

Polynomial Matrix Decompositions with Engineering Applications

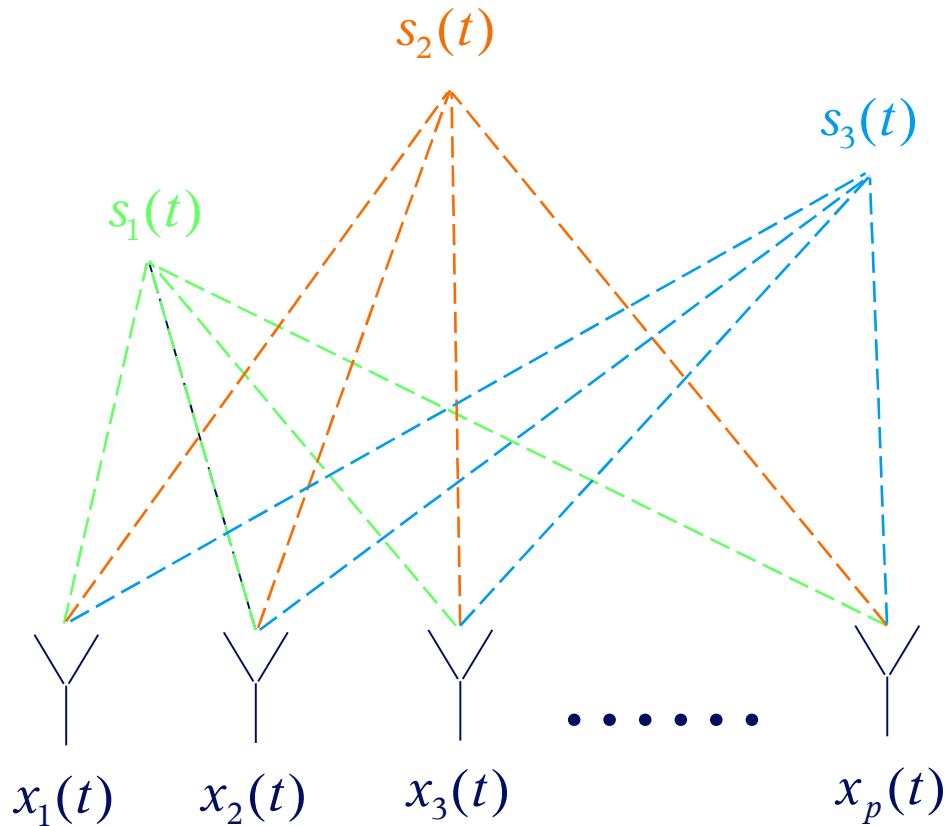
Prof J G McWhirter FRS
FREng
School of Engineering
Cardiff University

Dr Joanne Foster MSc PhD
Electronic & Electrical Engineering
Loughborough University

Outline of Talk

- Convolutive mixing and polynomial matrices
 - strong decorrelation by PEVD
- Sequential best rotation algorithm (SBR2)
 - convolutive signal extraction
- Polynomial matrix QR decomposition (PQRD)
 - MIMO channel equalisation

Instantaneous Mixing



- Signal model

$$x(t) = \mathbf{A}s(t) + n(t)$$

- Data matrix

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{N}$$

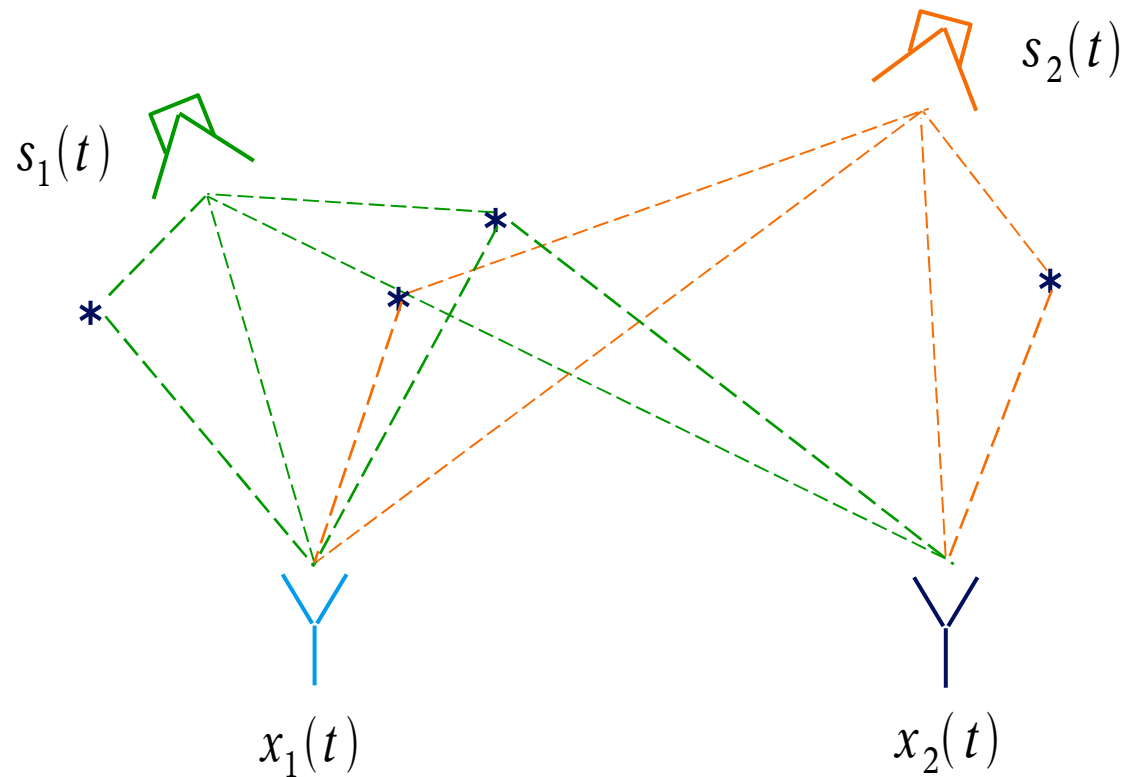
- Sample covariance matrix

$$\mathbf{M} = \mathbf{X}\mathbf{X}^H \quad m_{ij} = \sum_t x_i(t) x_j^*(t)$$

- Decorrelation corresponds to EVD of sample covariance matrix

Convolutional Mixing

- Effects of multipath, dispersion etc



Channel Model

- Weighted sum of delayed samples (convolution)

