

Polynomial Matrices and Decompositions; Tutorial Examples Including Beamforming and Source Localisation

Stephan Weiss

Centre for Signal & Image Processing
Department of Electronic & Electrical Engineering
University of Strathclyde, Glasgow, Scotland, UK

UDRC Summer School, University of Surrey, 29 June 2017

With many thanks to:

J.G. McWhirter, I.K. Proudler, J. Corr, Z. Wang, and F.K. Coutts

This work was supported by QinetiQ, the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/K014307/1 and the MOD University Defence Research Collaboration in Signal Processing.

Presentation Overview

1. Overview

Part I: Polynomial Matrices and Decompositions

2. Polynomial matrices and basic operations
 - 2.1 occurrence: MIMO systems, filter banks, space-time covariance
 - 2.2 basic properties and operations
3. Polynomial eigenvalue decomposition (PEVD)
4. Iterative PEVD algorithms
 - 4.1 sequential best rotation (SBR2)
 - 4.2 sequential matrix diagonalisation (SMD)
5. PEVD Matlab toolbox

Part II: Beamforming & Source Separation Applications

6. Broadband MIMO decoupling
7. Broadband angle of arrival estimation
 - 7.1 broadband / polynomial subspace decomposition
 - 7.2 polynomial MUSIC
8. Broadband beamforming
9. Summary and materials

What is a Polynomial Matrix?

- ▶ A polynomial matrix is a polynomial with matrix-valued coefficients, e.g.:

$$\mathbf{A}(z) = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} z^{-1} + \begin{bmatrix} -1 & 2 \\ 1 & -1 \end{bmatrix} z^{-2} \quad (1)$$

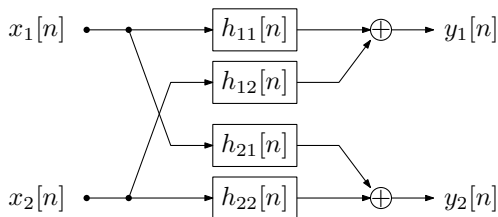
- ▶ a polynomial matrix can equivalently be understood a matrix with polynomial entries, i.e.

$$\mathbf{A}(z) = \begin{bmatrix} 1 + z^{-1} - z^{-2} & -1 + z^{-1} + 2z^{-2} \\ -1 + z^{-1} + z^{-2} & 2 - z^{-1} - z^{-2} \end{bmatrix} \quad (2)$$

- ▶ polynomial matrices could also contain rational polynomials, but the notation would not be as easily interchangeable as (1) and (2).

Where Do Polynomial Matrices Arise?

- ▶ A multiple-input multiple-output (MIMO) system could be made up of a number of finite impulse response (FIR) channels:

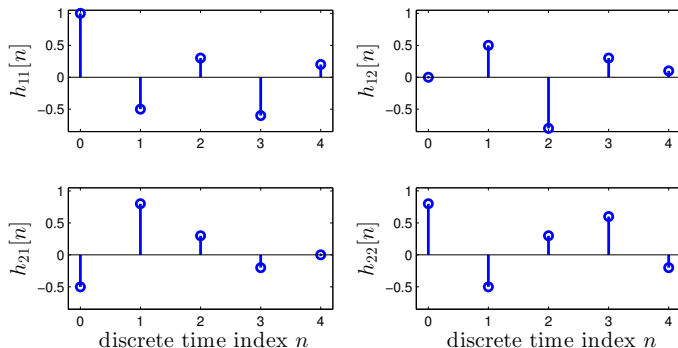


- ▶ writing this as a matrix of impulse responses:

$$\mathbf{H}[n] = \begin{bmatrix} h_{11}[n] & h_{12}[n] \\ h_{21}[n] & h_{22}[n] \end{bmatrix} \quad (3)$$

Transfer Function of a MIMO System

- ▶ Example for MIMO matrix $\mathbf{H}[n]$ of impulse responses:

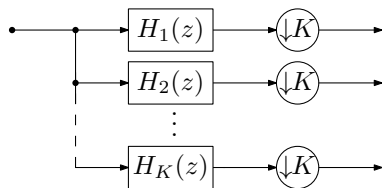


- ▶ the transfer function of this MIMO system is a polynomial matrix:

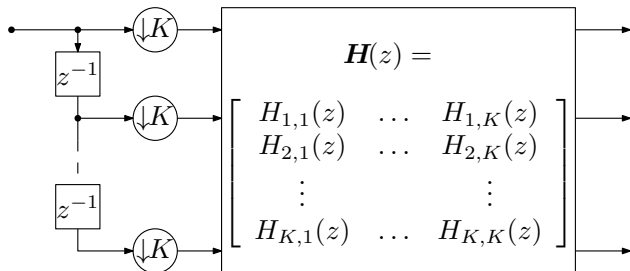
$$\mathbf{H}(z) = \sum_{n=-\infty}^{\infty} \mathbf{H}[n]z^{-n} \quad \text{or} \quad \mathbf{H}(z) \bullet \text{---} \circ \mathbf{H}[n] \quad (4)$$

Analysis Filter Bank

- ▶ Critically decimated K -channel analysis filter bank:



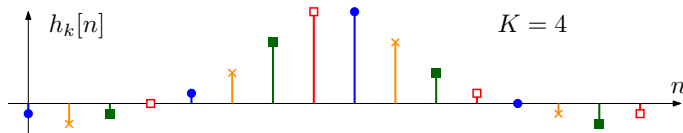
- ▶ equivalent polyphase representation:



Polyphase Analysis Matrix

- ▶ With the K -fold polyphase decomposition of the analysis filters

$$H_k(z) = \sum_{n=1}^K H_{k,n}(z^K) z^{-n+1} \quad (5)$$

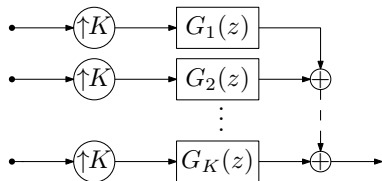


- ▶ the polyphase analysis matrix is a polynomial matrix:

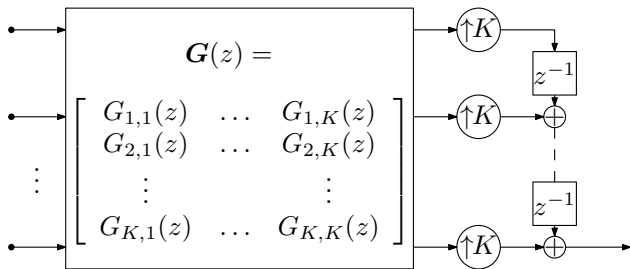
$$\mathbf{H}(z) = \begin{bmatrix} H_{1,1}(z) & H_{1,2}(z) & \dots & H_{1,K}(z) \\ H_{2,1}(z) & H_{2,2}(z) & \dots & H_{2,K}(z) \\ \vdots & \vdots & \ddots & \vdots \\ H_{K,1}(z) & H_{K,2}(z) & \dots & H_{K,K}(z) \end{bmatrix} \quad (6)$$

Synthesis Filter Bank

- ▶ Critically decimated K -channel synthesis filter bank:



- ▶ equivalent polyphase representation:



Polyphase Synthesis Matrix

- ▶ Analogous to analysis filter bank, the synthesis filters $G_k(z)$ can be split into K polyphase components, creating a polyphase synthesis matrix

$$\mathbf{G}(z) = \begin{bmatrix} G_{1,1}(z) & G_{1,2}(z) & \dots & G_{1,K}(z) \\ G_{2,1}(z) & G_{2,2}(z) & \dots & G_{2,K}(z) \\ \vdots & \vdots & \ddots & \vdots \\ G_{K,1}(z) & G_{K,2}(z) & \dots & G_{K,K}(z) \end{bmatrix} \quad (7)$$

- ▶ operating analysis and synthesis back-to-back, perfect reconstruction is achieved if

$$\mathbf{G}(z)\mathbf{H}(z) = \mathbf{I}; \quad (8)$$

- ▶ i.e. for perfect reconstruction, the polyphase analysis matrix must be invertible: $\mathbf{G}(z) = \mathbf{H}^{-1}(z)$.

Space-Time Covariance Matrix

- ▶ Measurements obtained from M sensors are collected in a vector $\mathbf{x}[n] \in \mathbb{C}^M$:

$$\mathbf{x}^T[n] = [x_1[n] \ x_2[n] \ \dots \ x_M[n]] ; \quad (9)$$

- ▶ with the expectation operator $\mathcal{E}\{\cdot\}$, the spatial correlation is captured by $\mathbf{R} = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^H[n]\}$;
- ▶ for spatial and temporal correlation, we require a space-time covariance matrix

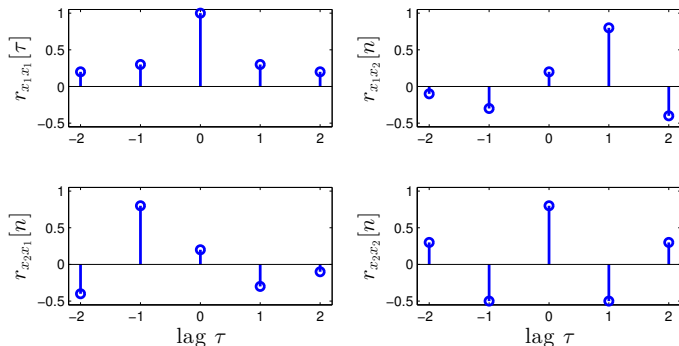
$$\mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^H[n - \tau]\} \quad (10)$$

- ▶ this space-time covariance matrix contains auto- and cross-correlation terms, e.g. for $M = 2$

$$\mathbf{R}[\tau] = \begin{bmatrix} \mathcal{E}\{x_1[n]x_1^*[n - \tau]\} & \mathcal{E}\{x_1[n]x_2^*[n - \tau]\} \\ \mathcal{E}\{x_2[n]x_1^*[n - \tau]\} & \mathcal{E}\{x_2[n]x_2^*[n - \tau]\} \end{bmatrix} \quad (11)$$

Cross-Spectral Density Matrix

- ▶ example for a space-time covariance matrix $\mathbf{R}[\tau] \in \mathbb{R}^{2 \times 2}$:



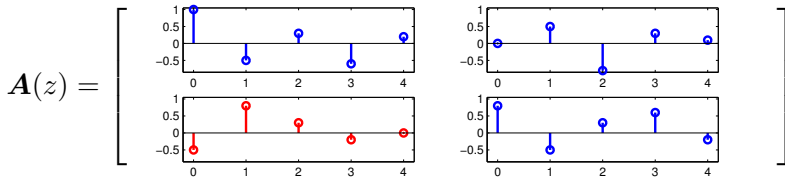
- ▶ the cross-spectral density (CSD) matrix

$$\mathbf{R}(z) \circ \bullet \mathbf{R}[\tau] \quad (12)$$

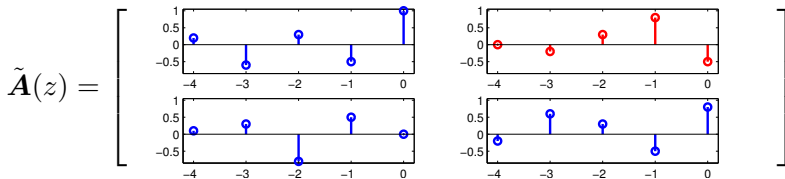
is a polynomial matrix.

ParaHermitian Operator

- ▶ A paraHermitian operation is indicated by $\{\tilde{\cdot}\}$, and compared to the Hermitian (= complex conjugate transpose) of a matrix additionally performs a time-reversal;
- ▶ example:

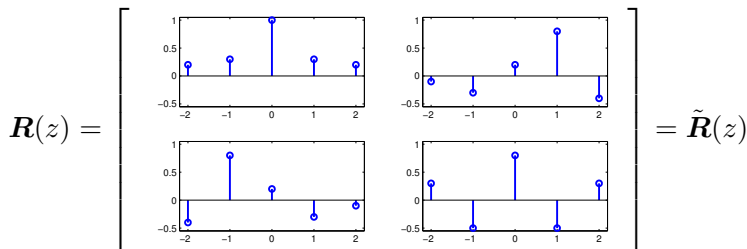


- ▶ paraHermitian $\tilde{\mathbf{A}}(z) = \mathbf{A}^H(z^{-1})$:



ParaHermitian Property

- ▶ A polynomial matrix $\mathbf{A}(z)$ is paraHermitian if $\tilde{\mathbf{A}}(z) = \mathbf{A}(z)$;
- ▶ this is an extension of the symmetric (if $\mathbf{A} \in \mathbb{R}$) or Hermitian (if $\mathbf{A} \in \mathbb{C}$) property to the polynomial case: transposition, complex conjugation and time reversal (in any order) do not alter a paraHermitian $\mathbf{A}(z)$;
- ▶ any CSD matrix is paraHermitian;
- ▶ example:



Paraunitary Matrices

- ▶ Recall that $\mathbf{A} \in \mathbb{C}$ (or $\mathbf{A} \in \mathbb{R}$) is a unitary (or orthonormal) matrix if $\mathbf{A}\mathbf{A}^H = \mathbf{A}^H\mathbf{A} = \mathbf{I}$;
- ▶ in the polynomial case, $\mathbf{A}(z)$ is paraunitary if

$$\mathbf{A}(z)\tilde{\mathbf{A}}(z) = \tilde{\mathbf{A}}(z)\mathbf{A}(z) = \mathbf{I} \quad (13)$$

- ▶ therefore, if $\mathbf{A}(z)$ is paraunitary, then the polynomial matrix inverse is simple:

$$\mathbf{A}^{-1}(z) = \tilde{\mathbf{A}}(z) \quad (14)$$

- ▶ example: polyphase analysis or synthesis matrices of perfectly reconstructing (or lossless) filter banks are usually paraunitary.

