# Polynomial Matrix Decompositions: Algorithms and Applications

**Joanne Foster – Loughborough University**

**John McWhirter – Cardiff University**

**Jonathon Chambers – Loughborough University**

http://www.lboro.ac.uk/departments/el/research/asp/ProjSite/index.htm

Email: J.A.Foster@Lboro.ac.uk

# Talk Outline

- What are polynomial matrices?
- Three types of polynomial matrix decompositions:
  - ✓ Eigenvalue decomposition (SBR2 Algorithm)
  - QR decomposition
  - Singular value decomposition
- The potential applications of these decompositions to MIMO communication problems

# What is a Polynomial Matrix?

- 2x2 example:

$$\underline{A}(z)=\begin{bmatrix} 1+z^{-1} & 2+3z^{-2} \\ 2+z^{-1}+z^{-2} & 4+z^{-1} \end{bmatrix}$$

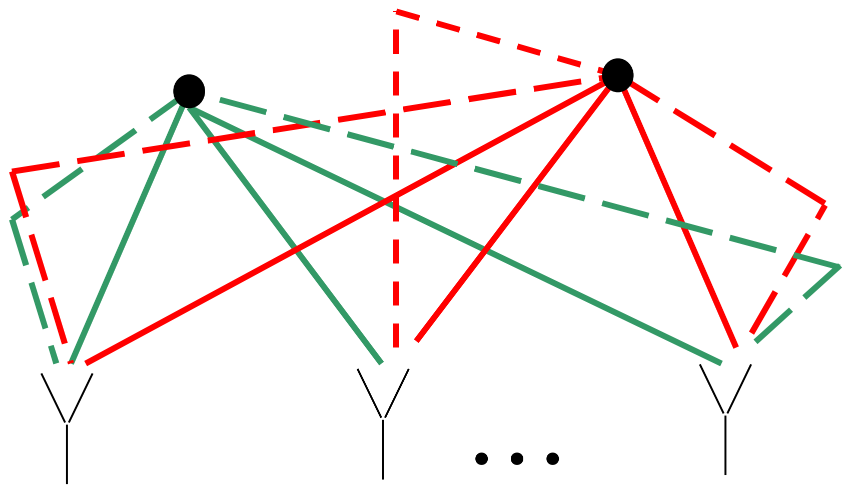
- General *pxq* polynomial matrix form:

$$\underline{A}(z)=\begin{bmatrix} \underline{a}_{11}(z) & \cdots & \underline{a}_{1q}(z) \\ \vdots & & \vdots \\ \underline{a}_{p1}(z) & \cdots & \underline{a}_{pq}(z) \end{bmatrix} = \sum_{t=T_{\min}}^{T_{\max}} A(t)z^{-t}$$

$$T_{\min} \le T_{\max}$$

# How do polynomial matrices arise?

- Used to describe convolutive mixing:



- Source signals received at sensors over multiple paths and with different delays
- Polynomial mixing matrix required, where each element is an FIR filter
- More realistic mixing situation

# Polynomial Matrix Formulation

- 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.

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

# Special Polynomial Matrices

- Unimodular Matrix: $$\det[A(z)] = \text{constant}$$
- Paraconjugation: $$\tilde{A}(z) = A_{\text{¿}}^{T}(1/z)$$
- 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

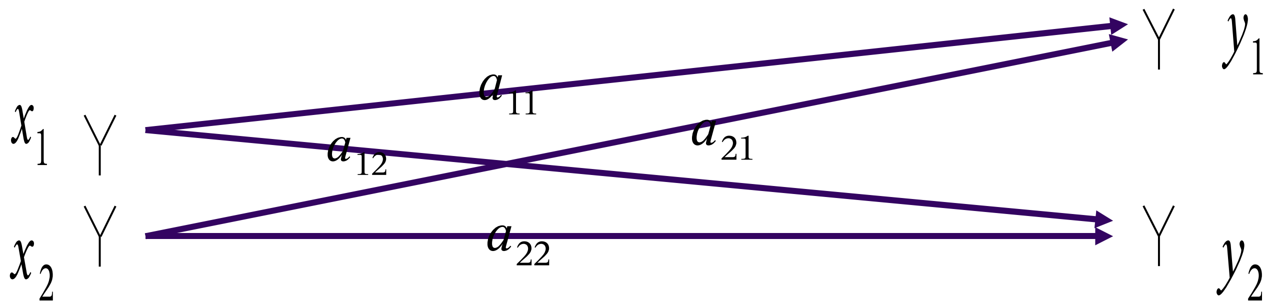
$$\underline{A}(z)=\begin{bmatrix}1+z^{-1} & z+2\\ z^{-1} & 2\end{bmatrix}$$

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

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

- Holds for all values of z - time domain processing

# Motivation: the QR decomposition in narrowband signal processing




- Can be expressed as: $$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (*)$$
- Calculate the decomposition of **A**: $$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{bmatrix}$$
- Rearrange (*): $y' = Q^H y = \mathbf{Rx}$ then $$y'_2 = r_{22} x_2$$ $$y'_1 = r_{11} x_1 + r_{12} x_2$$
- This application could be extended to sets of polynomial equations, but would require new algorithms for formulating the decompositions

# The Polynomial Matrix QR Decomposition (PQRD)

- The PQRD of a *pxq* polynomial matrix **A**(*z*) can be expressed as

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

- where **Q**(*z*) is a *pxp* paraunitary matrix,

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

- and **R**(z) is a *pxq* upper triangular polynomial matrix.

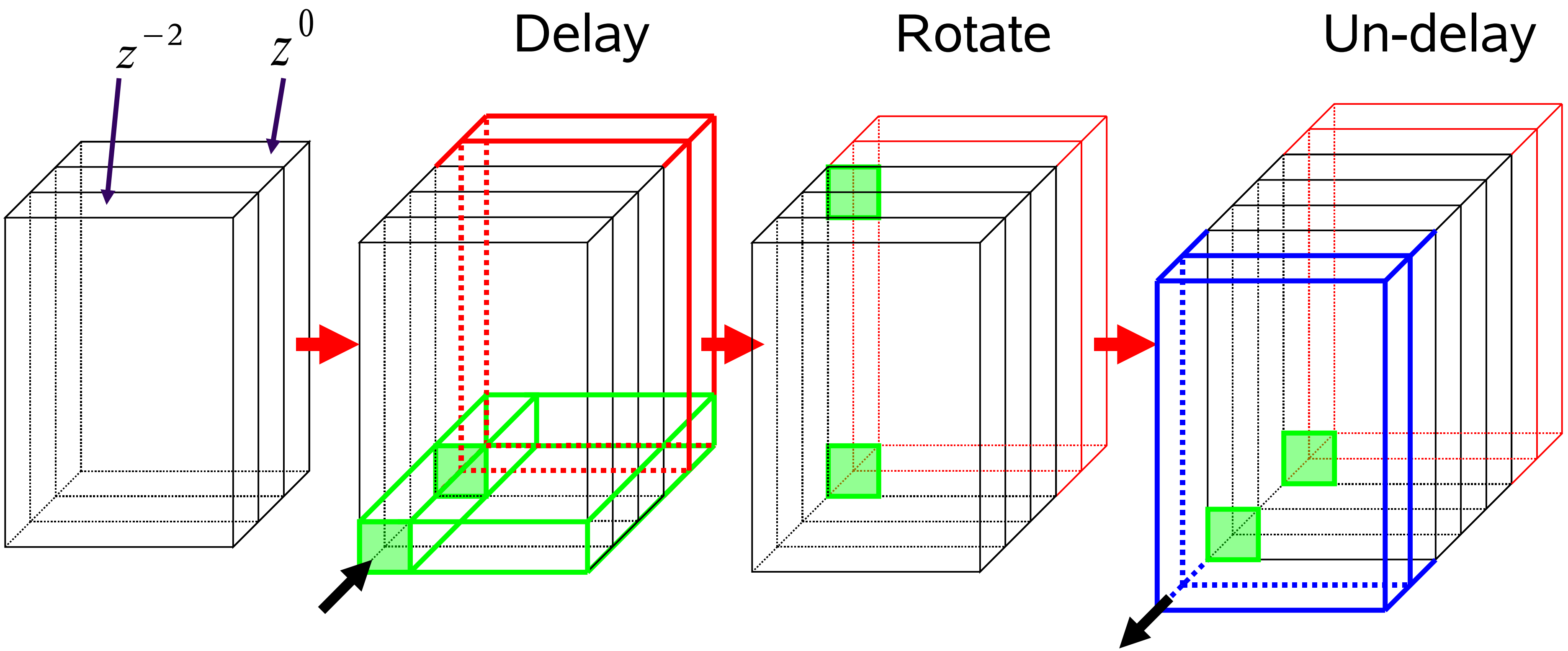
# Elementary Polynomial Givens Rotations (EPR)

$$G^{(\alpha,\theta,\varphi,t)}(z)=\underbrace{\begin{bmatrix}1 & 0\\ 0 & z^{-t}\end{bmatrix}}_{\text{Un-delay}}\underbrace{\begin{bmatrix}\cos(\theta)e^{i\alpha} & \sin(\theta)e^{i\varphi}\\ -\sin(\theta)e^{-i\varphi} & \cos(\theta)e^{-i\alpha}\end{bmatrix}}_{\text{Rotate}}\underbrace{\begin{bmatrix}1 & 0\\ 0 & z^{t}\end{bmatrix}}_{\text{Delay}}$$

$$=\begin{bmatrix}\cos(\theta)e^{i\alpha} & \sin(\theta)e^{i\varphi}z^{t}\\ -\sin(\theta)e^{-i\varphi}z^{-t} & \cos(\theta)e^{-i\alpha}\end{bmatrix}$$

**This matrix is paraunitary.**

# Application of an EPGR on a Polynomial Matrix



# Example of an EPGR

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

**Delay:**

$$\begin{bmatrix} 1 & 0 \\ 0 & z^{+1} \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2z^{-1} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & z^{+1} \end{bmatrix}$$

**Rotate:**

$$\frac{1}{\sqrt{5}} \begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & z^{+1} \end{bmatrix} = \frac{1}{\sqrt{5}} \begin{bmatrix} 5+2z^{-1} & 2+2z^{+1} \\ 0 & -4+z^{+1} \end{bmatrix}$$

**Un-delay:**

$$\frac{1}{\sqrt{5}} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} 5+2z^{-1} & 2+2z^{+1} \\ 0 & -4+z^{+1} \end{bmatrix} = \frac{1}{\sqrt{5}} \begin{bmatrix} 5+2z^{-1} & 2+2z^{+1} \\ 0 & -4z^{-1}+1 \end{bmatrix}$$

# The PQRD By Columns Algorithm

$$\underline{A}(z)=\begin{bmatrix} a_{11}(z) & \boldsymbol{a}_{12}(z) & \boldsymbol{a}_{13}(z) \\ a_{21}(z) & \boldsymbol{a}_{22}(z) & \boldsymbol{a}_{23}(z) \\ a_{31}(z) & \boldsymbol{a}_{32}(z) & \boldsymbol{a}_{33}(z) \\ a_{41}(z) & \boldsymbol{a}_{42}(z) & \boldsymbol{a}_{43}(z) \end{bmatrix}$$

**Step 1:** To drive all elements beneath the diagonal of column 1 to zero. Iterative process – each iteration apply an EPGR (polynomial rotation matrix) to zero the largest coefficient (in magnitude) beneath the diagonal.