$$x(n) = a_0 s(n) + a_1 s(n-1) + \dots + a_p s(n-p)$$

- Express in *polynomial* form (c.f. z-transform)

$$\underline{a}(z) = a_0 + a_1 z^{-1} + \dots + a_p z^{-p}$$

$$\underline{s}(z) = s(0) + s(1)z^{-1} + \dots + s(n)z^{-n} + \dots$$

$$\underline{x}(z) = x(0) + x(1)z^{-1} + \dots + x(n)z^{-n} + \dots$$

- Convolution becomes *simple product*

$$\underline{x}(z) = \underline{a}(z) \underline{s}(z)$$

- Underscore used to denote polynomial quantities

Polynomial Matrix Formulation

- Two signals and two sensors

$$\begin{aligned}\underline{x}_1(z) &= \underline{a}_{11}(z) \underline{s}_1(z) + \underline{a}_{12}(z) \underline{s}_2(z) \\ \underline{x}_2(z) &= \underline{a}_{21}(z) \underline{s}_1(z) + \underline{a}_{22}(z) \underline{s}_2(z)\end{aligned}$$

- Polynomial matrix form

$$\begin{bmatrix} \underline{x}_1(z) \\ \underline{x}_2(z) \end{bmatrix} = \begin{bmatrix} \underline{a}_{11}(z) & \underline{a}_{12}(z) \\ \underline{a}_{21}(z) & \underline{a}_{22}(z) \end{bmatrix} \begin{bmatrix} \underline{s}_1(z) \\ \underline{s}_2(z) \end{bmatrix}$$

- i.e

$$\underline{x}(z) = \underline{A}(z) \underline{s}(z)$$

Special Polynomial Matrices

- Unimodular Matrix $\det [A(z)] = \text{const}$

- Paraconjugation $\tilde{A}(z) = A_{\ast T}(1/z)$

- Para-Hermitian Matrix $\tilde{A}(z) = A(z)$

- Paraunitary matrix (defines multichannel all-pass filter)

$$H(z)\tilde{H}(z) = \tilde{H}(z)H(z) = I$$

Polynomial Matrix Examples

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

- Paraconjugate

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

- Inverse

$$\underline{A}^{-1}(z) = \begin{bmatrix} 2 & -(z + 2) \\ -z^{-1} & 1 + z^{-1} \end{bmatrix} \quad \det[\underline{A}(z)] = 1$$

- Holds for all values of z - time domain processing

Strong Decorrelation by Paraunitary Matrix

- Paraunitary transformation $H(z)x(z) = v(z)$

hence $R_{vv}(z) = H(z)R_{xx}(z)\tilde{H}(z)$ where $R_{xx}(z) = \sum_{\tau} R_{xx}(\tau)z^{-\tau}$

and $[R_{xx}(\tau)]_{ij} = E\{x_i(t)x_j^*(t-\tau)\}$

- Design $H(z)$ to achieve strong decorrelation i.e. $[R_{vv}(\tau)]_{ij} = E\{v_i(t)v_j^*(t-\tau)\} = d_i(\tau)\delta_{ij}$

- or $R_{vv}(z) = \begin{bmatrix} \underline{d}_1(z) & 0 \\ 0 & \underline{d}_p(z) \end{bmatrix}$ where $\underline{d}_i(z) = \sum_{\tau} d_i(\tau)z^{-\tau}$

Polynomial Matrix EVD

- Corresponds to diagonalising cross spectral density matrix (Laurent polynomial) by paraunitary similarity transform

$$\underline{R}_{vv}(z) = \underline{H}(z) \underline{R}_{xx}(z) \tilde{\underline{H}}(z) = \begin{bmatrix} \underline{d}_1(z) & 0 \\ 0 & \underline{d}_p(z) \end{bmatrix} \quad (z = e^{j\omega} \quad \forall \omega)$$

- Taken as definition of polynomial matrix EVD (PEVD)
- For polynomials of order zero \rightarrow scalar EVD

Elementary Paraunitary Transformation

- Applies integer delay followed by scalar rotation

$$\underline{\Gamma}(z) = \begin{bmatrix} c & s \\ -s^i & c \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^{-N} \end{bmatrix}$$

- Note that

$$\underline{\Gamma}(z) \tilde{\underline{\Gamma}}(z) = \begin{bmatrix} c & s \\ -s^i & c \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^{-N} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^N \end{bmatrix} \begin{bmatrix} c & -s \\ s^i & c \end{bmatrix} = I$$

Elementary Paraunitary Operation

- Covariance domain

$$\underline{\Gamma}(z) \begin{bmatrix} \underline{r}_{ii}(z) & \underline{r}_{ij}(z) \\ \underline{r}_{ji}(z) & \underline{r}_{jj}(z) \end{bmatrix} \tilde{\underline{\Gamma}}(z) = \begin{bmatrix} \underline{r}''_{ii}(z) & \underline{r}''_{ij}(z) \\ \underline{r}''_{ji}(z) & \underline{r}''_{jj}(z) \end{bmatrix}$$

where

$$\underline{r}_{ij}(z) = \sum_{\tau} r_{ij}(\tau) z^{-\tau} \quad r_{ij}(\tau) = E\{x_i(t)x_{j^i}(t-\tau)\}$$

and

$$\underline{r}''_{ij}(0) = 0 = \underline{r}''_{ji}(0)$$

SBR2 Algorithm

- Given the estimated space-time covariance matrix

$$r_{ij}(\tau) = \sum_{t=1}^T x_i(t) x_j(t-\tau) / T$$

- Locate dominant cross-correlation coefficient $r_{kl}(t)$
- Shift to zero-lag position by applying relative delay operator
- Apply rotation to achieve decorrelation at zero lag

$$\tan(2\theta) = 2|r_{kl}(0)| / (r_{kk}(0) - r_{ll}(0))$$

$$r'_{kl}(0)$$

- Drives $r'_{kl}(0)$ to zero and updates covariance matrix
- Elementary paraunitary operation is repeated iteratively

