Polynomial Matrix Decompositions Motivated by Defence Applications

Prof J G McWhirter FRS
FREng
School of Engineering
Cardiff University

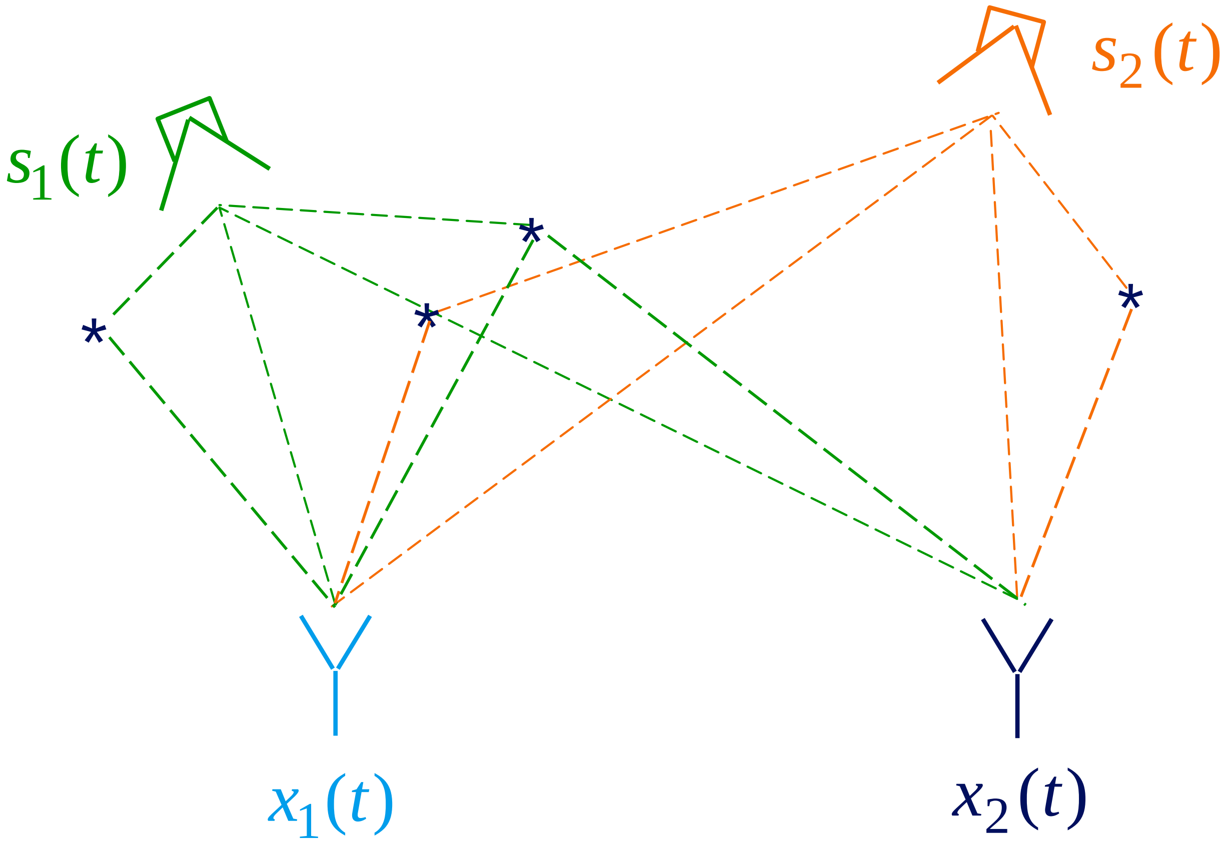
IMA, "Mathematics in Defence", Farnborough 19/11/2009

Outline of Talk

- Convolutive mixing and polynomial matrices
  - strong decorrelation by PEVD
- Sequential best rotation algorithm (SBR2)
  - convolutive signal extraction
- Application to sonobuoy array processing
- Polynomial matrix QR decomposition (PQRD)
  - MIMO channel equalisation
- Other potential military applications
- Problem with coherent research funding

Convolutive Mixing

- Effects of multipath, dispersion etc



Channel Model

- Weighted sum of delayed samples (convolution)