**Step 2:** To drive all elements beneath the diagonal of column 2 to zero.

**Step 3:** To drive all elements beneath the diagonal of column 3 to zero.

**Repeat Steps 1 - 3:** if any coefficients are larger beneath the diagonal are larger than a stopping criterion.

# Example 1

**Input to PQRD:**

$$\underline{A}(z)=\begin{bmatrix} 2 & 0 & 2z^{1} \\ z^{1} & 1 & 0 \\ 0 & z^{-1} & 1 \end{bmatrix}$$

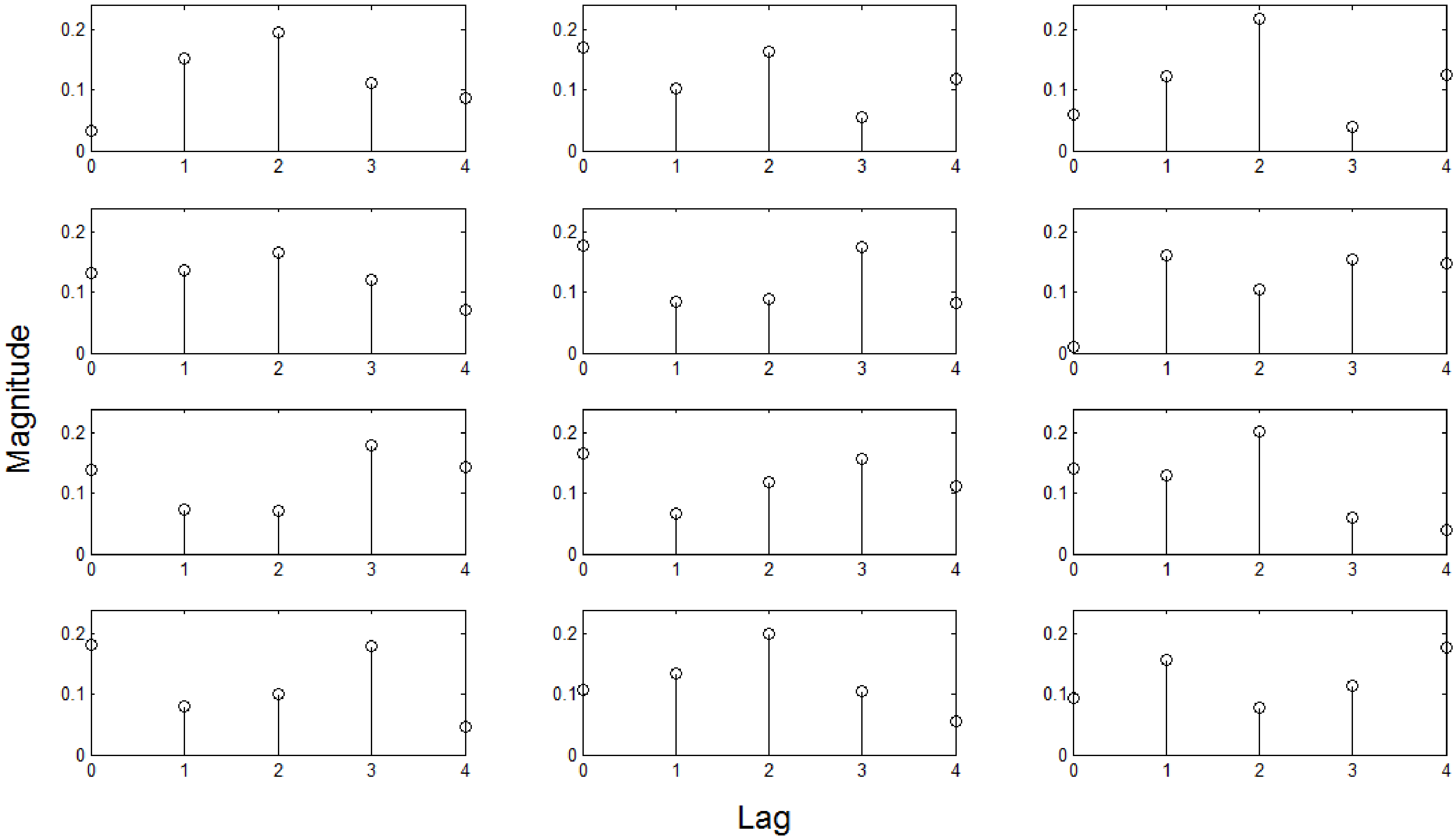
$$\underline{Q}(z)=\begin{bmatrix} 0.8944 & 0.4472z^{-1} & 0 \\ -0.2981z^{1} & 0.5963 & 0.7454z^{1} \\ 0.3333z^{1} & -0.6667 & 0.6667z^{2} \end{bmatrix}$$

**Paraunitary**

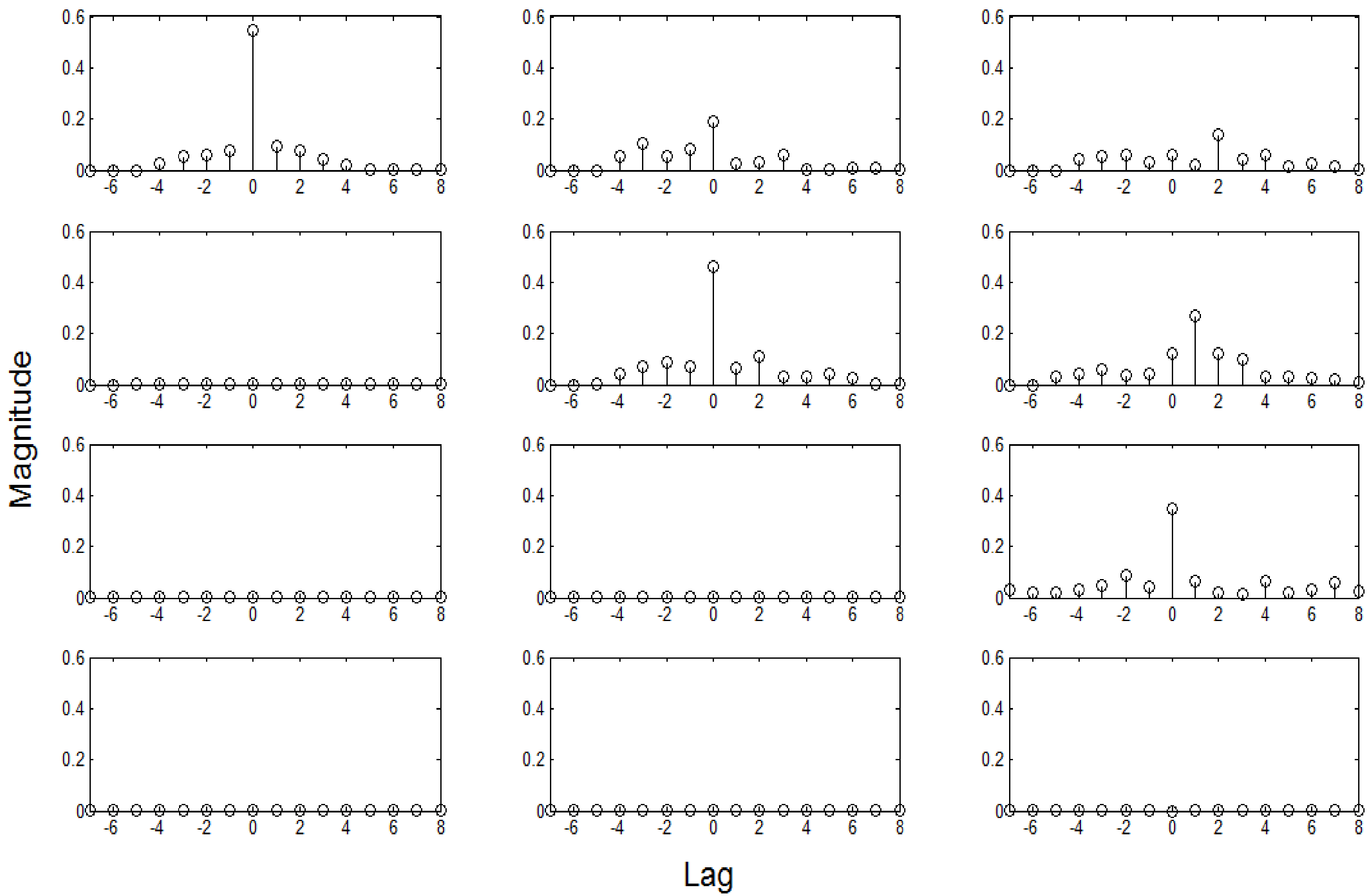
$$\underline{R}(z)=\begin{bmatrix} 2.2361 & 0.4472z^{-1} & 1.7889z^{1} \\ 0 & 1.3416 & 0.7454z^{1}-0.5963z^{2} \\ 0 & 0 & 0.6667z^{1}+0.6667z^{2} \end{bmatrix}$$