Attempt of Gaussian Elimination

- ▶ System of polynomial equations:

$$\begin{bmatrix} A_{11}(z) & A_{12}(z) \\ A_{21}(z) & A_{22}(z) \end{bmatrix} \cdot \begin{bmatrix} X_1(z) \\ X_2(z) \end{bmatrix} = \begin{bmatrix} B_1(z) \\ B_2(z) \end{bmatrix} \quad (15)$$

- ▶ modification of 2nd row:

$$\begin{bmatrix} A_{11}(z) & A_{12}(z) \\ A_{11}(z) & \frac{A_{11}(z)}{A_{21}(z)}A_{22}(z) \end{bmatrix} \cdot \begin{bmatrix} X_1(z) \\ X_2(z) \end{bmatrix} = \begin{bmatrix} B_1(z) \\ \frac{A_{11}(z)}{A_{21}(z)}B_2(z) \end{bmatrix} \quad (16)$$

- ▶ upper triangular form by subtracting 1st row from 2nd:

$$\begin{bmatrix} A_{11}(z) & A_{12}(z) \\ 0 & \frac{A_{11}(z)A_{22}(z) - A_{12}(z)A_{21}(z)}{A_{21}(z)} \end{bmatrix} \cdot \begin{bmatrix} X_1(z) \\ X_2(z) \end{bmatrix} = \begin{bmatrix} B_1(z) \\ \bar{B}_2(z) \end{bmatrix} \quad (17)$$

- ▶ penalty: we end up with rational polynomials.

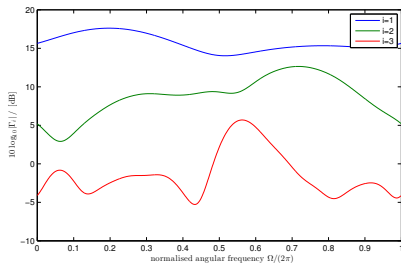
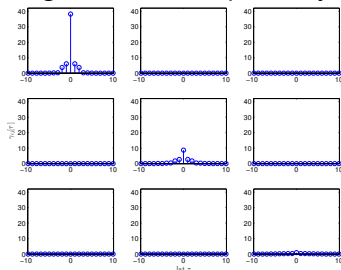
Polynomial Eigenvalue Decomposition

[McWhirter *et al.*, IEEE TSP 2007]

- ▶ Polynomial EVD of the CSD matrix

$$\mathbf{R}(z) \approx \mathbf{Q}(z) \mathbf{\Lambda}(z) \tilde{\mathbf{Q}}(z) \quad (18)$$

- ▶ with paraunitary $\mathbf{Q}(z)$, s.t. $\mathbf{Q}(z)\tilde{\mathbf{Q}}(z) = \mathbf{I}$;
- ▶ diagonalised and spectrally majorised $\mathbf{\Lambda}(z)$:



- ▶ approximation in (18) can be close with an FIR $\mathbf{Q}(z)$ of sufficiently high order [Icart & Comon 2012].

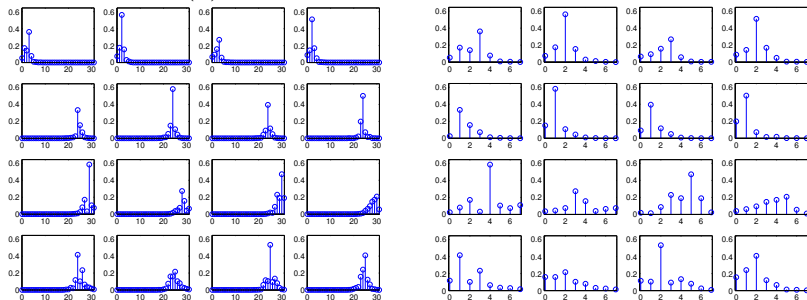
PEVD Ambiguity

[Corr *et al.*, EUSIPCO 2015]

- ▶ We believe diagonalised and spectral majorised $\Lambda(z)$ is unique;
- ▶ but there is ambiguity w.r.t. the paraunitary matrix $Q(z)$;
- ▶ set $\bar{Q}(z) = Q(z)\Gamma(z)$, with a diagonal allpass $\Gamma(z)$:

$$\begin{aligned}
 R(z) &= \bar{Q}(z)\Lambda(z)\bar{Q}(z) = Q(z)\Gamma(z)\Lambda(z)\tilde{\Gamma}(z)\tilde{Q}(z) \\
 &= Q(z)\Lambda(z)\Gamma(z)\tilde{\Gamma}(z)\tilde{Q}(z) = Q(z)\Lambda(z)\tilde{Q}(z) \quad (19)
 \end{aligned}$$

- ▶ example for $\tilde{Q}(z)$ — note different orders:



Iterative PEVD Algorithms

- ▶ Second order sequential best rotation (SBR2, McWhirter 2007);
- ▶ iterative approach based on an elementary paraunitary operation:

$$\begin{aligned}
 \mathbf{S}^{(0)}(z) &= \mathbf{R}(z) \\
 &\vdots \\
 \mathbf{S}^{(i+1)}(z) &= \tilde{\mathbf{H}}^{(i+1)}(z)\mathbf{S}^{(i)}(z)\mathbf{H}^{(i+1)}(z)
 \end{aligned}$$

- ▶ $\mathbf{H}^{(i)}(z)$ is an elementary paraunitary operation, which at the i th step eliminates the largest off-diagonal element in $\mathbf{s}^{(i-1)}(z)$;
- ▶ stop after L iterations:

$$\hat{\mathbf{\Lambda}}(z) = \mathbf{S}^{(L)}(z) \quad , \quad \mathbf{Q}(z) = \prod_{i=1}^L \mathbf{H}^{(i)}(z)$$

- ▶ sequential matrix diagonalisation (SMD) and
- ▶ multiple-shift SMD (MS-SMD) will follow the same scheme ...

Elementary Paraunitary Operation

- ▶ An elementary paraunitary matrix [Vaidyanathan] is defined as

$$\mathbf{H}^{(i)}(z) = \mathbf{I} - \mathbf{v}^{(i)} \mathbf{v}^{(i),H} + z^{-1} \mathbf{v}^{(i)} \mathbf{v}^{(i),H}, \quad \|\mathbf{v}^{(i)}\|_2 = 1$$

- ▶ we utilise a different definition:

$$\mathbf{H}^{(i)}(z) = \mathbf{D}^{(i)}(z) \mathbf{Q}^{(i)}$$

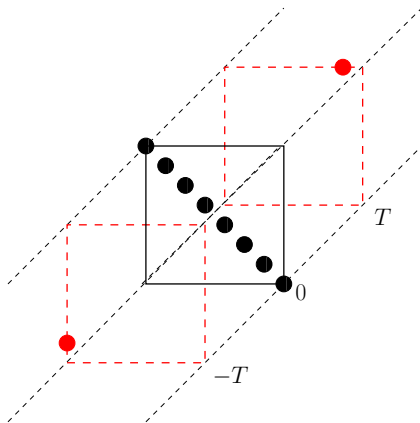
- ▶ $\mathbf{D}^{(i)}(z)$ is a delay matrix:

$$\mathbf{D}^{(i)}(z) = \text{diag}\{1 \dots 1 z^{-\tau} 1 \dots 1\}$$

- ▶ $\mathbf{Q}^{(i)}(z)$ is a Givens rotation.

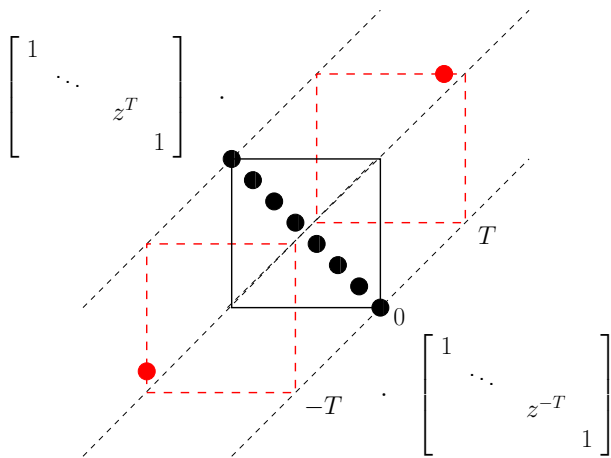
Sequential Best Rotation Algorithm (McWhirter)

- At iteration i , consider $\mathbf{S}^{(i-1)}(z) \circ \mathbf{S}^{(i-1)}[\tau]$



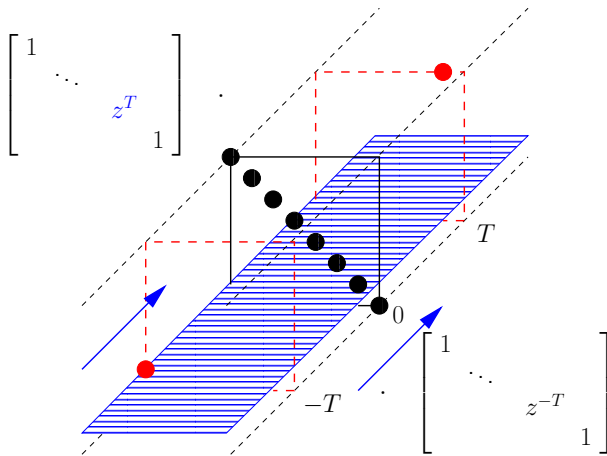
Sequential Best Rotation Algorithm (McWhirter)

► $\tilde{D}^{(i)}(z) \mathbf{S}^{(i-1)}(z) \mathbf{D}^{(i)}(z)$



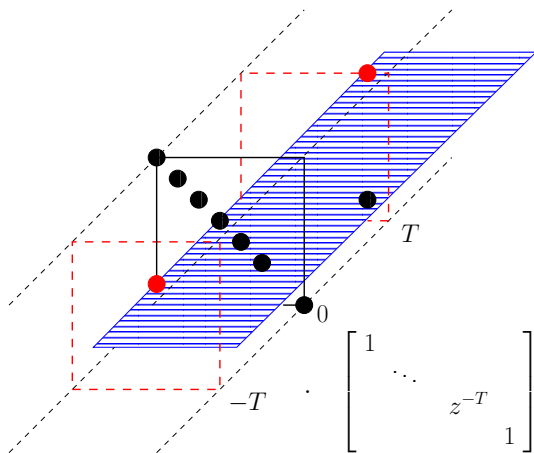
Sequential Best Rotation Algorithm (McWhirter)

- ▶ $\tilde{D}^{(i)}(z)$ advances a row-slice of $S^{(i-1)}(z)$ by T



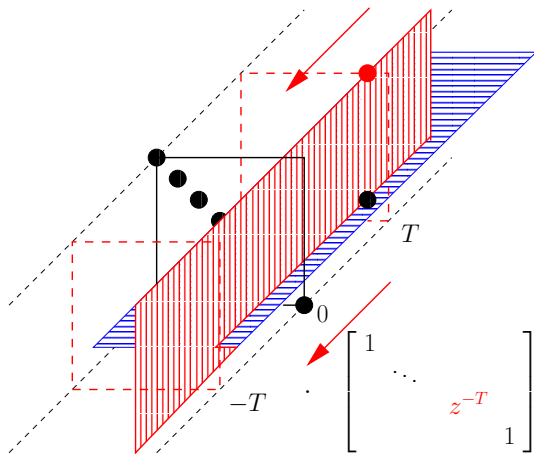
Sequential Best Rotation Algorithm (McWhirter)

- ▶ the off-diagonal element at $-T$ has now been translated to lag zero



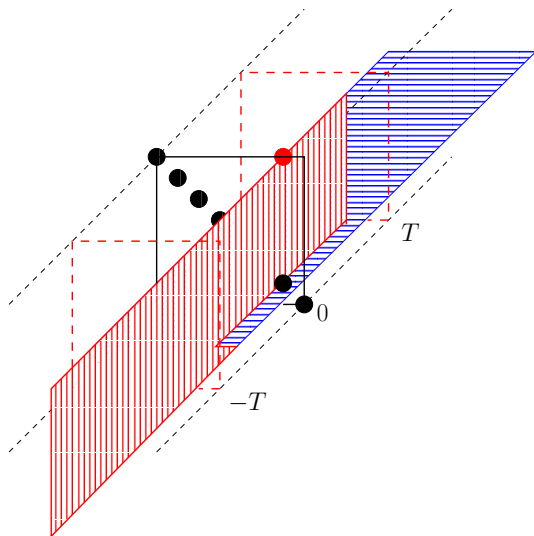
Sequential Best Rotation Algorithm (McWhirter)

- ▶ $\mathbf{D}^{(i)}(z)$ delays a column-slice of $\mathbf{S}^{(i-1)}(z)$ by T



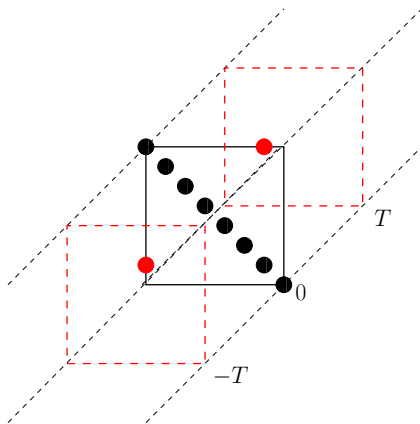
Sequential Best Rotation Algorithm (McWhirter)

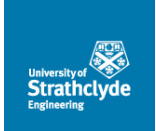
- ▶ the off-diagonal element at $-T$ has now been translated to lag zero



Sequential Best Rotation Algorithm (McWhirter)

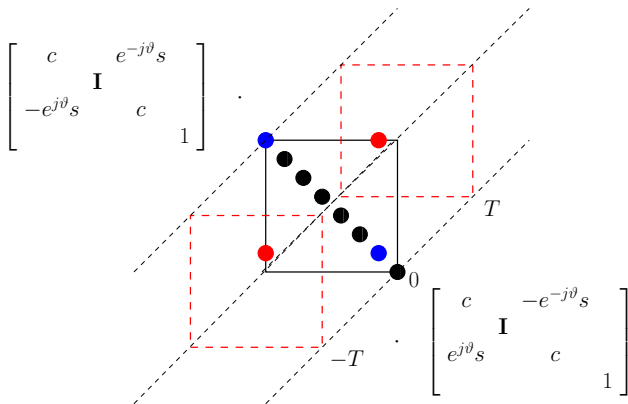
- ▶ the step $\tilde{\mathbf{D}}^{(i)}(z)\mathbf{S}^{(i-1)}(z)\mathbf{D}_{(i)}(z)$ has brought the largest off-diagonal elements to lag 0.