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

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

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

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

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

- Convolution becomes *simple product*

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

- Underscore used to denote polynomial quantities

Polynomial Matrix Formulation

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

- Two signals and two sensors

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

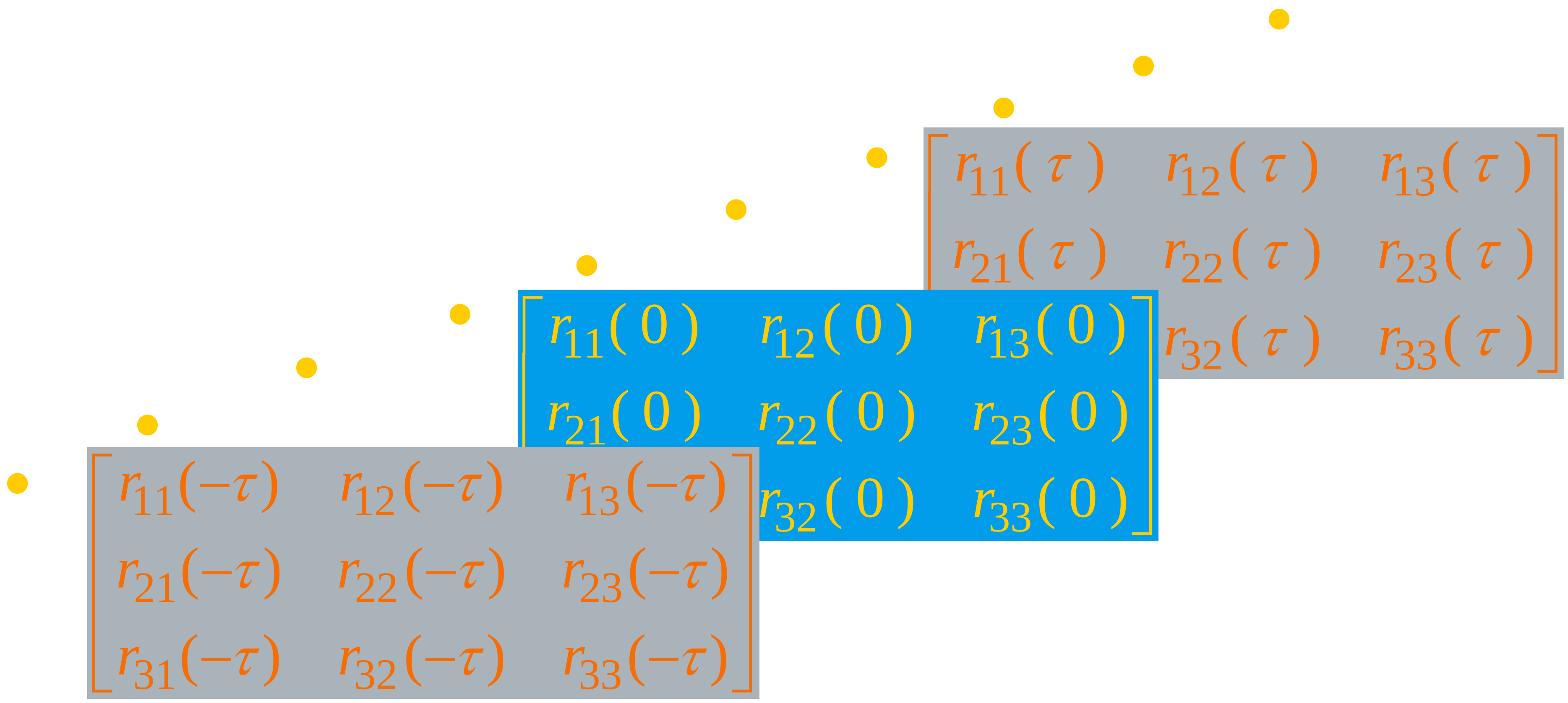
- Polynomial matrix form

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

- i.e

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

(Matrix) Polynomial (Matrix)



Special Polynomial Matrices

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

- Paraconjugation $\tilde{\underline{\mathbf{A}}}(z) = \underline{\mathbf{A}}_*{}^T(1/z)$

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

- Paraunitary matrix (defines multichannel all-pass filter)

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

Polynomial Matrix Examples

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

- Paraconjugate

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

- Inverse

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

- Holds for all values of z - time domain processing

Polynomial Matrix EVD

$$\underline{\mathbf{R}}_{vv}(z) = \underline{\mathbf{H}}(z)\underline{\mathbf{R}}_{xx}(z)\underline{\tilde{\mathbf{H}}}(z) = \begin{bmatrix} \underline{d}_1(z) & 0 \\ 0 & \underline{d}_p(z) \end{bmatrix}$$

- $\underline{\mathbf{H}}(z)$ is a paraunitary; $\underline{\mathbf{R}}_{xx}(z)$ is para-Hermitian
- Taken as definition of polynomial matrix EVD (PEVD)
- For polynomials of order zero $\longrightarrow$ scalar EVD
- Corresponds to diagonalising cross spectral density matrix (Laurent polynomial) by paraunitary similarity transform

$$[\mathbf{R}_{xx}(\tau)]_{ij} = \mathrm{E}\{x_i(t)x_j^*(t-\tau)\} \qquad \underline{\mathbf{R}}_{xx}(z) = \sum_{\tau}\mathbf{R}_{xx}(\tau)z^{-\tau}$$

Elementary Paraunitary Transformation

- Applies integer delay followed by scalar rotation

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

- Note that

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

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

$$\tan(2\theta) = 2\left|r'_{kl}(0)\right| / \left(r_{kk}(0) - r_{ll}(0)\right)$$

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

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

SBR2 Algorithm

- Initial Polynomial covariance matrix

$$\begin{bmatrix} r_{11}(\tau) & r_{12}(\tau) & r_{13}(\tau) \\ r_{21}(\tau) & r_{22}(\tau) & r_{23}(\tau) \\ & r_{32}(\tau) & r_{33}(\tau) \end{bmatrix}$$

$$\begin{bmatrix} r_{11}(0) & r_{12}(0) & r_{13}(0) \\ r_{21}(0) & r_{22}(0) & r_{23}(0) \\ & r_{32}(0) & r_{33}(0) \end{bmatrix}$$

$$\begin{bmatrix} r_{11}(-\tau) & r_{12}(-\tau) & r_{13}(-\tau) \\ r_{21}(-\tau) & r_{22}(-\tau) & r_{23}(-\tau) \\ r_{31}(-\tau) & r_{32}(-\tau) & r_{33}(-\tau) \end{bmatrix}$$

SBR2 Algorithm

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

$$\begin{bmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s^* & 0 & c \end{bmatrix}$$

$$\begin{bmatrix} r_{11}(\tau) & r_{12}(\tau) & r_{13}(\tau+1) \\ r_{21}(\tau) & r_{22}(\tau) & r_{23}(\tau+1) \\ & r_{32}(\tau-1) & r_{33}(\tau) \end{bmatrix}$$