**Upper triangular matrix**

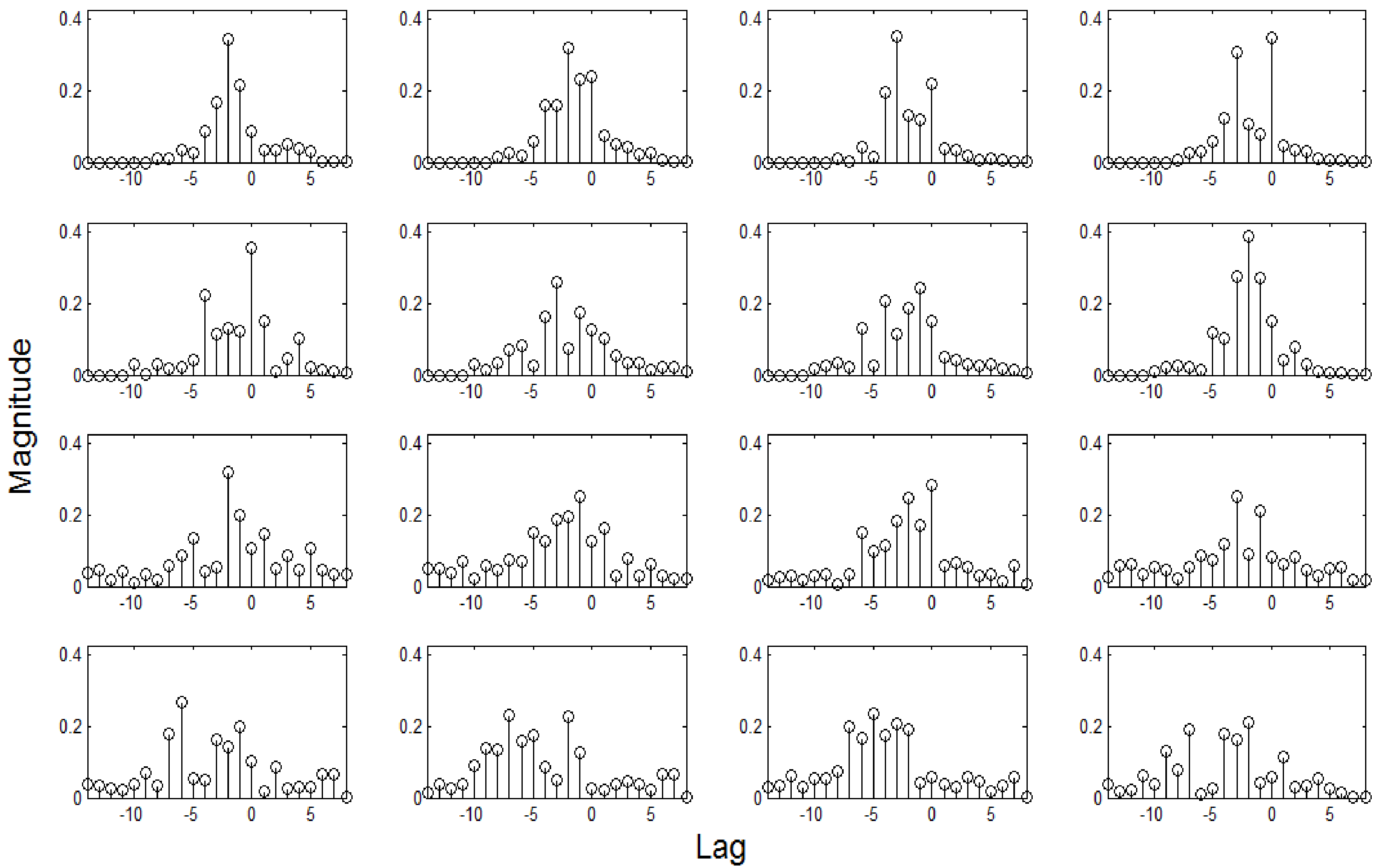
# Example 2



# Upper Triangular Matrix R(z)



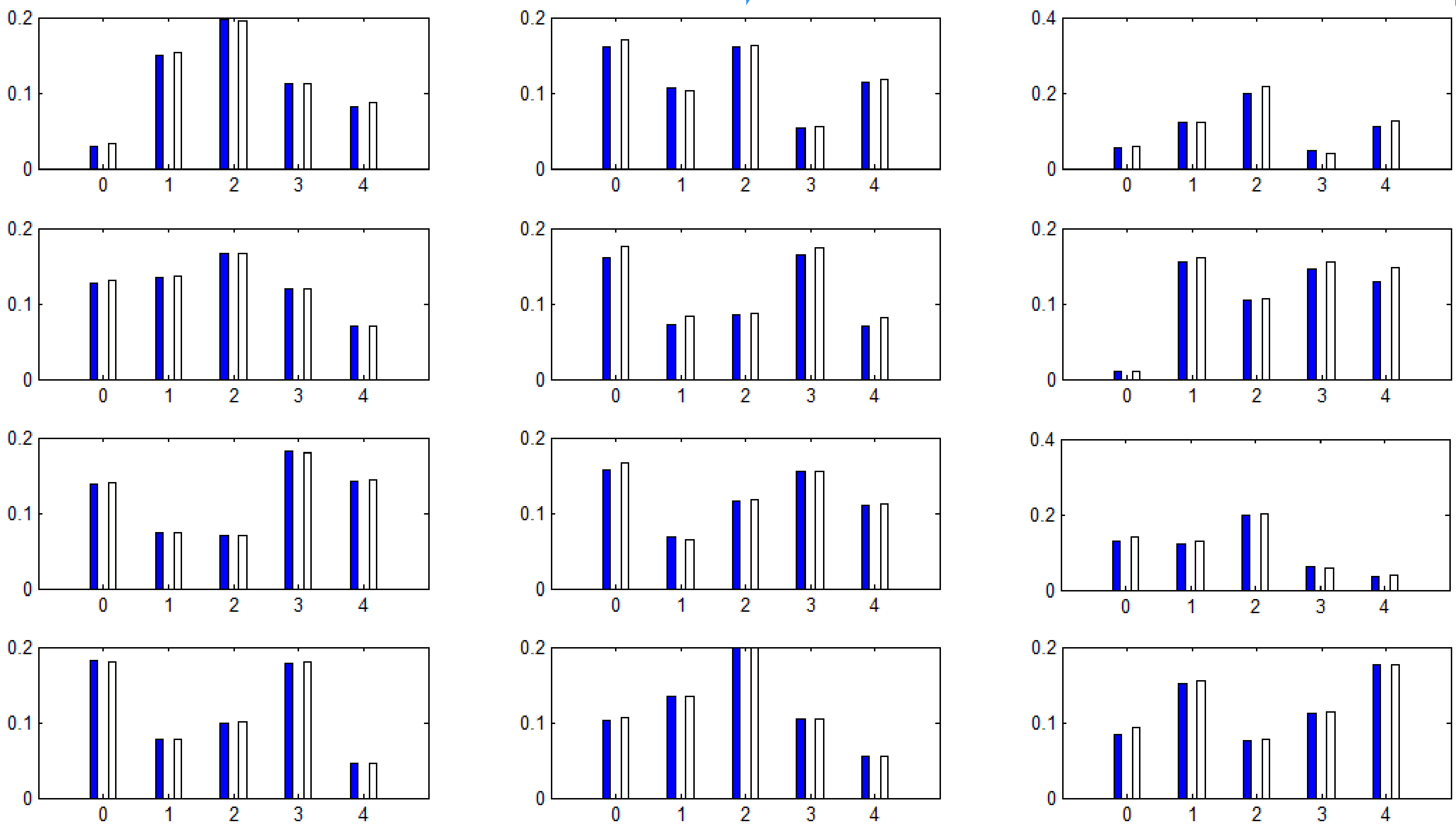
# Paraunitary matrix Q(z)