Sequential Best Rotation Algorithm (McWhirter)

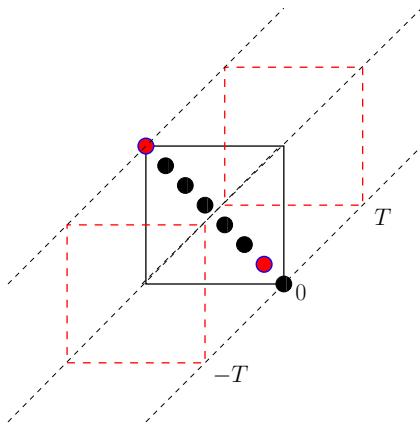
- Jacobi step to eliminate largest off-diagonal elements by $\mathbf{Q}^{(i)}$



Sequential Best Rotation Algorithm (McWhirter)

- iteration i is completed, having performed

$$\mathbf{S}^{(i)}(z) = \mathbf{Q}^{(i)} \mathbf{D}^{(i)}(z) \mathbf{S}^{(i-1)}(z) \tilde{\mathbf{D}}^{(i)}(z) \tilde{\mathbf{Q}}^{(i)}(z)$$



SBR2 Outcome

- ▶ At the i th iteration, the zeroing of off-diagonal elements achieved during previous steps may be partially undone;
- ▶ however, the algorithm has been shown to converge, transferring energy onto the main diagonal at every step (McWhirter 2007);
- ▶ after L iterations, we reach an approximate diagonalisation

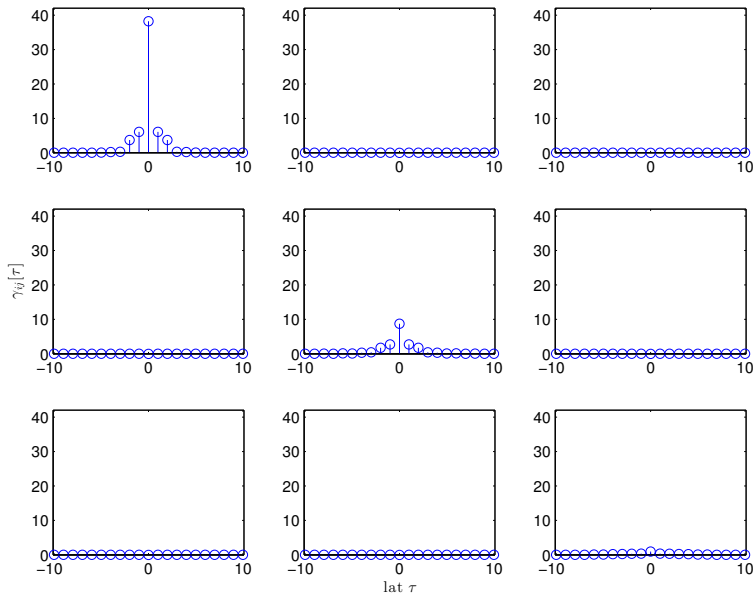
$$\hat{\mathbf{A}}(z) = \mathbf{S}^{(L)}(z) = \tilde{\mathbf{Q}}(z)\mathbf{R}(z)\mathbf{Q}(z)$$

with

$$\mathbf{Q}(z) = \prod_{i=1}^L \mathbf{D}^{(i)}(z)\mathbf{Q}^{(i)}$$

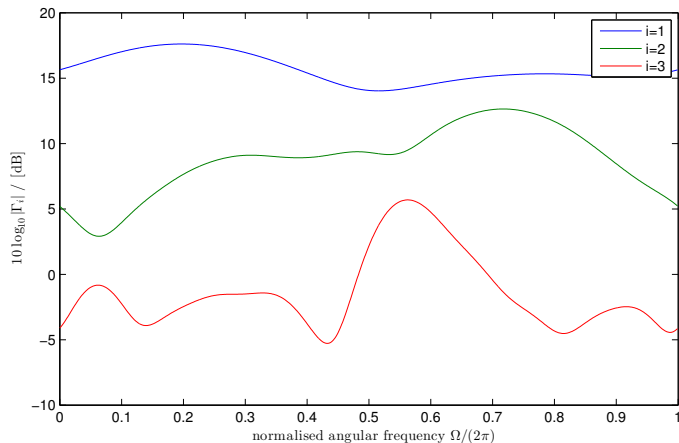
- ▶ diagonalisation of the previous 3×3 polynomial matrix ...

SBR2 Example — Diagonalisation



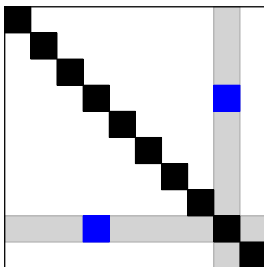
SBR2 Example — Spectral Majorisation

- ▶ The on-diagonal elements are spectrally majorised



SBR2 — Givens Rotation

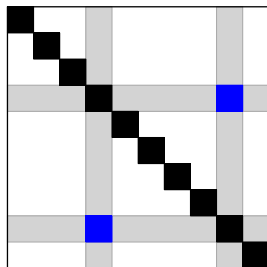
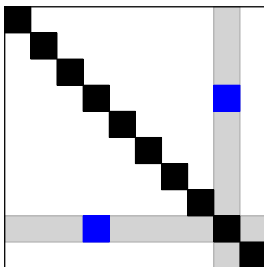
- ▶ A Givens rotation eliminates the maximum off-diagonal element once brought onto the lag-zero matrix;
- ▶ note I: in the lag-zero matrix, one column and one row are modified by the shift:



- ▶ note II: a Givens rotation only affects two columns and two rows in every matrix;
- ▶ Givens rotation is relatively low in computational cost!

SBR2 — Givens Rotation

- ▶ A Givens rotation eliminates the maximum off-diagonal element once brought onto the lag-zero matrix;
- ▶ note I: in the lag-zero matrix, one column and one row are modified by the shift:

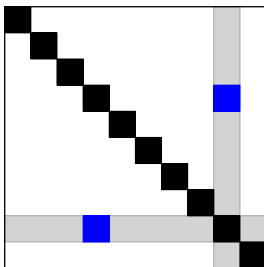


- ▶ note II: a Givens rotation only affects two columns and two rows in every matrix;
- ▶ Givens rotation is relatively low in computational cost!

Sequential Matrix Diagonalisation (SMD)

[Redif *et al.*, IEEE Trans SP 2015]

- ▶ Main idea — the zero-lag matrix is diagonalised in every step;
- ▶ initialisation: diagonalise $\mathbf{R}[0]$ by EVD and apply modal matrix to all matrix coefficients $\rightarrow \mathbf{S}^{(0)}$;
- ▶ at the i th step as in SBR2, the maximum element (or column with max. norm) is shifted to the lag-zero matrix:

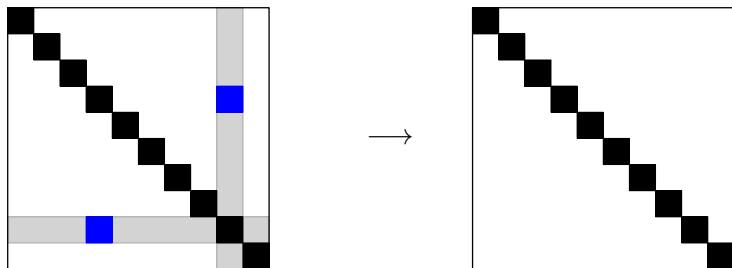


- ▶ an EVD is used to re-diagonalise the zero-lag matrix;
- ▶ a full modal matrix is applied at all lags — more costly than SBR2.

Sequential Matrix Diagonalisation (SMD)

[Redif *et al.*, IEEE Trans SP 2015]

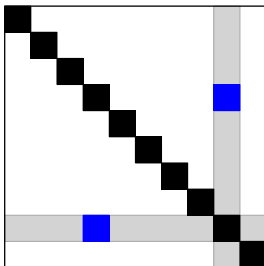
- ▶ Main idea — the zero-lag matrix is diagonalised in every step;
- ▶ initialisation: diagonalise $\mathbf{R}[0]$ by EVD and apply modal matrix to all matrix coefficients $\rightarrow \mathbf{S}^{(0)}$;
- ▶ at the i th step as in SBR2, the maximum element (or column with max. norm) is shifted to the lag-zero matrix:



- ▶ an EVD is used to re-diagonalise the zero-lag matrix;
- ▶ a full modal matrix is applied at all lags — more costly than SBR2.

Multiple Shift SMD (SMD)

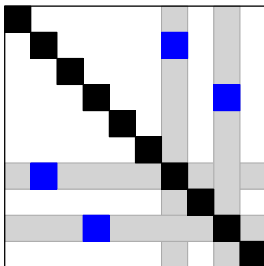
- ▶ SMD **converges faster** than SBR2 — more energy is transferred per iteration step;
- ▶ SMD is **more expensive** than SBR2 — full matrix multiplication at every lag;
- ▶ this cost will not increase further if more columns / rows are shifted into the lag-zero matrix at every iteration



- ▶ MS-SMD will transfer yet more off-diagonal energy per iteration;
- ▶ because the total energy must remain constant under paraunitary operations, SBR2, SMD and MS-SMD can be proven to converge.

Multiple Shift SMD (SMD)

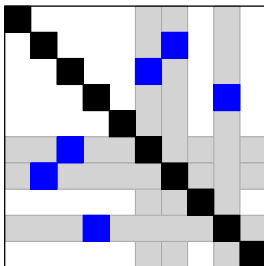
- ▶ SMD **converges faster** than SBR2 — more energy is transferred per iteration step;
- ▶ SMD is **more expensive** than SBR2 — full matrix multiplication at every lag;
- ▶ this cost will not increase further if more columns / rows are shifted into the lag-zero matrix at every iteration



- ▶ MS-SMD will transfer yet more off-diagonal energy per iteration;
- ▶ because the total energy must remain constant under paraunitary operations, SBR2, SMD and MS-SMD can be proven to converge.

Multiple Shift SMD (SMD)

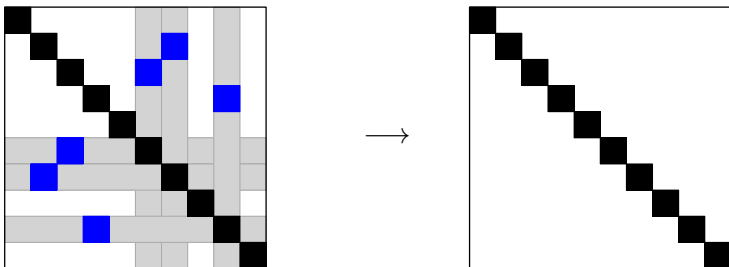
- ▶ SMD **converges faster** than SBR2 — more energy is transferred per iteration step;
- ▶ SMD is **more expensive** than SBR2 — full matrix multiplication at every lag;
- ▶ this cost will not increase further if more columns / rows are shifted into the lag-zero matrix at every iteration



- ▶ MS-SMD will transfer yet more off-diagonal energy per iteration;
- ▶ because the total energy must remain constant under paraunitary operations, SBR2, SMD and MS-SMD can be proven to converge.

Multiple Shift SMD (SMD)

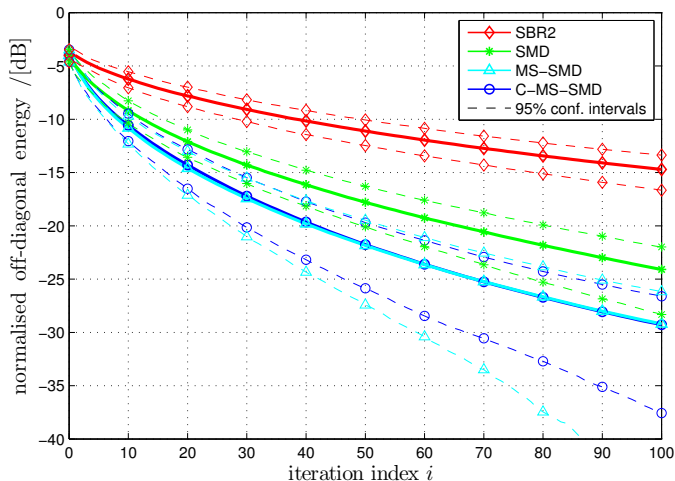
- ▶ SMD **converges faster** than SBR2 — more energy is transferred per iteration step;
- ▶ SMD is **more expensive** than SBR2 — full matrix multiplication at every lag;
- ▶ this cost will not increase further if more columns / rows are shifted into the lag-zero matrix at every iteration



- ▶ MS-SMD will transfer yet more off-diagonal energy per iteration;
- ▶ because the total energy must remain constant under paraunitary operations, SBR2, SMD and MS-SMD can be proven to converge.