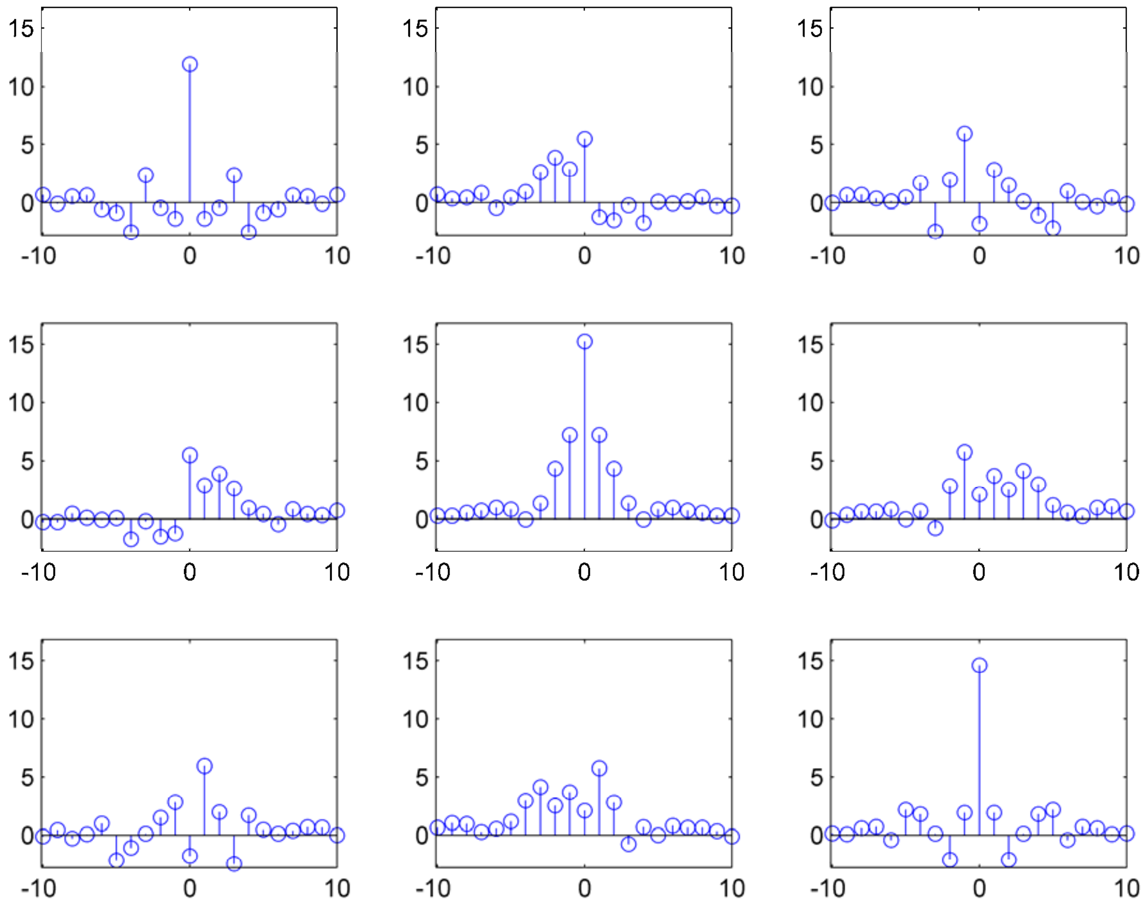
$$\begin{bmatrix} r_{11}(0) & r_{12}(0) & r_{13}(1) \\ r_{21}(0) & r_{22}(0) & r_{23}(1) \\ & & r_{33}(0) \end{bmatrix}$$