# Inverse Decomposition

$$R(z) = \tilde{Q}(z)A(z) \;\Rightarrow\; \hat{A}(z) = Q(z)\hat{R}(z)$$

Strictly upper triangular



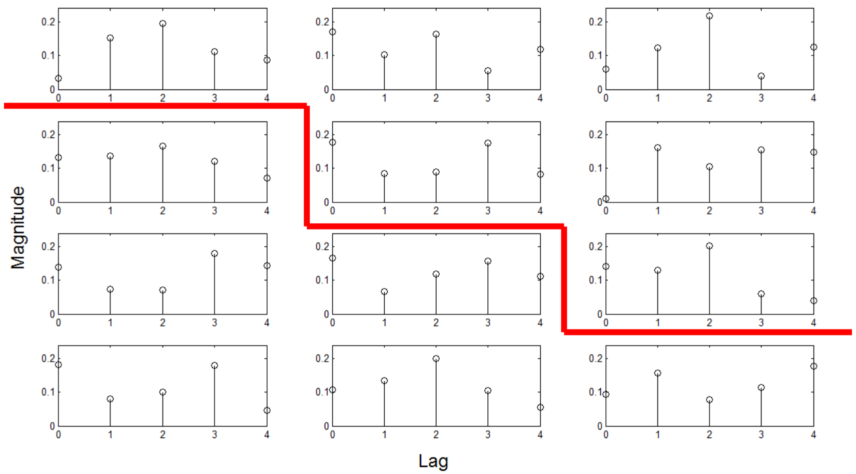
Input matrix

atrix obtained from inverse decomposition

# Performance Measure

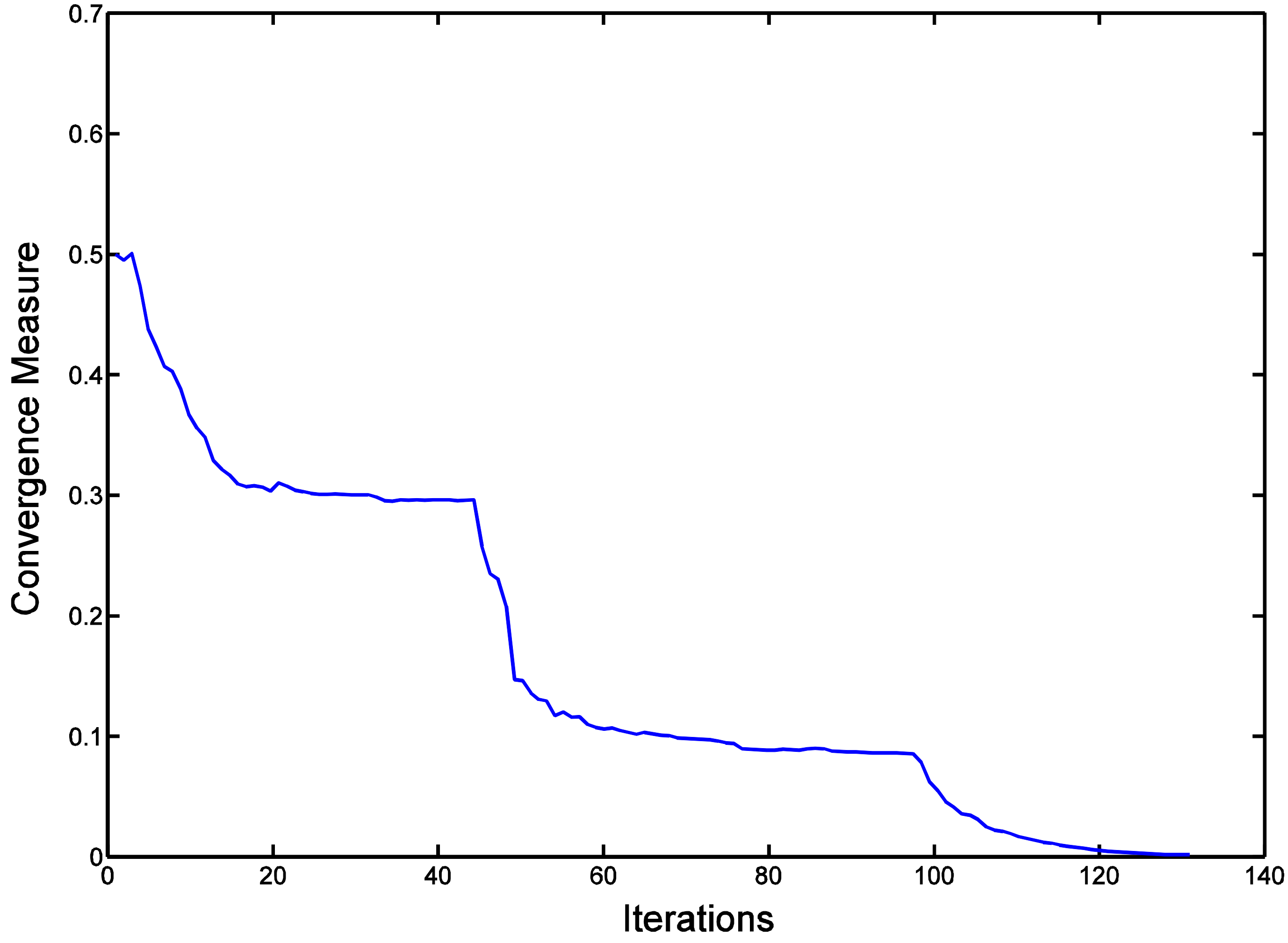
- The proportion of the Frobenius norm of matrix positioned beneath the diagonal of the matrix:

$$\frac{\sum_{t} \sum_{j=2}^{4} \sum_{k=1}^{j-1} |a'_{jk}(t)|^2}{\sum_{t=0}^{4} \sum_{j=1}^{4} \sum_{k=1}^{4} |a_{jk}(t)|^2}$$




- Measure of the upper triangularity of the polynomial matrix.

# Performance Measure

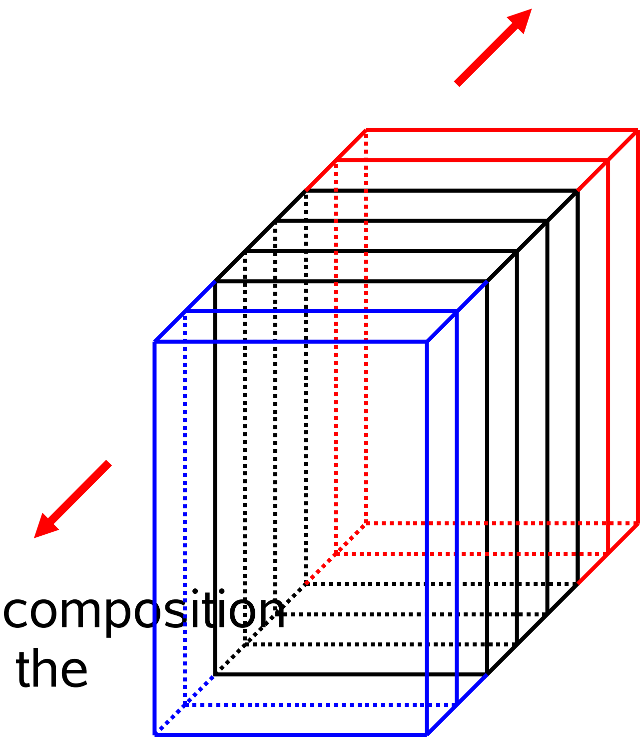


# Truncation Method

- Order of both polynomial matrices grows at each iteration
- Many outer layers consisting of small values
- Find max value of $T_1$ and min value of $T_2$ such that:

$$\frac{\sum_{t=T_2}^{b}\sum_{j=1}^{p}\sum_{k=1}^{p}|a'_{jk}(t)|^2}{\sum_{t}\sum_{j=1}^{p}\sum_{k=1}^{p}|a_{jk}(t)|^2} \text{¿} \frac{\mu}{2} \qquad \frac{\sum_{t=a}^{T_1}\sum_{j=1}^{p}\sum_{k=1}^{p}|a'_{jk}(t)|^2}{\sum_{t}\sum_{j=1}^{p}\sum_{k=1}^{p}|a_{jk}(t)|^2} \text{¿} \frac{\mu}{2}$$

- Keep only coefficients of $z^{T_1+1}, \ldots, z^{T_2+1}$
- This ensures that the accuracy of the decomposition is not significantly compromised, whilst reducing the computational load.



# The Polynomial Matrix Singular Value Decomposition (PSVD)

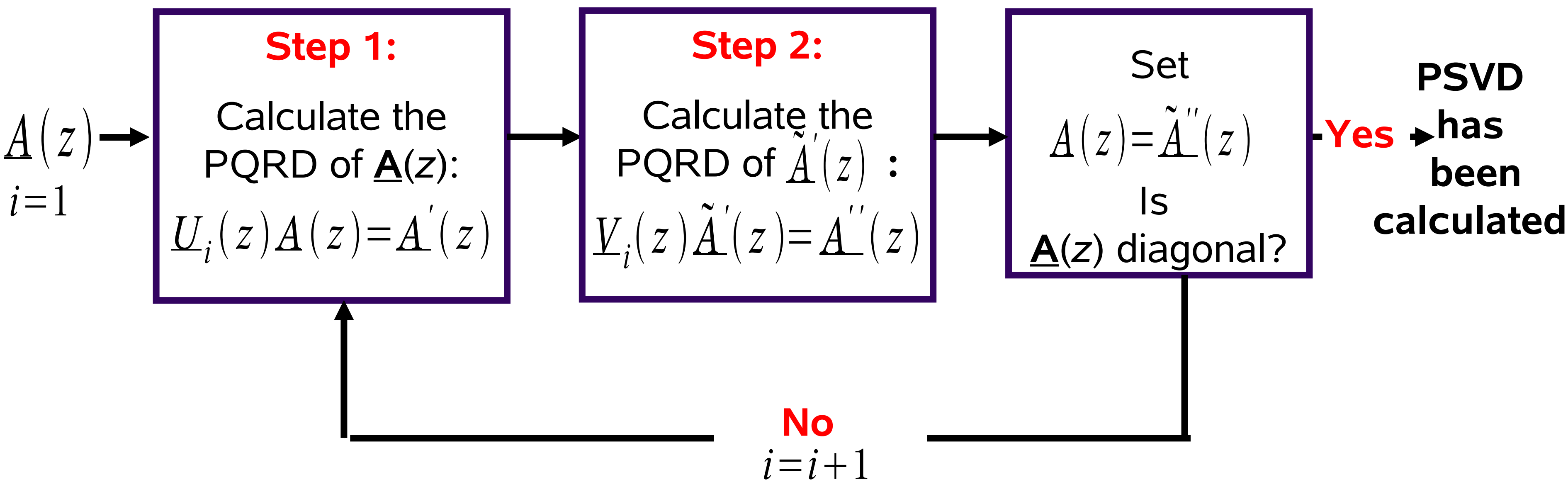
- The PSVD of a *pxq* polynomial matrix **A**(*z*) can be expressed as

$$\underline{U}(z)\underline{A}(z)\underline{\tilde{V}}(z)=\underline{\Sigma}(z)$$

where

- **U**(*z*) is a *pxp* paraunitary matrix,
- **V**(*z*) is a *qxq* paraunitary matrix and
- **S**(z) is a *pxq* diagonal polynomial matrix.

