mathbf{y} = \mathbf{A}^T [\mathbf{A} \mathbf{B} \alpha + \mathbf{z}] = \underbrace{\mathbf{A}^T \mathbf{A}}_{\approx \mathbf{I}} \underbrace{\mathbf{B} \alpha}_{=\mathbf{x}} + \mathbf{A}^T \mathbf{z}$$

because the matrix $\mathbf{A}$ typically is "sort of orthogonal" and with some really small eigenvalues (that will magnify upon inversion). We can also add a weighing to $\mathbf{D}$ to the matrix $\mathbf{A}$ without this changing the scenario much.

Robustness may be one of the reasons traditional Delay-and-sum approaches are still in extensive use

Lets try to fix this without relying on the transpose:

$$\mathbf{x}_i = \mathbf{B}_i \alpha_i + \mathbf{U}_i \gamma_i \Rightarrow \mathbf{x} = \mathbf{B} \alpha + \mathbf{U} \gamma$$

where $R(\mathbf{B}_i) \cup R(\mathbf{U}_i) = R^{P_i}$, i.e. together they span the full space. This gives the relation

$$\mathbf{y} = \mathbf{A} \mathbf{x} = \mathbf{A} [\mathbf{B} \mathbf{U}] \begin{bmatrix} \alpha \\ \gamma \end{bmatrix}$$

and then we try to solved for $[\alpha^T \gamma^T]^T$.

113

Lets try to fix this without relying on the transpose:

$$\mathbf{x}_i = \mathbf{B}_i \alpha_i + \mathbf{U}_i \gamma_i \Rightarrow \mathbf{x} = \mathbf{B} \alpha + \mathbf{U} \gamma$$

where $R(\mathbf{B}_i) \cup R(\mathbf{U}_i) = R^{P_i}$, i.e. together they span the full space. This gives the relation

$$\mathbf{y} = \mathbf{A} \mathbf{x} = \mathbf{A}[\mathbf{B} \mathbf{U}] \begin{bmatrix} \alpha \\ \gamma \end{bmatrix}$$

and then we try to solved for $[\alpha^T \gamma^T]^T$. But then we are really just back to modelling $\mathbf{x}$ - whatever it may be. Unless:

$$\mathbf{y} = \mathbf{A} \mathbf{x} = \mathbf{A}[\mathbf{B} \mathbf{U}] \begin{bmatrix} \alpha \\ \gamma \end{bmatrix}$$

we *know* that $\gamma = 0$ or $\gamma \approx 0$. Is there a way to *force* γ in this direction?

114

Recall our starting point (for a sensor i):

$$\mathbf{y}_i = \sum_{j=1}^N \mathbf{A}_{ij} \mathbf{x}_j$$

Now, $\mathbf{A}_{ij}$ is a banded matrix, and there also a *filter*. Therefore we can rewrite, in time-domain,

$$y_i(t) = \sum_{j=1}^N a_{ij}(t) * x_j(t)$$

Is there a way to *change* $x_j(t)$ so that its vector parallell $\mathbf{x}$ resides in a subspace (and not all of R^{P_i})?

115

The answer is yes, by applying a filter $f(t)$ to $y_i(t)$:

$$\begin{aligned} \underbrace{f(t) * y_i(t)}_{\tilde{y}_i(t)} &= f(t) * \sum_{j=1}^N a_{ij}(t) * x_j(t) \\ &= \sum_{j=1}^N f(t) * a_{ij}(t) * x_j(t) \\ &= \sum_{j=1}^N a_{ij}(t) * \underbrace{f(t) * x_j(t)}_{\tilde{x}_j(t)} \end{aligned}$$

where we have utilized that filters (or convolutions) are *commutative*.

We are now working on a *different version* of $x_j(t)$ but that may be quite ok: if we known i.e. that $x_j(t)$ is smooth, there is no harm in applying a smoothing filter $f(t)$ to $y(t)$, and indirectly, to $x_j(t)$.