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

$$\begin{bmatrix} c & 0 & -s \\ 0 & 1 & 0 \\ s^* & 0 & c \end{bmatrix}$$

Initial Polynomial Covariance Matrix

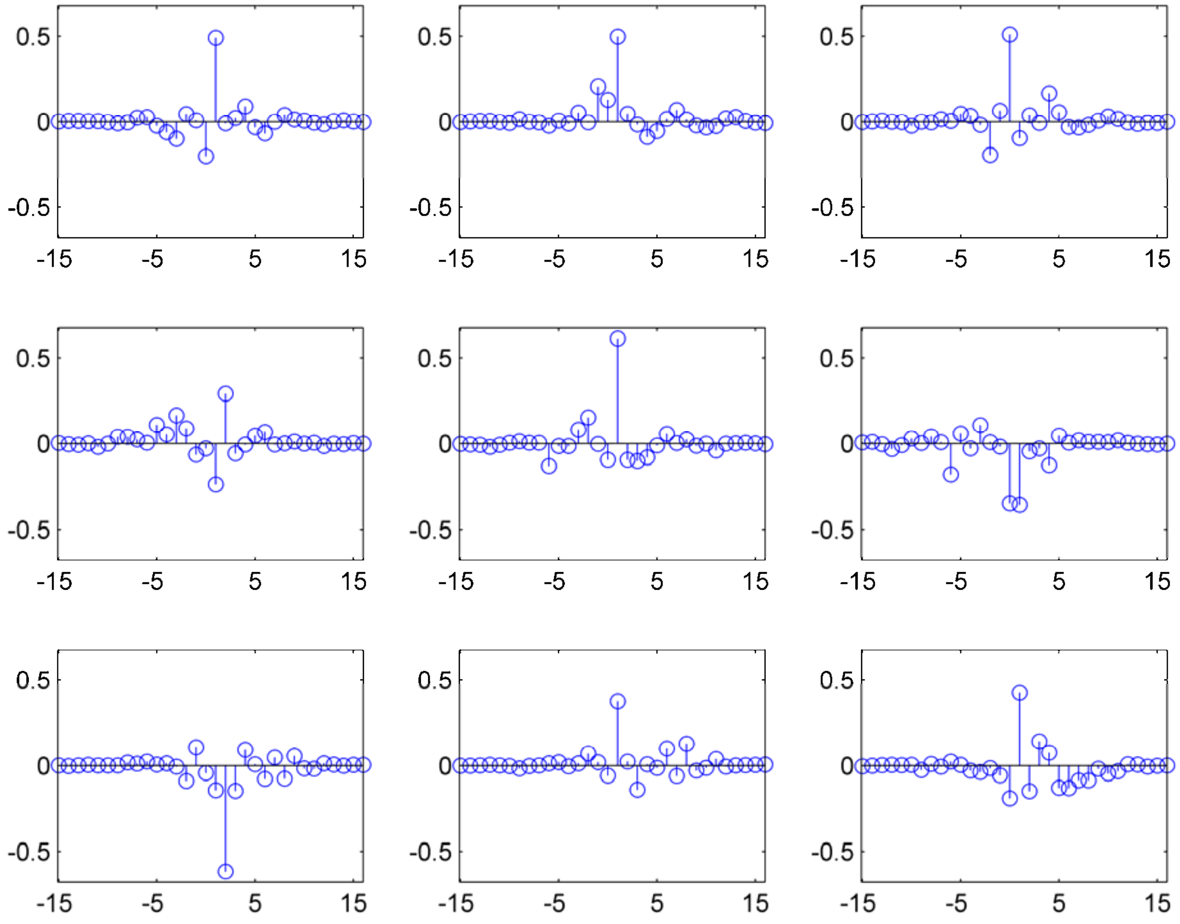
$\underline{\mathbf{R}}_{xx}(z)$



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

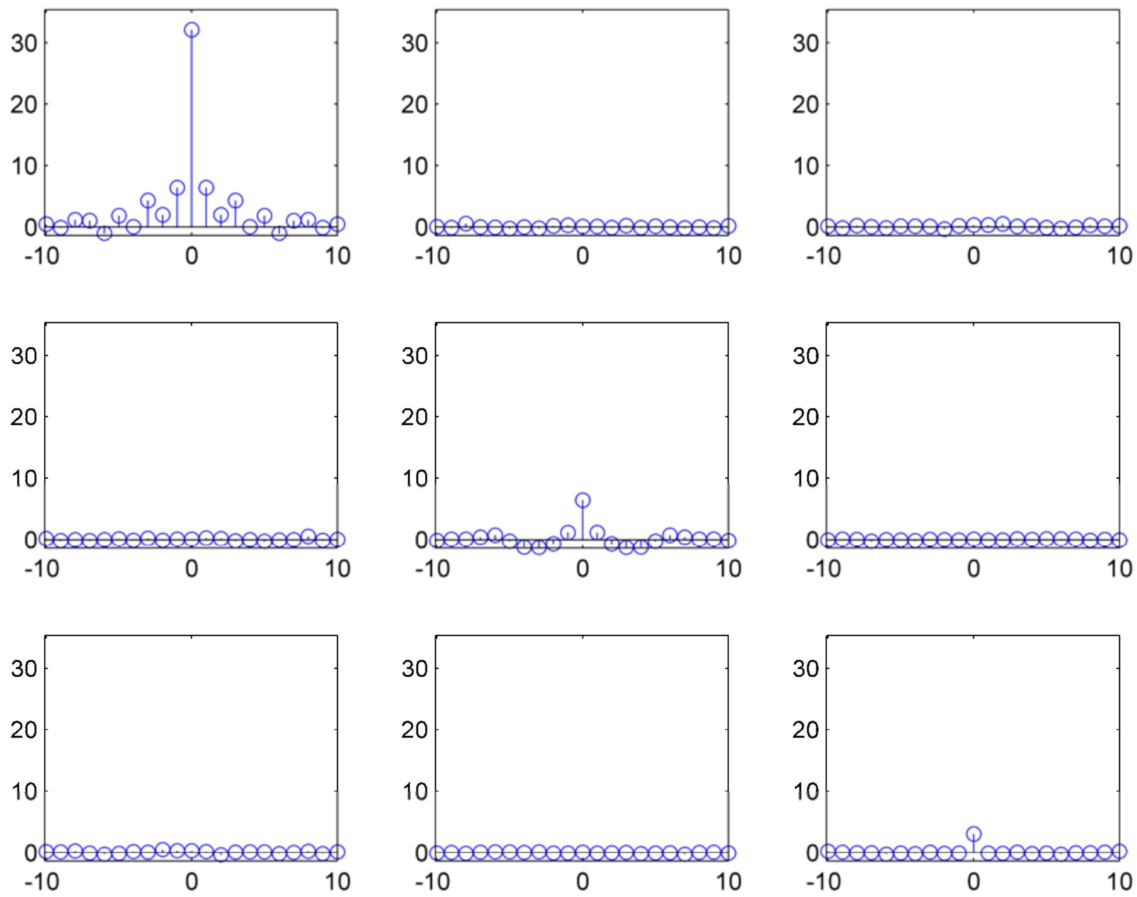
Paraunitary Transformation Matrix

$\underline{\mathbf{H}}(z)$

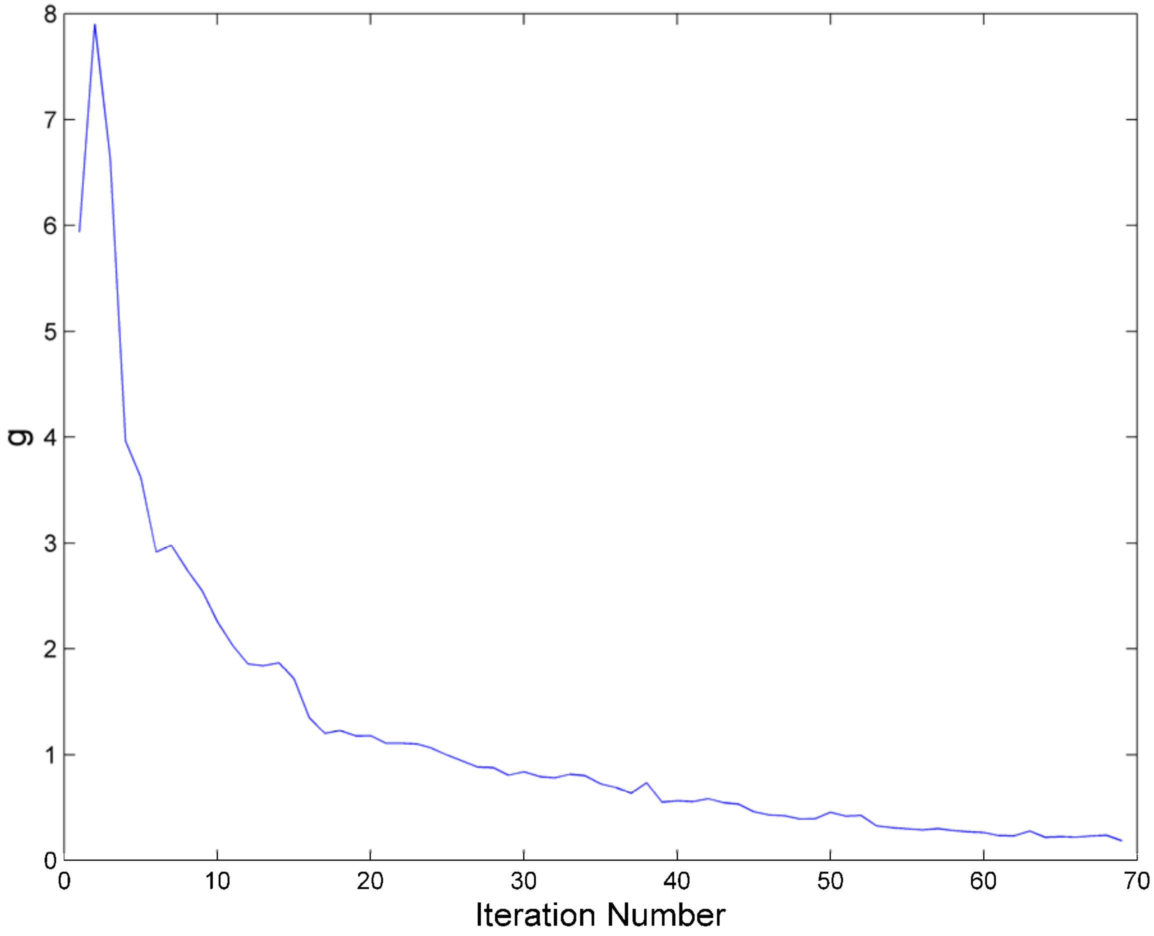


Effect of Strong Decorrelation

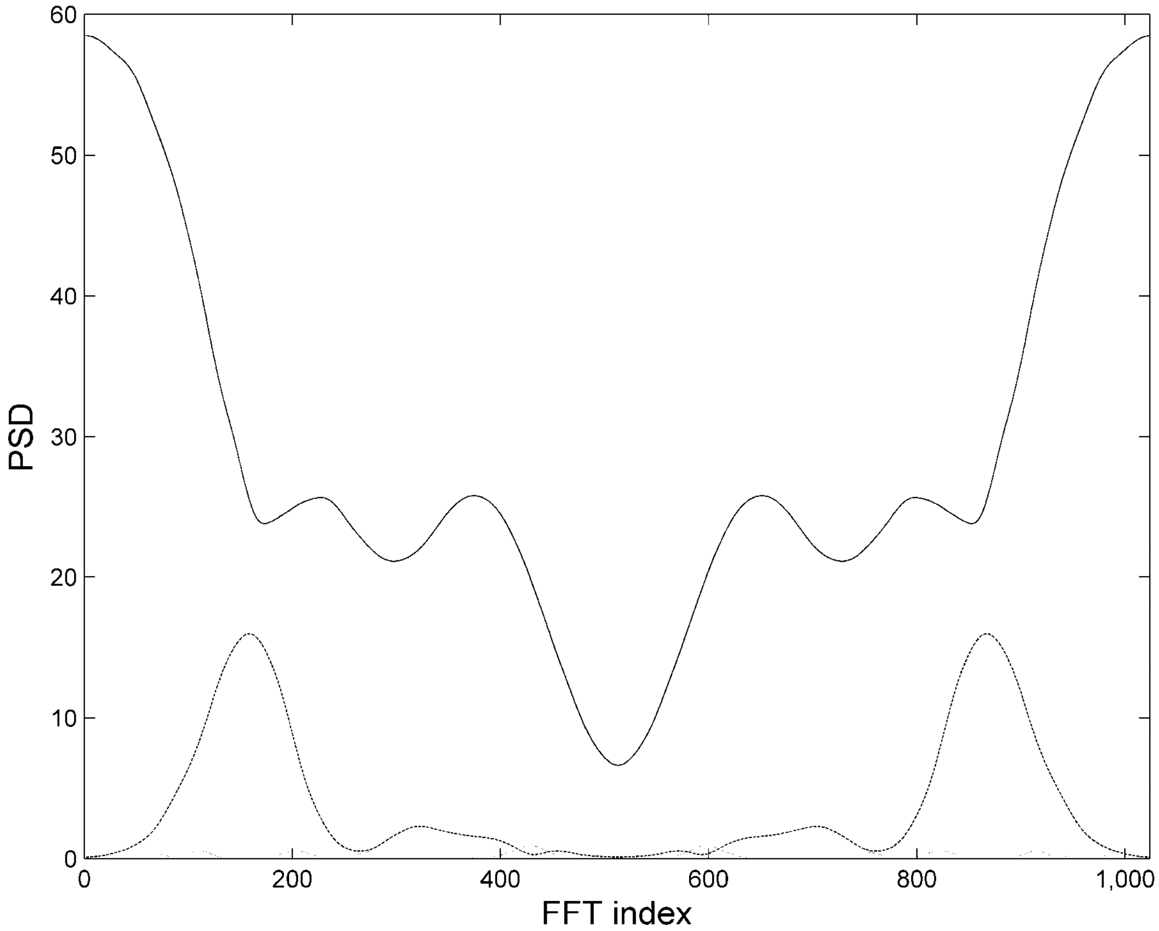
$\underline{\mathbf{R}}_{vv}(z)$



Convergence of SBR2 Algorithm