SBR2 Algorithm

- Single stage (assuming dominant coefficient is $r_{13}(1)$)
 - modifies every plane in polynomial matrix
 - order of polynomial grows (zero padding)

$$\begin{bmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s^i & 0 & c \end{bmatrix} \begin{bmatrix} r_{11}(0) & r_{12}(0) & r_{31}(\tau-1) \\ r_{21}(0) & r_{22}(0) & r_{32}(\tau-1) \\ r_{31}(-1) & r_{32}(-1) & r_{33}(0) \end{bmatrix} \begin{bmatrix} r_{11}(\tau) & r_{12}(\tau) & r_{13}(\tau+1) \\ r_{21}(\tau) & r_{22}(\tau) & r_{23}(\tau+1) \\ r_{32}(\tau-1) & r_{33}(\tau) & \end{bmatrix}$$

$$\begin{bmatrix} r_{11}(-\tau) & r_{12}(-\tau) & r_{13}(1-\tau) \\ r_{21}(-\tau) & r_{22}(-\tau) & r_{23}(1-\tau) \\ r_{31}(-1-\tau) & r_{32}(-1-\tau) & r_{33}(-\tau) \end{bmatrix} \begin{bmatrix} c & 0 & -s \\ 0 & 1 & 0 \\ s^i & 0 & c \end{bmatrix}$$

Cost Function for SBR2

- Zero-lag measures

$$N_1 = \sum_i |r_{ii}(0)|^2 \quad \text{invariant to delay}$$

$$N_2 = \sum_i \sum_j |r_{ij}(0)|^2 \quad \text{invariant to rotation}$$

$$N_3 = N_2 - N_1 = \sum_i \sum_{j \neq i} |r_{ij}(0)|^2$$

- Strong decorrelation measure

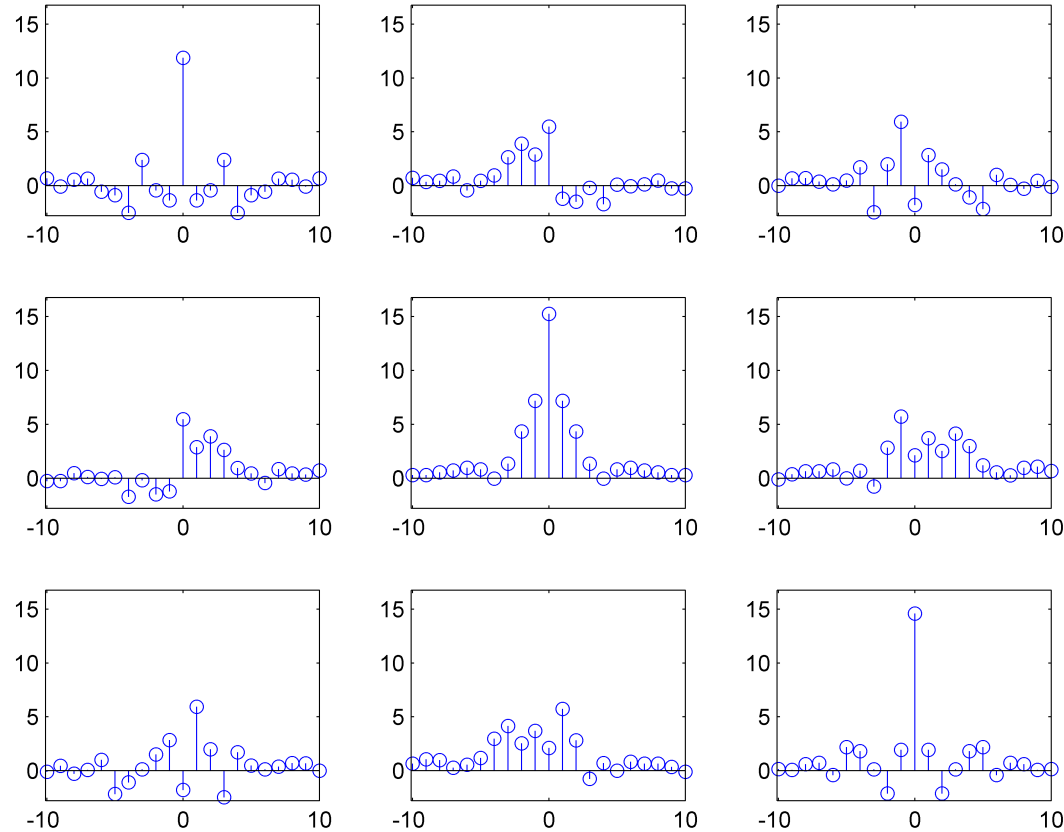
$$M = \max_{i, j, \tau} \{|r_{ij}(\tau)|\} \quad (i \neq j)$$