116

Why is this useful? Going back to vector notation we note that $\tilde{x}_j = f(t) * x_j(t)$ is replaced by $\tilde{\mathbf{x}}_i = \mathbf{F} \mathbf{x}_i$. Let us assume that we know or guess on the distribution of the relative coefficients of the vector $\mathbf{x}_j$:

$$\begin{aligned} E(\mathbf{x}_j \mathbf{x}_j^T) &= \gamma \mathbf{I} \\ E(\mathbf{x}_j \mathbf{x}_j^T) &= \mathbf{C} \end{aligned}$$

which correspond to white (uncorrelated) time-samples, or, a more general correlation structure held in $\mathbf{C}$. We then have

$$E(\tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_j^T) = E(\mathbf{F} \mathbf{x}_j \mathbf{x}_j^T \mathbf{F}^T) = \mathbf{F} E(\mathbf{x}_j \mathbf{x}_j^T) \mathbf{F}^T = \mathbf{F} \mathbf{C} \mathbf{F}^T$$

117

Why is this useful? Going back to vector notation we note that $\tilde{x}_j = f(t) * x_j(t)$ is replaced by $\tilde{\mathbf{x}}_j = \mathbf{F}\mathbf{x}_j$. Let us assume that we know or guess on the distribution of the relative coefficients of the vector $\mathbf{x}_j$:

$$\begin{aligned} E(\mathbf{x}_j \mathbf{x}_j^T) &= \gamma \mathbf{I} \\ E(\mathbf{x}_j \mathbf{x}_j^T) &= \mathbf{C} \end{aligned}$$

which correspond to white (uncorrelated) time-samples, or, a more general correlation structure held in $\mathbf{C}$. We then have

$$E(\tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_j^T) = E(\mathbf{F} \mathbf{x}_j \mathbf{x}_j^T \mathbf{F}^T) = \mathbf{F} E(\mathbf{x}_j \mathbf{x}_j^T) \mathbf{F}^T = \mathbf{F} \mathbf{C} \mathbf{F}^T$$

Typically, the filter matrix $\mathbf{F}$ is chosen so that it reduces the dimensionality of the signal $x_j(t)$ (or: the dimensionality in its locally, time-sampled vectors). We can therefore expect that, or rather *design* $\mathbf{F}$, so that

$$\mathbf{F} \mathbf{C} \mathbf{F}^T = \mathbf{V}_0 \mathbf{S}_0^2 \mathbf{V}_0^T$$

where $\mathbf{V}_0$ is an orthogonal matrix of reduced column rank.

118

We then have our basis for $\tilde{\mathbf{x}}_j$:

$$\tilde{\mathbf{x}}_j = \mathbf{B}_j \alpha_j = \mathbf{V}_0 \alpha_j$$

i.e. $\mathbf{B}_j = \mathbf{V}_0$, and we don't have to worry about the orthogonal complement of $\mathbf{B}_j$, because its coefficient, the γ , has already been set to 0 by the filtering operation with $f(t)$.

We can also take the Bayesian approach one step further, and include the weighing-in of the distribution parameters held in $\mathbf{S}_0$.

119

We then have our basis for $\tilde{\mathbf{x}}_j$:

$$\tilde{\mathbf{x}}_j = \mathbf{B}_j \alpha_j = \mathbf{V}_0 \alpha_j$$

i.e. $\mathbf{B}_j = \mathbf{V}_0$, and we don't have to worry about the orthogonal complement of $\mathbf{B}_j$, because its coefficient, the γ , has already been set to 0 by the filtering operation with $f(t)$.

We can also take the Bayesian approach one step further, and include the weighing-in of the distribution parameters held in $\mathbf{S}_0$.