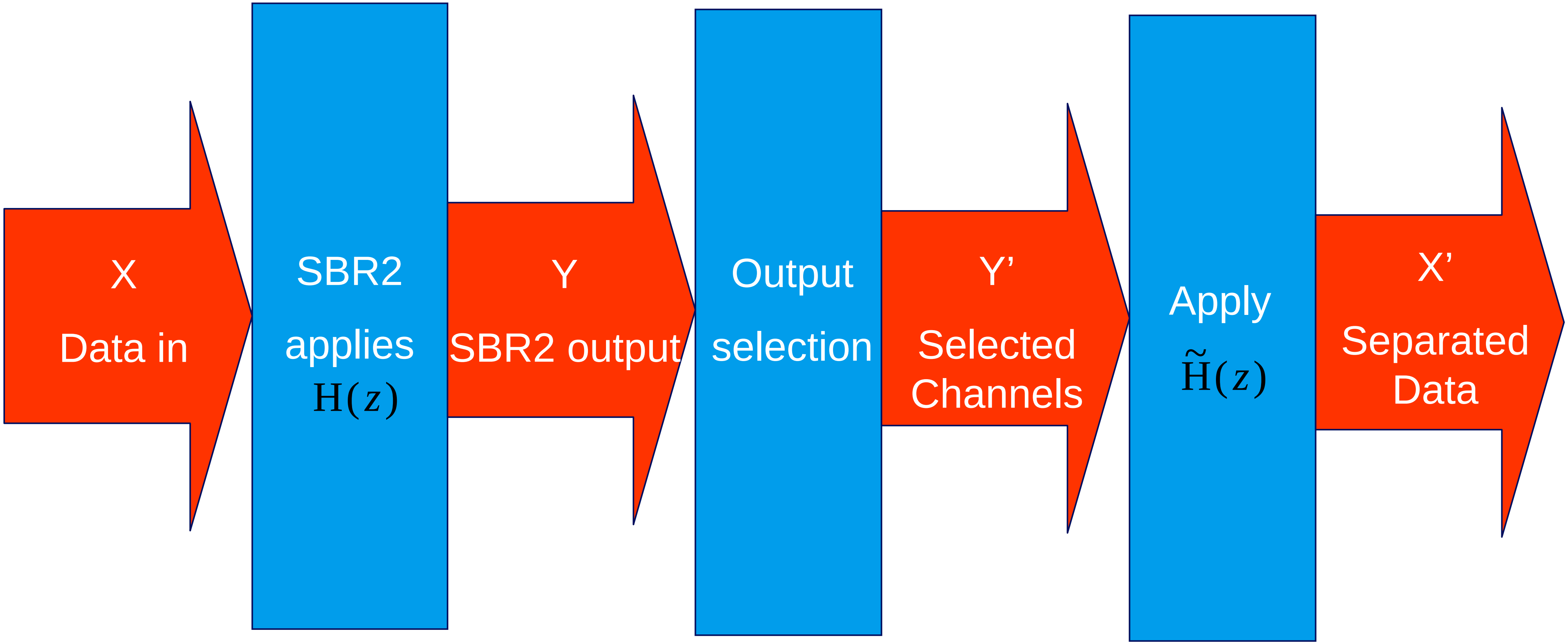


Spectral Majorisation of Output Signals

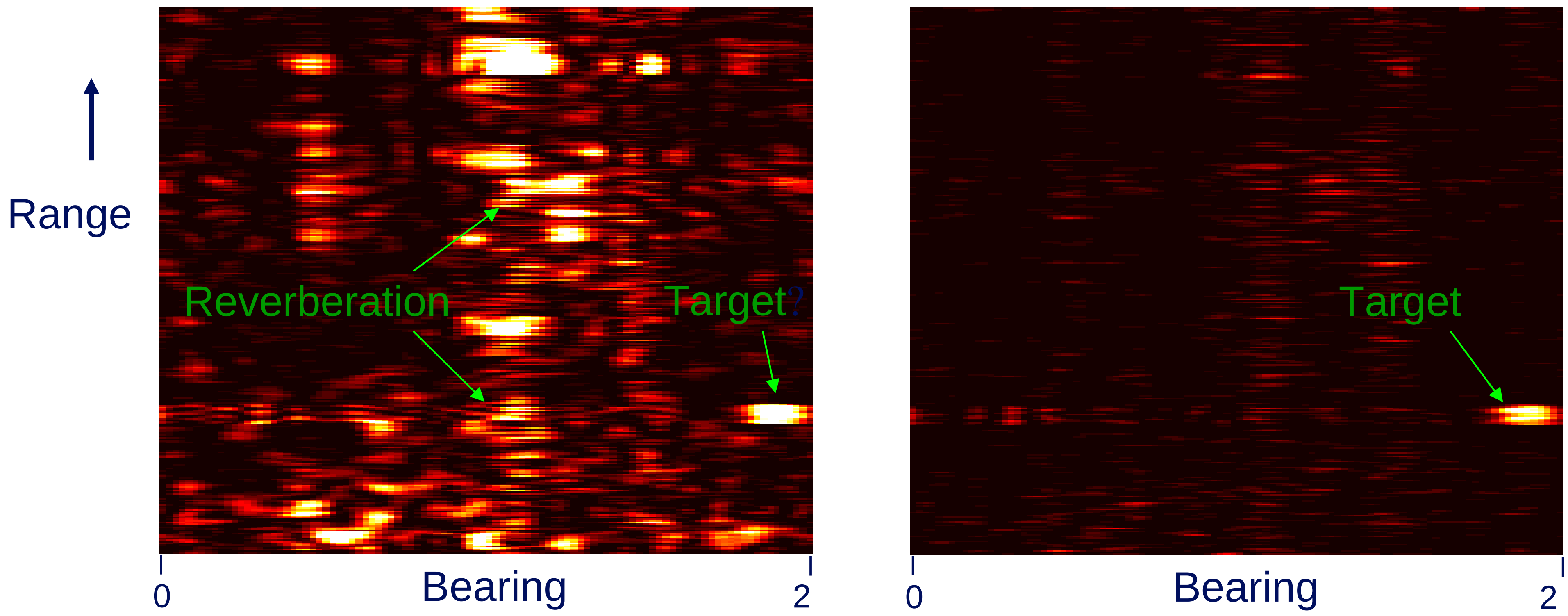


Sonobuoy Array

Signal separation using SBR2



Sonar reverberation suppression



- The SBR2 algorithm can been used to reduce reverberation considerably
- A target is revealed at a bearing of ~343º

QR Decomposition of a Polynomial Matrix

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

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

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

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

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

MIMO Channel Equalisation

Convolutive Mixing Model

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

Or expressed in polynomial form

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

Assume channel matrix is known, then

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

Rearranging

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

2x2 Example

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

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

Use **back substitution** to...