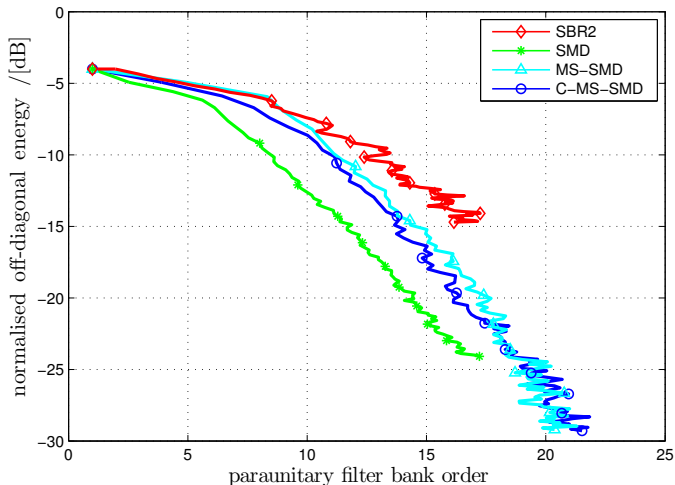
SBR2/SMD/MS-SMD Convergence

- Measuring the remaining normalised off-diagonal energy over an ensemble of space-time covariance matrices:



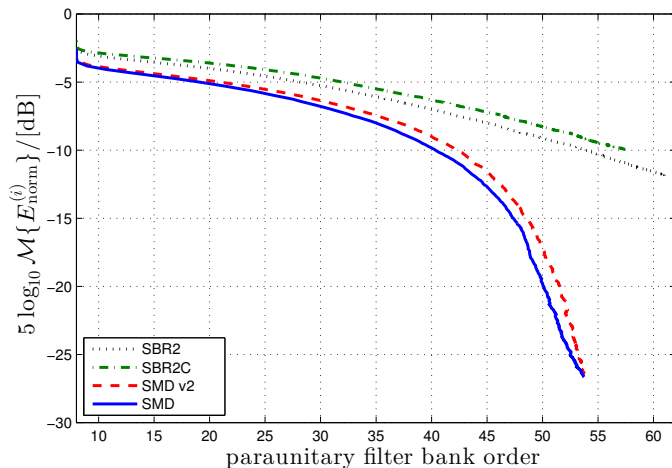
SBR2/SMD/MS-SMD Application Cost 1

- ▶ Ensemble average of remaining off-diagonal energy vs. order of paraunitary filter banks to decompose $4 \times 4 \times 16$ matrices:



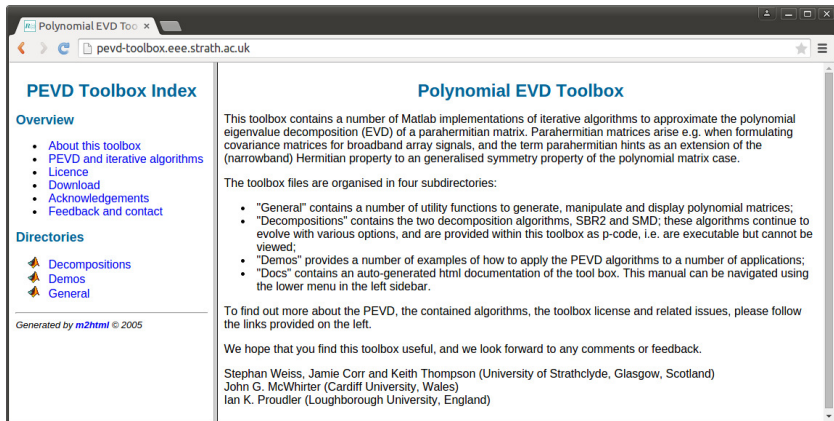
SBR2/SMD/MS-SMD Application Cost 2

- ▶ Ensemble average of remaining off-diagonal energy vs. order of paraunitary filter banks to decompose $8 \times 8 \times 64$ matrices:



MATLAB Polynomial EVD Toolbox

- ▶ The MATLAB polynomial EVD toolbox can be downloaded from pevd-toolbox.eee.strath.ac.uk



The screenshot shows a web browser window with the URL pevd-toolbox.eee.strath.ac.uk. The page is titled "Polynomial EVD Toolbox" and features a sidebar with a "PEVD Toolbox Index" and an "Overview" section. The main content area contains an introduction to the toolbox, a list of subdirectories, and contact information for the developers.

PEVD Toolbox Index

Overview

- [About this toolbox](#)
- [PEVD and iterative algorithms](#)
- [Licence](#)
- [Download](#)
- [Acknowledgements](#)
- [Feedback and contact](#)

Directories

- ▶ [Decompositions](#)
- ▶ [Demos](#)
- ▶ [General](#)

Generated by [m2html](#) © 2005

Polynomial EVD Toolbox

This toolbox contains a number of Matlab implementations of iterative algorithms to approximate the polynomial eigenvalue decomposition (EVD) of a parahermitian matrix. Parahermitian matrices arise e.g. when formulating covariance matrices for broadband array signals, and the term parahermitian hints as an extension of the (narrowband) Hermitian property to a generalised symmetry property of the polynomial matrix case.

The toolbox files are organised in four subdirectories:

- "General" contains a number of utility functions to generate, manipulate and display polynomial matrices;
- "Decompositions" contains the two decomposition algorithms, SBR2 and SMD; these algorithms continue to evolve with various options, and are provided within this toolbox as p-code, i.e. are executable but cannot be viewed;
- "Demos" provides a number of examples of how to apply the PEVD algorithms to a number of applications;
- "Docs" contains an auto-generated html documentation of the tool box. This manual can be navigated using the lower menu in the left sidebar.

To find out more about the PEVD, the contained algorithms, the toolbox license and related issues, please follow the links provided on the left.

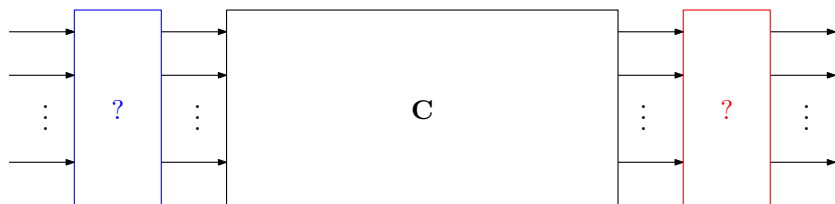
We hope that you find this toolbox useful, and we look forward to any comments or feedback.

Stephan Weiss, Jamie Corr and Keith Thompson (University of Strathclyde, Glasgow, Scotland)
John G. McWhirter (Cardiff University, Wales)
Ian K. Proudler (Loughborough University, England)

- ▶ the toolbox contains a number of iterative algorithms to calculate an approximate PEVD, related functions, and demos.

Narrowband MIMO Communications

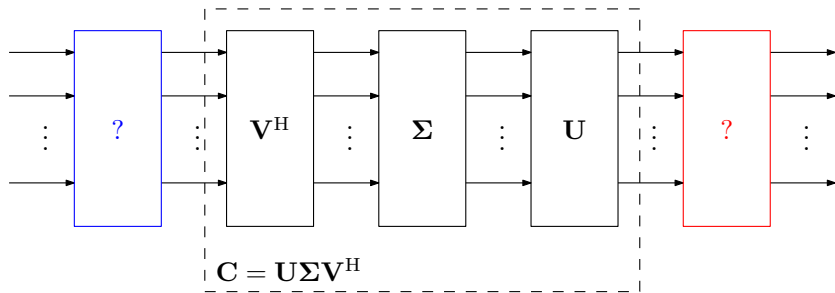
- ▶ a narrowband channel is characterised by a matrix \mathbf{C} containing complex gain factors;
- ▶ problem: how to select the **precoder** and **equaliser**?



- ▶ overall system;

Narrowband MIMO Communications

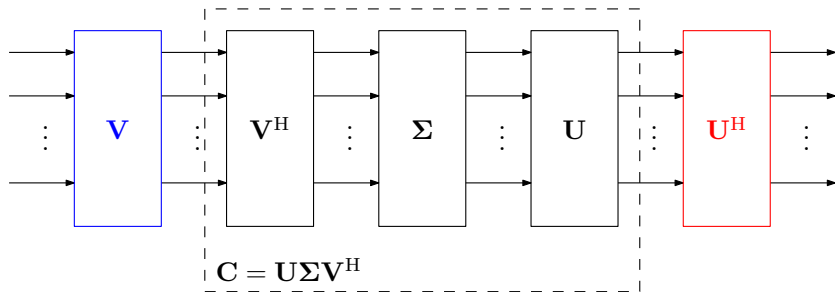
- ▶ a narrowband channel is characterised by a matrix \mathbf{C} containing complex gain factors;
- ▶ problem: how to select the **precoder** and **equaliser**?



- ▶ the singular value decomposition (SVD) factorises \mathbf{C} into two unitary matrices \mathbf{U} and \mathbf{V}^H and a diagonal matrix Σ ;

Narrowband MIMO Communications

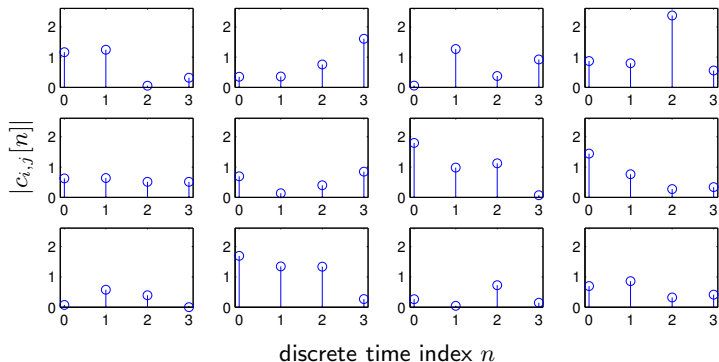
- ▶ a narrowband channel is characterised by a matrix \mathbf{C} containing complex gain factors;
- ▶ problem: how to select the precoder and equaliser?



- ▶ we select the precoder and the equaliser from the unitary matrices provided by the channel's SVD;
- ▶ the overall system is diagonalised, decoupling the channel into independent single-input single-output systems by means of unitary matrices.

Broadband MIMO Channel

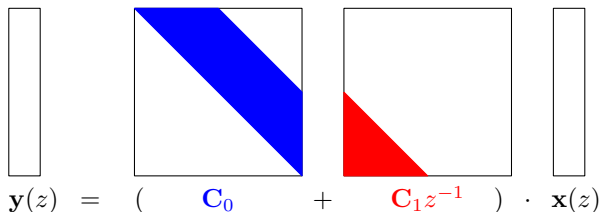
- ▶ The channel is now a matrix of FIR filters; example for a 3×4 MIMO system $\mathbf{C}[n]$:



- ▶ the transfer function $\mathbf{C}(z) \bullet \text{---} \circ \mathbf{C}[n]$ is a polynomial matrix;
- ▶ an SVD can only diagonalise $\mathbf{C}[n]$ for one particular lag n .

Standard Broadband MIMO Approaches

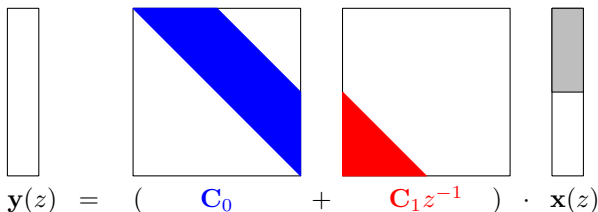
- ▶ OFDM (if approximate channel length is known):
 1. divide spectrum into narrowband channels;
 2. address each narrowband channel independently using narrowband-optimal techniques;
 drawback: ignores spectral coherence across frequency bins;
- ▶ optimum filter bank transceiver (if channel itself is known):
 1. block processing;
 2. inter-block interference is eliminated by guard intervals;
 3. resulting matrix can be diagonalised by SVD;
- ▶ both techniques invest DOFs into the guard intervals, which are generally not balanced against other error sources.



$$\mathbf{y}(z) = \left(\mathbf{C}_0 + \mathbf{C}_1 z^{-1} \right) \cdot \mathbf{x}(z)$$

Standard Broadband MIMO Approaches

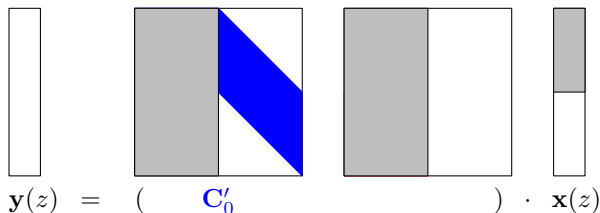
- ▶ OFDM (if approximate channel length is known):
 1. divide spectrum into narrowband channels;
 2. address each narrowband channel independently using narrowband-optimal techniques;
 drawback: ignores spectral coherence across frequency bins;
- ▶ optimum filter bank transceiver (if channel itself is known):
 1. block processing;
 2. inter-block interference is eliminated by guard intervals;
 3. resulting matrix can be diagonalised by SVD;
- ▶ both techniques invest DOFs into the guard intervals, which are generally not balanced against other error sources.



$$\mathbf{y}(z) = \left(\mathbf{C}_0 + \mathbf{C}_1 z^{-1} \right) \cdot \mathbf{x}(z)$$