By using filtering, we have "shaved off" a subspace that made our modelling approach difficult

120

Why is this useful? Going back to vector notation we note that $\tilde{x}_j = f(t) * x_j(t)$ is replaced by $\tilde{\mathbf{x}}_j = \mathbf{F}\mathbf{x}_j$. Let us assume that we know or guess on the distribution of the relative coefficients of the vector $\mathbf{x}_j$:

$$\begin{aligned} E(\mathbf{x}_j \mathbf{x}_j^T) &= \gamma \mathbf{I} \\ E(\mathbf{x}_j \mathbf{x}_j^T) &= \mathbf{C} \end{aligned}$$

which correspond to white (uncorrelated) time-samples, or, a more general correlation structure held in $\mathbf{C}$. We then have

$$E(\tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_j^T) = E(\mathbf{F} \mathbf{x}_j \mathbf{x}_j^T \mathbf{F}^T) = \mathbf{F} E(\mathbf{x}_j \mathbf{x}_j^T) \mathbf{F}^T = \mathbf{F} \mathbf{C} \mathbf{F}^T$$

Typically, the filter matrix $\mathbf{F}$ is chosen so that it reduces the dimensionality of the signal $x_j(t)$ (or: the dimensionality in its locally, time-sampled vectors). We can therefore expect that, or rather *design* $\mathbf{F}$, so that

$$\mathbf{F} \mathbf{C} \mathbf{F}^T = \mathbf{V}_0 \mathbf{S}_0^2 \mathbf{V}_0^T$$

where $\mathbf{V}_0$ is an orthogonal matrix of reduced column rank.

121

Why is this useful? Going back to vector notation we note that $\tilde{x}_j = f(t) * x_j(t)$ is replaced by $\tilde{\mathbf{x}}_j = \mathbf{F}\mathbf{x}_j$. Let us assume that we know or guess on the distribution of the relative coefficients of the vector $\mathbf{x}_j$:

$$\begin{aligned} E(\mathbf{x}_j \mathbf{x}_j^T) &= \gamma \mathbf{I} \\ E(\mathbf{x}_j \mathbf{x}_j^T) &= \mathbf{C} \end{aligned}$$

which correspond to white (uncorrelated) time-samples, or, a more general correlation structure held in $\mathbf{C}$. We then have

$$E(\tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_j^T) = E(\mathbf{F} \mathbf{x}_j \mathbf{x}_j^T \mathbf{F}^T) = \mathbf{F} E(\mathbf{x}_j \mathbf{x}_j^T) \mathbf{F}^T = \mathbf{F} \mathbf{C} \mathbf{F}^T$$

Typically, the filter matrix $\mathbf{F}$ is chosen so that it reduces the dimensionality of the signal $x_j(t)$ (or: the dimensionality in its locally, time-sampled vectors). We can therefore expect that, or rather *design* $\mathbf{F}$, so that

$$\mathbf{F} \mathbf{C} \mathbf{F}^T = \mathbf{V}_0 \mathbf{S}_0^2 \mathbf{V}_0^T$$

In speech recognition applications, this distribution is at least partially learnable – via $\mathbf{y}$ – by utilizing "silent" or single-speaker situations, or by using more complex approaches utilizing the **NON-STATIONARITY** of the sources – ICA, BSS,...

122

Why is this useful? Going back to vector notation we note that $\tilde{x}_j = f(t) * x_j(t)$ is replaced by $\tilde{\mathbf{x}}_j = \mathbf{F}\mathbf{x}_j$. Let us assume that we know or guess on the distribution of the relative coefficients of the vector $\mathbf{x}_j$:

$$\begin{aligned} E(\mathbf{x}_j \mathbf{x}_j^T) &= \gamma \mathbf{I} \\ E(\mathbf{x}_j \mathbf{x}_j^T) &= \mathbf{C} \end{aligned}$$

which correspond to white (uncorrelated) time-samples, or, a more general correlation structure held in $\mathbf{C}$. We then have

$$E(\tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_j^T) = E(\mathbf{F} \mathbf{x}_j \mathbf{x}_j^T \mathbf{F}^T) = \mathbf{F} E(\mathbf{x}_j \mathbf{x}_j^T) \mathbf{F}^T = \mathbf{F} \mathbf{C} \mathbf{F}^T$$