1. Estimate source 2

$$\underline{x}_2^{'}(z) = \underline{r}_{22}(z)\underline{s}_2(z) + \underline{n}_2^{'}(z)$$

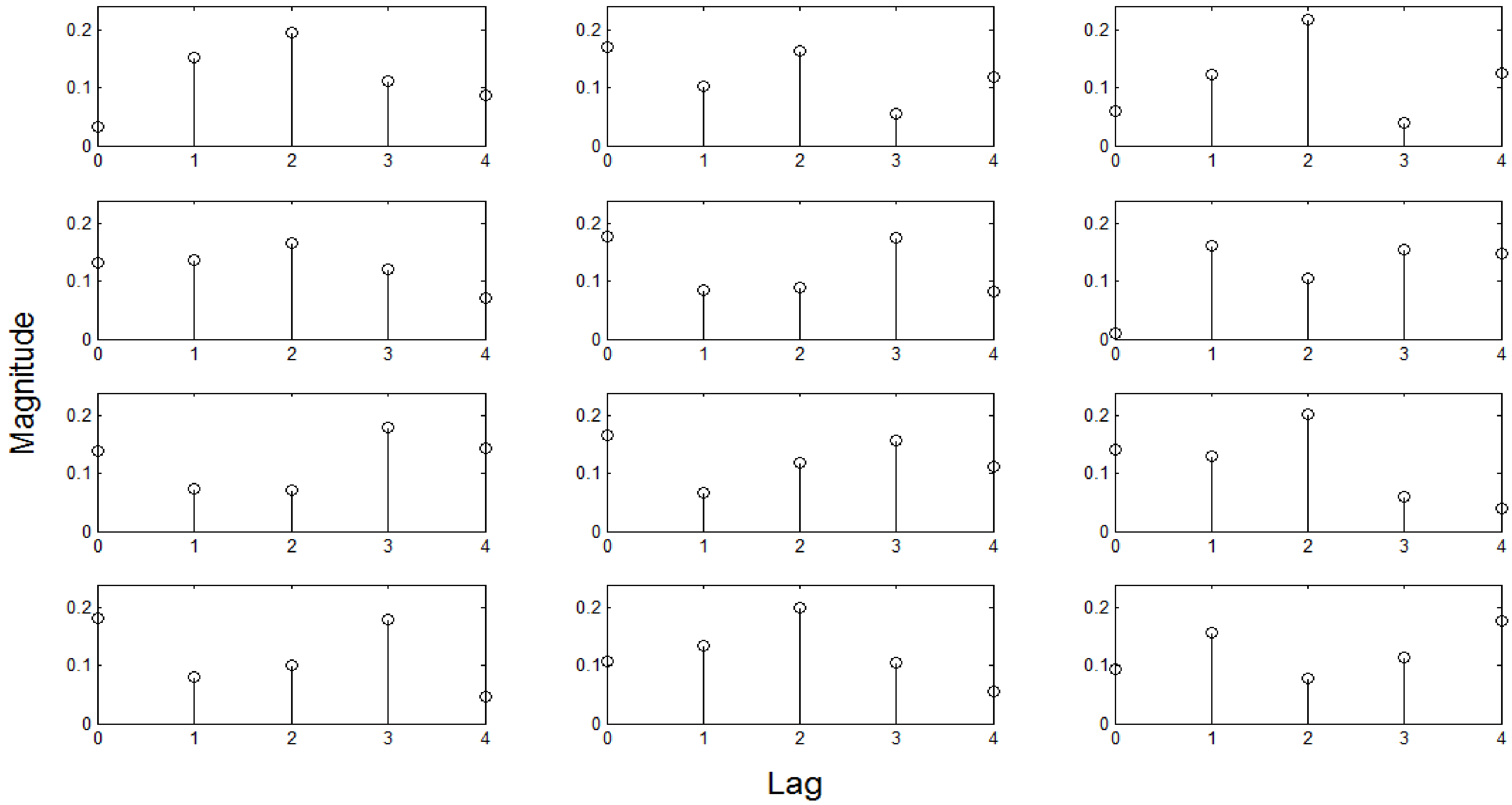
Single channel equalisation problem – solve using a maximum likelihood sequence estimator

2. Estimate source 1

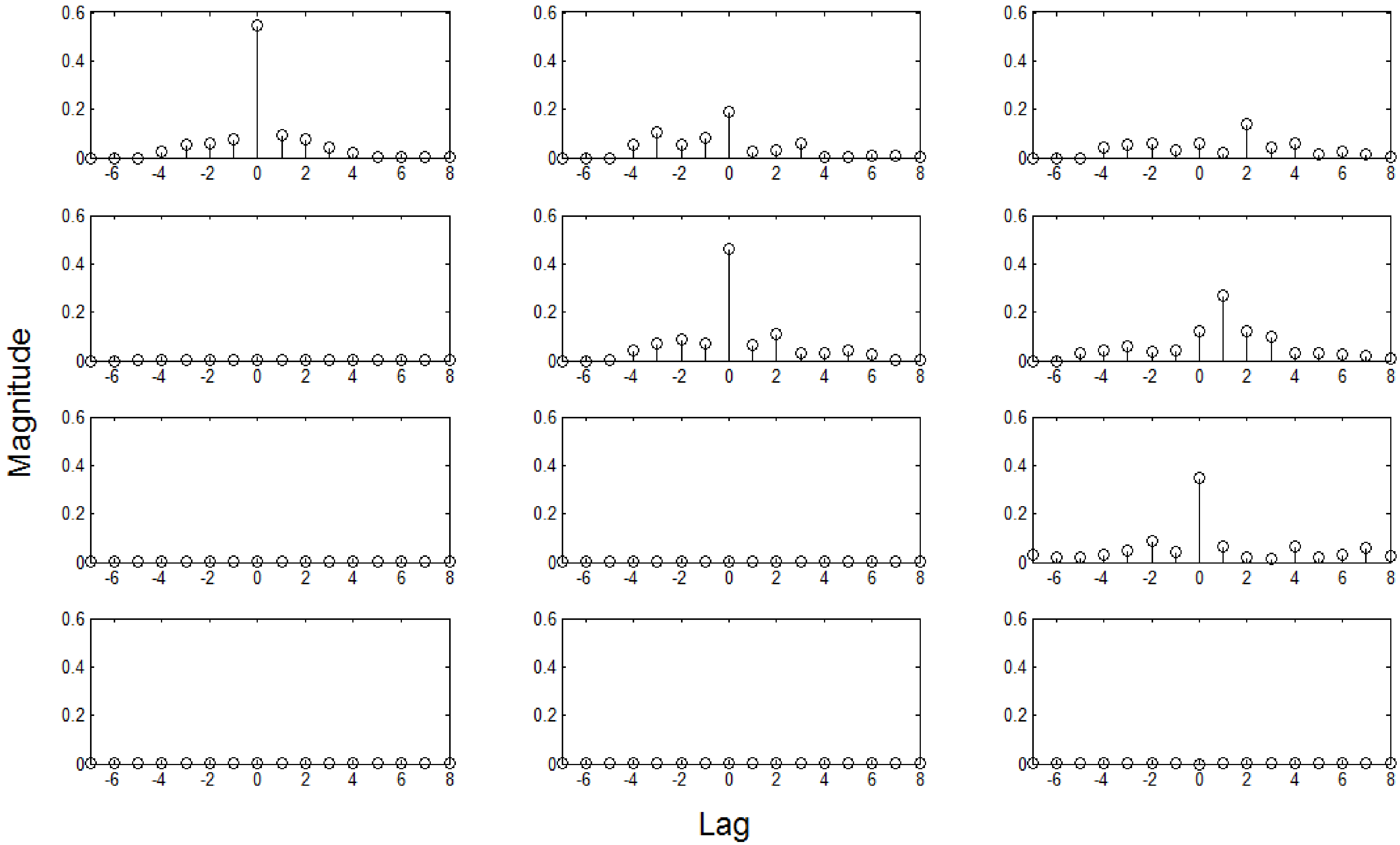
$$\underline{x}_1^{'}(z) - \underline{r}_{12}(z)\underline{s}_2(z) = \underline{r}_{11}(z)\underline{s}_1(z) + \underline{n}_1^{'}(z)$$