Standard Broadband MIMO Approaches

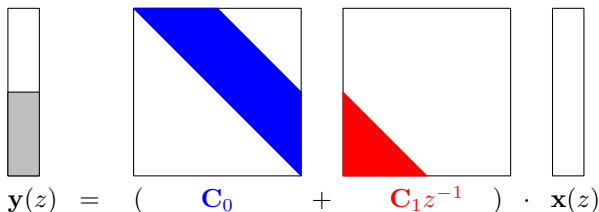
- ▶ OFDM (if approximate channel length is known):
 1. divide spectrum into narrowband channels;
 2. address each narrowband channel independently using narrowband-optimal techniques;
 drawback: ignores spectral coherence across frequency bins;
- ▶ optimum filter bank transceiver (if channel itself is known):
 1. block processing;
 2. inter-block interference is eliminated by guard intervals;
 3. resulting matrix can be diagonalised by SVD;
- ▶ both techniques invest DOFs into the guard intervals, which are generally not balanced against other error sources.



$$\mathbf{y}(z) = \left(\begin{array}{c} \text{grey block} \\ \text{blue diagonal} \\ \text{grey block} \end{array} \right) \cdot \mathbf{x}(z)$$

Standard Broadband MIMO Approaches

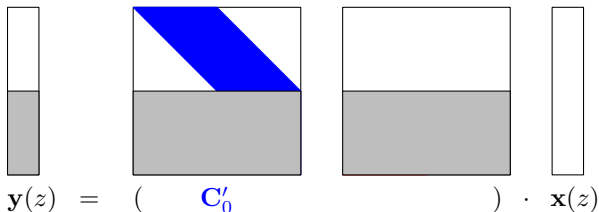
- ▶ OFDM (if approximate channel length is known):
 1. divide spectrum into narrowband channels;
 2. address each narrowband channel independently using narrowband-optimal techniques;
 drawback: ignores spectral coherence across frequency bins;
- ▶ optimum filter bank transceiver (if channel itself is known):
 1. block processing;
 2. inter-block interference is eliminated by guard intervals;
 3. resulting matrix can be diagonalised by SVD;
- ▶ both techniques invest DOFs into the guard intervals, which are generally not balanced against other error sources.



$$\mathbf{y}(z) = \left(\mathbf{C}_0 + \mathbf{C}_1 z^{-1} \right) \cdot \mathbf{x}(z)$$

Standard Broadband MIMO Approaches

- ▶ OFDM (if approximate channel length is known):
 1. divide spectrum into narrowband channels;
 2. address each narrowband channel independently using narrowband-optimal techniques;
 drawback: ignores spectral coherence across frequency bins;
- ▶ optimum filter bank transceiver (if channel itself is known):
 1. block processing;
 2. inter-block interference is eliminated by guard intervals;
 3. resulting matrix can be diagonalised by SVD;
- ▶ both techniques invest DOFs into the guard intervals, which are generally not balanced against other error sources.



$$\mathbf{y}(z) = \begin{pmatrix} \mathbf{C}'_0 \end{pmatrix} \cdot \mathbf{x}(z)$$

Polynomial Singular Value Decompositions

- ▶ Iterative algorithms have been developed to determine a polynomial eigenvalue decomposition (EVD) for a parahermitian matrix $\mathbf{R}(z) = \tilde{\mathbf{R}}(z) = \mathbf{R}^H(z^{-1})$:

$$\mathbf{R}(z) \approx \mathbf{H}(z)\mathbf{\Gamma}(z)\tilde{\mathbf{H}}(z)$$

- ▶ paraunitary $\mathbf{H}(z)\tilde{\mathbf{H}}(z) = \mathbf{I}$, diagonal and spectrally majorised $\mathbf{\Gamma}(z)$;
- ▶ polynomial SVD of channel $\mathbf{C}(z)$ can be obtained via two EVDs:

$$\mathbf{C}(z)\tilde{\mathbf{C}}(z) = \mathbf{U}(z)\mathbf{\Sigma}^+(z)\mathbf{\Sigma}^-(z)\tilde{\mathbf{U}}(z)$$

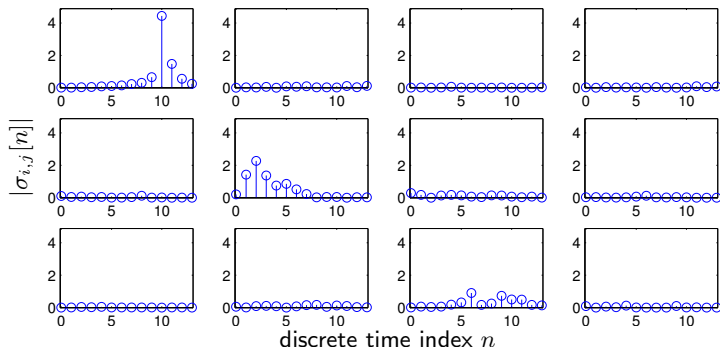
$$\tilde{\mathbf{C}}(z)\mathbf{C}(z) = \mathbf{V}(z)\mathbf{\Sigma}^-(z)\mathbf{\Sigma}^+(z)\tilde{\mathbf{V}}(z)$$

finally:

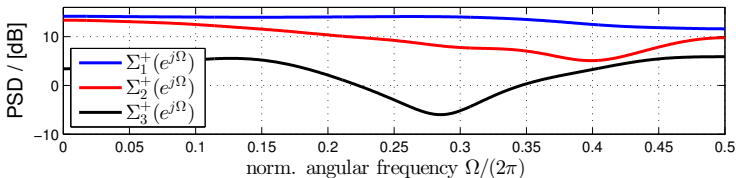
$$\mathbf{C}(z) = \mathbf{U}(z)\mathbf{\Sigma}^+(z)\tilde{\mathbf{V}}(z)$$

MIMO Application Example

- Polynomial SVD of the previous $C(z) \in \mathbb{C}^{3 \times 4}$ channel matrix:

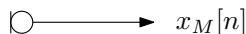
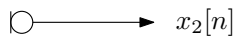
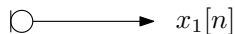
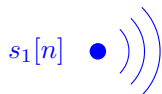


- the singular value spectra are majorised:



Narrowband Source Model

- ▶ Scenario with sensor array and far-field sources:

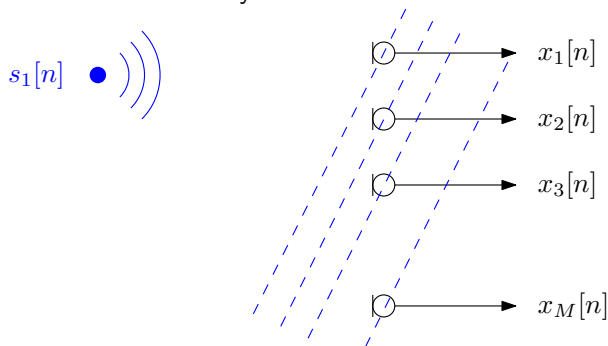


- ▶ for the narrowband case, the source signals arrive with delays, expressed by phase shifts in a steering vector
- ▶ data model:

$$\mathbf{x}[n] =$$

Narrowband Source Model

- ▶ Scenario with sensor array and far-field sources:

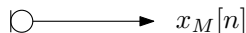
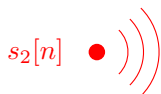
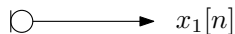
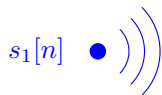


- ▶ for the narrowband case, the source signals arrive with delays, expressed by phase shifts in a steering vector \mathbf{s}_1
- ▶ data model:

$$\mathbf{x}[n] = s_1[n] \cdot \mathbf{s}_1$$

Narrowband Source Model

- ▶ Scenario with sensor array and far-field sources:

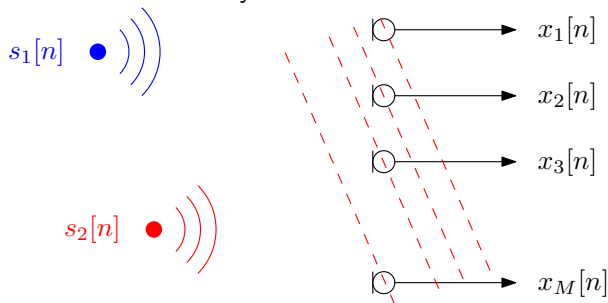


- ▶ for the narrowband case, the source signals arrive with delays, expressed by phase shifts in a steering vector \mathbf{s}_1
- ▶ data model:

$$\mathbf{x}[n] = \mathbf{s}_1[n] \cdot \mathbf{s}_1$$

Narrowband Source Model

- Scenario with sensor array and far-field sources:

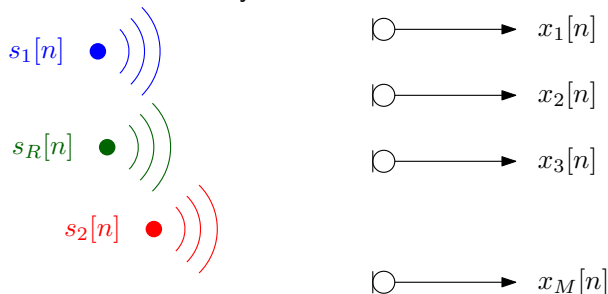


- for the narrowband case, the source signals arrive with delays, expressed by phase shifts in a steering vector \mathbf{s}_1 , \mathbf{s}_2
- data model:

$$\mathbf{x}[n] = s_1[n] \cdot \mathbf{s}_1 + s_2[n] \cdot \mathbf{s}_2$$

Narrowband Source Model

- Scenario with sensor array and far-field sources:

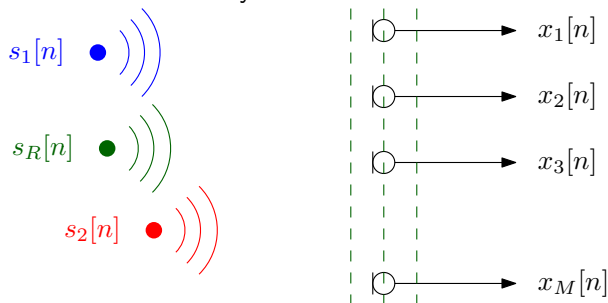


- for the narrowband case, the source signals arrive with delays, expressed by phase shifts in a steering vector \mathbf{s}_1 , \mathbf{s}_2
- data model:

$$\mathbf{x}[n] = s_1[n] \cdot \mathbf{s}_1 + s_2[n] \cdot \mathbf{s}_2$$

Narrowband Source Model

- Scenario with sensor array and far-field sources:



- for the narrowband case, the source signals arrive with delays, expressed by phase shifts in a steering vector $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_R$;
- data model:

$$\mathbf{x}[n] = s_1[n] \cdot \mathbf{s}_1 + s_2[n] \cdot \mathbf{s}_2 + \dots + s_R[n] \cdot \mathbf{s}_R = \sum_{r=1}^R s_r[n] \cdot \mathbf{s}_r$$

Steering Vector

- ▶ A signal $s[n]$ arriving at the array can be characterised by the delays of its wavefront (neglecting attenuation):

$$\begin{bmatrix} x_0[n] \\ x_1[n] \\ \vdots \\ x_{M-1}[n] \end{bmatrix} = \begin{bmatrix} s[n - \tau_0] \\ s[n - \tau_1] \\ \vdots \\ s[n - \tau_{M-1}] \end{bmatrix} = \begin{bmatrix} \delta[n - \tau_0] \\ \delta[n - \tau_1] \\ \vdots \\ \delta[n - \tau_{M-1}] \end{bmatrix} * s[n] \quad \text{---} \bullet \quad \mathbf{a}_\vartheta(z) S(z)$$

- ▶ if evaluated at a narrowband normalised angular frequency Ω_i , the time delays τ_m in the **broadband steering vector** $\mathbf{a}_\vartheta(z)$ collapse to phase shifts in the **narrowband steering vector** $\mathbf{a}_{\vartheta, \Omega_i}$,

$$\mathbf{a}_{\vartheta, \Omega_i} = \mathbf{a}_\vartheta(z) \Big|_{z=e^{j\Omega_i}} = \begin{bmatrix} e^{-j\tau_0\Omega_i} \\ e^{-j\tau_1\Omega_i} \\ \vdots \\ e^{-j\tau_{M-1}\Omega_i} \end{bmatrix} \cdot$$

Data and Covariance Matrices

- ▶ A data matrix $\mathbf{X} \in \mathbb{C}^{M \times L}$ can be formed from L measurements:

$$\mathbf{X} = [\mathbf{x}[n] \quad \mathbf{x}[n+1] \quad \dots \quad \mathbf{x}[n+L-1]]$$

- ▶ assuming that all $x_m[n]$, $m = 1, 2, \dots, M$ are zero mean, the (instantaneous) data covariance matrix is

$$\mathbf{R} = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^H[n]\} \approx \frac{1}{L}\mathbf{X}\mathbf{X}^H$$

where the approximation assumes ergodicity and a sufficiently large L ;

- ▶ Problem: can we tell from \mathbf{X} or \mathbf{R} (i) the number of sources and (ii) their origin / time series?
- ▶ w.r.t. Jonathon Chamber's introduction, we here only consider the underdetermined case of more sensors than sources, $M \geq K$, and generally $L \gg M$.