Typically, the filter matrix $\mathbf{F}$ is chosen so that it reduces the dimensionality of the signal $x_j(t)$ (or: the dimensionality in its locally, time-sampled vectors). We can therefore expect that, or rather *design* $\mathbf{F}$, so that

If nothing else is known or estimated, we can still define the filter so that we "zoom in" on a certain sub-band of interest (or utilize sampling system knowledge to define smoothness levels) – and we thereby **enable inversion without transient issues**.

In speech recognition applications, this distribution is at least partially learnable – via $\mathbf{y}$ – by utilizing "silent" or single-speaker situations, or by using more complex approaches utilizing the **NON-STATIONARITY** of the sources – ICA, BSS,...

123

Why is this useful? Going back to vector notation we note that $\tilde{x}_j = f(t) * x_j(t)$ is replaced by $\tilde{\mathbf{x}}_j = \mathbf{F}\mathbf{x}_j$. Let us assume that we know or guess on the distribution of the relative coefficients of the vector $\mathbf{x}_j$:

$$\begin{aligned} E(\mathbf{x}_j \mathbf{x}_j^T) &= \gamma \mathbf{I} \\ E(\mathbf{x}_j \mathbf{x}_j^T) &= \mathbf{C} \end{aligned}$$

which correspond to white (uncorrelated) time-samples, or, a more general correlation structure held in $\mathbf{C}$. We then have

$$E(\tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_j^T) = E(\mathbf{F} \mathbf{x}_j \mathbf{x}_j^T \mathbf{F}^T) = \mathbf{F} E(\mathbf{x}_j \mathbf{x}_j^T) \mathbf{F}^T = \mathbf{F} \mathbf{C} \mathbf{F}^T$$

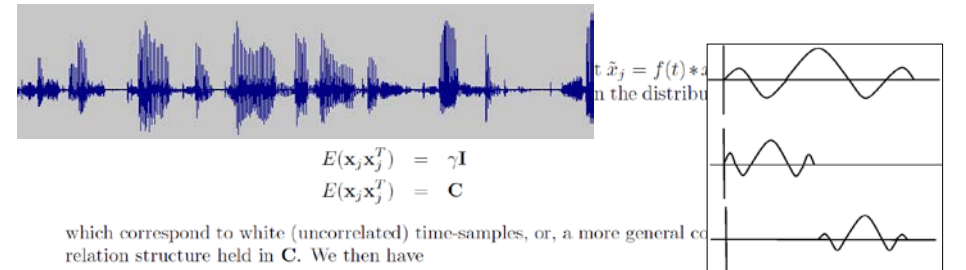
Typically, the filter matrix $\mathbf{F}$ is chosen so that it reduces the dimensionality of the signal $x_j(t)$ (or: the dimensionality in its locally, time-sampled vectors). We can therefore expect that, or rather *design* $\mathbf{F}$, so that

$$\mathbf{F} \mathbf{C} \mathbf{F}^T = \mathbf{V}_0 \mathbf{S}_0^2 \mathbf{V}_0^T$$

where $\mathbf{V}_0$ is an orthogonal matrix of reduced column rank.

As long as we span the subspaces $R(\mathbf{F})$ and $R(\mathbf{F}^T)$ we can pick any set of basis vectors for those spaces in the modelling to follow

124



which correspond to white (uncorrelated) time-samples, or, a more general correlation structure held in $\mathbf{C}$. We then have

$$E(\tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_j^T) = E(\mathbf{F} \mathbf{x}_j \mathbf{x}_j^T \mathbf{F}^T) = \mathbf{F} E(\mathbf{x}_j \mathbf{x}_j^T) \mathbf{F}^T = \mathbf{F} \mathbf{C} \mathbf{F}^T$$

Typically, the filter matrix $\mathbf{F}$ is chosen so that it reduces the dimensionality of the signal $x_j(t)$ (or: the dimensionality in its locally, time-sampled vectors). We can therefore expect that, or rather *design* $\mathbf{F}$, so that

$$\mathbf{F} \mathbf{C} \mathbf{F}^T = \mathbf{V}_0 \mathbf{S}_0^2 \mathbf{V}_0^T$$

where $\mathbf{V}_0$ is an orthogonal matrix of reduced column rank.