Simple Example

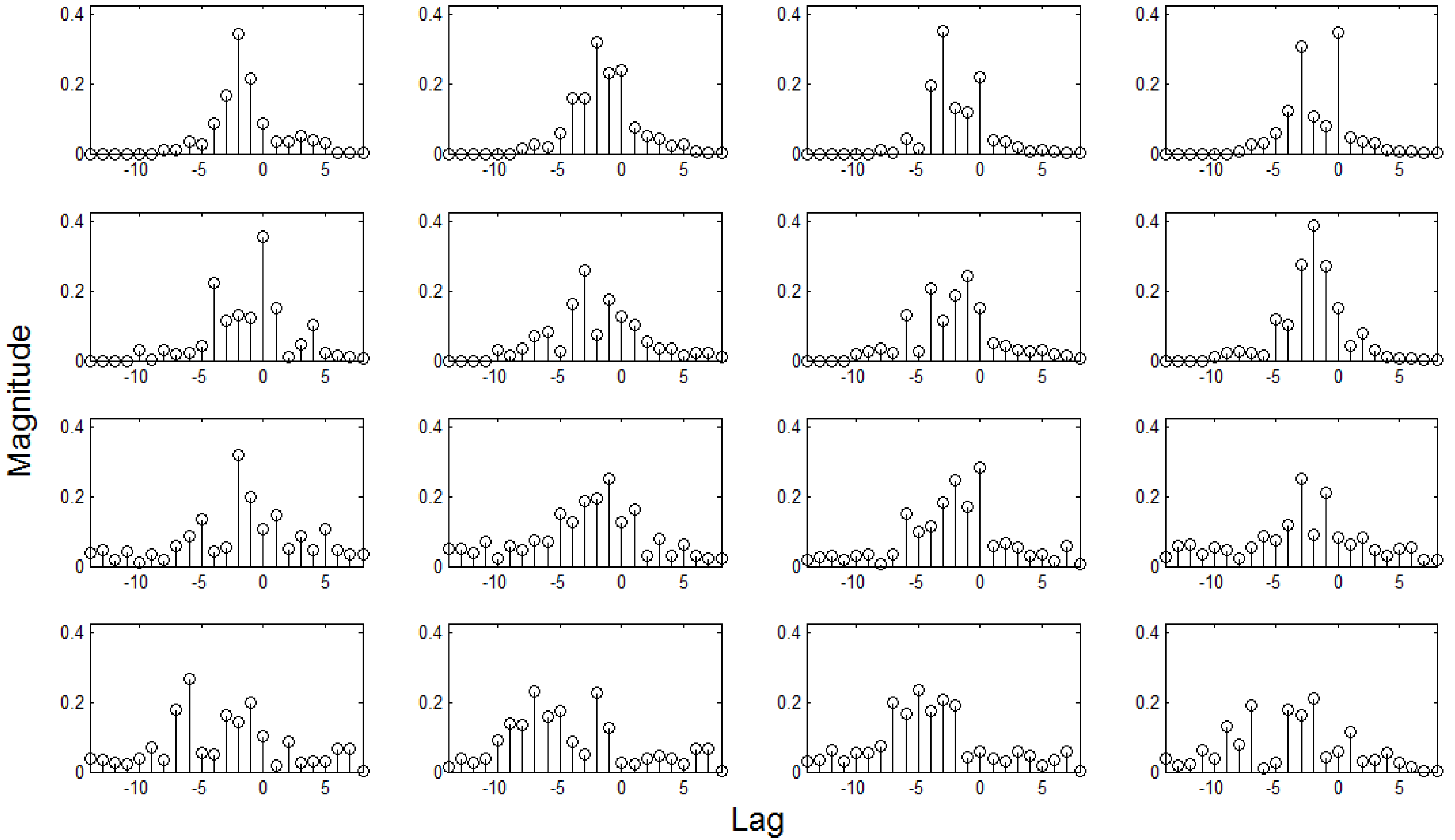
Input Matrix $\underline{\mathbf{A}}(z)$



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



Paraunitary Matrix **Q**(*z*)



Potential Military Applications

- Sonobuoy array processing
- Sonar towed array processing
- Sonar flank array processing
- Underwater acoustic communications
- STAP radar
- MIMO communication networks
- Monitoring seismic events (nuclear test ban treaty)
- Acoustic monitoring for ground asset or harbour protection

Important aspects to note

- Mathematical research motivated by military problems
- Initial research produces some “practical” algorithms
- Early simulations demonstrate potential of new approach
- Need for more fundamental mathematical analysis
- Opens up possibility of whole new research topic
  – Other numerical algorithms and approaches?
  – Stronger convergence and approximation results?
- Numerous applications to be investigated
- But still very difficult to obtain coherent funding

Acknowledgements

- This research was sponsored by the United Kingdom Ministry of Defence, Corporate Research Programme and also by EPSRC.
- Thanks to co-authors from QinetiQ and Loughborough University whose work is also presented here.
- 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).