SVD of Data Matrix

- ▶ Singular value decomposition of \mathbf{X} :

$$\boxed{\mathbf{X}} = \boxed{\mathbf{U}} \boxed{\Sigma} \boxed{\mathbf{V}^H}$$

- ▶ unitary matrices $\mathbf{U} = [\mathbf{u}_1 \dots \mathbf{u}_M]$ and $\mathbf{V} = [\mathbf{v}_1 \dots \mathbf{v}_L]$;
- ▶ diagonal Σ contains the real, positive semidefinite singular values of \mathbf{X} in descending order:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \ddots & \vdots & \vdots & & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & & \vdots \\ 0 & & 0 & \sigma_M & 0 & \dots & 0 \end{bmatrix}$$

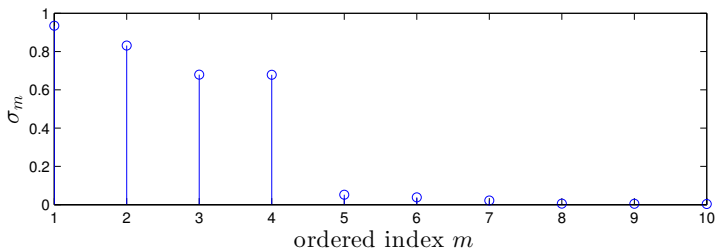
with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_M \geq 0$.

Singular Values

- ▶ If the array is illuminated by $R \leq M$ linearly independent sources, the rank of the data matrix is

$$\text{rank}\{\mathbf{X}\} = R$$

- ▶ only the first R singular values of \mathbf{X} will be non-zero;
- ▶ in practice, noise often will ensure that $\text{rank}\{\mathbf{X}\} = M$, with $M - R$ trailing singular values that define the noise floor:



- ▶ therefore, by thresholding singular values, it is possible to estimate the number of linearly independent sources R .

Subspace Decomposition

- ▶ If $\text{rank}\{\mathbf{X}\} = R$, the SVD can be split:

$$\mathbf{X} = [\mathbf{U}_s \quad \mathbf{U}_n] \begin{bmatrix} \Sigma_s & \mathbf{0} \\ \mathbf{0} & \Sigma_n \end{bmatrix} \begin{bmatrix} \mathbf{V}_s^H \\ \mathbf{V}_n^H \end{bmatrix}$$

- ▶ with $\mathbf{U}_s \in \mathbb{C}^{M \times R}$ and $\mathbf{V}_s^H \in \mathbb{C}^{R \times L}$ corresponding to the R largest singular values;
- ▶ \mathbf{U}_s and \mathbf{V}_s^H define the **signal-plus-noise subspace** of \mathbf{X} :

$$\mathbf{X} = \sum_{m=1}^M \sigma_m \mathbf{u}_m \mathbf{v}_m^H \approx \sum_{m=1}^R \sigma_m \mathbf{u}_m \mathbf{v}_m^H$$

- ▶ the complements \mathbf{U}_n and \mathbf{V}_n^H ,

$$\mathbf{U}_s^H \mathbf{U}_n = \mathbf{0} \quad , \quad \mathbf{V}_s \mathbf{V}_n^H = \mathbf{0}$$

define the **noise-only subspace** of \mathbf{X} .

SVD via Two EVDs

- ▶ Any Hermitian matrix $\mathbf{A} = \mathbf{A}^H$ allows an eigenvalue decomposition

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H$$

with \mathbf{Q} unitary and the eigenvalues in $\mathbf{\Lambda}$ real valued and positive semi-definite;

- ▶ postulating $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, therefore:

$$\mathbf{X}\mathbf{X}^H = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^H)(\mathbf{V}\mathbf{\Sigma}^H\mathbf{U}^H) = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H \quad (20)$$

$$\mathbf{X}^H\mathbf{X} = (\mathbf{V}\mathbf{\Sigma}^H\mathbf{U}^H)(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^H) = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H \quad (21)$$

- ▶ (ordered) eigenvalues relate to the singular values: $\lambda_m = \sigma_m^2$;
- ▶ the covariance matrix $\mathbf{R} = \frac{1}{L}\mathbf{X}\mathbf{X}$ has the same rank as the data matrix \mathbf{X} , and with \mathbf{U} provides access to the same spatial subspace decomposition.

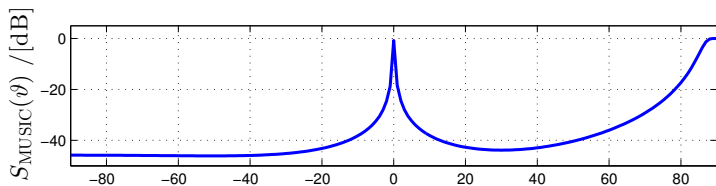
Narrowband MUSIC Algorithm

- ▶ EVD of the narrowband covariance matrix identifies signal-plus-noise and noise-only subspaces

$$\mathbf{R} = [\mathbf{U}_s \quad \mathbf{U}_n] \begin{bmatrix} \mathbf{\Lambda}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_n \end{bmatrix} \begin{bmatrix} \mathbf{U}_s^H \\ \mathbf{U}_n^H \end{bmatrix}$$

- ▶ scanning the signal-plus-noise subspace could only help to retrieve sources with orthogonal steering vectors;
- ▶ therefore, the multiple signal classification (MUSIC) algorithm scans the noise-only subspace for minima, or maxima of its reciprocal

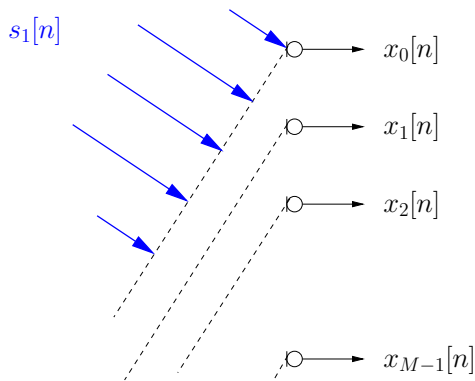
$$S_{\text{MUSIC}}(\vartheta) = \frac{1}{\|\mathbf{U}_n \mathbf{a}_{\vartheta, \Omega_i}\|_2^2}$$



Narrowband Source Separation

- ▶ Via SVD of the data matrix \mathbf{X} or EVD of the covariance matrix \mathbf{R} , we can determine the number of linearly independent sources R ;
- ▶ using the subspace decompositions offered by EVD/SVD, the directions of arrival can be estimated using e.g. MUSIC;
- ▶ based on knowledge of the angle of arrival, beamforming could be applied to \mathbf{X} to extract specific sources;
- ▶ overall: EVD (and SVD) can play a vital part in **narrowband source separation**;
- ▶ what about **broadband source separation**?

Broadband Array Scenario



- ▶ Compared to the narrowband case, time delays rather than phase shifts bear information on the direction of a source.

Broadband Steering Vector

- ▶ A signal $s[n]$ arriving at the array can be characterised by the delays of its wavefront (neglecting attenuation):

$$\begin{bmatrix} x_0[n] \\ x_1[n] \\ \vdots \\ x_{M-1}[n] \end{bmatrix} = \begin{bmatrix} s[n - \tau_0] \\ s[n - \tau_1] \\ \vdots \\ s[n - \tau_{M-1}] \end{bmatrix} = \begin{bmatrix} \delta[n - \tau_0] \\ \delta[n - \tau_1] \\ \vdots \\ \delta[n - \tau_{M-1}] \end{bmatrix} * s[n] \quad \text{---} \bullet \quad \mathbf{a}_\vartheta(z) S(z)$$

- ▶ if evaluated at a narrowband normalised angular frequency Ω_i , the time delays τ_m in the **broadband steering vector** $\mathbf{a}_\vartheta(z)$ collapse to phase shifts in the **narrowband steering vector** $\mathbf{a}_{\vartheta, \Omega_i}$,

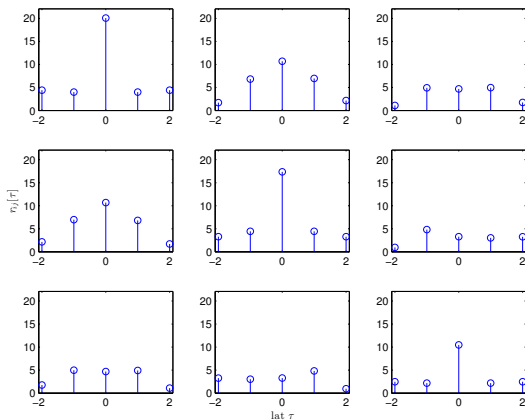
$$\mathbf{a}_{\vartheta, \Omega_i} = \mathbf{a}_\vartheta(z) \Big|_{z=e^{j\Omega_i}} = \begin{bmatrix} e^{-j\tau_0\Omega_i} \\ e^{-j\tau_1\Omega_i} \\ \vdots \\ e^{-j\tau_{M-1}\Omega_i} \end{bmatrix} \cdot$$

Space-Time Covariance Matrix

- ▶ If delays must be considered, the (space-time) covariance matrix must capture the lag τ :

$$\mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}[n] \cdot \mathbf{x}^H[n - \tau]\}$$

- ▶ $\mathbf{R}[\tau]$ contains auto- and cross-correlation sequences:



Cross Spectral Density Matrix

- ▶ z -transform of the space-time covariance matrix is given by

$$\mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}_n \mathbf{x}_{n-\tau}^H\} \quad \circ \text{---} \bullet \quad \mathbf{R}(z) = \sum_l S_l(z) \mathbf{a}_{\vartheta_l}(z) \tilde{\mathbf{a}}_{\vartheta_l}(z) + \sigma_N^2 \mathbf{I}$$

with ϑ_l the direction of arrival and $S_l(z)$ the PSD of the l th source;

- ▶ $\mathbf{R}(z)$ is the cross spectral density (CSD) matrix;
- ▶ the instantaneous covariance matrix (no lag parameter τ)

$$\mathbf{R} = \mathcal{E}\{\mathbf{x}_n \mathbf{x}_n^H\} = \mathbf{R}[0]$$

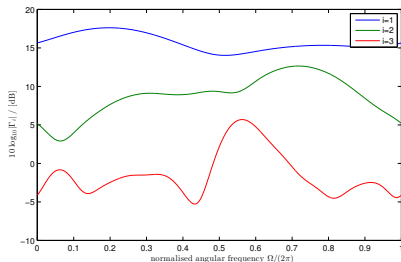
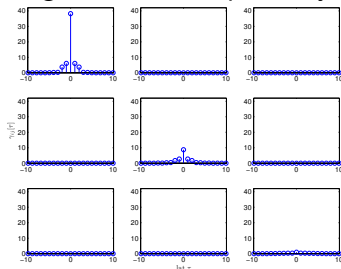
Polynomial MUSIC (PMUSIC)

[Alrmah, Weiss, Lambotharan, *EUSIPCO* (2011)]

- Based on the polynomial EVD of the broadband covariance matrix

$$\mathbf{R}(z) \approx \underbrace{[\mathbf{Q}_s(z) \quad \mathbf{Q}_n(z)]}_{\mathbf{Q}(z)} \underbrace{\begin{bmatrix} \mathbf{\Lambda}_s(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_n(z) \end{bmatrix}}_{\mathbf{\Lambda}(z)} \begin{bmatrix} \tilde{\mathbf{Q}}_s(z) \\ \tilde{\mathbf{Q}}_n(z) \end{bmatrix}$$

- paraunitary $\mathbf{Q}(z)$, s.t. $\mathbf{Q}(z)\tilde{\mathbf{Q}}(z) = \mathbf{I}$;
- diagonalised and spectrally majorised $\mathbf{\Lambda}(z)$:



PMUSIC cont'd

- ▶ Idea — scan the polynomial noise-only subspace $Q_n(z)$ with broadband steering vectors

$$\Gamma(z, \vartheta) = \tilde{\mathbf{a}}_{\vartheta}(z) \tilde{\mathbf{Q}}_n(z) \mathbf{Q}_n(z) \mathbf{a}_{\vartheta}(z)$$

- ▶ looking for minima leads to a spatio-spectral PMUSIC

$$S_{\text{PSS-MUSIC}}(\vartheta, \Omega) = (\Gamma(z, \vartheta)|_{z=e^{j\Omega}})^{-1}$$

- ▶ and a spatial-only PMUSIC

$$S_{\text{PS-MUSIC}}(\vartheta) = \left(2\pi \oint \Gamma(z, \vartheta)|_{z=e^{j\Omega}} d\Omega \right)^{-1} = \Gamma_{\vartheta}^{-1}[0]$$

with $\Gamma_{\vartheta}[\tau] \circ \bullet \Gamma(z, \vartheta)$.