As long as we span the subspaces $R(\mathbf{F})$ and $R(\mathbf{F}^T)$ we can pick any set of basis vectors for those spaces in the modelling to follow

If we believe our signals to be "sparse in time" – i.e. utterances followed by silent periods – we can model that!

125

LOCAL-ICA: INDEPENDENT COMPONENT ANALYSIS FOR SOURCES WITH NON-STATIONARY AMPLITUDES

Tobias Dahl, Ole C. Lingjærde, Nils Christophersen

Department of Informatics,
University of Oslo, P.O. Box 1080 Blindern
N-0316 Oslo, Norway

Torbjørn Ekman

Unik - University Graduate Center
P.O. Box 70
N-2007 Kjeller, Norway

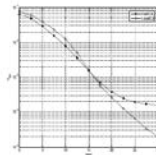
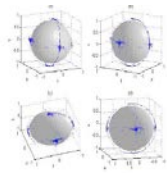
ABSTRACT

We propose a novel and conceptually simple method for blind source separation of linear mixtures. Sources are required to have independently fluctuating non-stationary amplitudes, where the latter are defined relative to a moving time frame. Utilizing the local temporal structure of the mixtures, the proposed method estimates the mixing matrix and the sources by repeated use of the singular value decomposition and k-means clustering. Performance is comparable to FastICA [2] on some typical test problems. Unlike the ICA algorithms that are based on higher order statistics, LocalICA is based only on second order statistics. LocalICA thus has the potential for separating mixtures of colored Gaussian sources.

that our estimator performs well in the presence of moderate levels of noise and deteriorates gracefully with decreasing signal-to-noise ratio, see below. In order to present the main ideas as succinctly as possible and to facilitate geometric intuitions, we focus in the following on the noise-free model in (1).

Suppose we observe the $x_j(t)$ for $t = 1, \dots, N$. Let $\mathbf{S}$ be the $n \times N$ -matrix with elements $s_{ij} = x_{ij}(t)$, $\mathbf{X}$ the $n \times N$ -matrix with elements $x_{ij}(t)$, and $\mathbf{H}$ the $n \times n$ -matrix with elements h_{ij} . Sources are assumed to be uncorrelated and with identical empirical variances, i.e. we have $\mathbf{SS}^T = \eta \mathbf{I}$. Note that the latter equation is equivalent to $\mathbf{SS}^T = \mathbf{I}$ where $\mathbf{S} = \eta^{-1/2} \mathbf{S}$; hence in the following we assume with no loss of generality that $\eta = 1$ and

$$\mathbf{SS}^T = \mathbf{I} \quad (2)$$



$$E(\mathbf{xx}^T) = \alpha_1 \mathbf{f}_1 \mathbf{f}_1^T + \alpha_2 \mathbf{f}_2 \mathbf{f}_2^T$$

Average or "static" covariance estimation (assumes stationarity)

$$E(\mathbf{xx}^T)(t) = \alpha_1(t) \mathbf{f}_1 \mathbf{f}_1^T + \alpha_2(t) \mathbf{f}_2 \mathbf{f}_2^T$$

Dynamic covariance estimation, utilizing amplitude fluctuations (=non-stationarity)

Can make much better filters when parts are computed!