- Proven convergence

$$M \rightarrow 0$$

Initial Polynomial Covariance Matrix

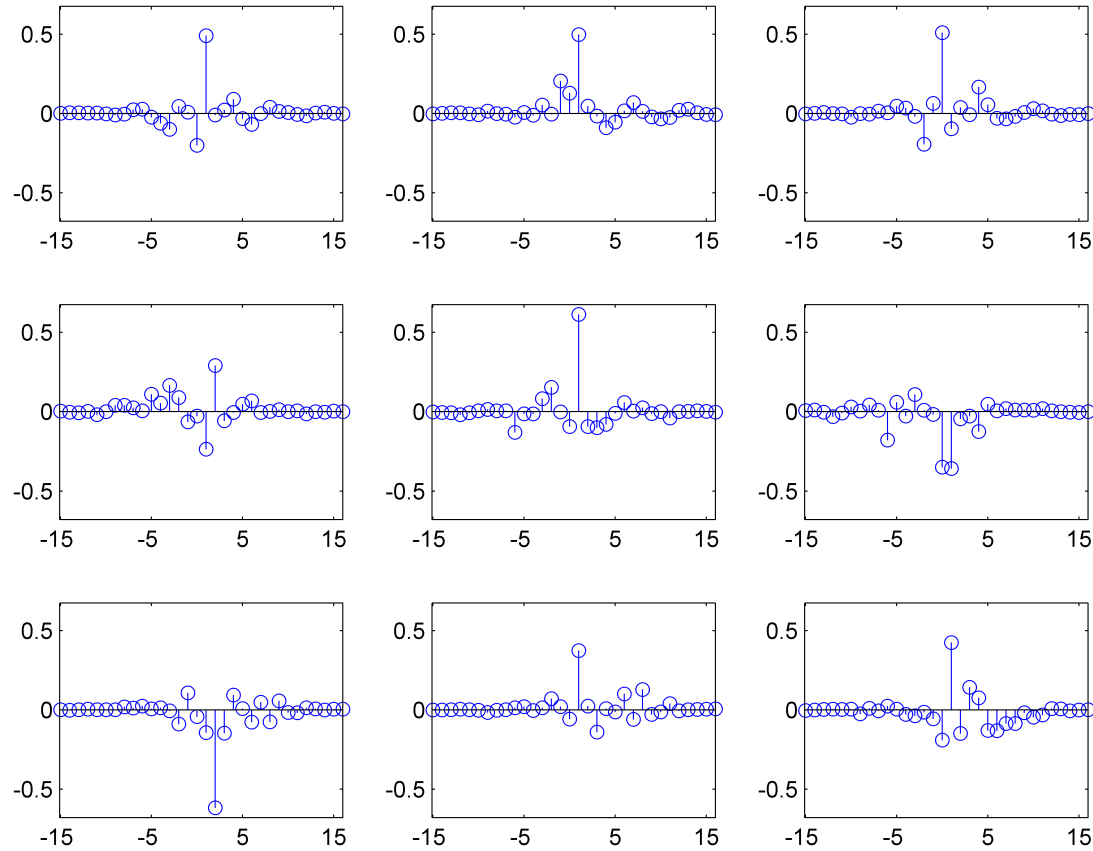
$$R_{xx}(z)$$



2 sources, 3 sensors, order 5 mixing filters, coefficients random in $[-1 \ 1]$

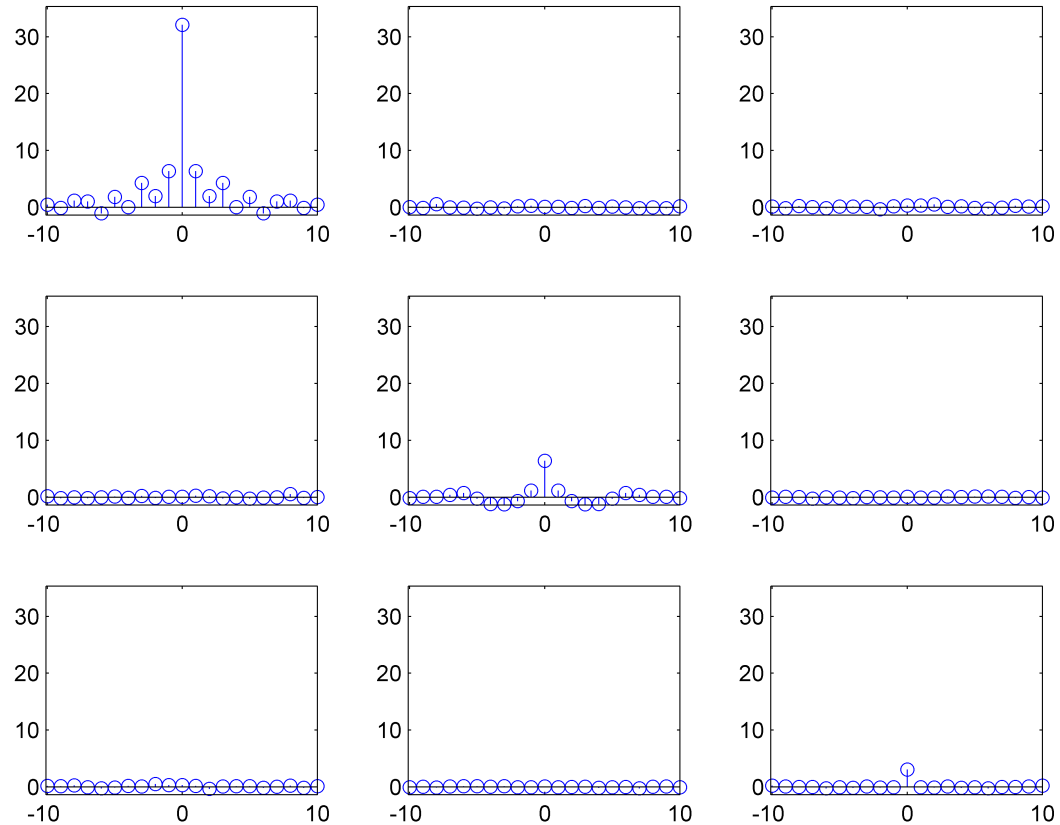
Paraunitary Transformation Matrix

$H(z)$

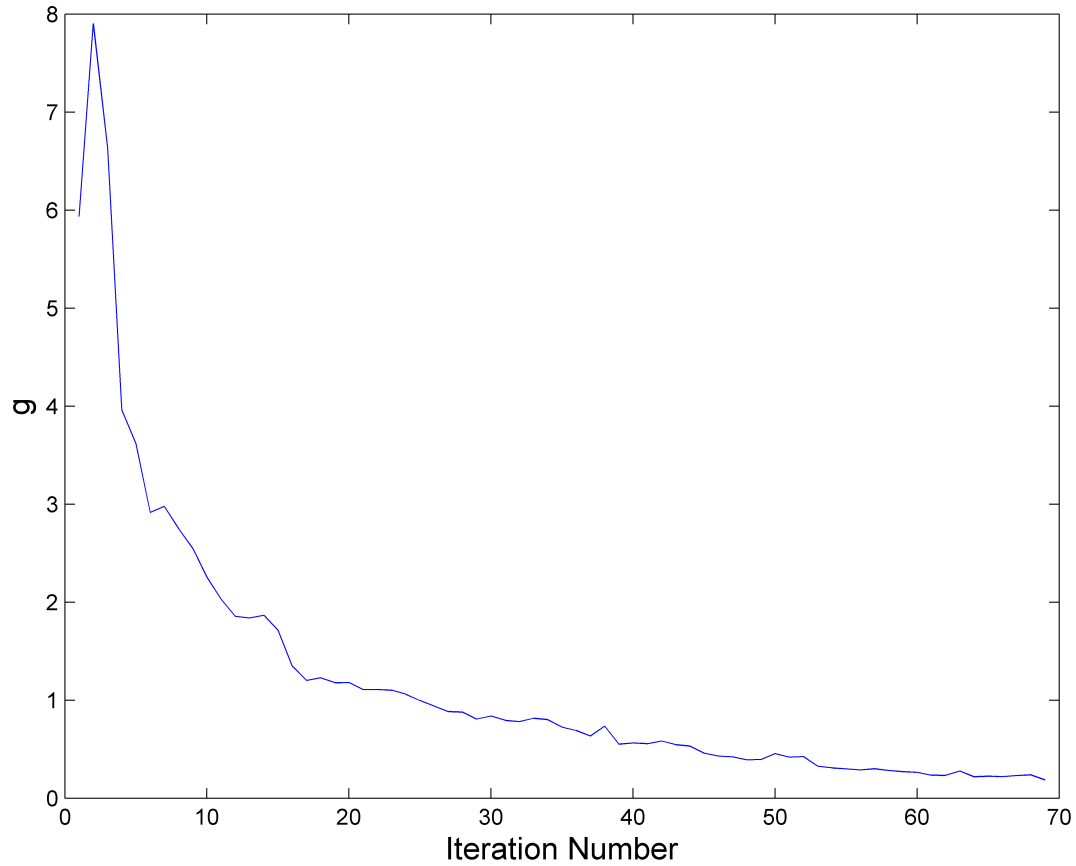


Effect of Strong Decorrelation

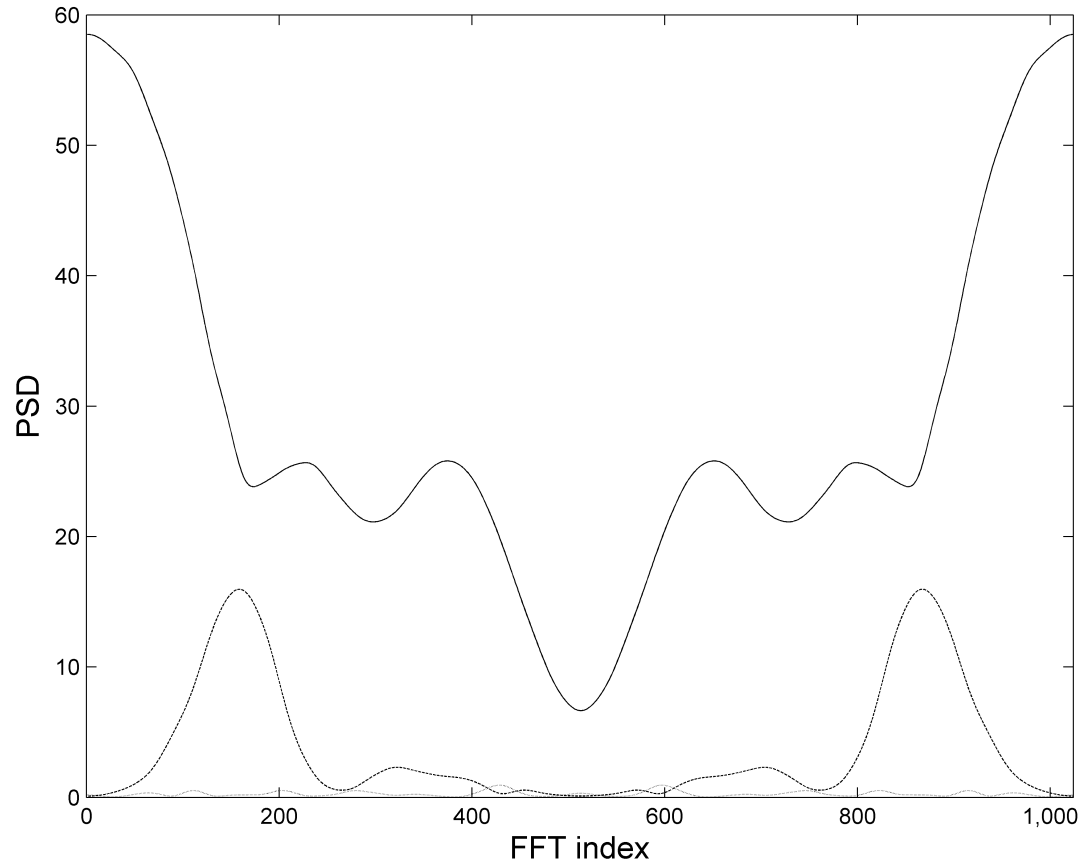
$$R_w(z)$$



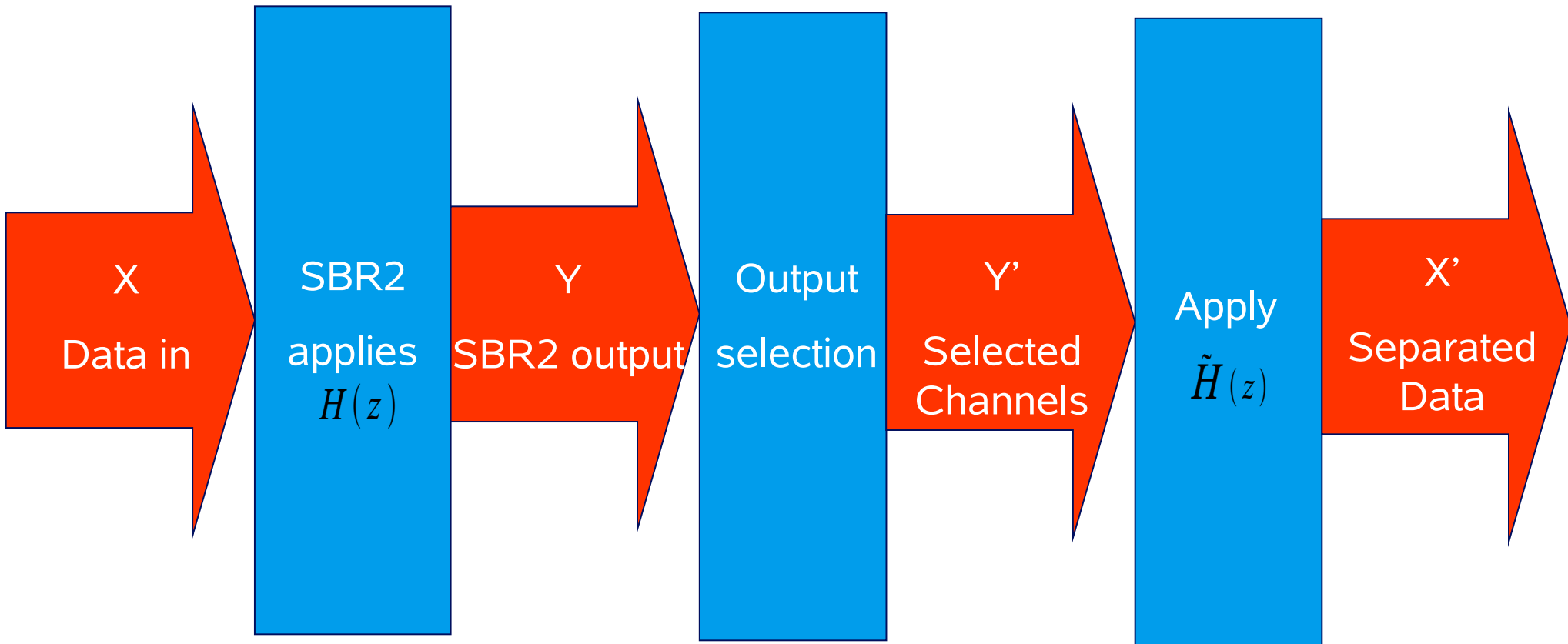
Convergence of SBR2 Algorithm



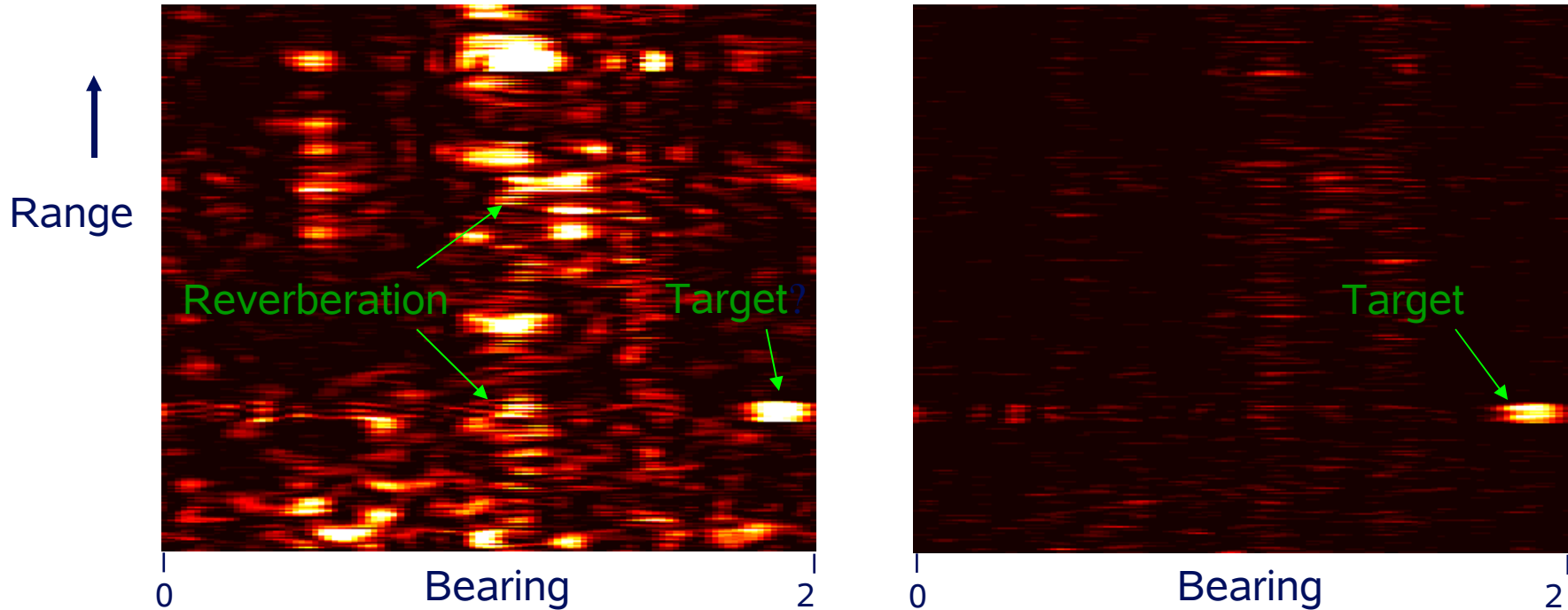
Spectral Majorisation of Output Signals



Signal separation using SBR2



Sonar reverberation suppression



- The SBR2 algorithm can be used to reduce reverberation considerably
- A target is revealed at a bearing of $\sim 343^\circ$

QR Decomposition of a Polynomial Matrix

Input matrix $\underline{A}(z) \in \underline{\mathbb{C}}^{p \times q}$