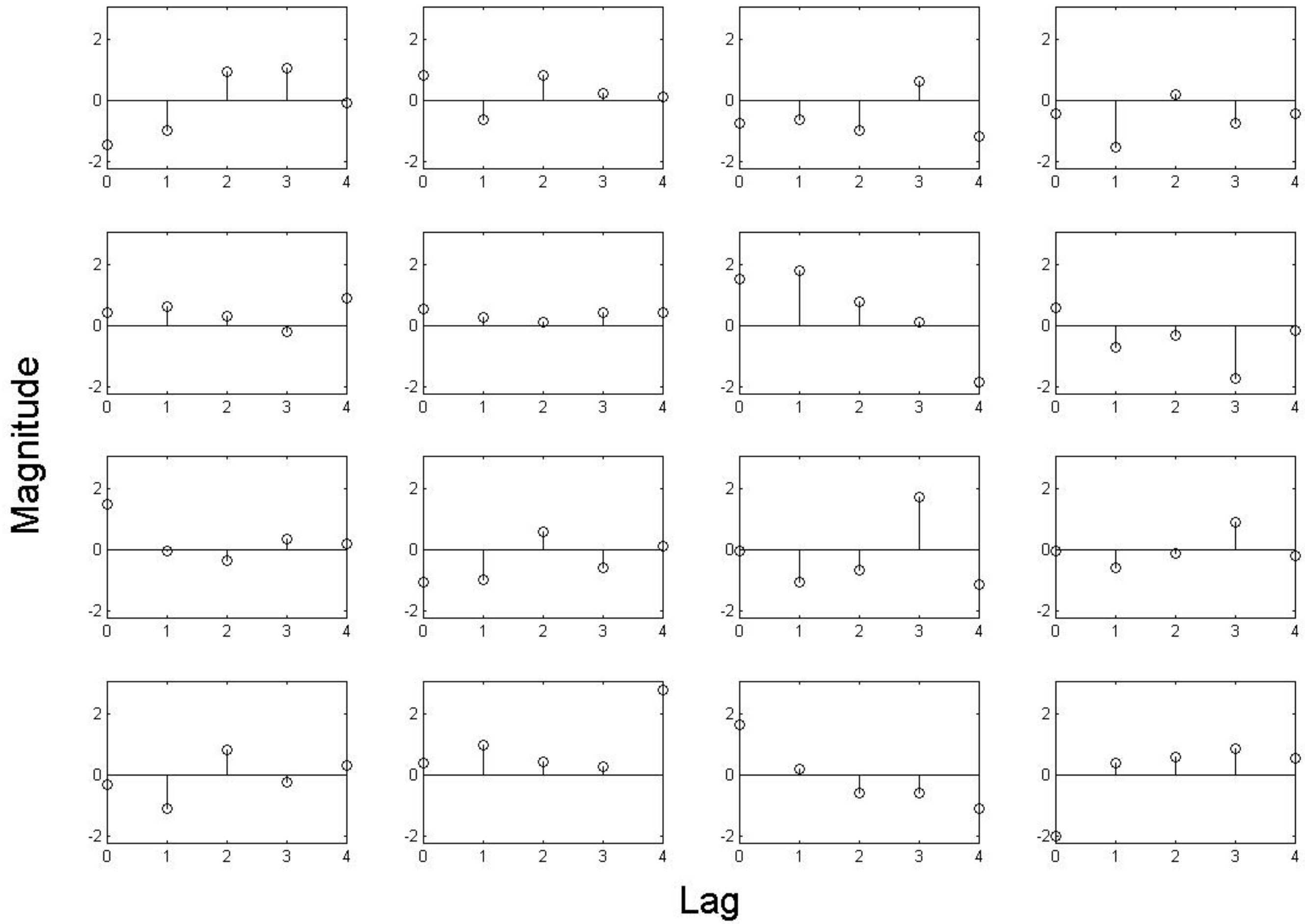
# The PSVD by PQRD Algorithm



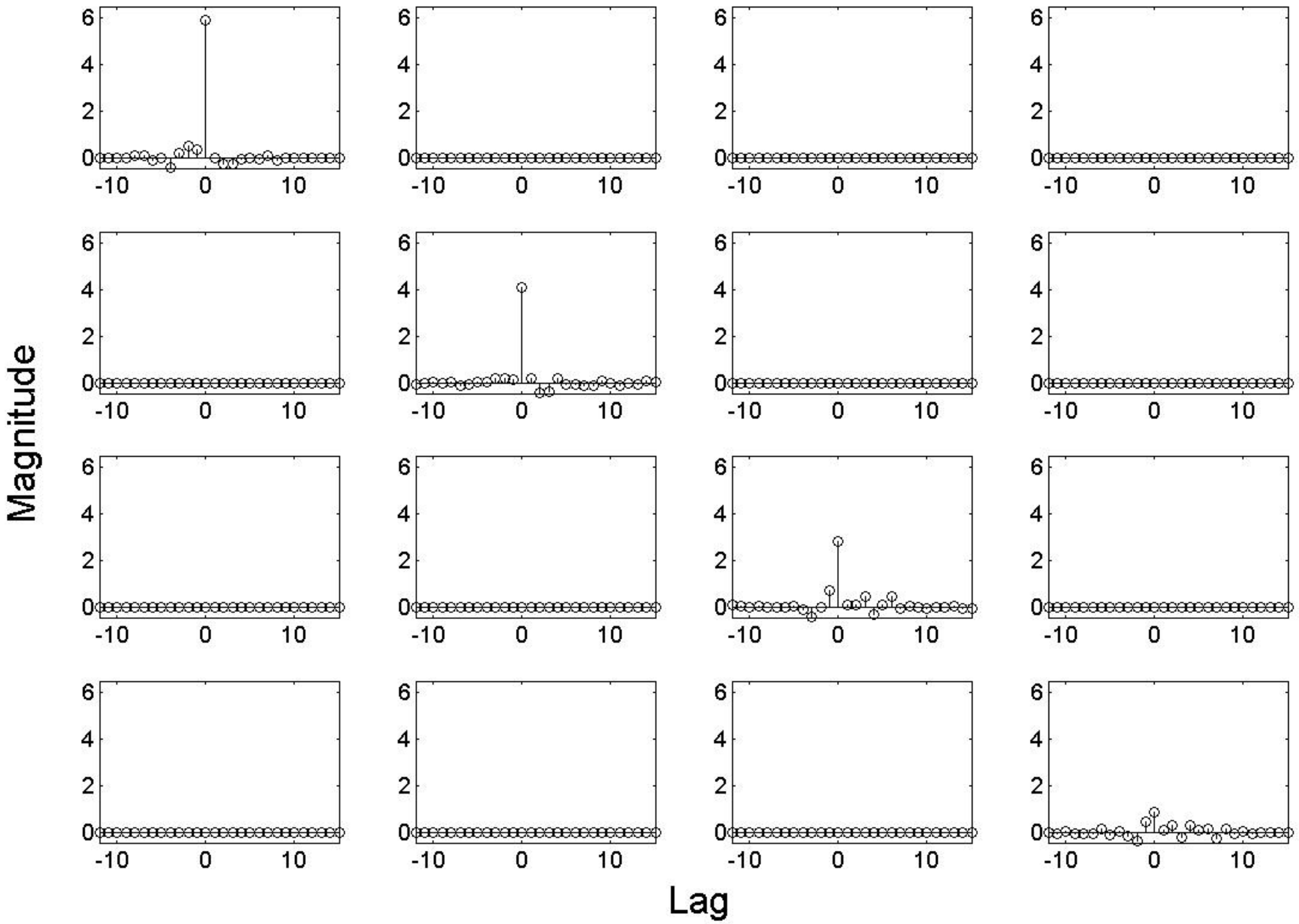

**Assuming the algorithm requires $N$ iterations to convergence**

$$U_N(z)\dots U_1(z)A(z)\tilde{V}_1(z)\dots\tilde{V}_N(z)=\Sigma(z)$$

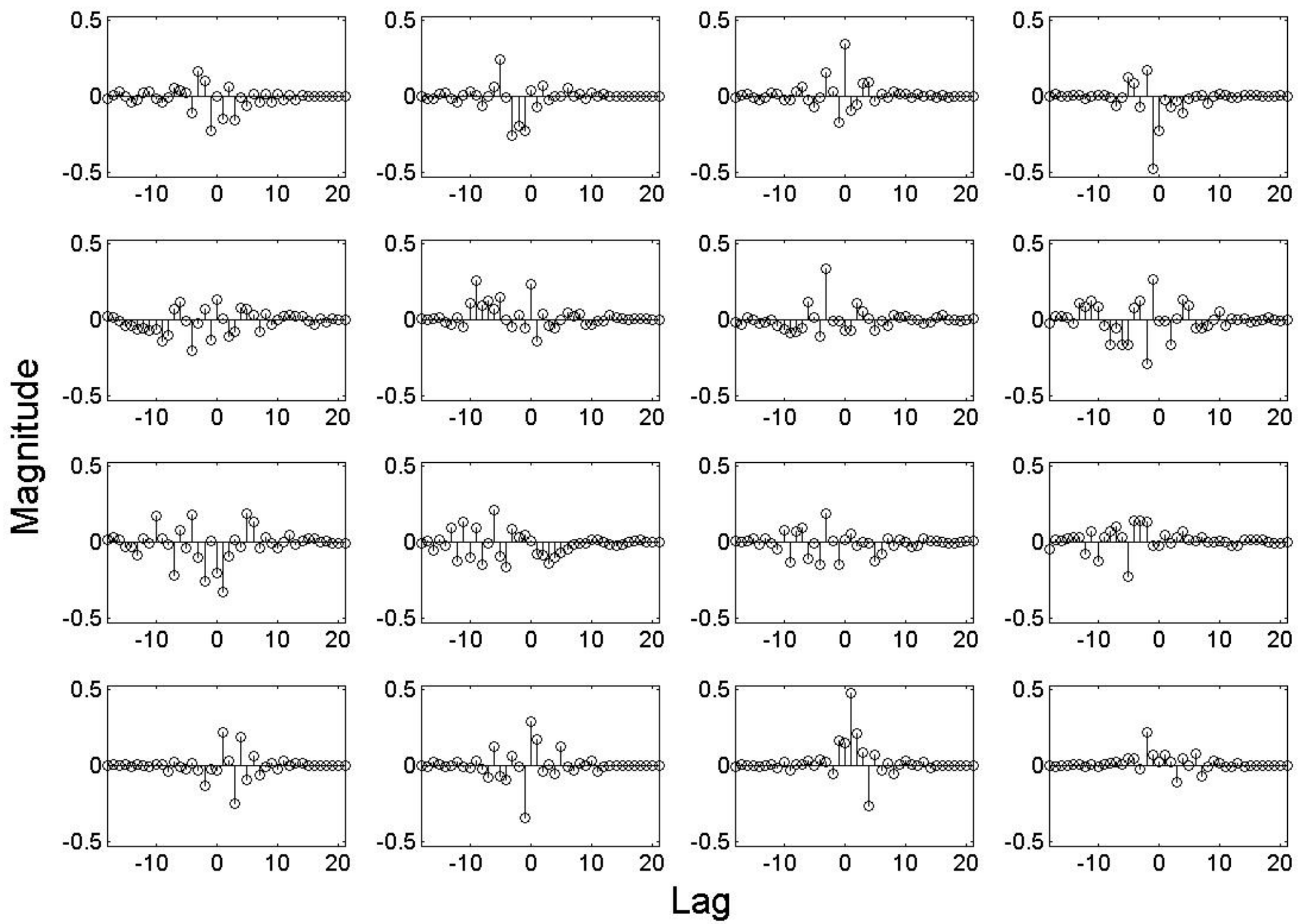
# Example 1



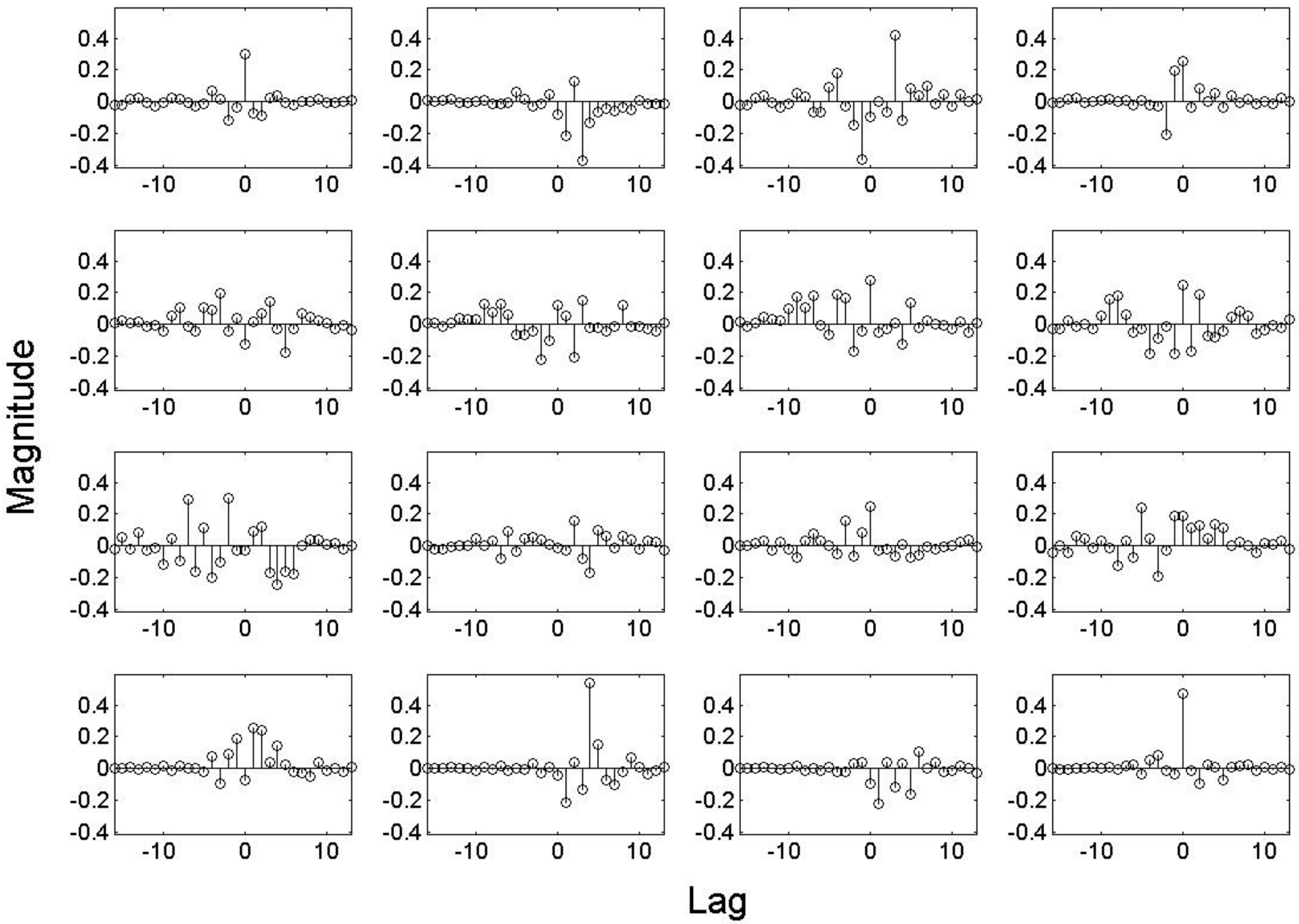