$$\mathbf{f}_1 \mathbf{f}_1^T \quad \mathbf{f}_2 \mathbf{f}_2^T$$

126

Model summary:

$$\begin{aligned} \mathbf{y} &= \mathbf{A}\mathbf{x} + \mathbf{n} \xrightarrow{f} \tilde{\mathbf{y}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{n} \\ \tilde{\mathbf{x}}_j &= \mathbf{B}_j \alpha_j \\ \tilde{\mathbf{x}} &= \mathbf{B}\alpha, \quad \mathbf{B} = \text{diag}\{\mathbf{B}_1, \dots, \mathbf{B}_Q\}, \alpha = [\alpha_1^T \dots, \alpha_Q^T]^T \\ \tilde{\mathbf{y}} &= \mathbf{A}\tilde{\mathbf{x}} + \mathbf{n} = \mathbf{A}\mathbf{B}\alpha + \mathbf{n} \end{aligned}$$

If we now use the (observed, estimated, guessed,...) covariance matrices

$$E(\tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_j^T) = \mathbf{M} = \mathbf{V}_j \mathbf{S}_j \mathbf{V}_j^T$$

where the right hand side is reduced-rank decomposition, and further make a guess that

$$E(\alpha_j \alpha_j^T) = \mathbf{S}_j^2$$

then we have the correct modelling of this expectation, because:

$$\begin{aligned} E(\tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_j^T) &= E(\mathbf{B}_j \alpha_j \alpha_j^T \mathbf{B}_j^T) = E(\mathbf{V}_j \alpha_j \alpha_j^T \mathbf{V}_j^T) \\ &= \mathbf{V}_j E(\alpha_j \alpha_j^T) \mathbf{V}_j^T = \mathbf{V}_j \mathbf{S}_j^2 \mathbf{V}_j^T = \mathbf{M} \end{aligned}$$

127

We now have a complete, parameterized and "transient-issue-free" model:

$$\tilde{\mathbf{y}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{n} = \underbrace{\mathbf{A}\mathbf{B}}_{\mathbf{T}} \alpha + \mathbf{n} = \mathbf{T}\alpha + \mathbf{n}$$

with distribution expectations:

$$E(\alpha \alpha^T) = \begin{pmatrix} \mathbf{S}_1 & \dots & 0 & 0 \\ 0 & \mathbf{S}_2 & 0 & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{S}_Q \end{pmatrix} \quad E(\mathbf{nn}^T) = \sigma^2 \mathbf{I}$$

We can plug this straight into Bayes formula, from the first sections.

128

We now have a complete, parameterized and "transient-issue-free" model:

$$\tilde{\mathbf{y}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{n} = \underbrace{\mathbf{A}\mathbf{B}}_{\mathbf{T}} \alpha + \mathbf{n} = \mathbf{T}\alpha + \mathbf{n}$$

with distribution expectations:

$$\begin{pmatrix} \mathbf{S}_1 & \dots & 0 & 0 \end{pmatrix}$$

Bayes:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \quad \begin{aligned} E(\mathbf{nn}^T) &= \mathbf{R} \\ E(\mathbf{xx}^T) &= \mathbf{C} \end{aligned} \Rightarrow \begin{aligned} \hat{\mathbf{x}} &= (\mathbf{A}^T \mathbf{C} \mathbf{A} + \mathbf{R})^{-1} \mathbf{A}^T \mathbf{C} \mathbf{y} \\ E(\hat{\mathbf{x}} \hat{\mathbf{x}}^T) &= \mathbf{C} - \mathbf{C} \mathbf{A} (\mathbf{A}^T \mathbf{C} \mathbf{A} + \mathbf{R})^{-1} \mathbf{A}^T \mathbf{C} \end{aligned} \quad \text{)} = \sigma^2 \mathbf{I}$$

$$E(\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^T = \mathbf{E} = \dots$$

$$E(\hat{x}_i - x_i) = E_{ii}$$

first sections.