Objective is to calculate $\underline{Q}(z) \in \underline{\mathbb{C}}^{p \times p}$ such that

$$\underline{A}(z) = \underline{Q}(z) \underline{R}(z)$$

where $\underline{R}(z) \in \underline{\mathbb{C}}^{p \times q}$ is upper triangular and $\underline{Q}(z)$ is paraunitary, i.e.

$$\tilde{\underline{Q}}(z) \underline{Q}(z) = \underline{Q}(z) \tilde{\underline{Q}}(z) = I$$

Complete Polynomial Givens Rotation

$$\underline{G}(z) \begin{bmatrix} \underline{a}_1(z) \\ \underline{a}_2(z) \end{bmatrix} \simeq \begin{bmatrix} \underline{a}'_1(z) \\ 0 \end{bmatrix}$$

- Series of EPGR's $\underline{G}(z) = \hat{G}_i(z) \dots \hat{G}_1(z)$
- This matrix is paraunitary $\tilde{\underline{G}}(z) \underline{G}(z) = \underline{G}(z) \tilde{\underline{G}}(z) = I$
- All coefficients of $\underline{a}_2(z)$ have been driven arbitrarily close to zero

$$|a_2(\hat{t})| < \varepsilon$$

MIMO Channel Equalisation

Convolutional Mixing Model

$$x(t) = \sum_{k=0}^N A(k) s(t-k) + n(t)$$

Or expressed in polynomial form

$$\underline{x}(z) = \underline{A}(z) \underline{s}(z) + \underline{n}(z)$$

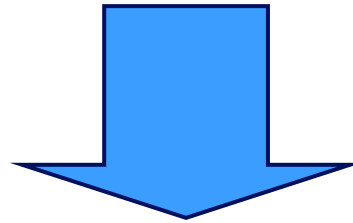
Assume channel matrix is known, then $\underline{A}(z) = \underline{Q}(z) \underline{R}(z)$

Rearranging

$$\underbrace{\tilde{\underline{Q}}(z) \underline{x}(z)}_{\underline{x}'(z)} = \underline{R}(z) \underline{s}(z) + \underbrace{\tilde{\underline{Q}}(z) \underline{n}(z)}_{\underline{n}'(z)}$$

2x2 Example

$$\begin{bmatrix} \underline{x}'_1(z) \\ \underline{x}'_2(z) \end{bmatrix} = \begin{bmatrix} \underline{r}_{11}(z) & \underline{r}_{12}(z) \\ 0 & \underline{r}_{22}(z) \end{bmatrix} \begin{bmatrix} \underline{s}_1(z) \\ \underline{s}_2(z) \end{bmatrix} + \begin{bmatrix} \underline{n}'_1(z) \\ \underline{n}'_2(z) \end{bmatrix}$$



Use **back substitution** to...