# Diagonal Matrix $\underline{\Sigma}(z)$



# Paraunitary matrix U(z)



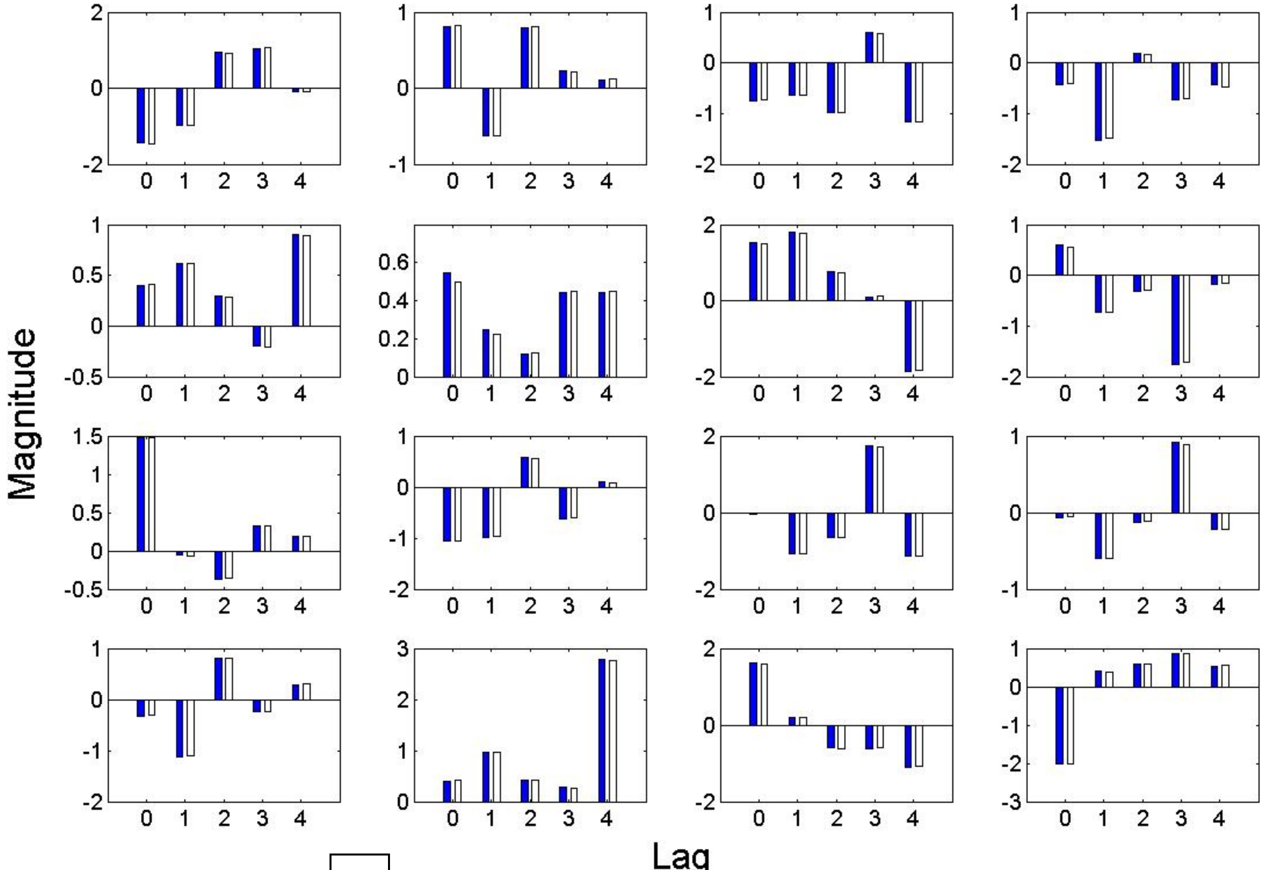
# Paraunitary matrix $\underline{V}(z)$



# Inverse Decomposition

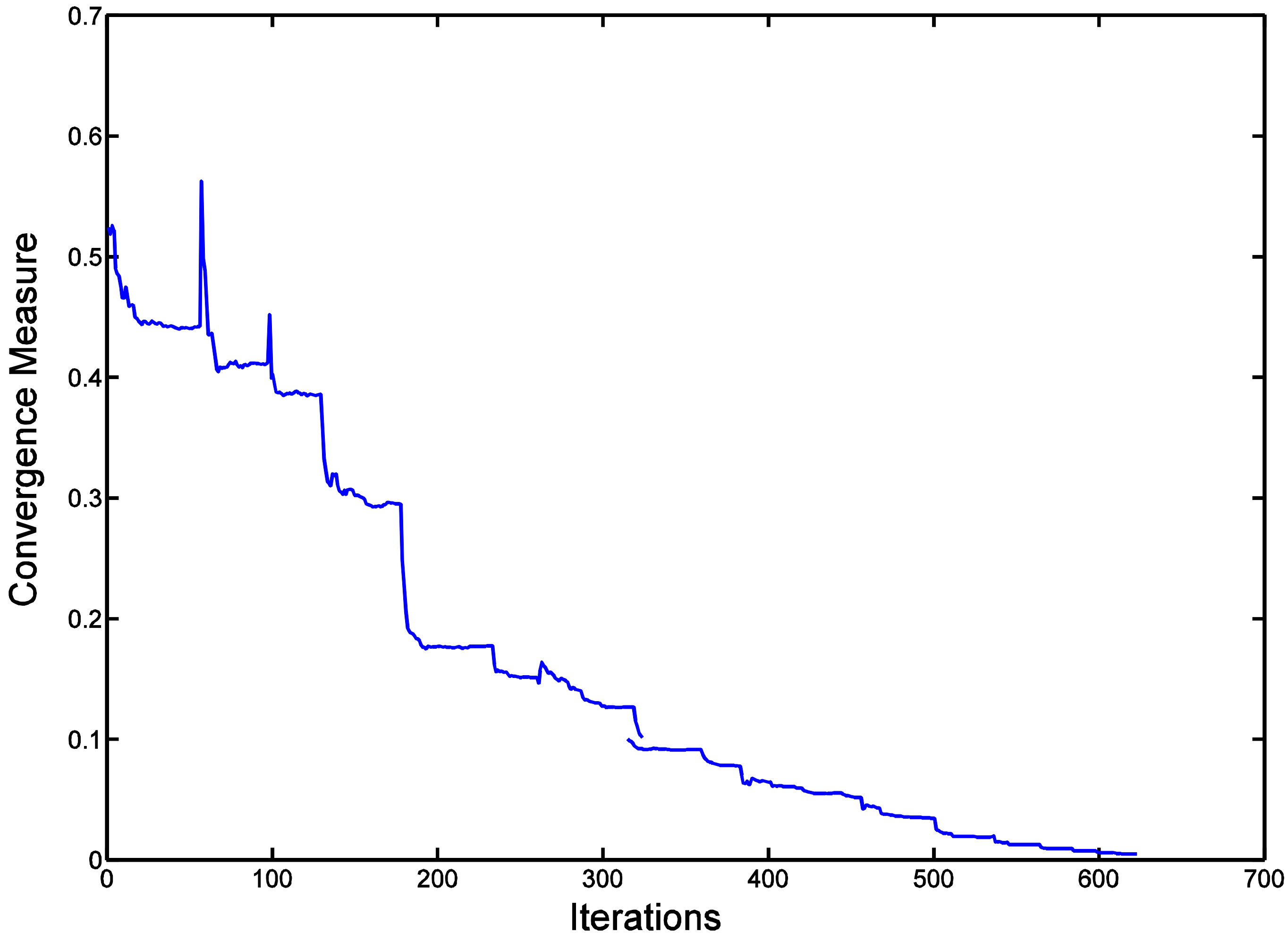
$$\underline{\Sigma}(z)=\underline{U}(z)\underline{A}(z)\underline{\tilde{V}}(z) \Rightarrow \underline{\hat{A}}(z)=\underline{\tilde{U}}(z)\underline{\hat{\Sigma}}(z)\underline{V}(z)$$