129

We now have a complete, parameterized and "transient-issue-free" model:

$$\tilde{\mathbf{y}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{n} = \underbrace{\mathbf{A}\mathbf{B}}_{\mathbf{T}}\boldsymbol{\alpha} + \mathbf{n} = \mathbf{T}\boldsymbol{\alpha} + \mathbf{n}$$

with distribution expectations:

$$\begin{pmatrix} \mathbf{S}_1 & \dots & 0 & 0 \end{pmatrix}$$

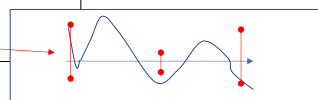
Bayes:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \quad \begin{matrix} E(\mathbf{n}\mathbf{n}^T) = \mathbf{R} \\ E(\mathbf{x}\mathbf{x}^T) = \mathbf{C} \end{matrix} \Rightarrow \begin{matrix} \hat{\mathbf{x}} \\ E(\hat{\mathbf{x}}\hat{\mathbf{x}}^T) \end{matrix} = \begin{matrix} (\mathbf{A}^T\mathbf{C}\mathbf{A} + \mathbf{R})^{-1}\mathbf{A}^T\mathbf{C}\mathbf{y} \\ \mathbf{C} - \mathbf{C}\mathbf{A}(\mathbf{A}^T\mathbf{C}\mathbf{A} + \mathbf{R})^{-1}\mathbf{A}^T\mathbf{C} \end{matrix} \quad \mathbf{I}) = \sigma^2\mathbf{I}$$

$$E(\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^T = \mathbf{E} = \dots$$

$$E(\hat{x}_i - x_i) = E_{ii}$$

Gives you an error (~confidence) of the signal coefficients estimated!



130

Summary on statistical estimation

- We can use a (multiband, broadband) basis with filters
- It will take care of the "transient"
 - Utilizes commutativity of convolution operation
- Can "model in" probability models (covariances) of speech signals
 - Can extend to dynamic models or mixture models of covariance matrices
 - Can utilize fluctuations in speech to tease out individual components
- Can "model in" expected shape of speech (wavelets, other bases).

Multi-narrowband is not the same as broadband beam forming

132

Method	Transient blow-up issue	Utilizes sparsity in space [if present in scene]	Utilizes learning (of covariance matrix) to improve estimates	Physics-mimicry: prefers physical direction	Facilitates connection modelling between frequencies	Facilitates utilization of sparse-time modelling
Direct inversion ($\mathbf{A}^*$) on original signal $\mathbf{x}$	No	No	No	No	Yes	Yes
Inversion in [Fourier] subspace	Yes	No	No	No	Yes	Yes
Delay-and-sum ($\mathbf{A}^*$) on original signal $\mathbf{x}$	No	No	No	Yes	Yes	No
Least squares per frequency	Yes	No	No	No	No	No
Least squares per frequency with knowledge of $E(\mathbf{x}\mathbf{x}^T)$	Yes(?)	Yes	Yes	No (perhaps)	No	No
Minimum Variance Estimation (MVDR) per frequency	?	Yes	Yes	Somewhat	No	No
L0/L1-norm minimization per frequency	?	Yes	[How?]	probably moderately	No	No
Bayes subspace & $E(\mathbf{x}\mathbf{x}^T)$	Yes	Yes	Yes	Yes	Yes	Yes – with Wavelet basis & mods

The (new) context: infinite compute!

- There used to be a time that everything had to be real-time on ARM
 - With speech, most crunching is already running on the cloud
 - Wouldn't be a big leap to make high-performance array algorithms there too
- Speech recognition is a game with "multiple lives"
 - If one set of parameters doesn't give you % recognition – try another!
 - You can also use % recognition for validation and 'steering'
- The same applies to other applications – like medical imaging
 - Not long ago, beam forming was in hardware
 - There are still realtime requirements to guide the probe
 - However, if image analysis is **also to be done with AI**, you can post-process your stored signals to get images with much higher quality – potentially.

133

Robert K. Merton

- The fallacy of the latest word
- The fallacy of the immediate interest