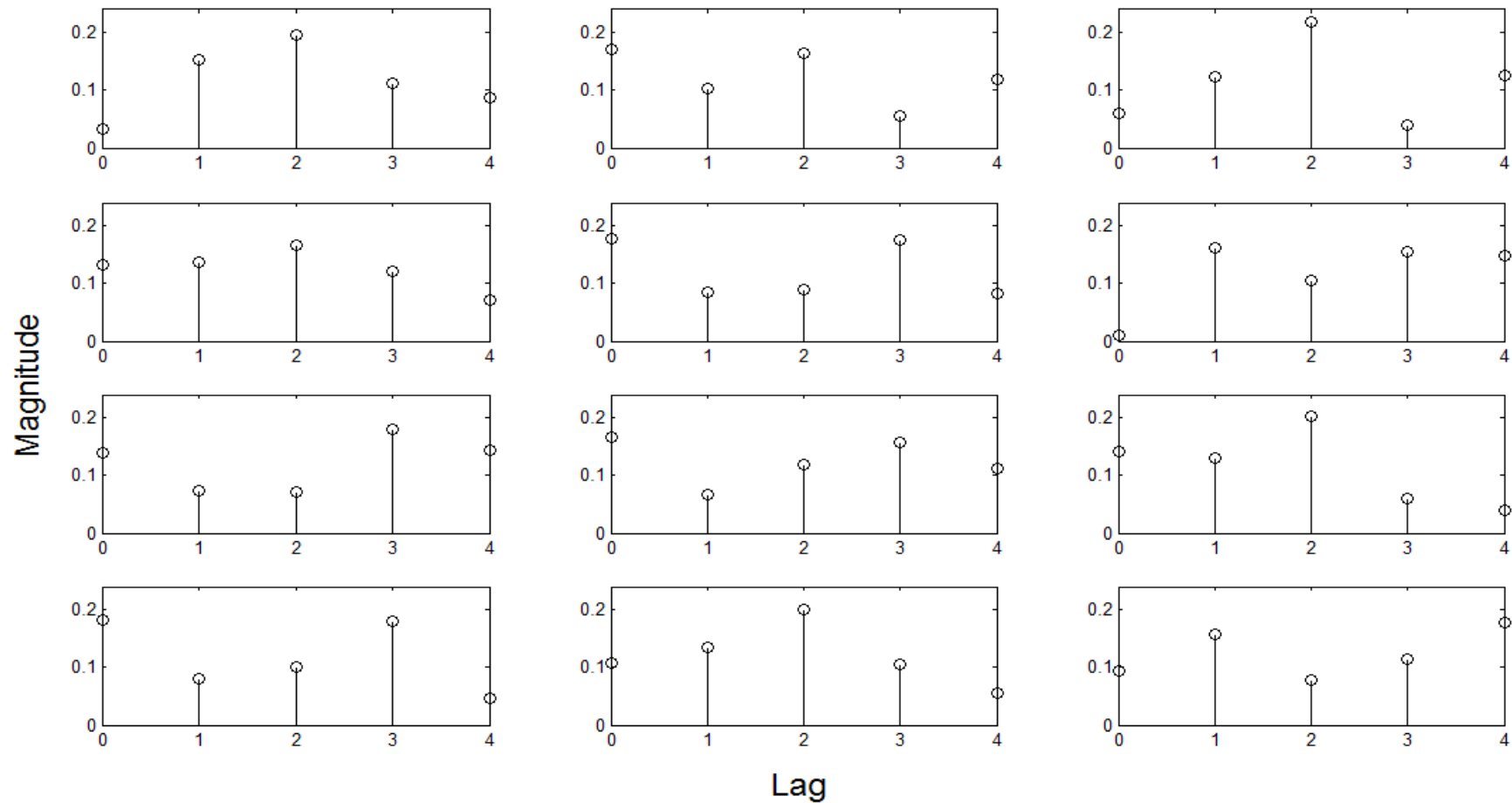
Single channel
equalisation
problem – solve
using a maximum
likelihood
sequence estimator

1. Estimate source 2 $\underline{x}'_2(z) = \underline{r}_{22}(z) \underline{s}_2(z) + \underline{n}'_2(z)$

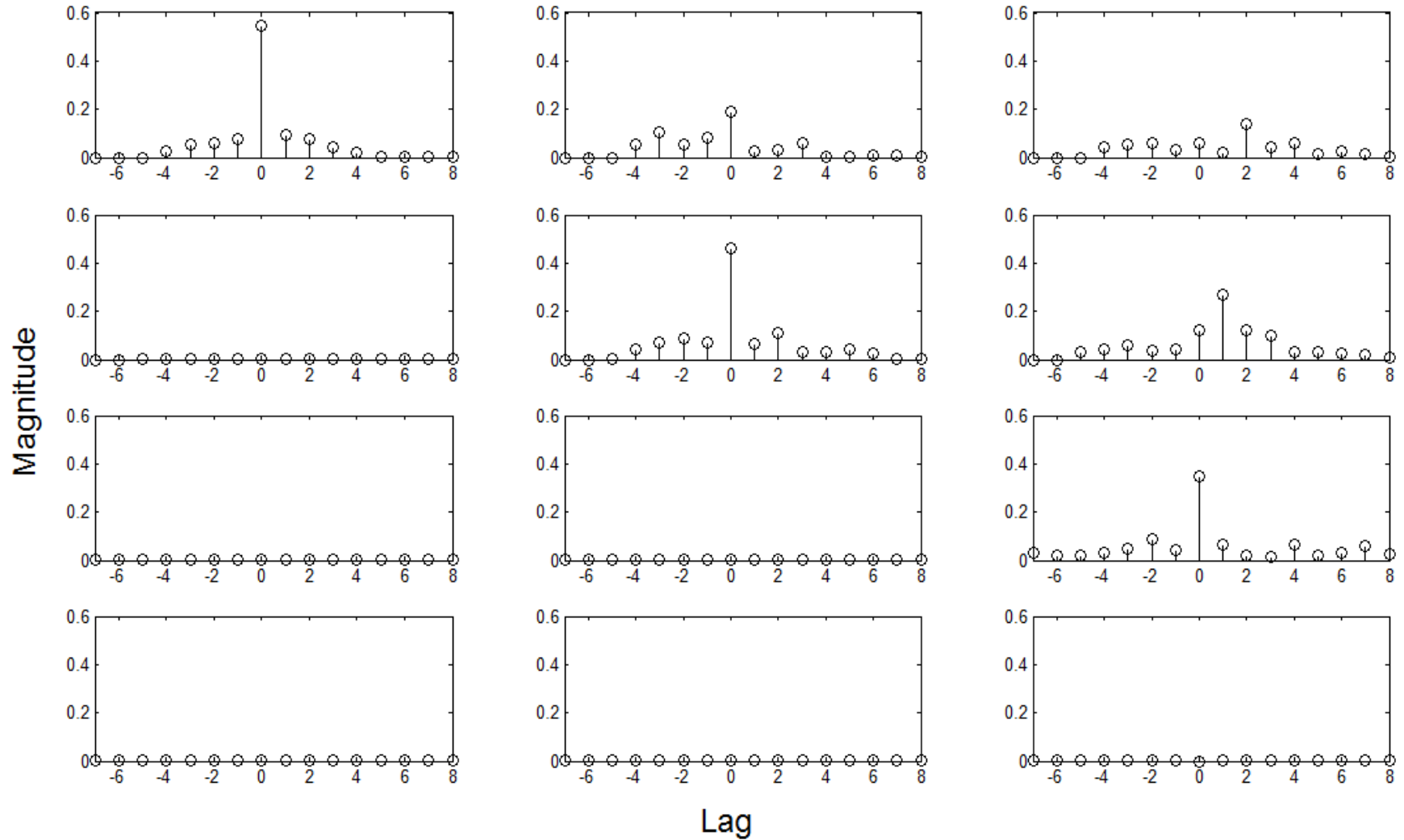
2. Estimate source 1 $\underline{x}'_1(z) - \underline{r}_{12}(z) \underline{s}_2(z) = \underline{r}_{11}(z) \underline{s}_1(z) + \underline{n}'_1(z)$