Strictly diagonal




Input matrix    Matrix obtained from inverse decomposition

# Convergence Measures



# Application to broadband MIMO communications

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

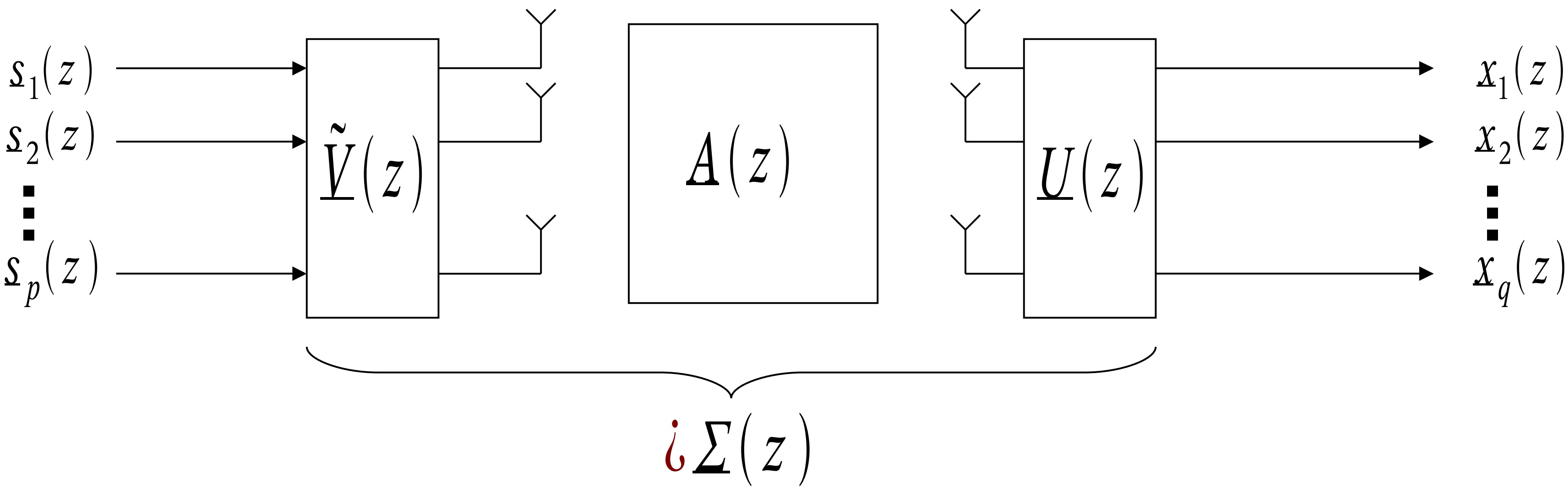
- Or expressed in polynomial form $\underline{x}(z)=\underline{A}(z)\underline{s}(z)+\underline{n}(z)$

- Assume channel matrix is known, then $\underline{U}(z)\underline{A}(z)\underline{\tilde{V}}(z)=\underline{\Sigma}(z)$

- Rearranging

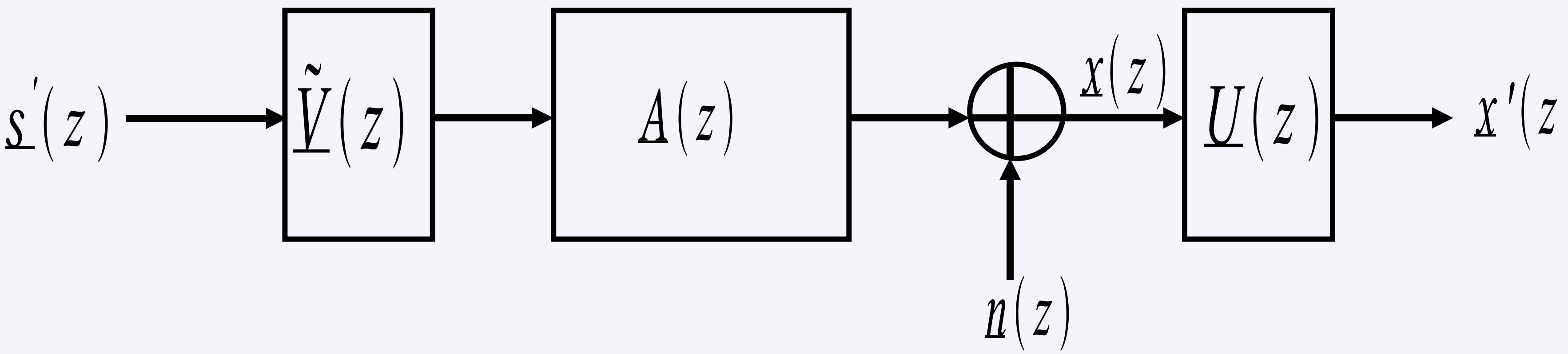
$$\underbrace{\underline{U}(z)\underline{x}(z)}_{\underline{x}'(z)}=\underline{U}(z)\underbrace{\underline{\tilde{U}}(z)\underline{\Sigma}(z)\underline{V}(z)}_{\underline{A}(z)}\underbrace{\left[\underline{\tilde{V}}(z)\underline{s}'(z)\right]}_{\underline{s}(z)}+\underbrace{\underline{U}(z)\underline{n}(z)}_{\underline{n}'(z)}$$

# Potential Application of the PSVD to MIMO Systems

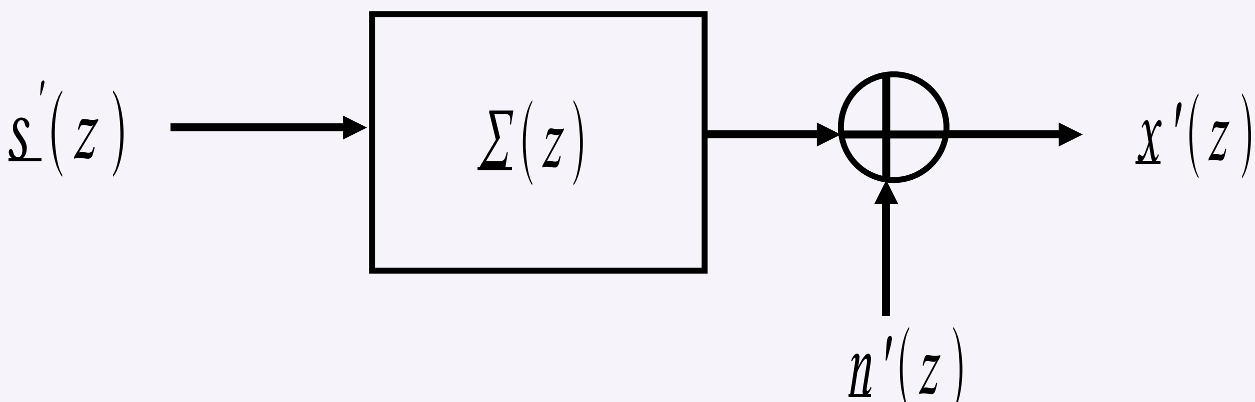


# The PSVD in a MIMO system:

Implement:




Equivalent system:



# 2x2 Example (PSVD)

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

1. Estimate source 1 $\underline{x}_1'(z) = \underline{\sigma}_{11}(z)\underline{s}_1'(z) + \underline{n}_1'(z)$

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

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

# Comparative Bit Error Rate Simulations

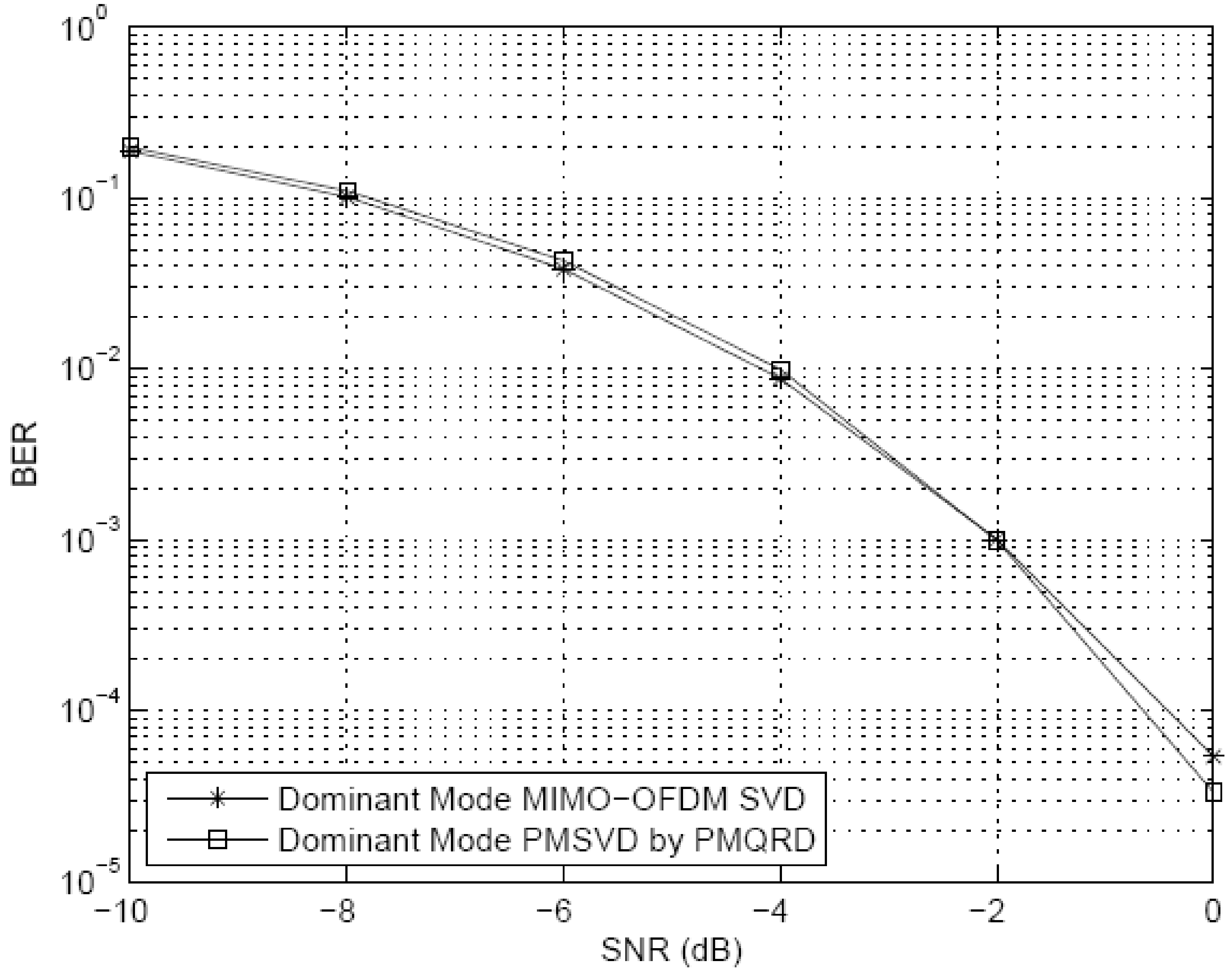
- **Benchmark scheme:** Dominant mode MIMO Orthogonal Frequency Division Multiplexing (OFDM) SVD [1].