Simulation I — Toy Problem

- ▶ Linear uniform array with critical spatial and temporal sampling;
- ▶ broadband steering vector for end-fire position:

$$\mathbf{a}_{\pi/2}(z) = [1 \quad z^{-1} \quad \dots \quad z^{-M+1}]^T$$

- ▶ covariance matrix

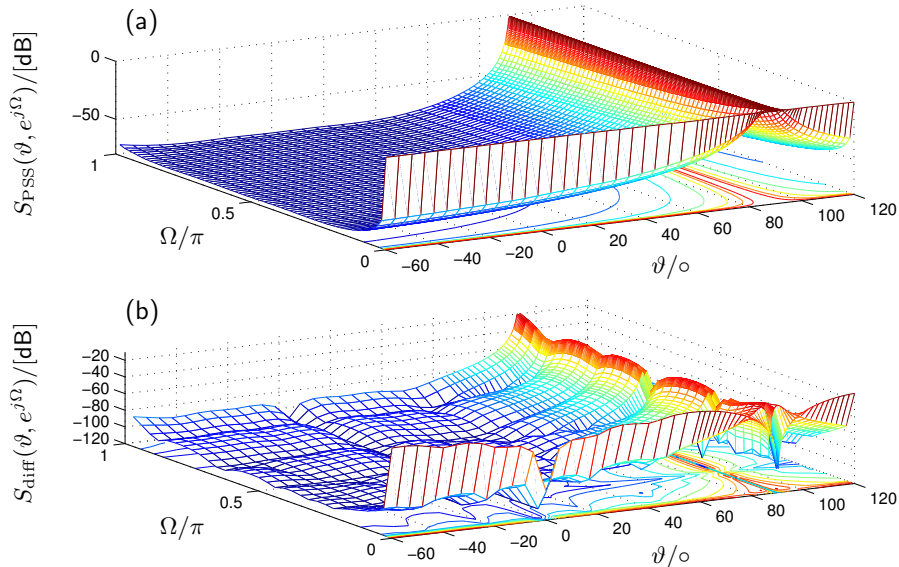
$$\mathbf{R}(z) = \mathbf{a}_{\pi/2}(z)\tilde{\mathbf{a}}_{\pi/2}(z) = \begin{bmatrix} 1 & z^1 & \dots & z^{M-1} \\ z^{-1} & 1 & & \vdots \\ \vdots & & \ddots & \vdots \\ z^{-M+1} & \dots & \dots & 1 \end{bmatrix} .$$

- ▶ PEVD (by inspection)

$$\mathbf{Q}(z) = \mathbf{T}_{\text{DFT}} \text{diag}\{1 \quad z^{-1} \quad \dots \quad z^{-M+1}\} \quad ; \quad \mathbf{\Lambda}(z) = \text{diag}\{1 \quad 0 \quad \dots \quad 0\}$$

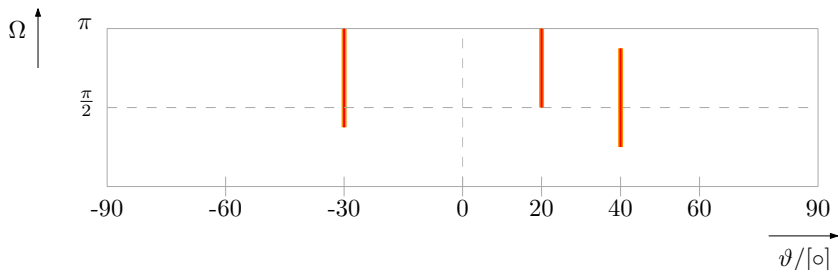
- ▶ simulations with $M = 4 \dots$

Simulation I — PSS-MUSIC



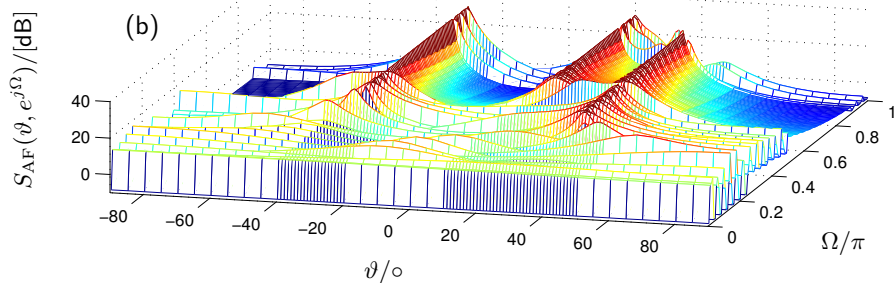
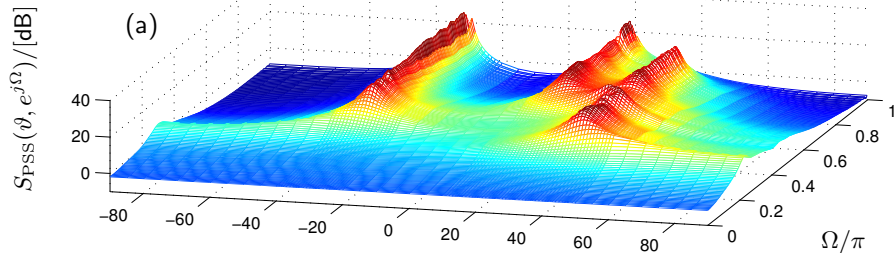
Simulation II

- ▶ $M = 8$ element sensor array illuminated by three sources;
- ▶ source 1: $\vartheta_1 = -30^\circ$, active over range $\Omega \in [\frac{3\pi}{8}; \pi]$;
- ▶ source 2: $\vartheta_2 = 20^\circ$, active over range $\Omega \in [\frac{\pi}{2}; \pi]$;
- ▶ source 3: $\vartheta_3 = 40^\circ$, active over range $\Omega \in [\frac{2\pi}{8}; \frac{7\pi}{8}]$; and



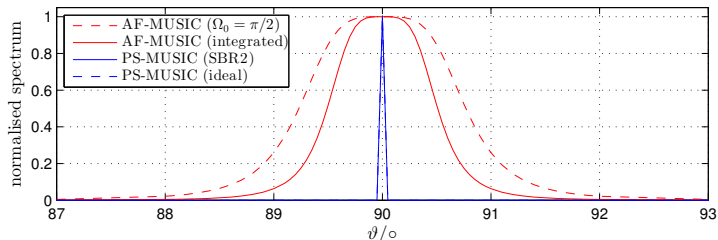
- ▶ filter banks as innovation filters, and broadband steering vectors to simulate AoA;
- ▶ space-time covariance matrix is estimated from 10^4 samples.

Simulation II — PSS-MUSIC

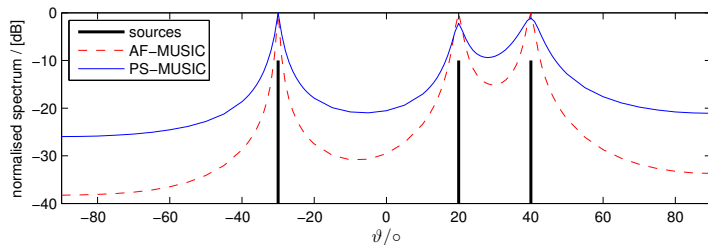


PS-MUSIC Comparison

- ▶ Simulation I (toy problem): peaks normalised to unity:



- ▶ Simulation II: inaccuracies on PEVD and broadband steering vector

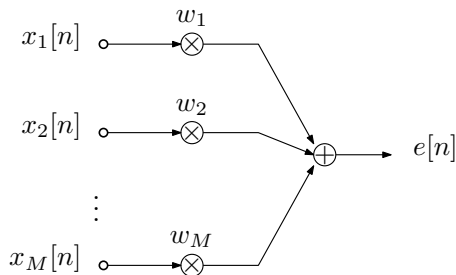


AoA Estimation — Conclusions

- ▶ We have considered the importance of SVD and EVD for narrowband source separation;
- ▶ narrowband matrix decomposition reveal the matrix rank and offer subspace decompositions on which angle-of-arrival estimation algorithms such as MUSIC can be based;
- ▶ broadband problems lead to a space-time covariance or CSD matrix;
- ▶ such polynomial matrices cannot be decomposed by standard EVD and SVD;
- ▶ a polynomial EVD has been defined;
- ▶ iterative algorithms such as SBR2 can be used to approximate the PEVD;
- ▶ this permits a number of applications, such as broadband angle of arrival estimation;
- ▶ broadband beamforming could then be used to separate broadband sources.

Narrowband Minimum Variance Distortionless Response Beamformer

- ▶ Scenario: an array of M sensors receives data $\mathbf{x}[n]$, containing a desired signal with frequency Ω_s and angle of arrival ϑ_s , corrupted by interferers;
- ▶ a narrowband beamformer applies a single coefficient to every of the M sensor signals:



Narrowband MVDR Problem

- ▶ Recall the space-time covariance matrix:

$$\mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^H[n - \tau]\}$$

- ▶ the MVDR beamformer minimises the output power of the beamformer:

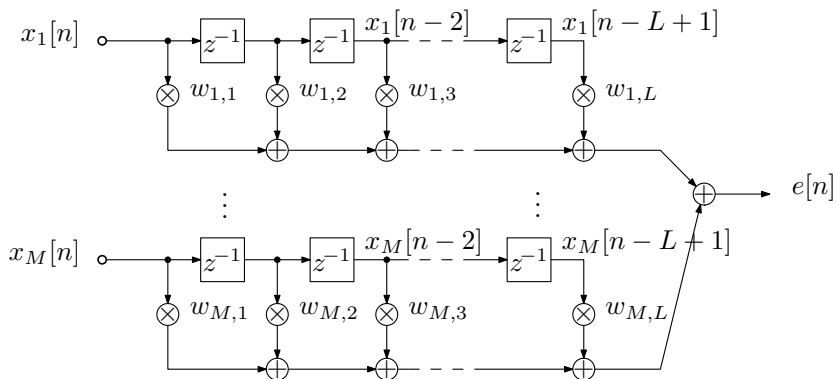
$$\min_{\mathbf{w}} \mathcal{E}\{|e[n]|^2\} = \min_{\mathbf{w}} \mathbf{w}^H \mathbf{R}[0] \mathbf{w} \quad (22)$$

$$\text{s.t. } \mathbf{a}^H(\vartheta_s, \Omega_s) \mathbf{w} = 1, \quad (23)$$

- ▶ this is subject to protecting the signal of interest by a constraint in look direction ϑ_s ;
- ▶ the steering vector $\mathbf{a}_{\vartheta_s, \Omega_s}$ defines the signal of interest's parameters.

Broadband MVDR Beamformer

- Each sensor is followed by a tap delay line of dimension L , giving a total of ML coefficients in a vector $\mathbf{v} \in \mathbb{C}^{ML}$



Broadband MVDR Beamformer

- ▶ A larger input vector $\mathbf{x}_n \in \mathbb{C}^{ML}$ is generated, also including lags;
- ▶ the general approach is similar to the narrowband system, minimising the power of $e[n] = \mathbf{v}^H \mathbf{x}_n$;
- ▶ however, we require several constraint equations to protect the signal of interest, e.g.

$$\mathbf{C} = [\mathbf{s}(\vartheta_s, \Omega_0), \mathbf{s}(\vartheta_s, \Omega_1) \dots \mathbf{s}(\vartheta_s, \Omega_{L-1})] \quad (24)$$

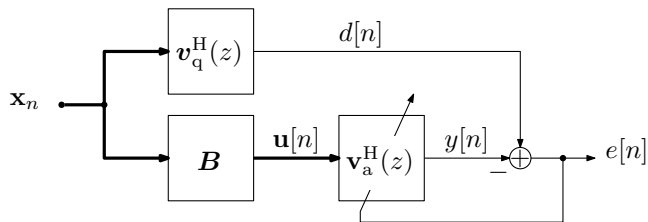
- ▶ these L constraints pin down the response to unit gain at L separate points in frequency:

$$\mathbf{C}^H \mathbf{v} = \mathbf{1} ; \quad (25)$$

- ▶ generally $\mathbf{C} \in \mathbb{C}^{ML \times L}$, but simplifications can be applied if the look direction is towards broadside.

Generalised Sidelobe Canceller

- ▶ A quiescent beamformer $\mathbf{v}_q = (\mathbf{C}^H)^\dagger \mathbf{1} \in \mathbb{C}^{ML}$ picks the signal of interest;
- ▶ the quiescent beamformer is optimal for AWGN but generally passes structured interference;
- ▶ the output of the blocking matrix \mathbf{B} contains interference only, which requires $[\mathbf{B}\mathbf{C}]$ to be unitary; hence $\mathbf{B} \in \mathbb{C}^{ML \times (M-1)L}$;
- ▶ an adaptive noise canceller $\mathbf{v}_a \in \mathbb{C}^{(M-1)L}$ aims to remove the residual interference:



- ▶ note: all dimensions are determined by $\{M, L\}$.

Polynomial Matrix MVDR Formulation

- ▶ Power spectral density of beamformer output:

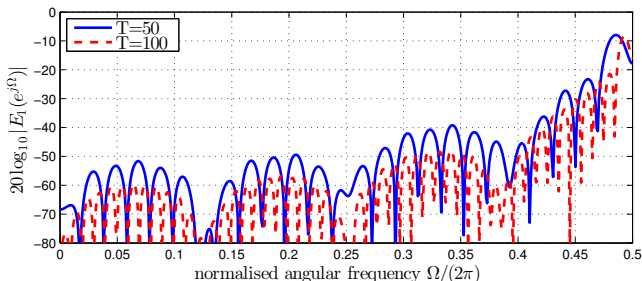
$$R_e(z) = \tilde{\mathbf{w}}(z)\mathbf{R}(z)\mathbf{w}(z)$$

- ▶ proposed broadband MVDR beamformer formulation:

$$\min_{\mathbf{w}(z)} \oint_{|z|=1} R_e(z) \frac{dz}{z} \quad (26)$$

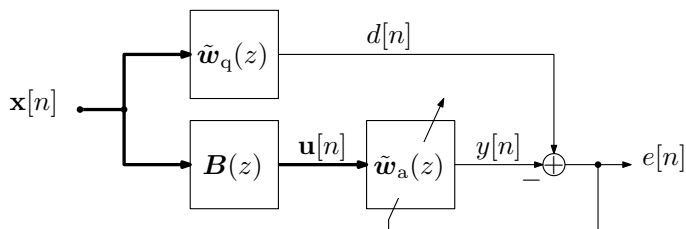
$$\text{s.t. } \tilde{\mathbf{a}}(\vartheta_s, z)\mathbf{w}(z) = F(z). \quad (27)$$

- ▶ precision of broadband steering vector, $|\tilde{\mathbf{a}}(\vartheta_s, z)\mathbf{a}(\vartheta_s, z) - 1|$, depends on the length T of the fractional delay filter:



Generalised Sidelobe Canceller

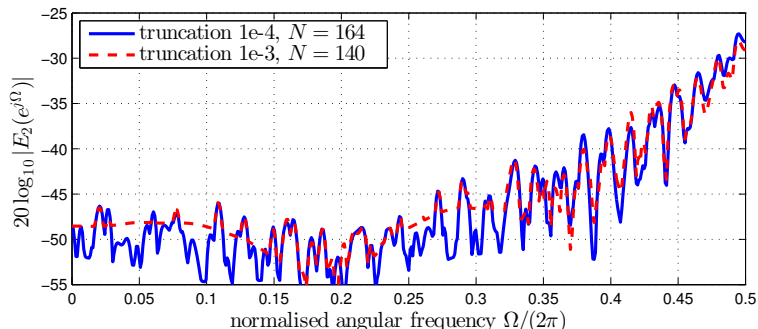
- ▶ Instead of performing constrained optimisation, the GSC projects the data and performs adaptive noise cancellation:



- ▶ the quiescent vector $\mathbf{w}_q(z)$ is generated from the constraints and passes signal plus interference;
- ▶ the blocking matrix $\mathbf{B}(z)$ has to be orthonormal to $\mathbf{w}_q(z)$ and only pass interference.