Simple Example

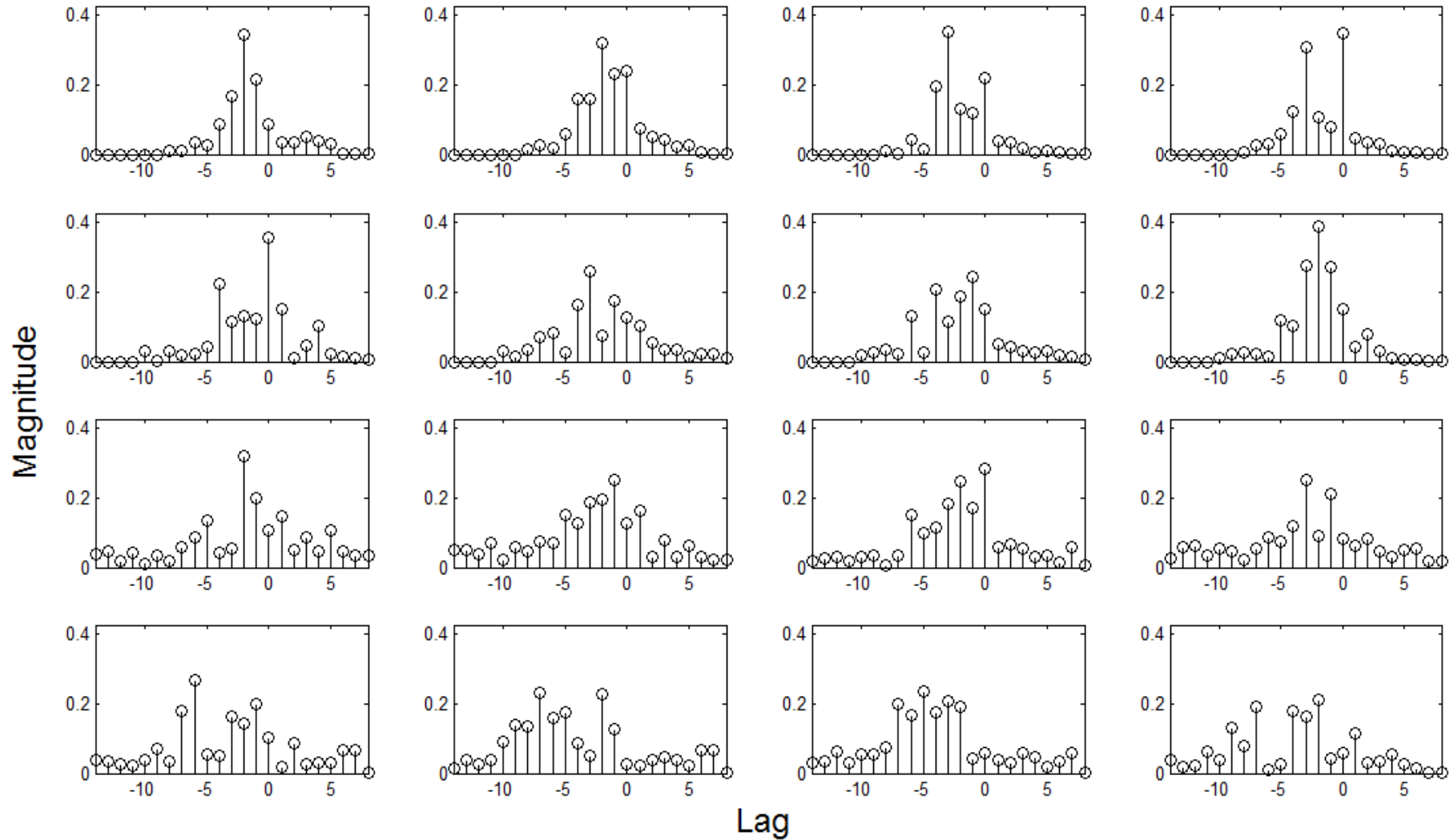
Input Matrix $\underline{A}(z)$



Upper Triangular Matrix $\underline{R}(z)$



Paraunitary Matrix $Q(z)$



Average BER for 3 Source Signals at Varying Noise Levels

RSNR (dB)	Average BER		
	Source 1	Source 2	Source 3
-5	0.1914	0.1938	0.2693
0	0.0644	0.0729	0.1470
5	0.0032	0.0056	0.0319
10	0	0	0.0007
15	0	0	0
20	0	0	0

Acknowledgements

- This research was sponsored by the United Kingdom Ministry of Defence, Corporate Research Programme.
- Reference
 - J G McWhirter, P D Baxter, T Cooper, S Redif and J Foster
“An EVD Algorithm for Para-Hermitian Polynomial Matrices”
IEEE Trans Signal Processing, Vol 55, No 6 (May 2007).
- Thanks to co-authors from QinetiQ and Cardiff University whose work is also presented here.