Design Considerations

- ▶ The blocking matrix can be obtained by completing a paraunitary matrix from $\mathbf{w}_q(z)$;
- ▶ this can be achieved by calculating a PEVD of the rank one matrix $\mathbf{w}_q(z)\tilde{\mathbf{w}}_q(z)$;
- ▶ this leads to a block matrix of order N that is typically greater than L ;
- ▶ maximum leakage of the signal of interest through the blocking matrix:



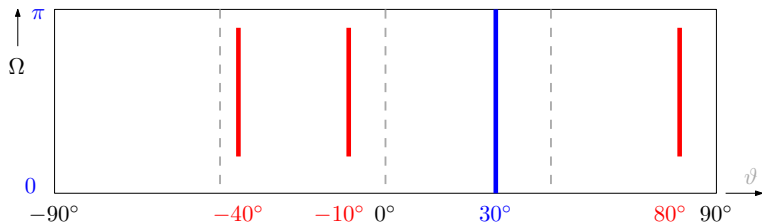
Computational Cost

- ▶ With M sensors and a TDL length of L , the complexity of a standard beamformer is dominated by the blocking matrix;
- ▶ in the proposed design, $\mathbf{w}_a \in \mathbb{C}^{M-1}$ has degree L ;
- ▶ the quiescent vector $\mathbf{w}_q(z) \in \mathbb{C}^M$ has degree T ;
- ▶ the blocking matrix $\mathbf{B}(z) \in \mathbb{C}^{(M-1) \times M}$ has degree N ;
- ▶ cost comparison in multiply-accumulates (MACs):

component	GSC cost	
	polynomial	standard
quiescent beamformer	MT	ML
blocking matrix	$M(M-1)N$	$M(M-1)L^2$
adaptive filter (NLMS)	$2(M-1)L$	$2(M-1)L$

Example

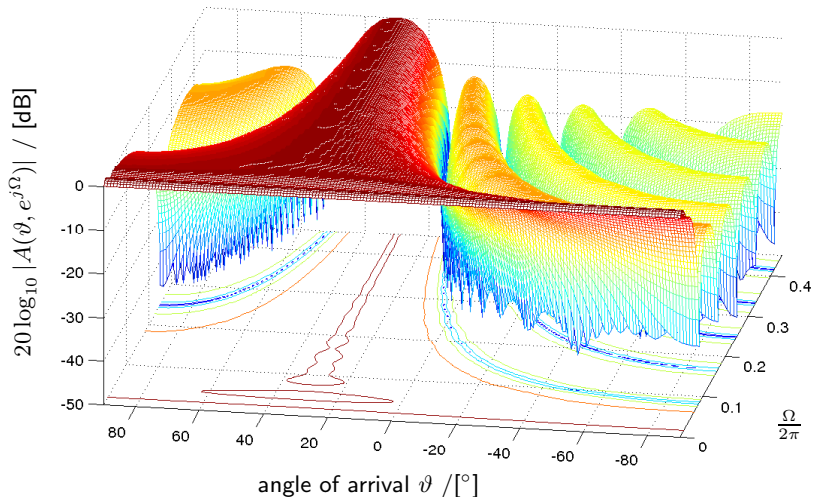
- ▶ We assume a **signal of interest** from $\vartheta = 30^\circ$;
- ▶ three **interferers** with angles $\vartheta_i \in \{-40^\circ, -10^\circ, 80^\circ\}$ active over the frequency range $\Omega = 2\pi \cdot [0.1; 0.45]$ at signal to interference ratio of -40 dB;



- ▶ $M = 8$ element linear uniform array is also corrupted by spatially and temporally white additive Gaussian noise at 20 dB SNR;
- ▶ parameters: $L = 175$, $T = 50$, and $N = 140$;
- ▶ cost per iteration: 10.7 kMACs (proposed) versus 1.72 MMACs (standard).

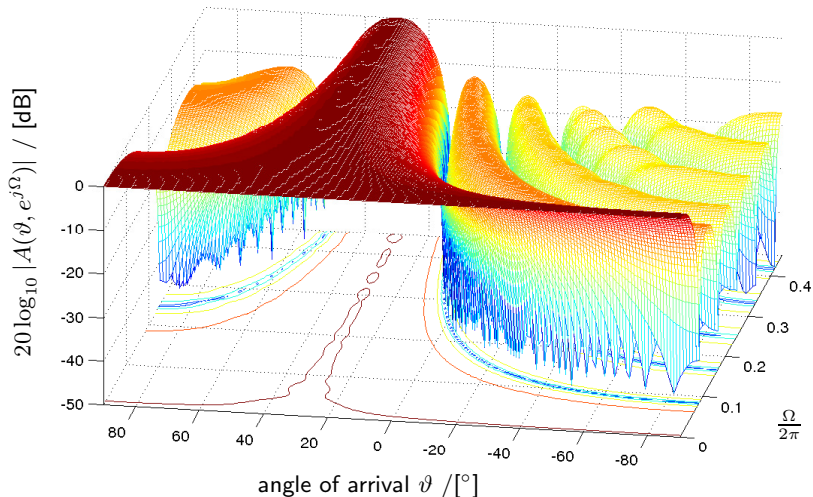
Quiescent Beamformer

- Directivity pattern of quiescent standard broadband beamformer:



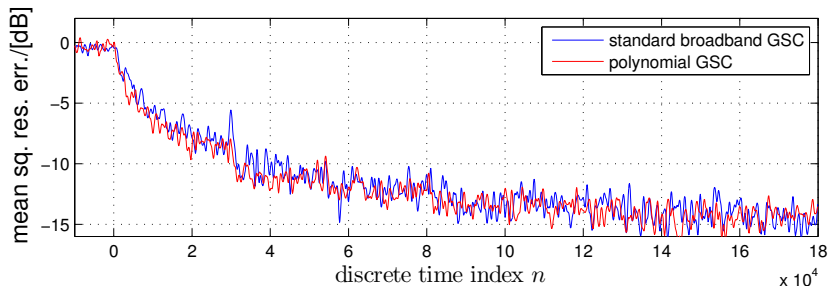
Quiescent Beamformer

- ▶ Directivity pattern of quiescent proposed broadband beamformer:



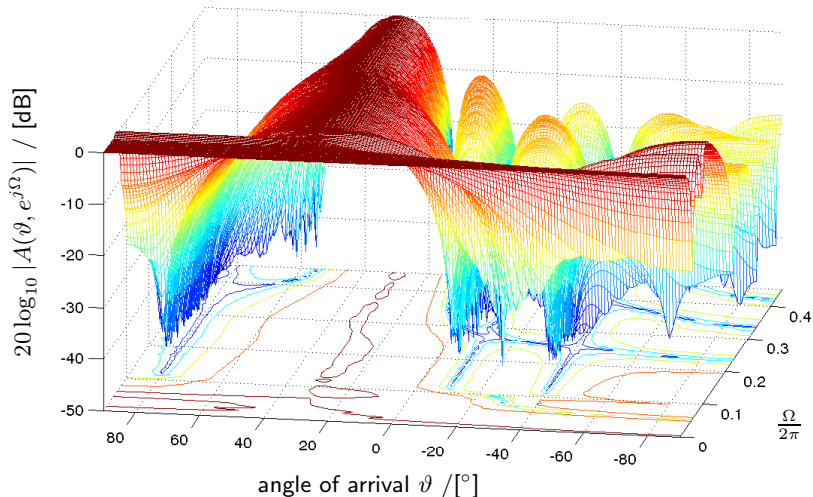
Adaptation

- ▶ Convergence curves of the two broadband beamformers, showing the residual mean squared error (i.e. beamformer output minus signal of interest):



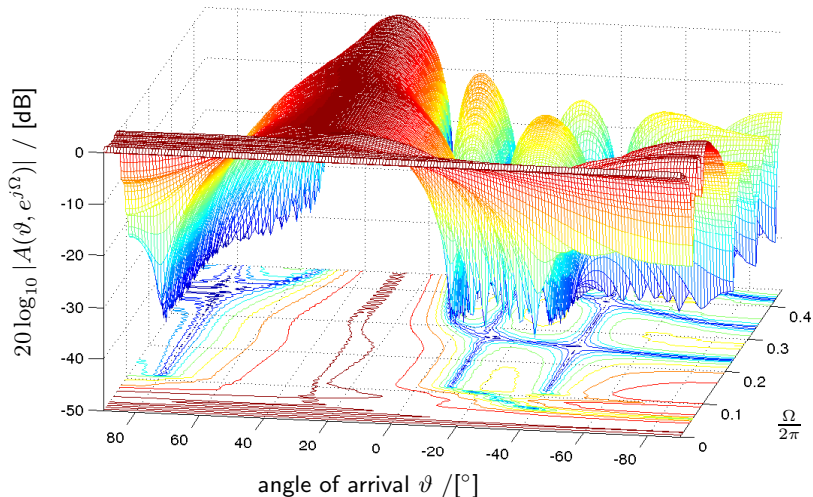
Adapted Beamformer

- Directivity pattern of adapted proposed broadband beamformer:



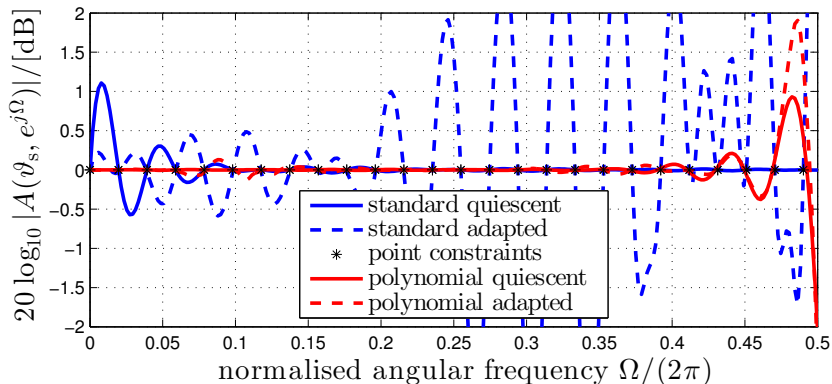
Adapted Beamformer

- ▶ Directivity pattern of adapted standard broadband beamformer:



Gain in Look Direction

- Gain in look direction $\vartheta_s = 30^\circ$ before and after adaptation:



- due to signal leakage, the standard broadband beamformer after adaptation only maintains the point constraints but deviates elsewhere.

Broadband Beamforming Conclusions

- ▶ Based on the previous AoA estimation, beamforming can help to extract source signals and thus perform “source separation”;
- ▶ broadband beamformers usually assume pre-steering such that the signal of interest lies at broadside;
- ▶ this is not always given, and difficult for arbitrary array geometries;
- ▶ the proposed beamformer using a polynomial matrix formulation can implement arbitrary constraints;
- ▶ the performance for such constraints is better in terms of the accuracy of the directivity pattern;
- ▶ because the proposed design decouples the complexities of the coefficient vector, the quiescent vector and block matrix, and the adaptive process, the cost is significantly lower than for a standard broadband adaptive beamformer.

Additional Material

- ▶ Key papers:
 - 1 J.G. McWhirter, P.D. Baxter, T. Cooper, S. Redif, and J. Foster: “An EVD Algorithm for Para-Hermitian Polynomial Matrices,” IEEE Trans SP, **55**(5): 2158-2169, May 2007.
 - 2 S. Redif, J.G. McWhirter, and S. Weiss: “Design of FIR Paraunitary Filter Banks for Subband Coding Using a Polynomial Eigenvalue Decomposition,” IEEE Trans SP, **59**(11): 5253-5264, Nov. 2011.
 - 3 S. Redif, S. Weiss, and J.G. McWhirter: “Sequential matrix diagonalisation algorithms for polynomial EVD of parahermitian matrices,” IEEE Trans SP, **63**(1): 81–89, Jan. 2015.
- ▶ If interested in the discussed methods and algorithms, please download the free Matlab PEVD toolbox from
pevd-toolbox.eee.strath.ac.uk
- ▶ for questions, please feel free to ask:
 - Stephan Weiss (stephan.weiss@strath.ac.uk) or
 - Jamie Corr (jamie.corr@strath.ac.uk)
 - Fraser Coutts (fraser.coutts@strath.ac.uk)