- **Our scheme:** Dominant mode PSVD with Turbo equalisation [2].

- **Simulation parameters:**
  - 5 *x* 5 exponentially decaying MIMO channels
  - Matrix order (delay spread) four.
  - Additive complex Gaussian noise with variance chosen to obtain a range of signal to noise values.

[1] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, 1st ed. University Press, Cambridge: Cambridge University Press, 2003.

[2] M. Davies, S. Lambotharan, J. Chambers and J. G. McWhirter, Broadband MIMO Beamforming For Frequency Selective Channels Using the Sequential Best Rotation Algorithm, *67th Vehicular Technology Conference*, Singapore, 2008.




- M. Davies, *Polynomial Matrix Decomposition Techniques for Frequency Selective MIMO Channels*, PhD Thesis, Department of Electronic and Electrical Engineering, Loughborough University, submitted 2009.

# Application to broadband MIMO communications

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

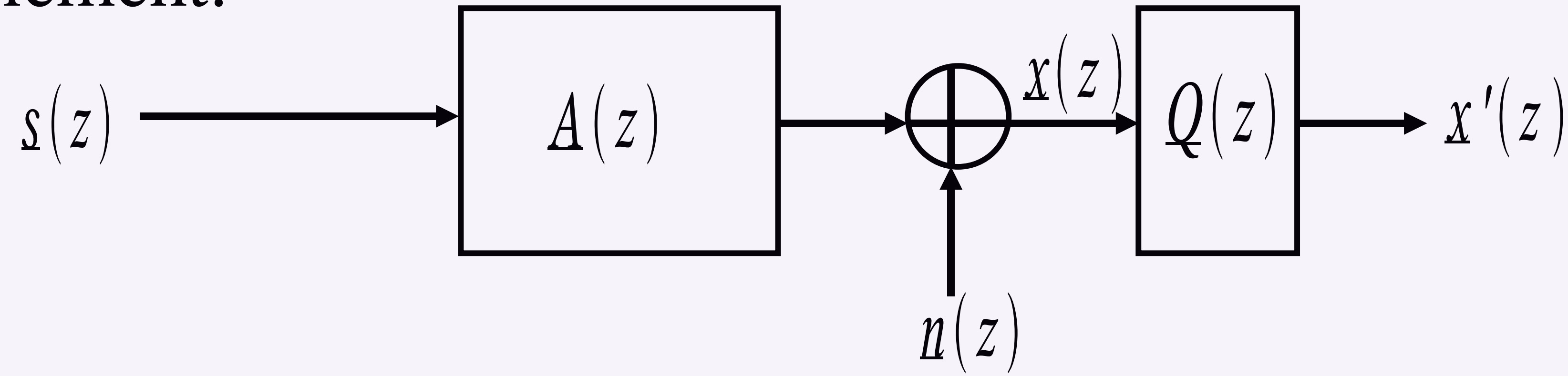
- Or expressed in polynomial form $$\underline{x}(z)=\underline{A}(z)\underline{s}(z)+\underline{n}(z)$$

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

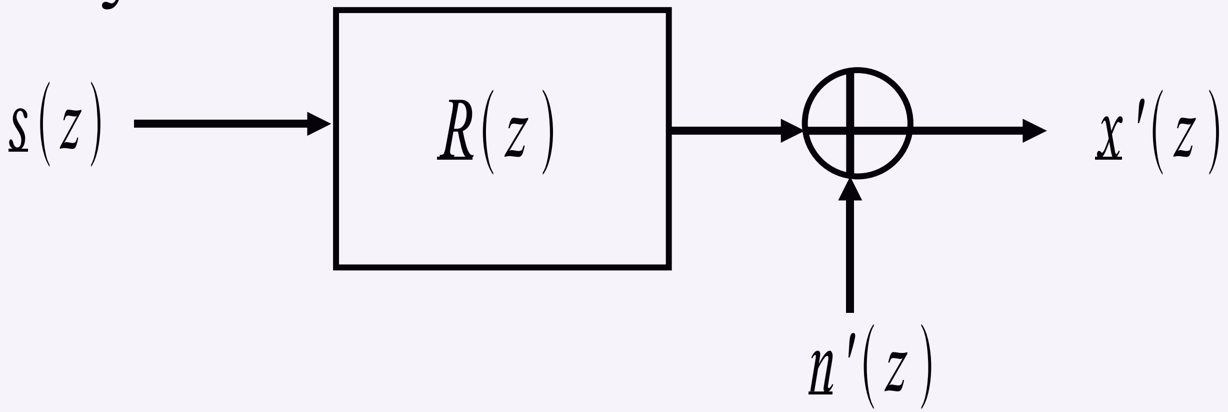
- Rearranging $$\underbrace{\underline{\tilde{Q}}(z)\underline{x}(z)}_{\underline{x}'(z)}=\underline{R}(z)\underline{s}(z)+\underbrace{\underline{\tilde{Q}}(z)\underline{n}(z)}_{\underline{n}'(z)}$$

# The PQRD in a MIMO system:

Implement:




Equivalent system:



# 2x2 Example (PQRD)

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

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

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

# Comparative Bit Error Rate Simulations

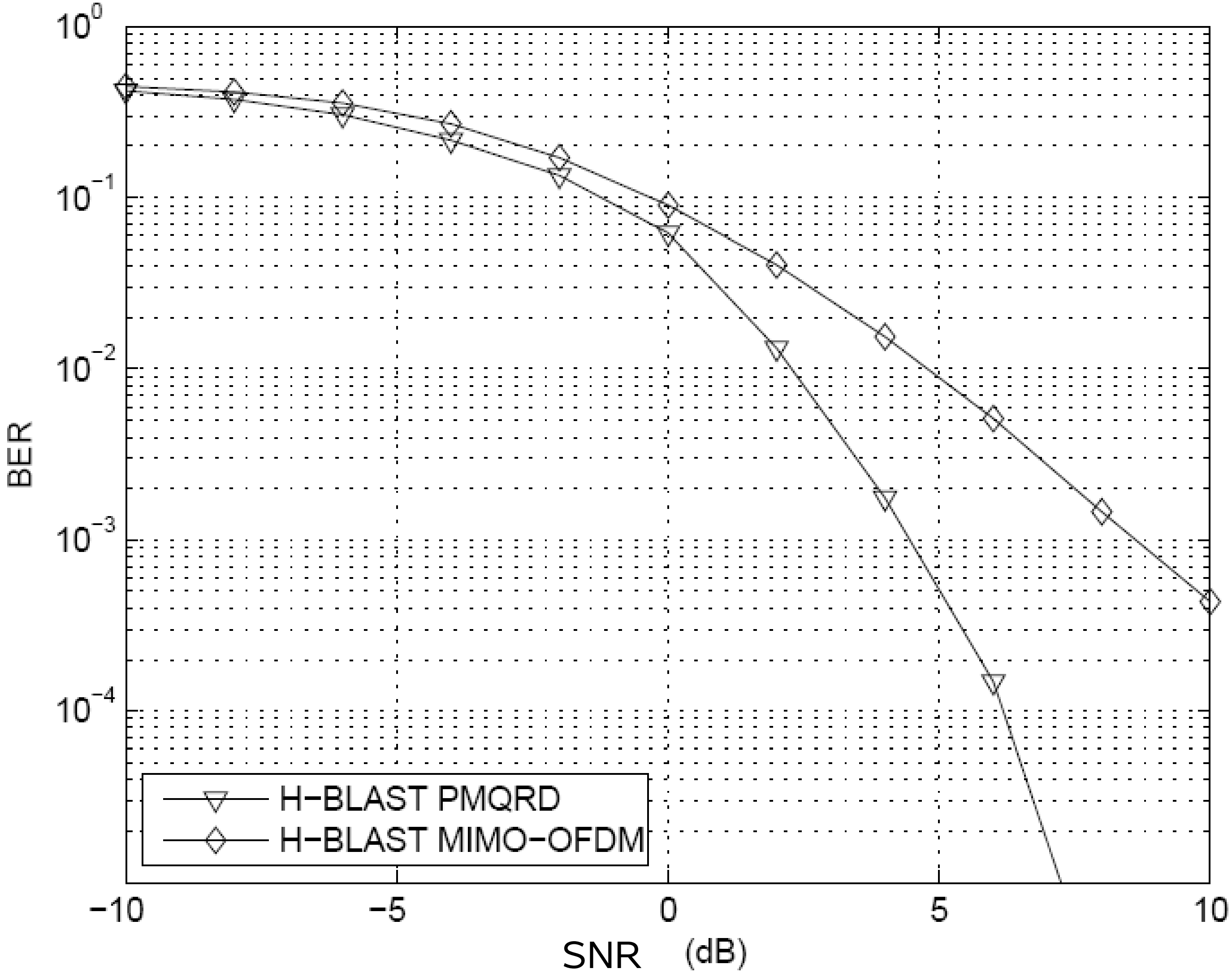
- **Benchmark scheme:** Horizontal Bell Laboratories Layered Space Time (H-BLAST) MIMO-OFDM QRD [1,2].

- **Our scheme:** H-BLAST PQRD.

- **Simulation parameters:**
  - 3 *x* 3 quasi static MIMO channels (polynomial)
  - Matrix order (delay spread) four.
  - Additive complex Gaussian noise with variance chosen to obtain a range of signal to noise values.

[1] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, 1st ed. University Press, Cambridge: Cambridge University Press, 2003.

[2] G. Foschini, *Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas*, Bell Labs Technical Journal (Autumn), pp. 41–59, 1996.




- M. Davies, *Polynomial Matrix Decomposition Techniques for Frequency Selective MIMO Channels*, PhD Thesis, Department of Electronic and Electrical Engineering, Loughborough University, submitted 2009.

# Conclusions

- Algorithms developed for calculating the following decompositions of a polynomial matrix:
  - QR decomposition
  - Eigenvalue decomposition
  - Singular value decomposition
- Decompositions have been shown to converge, the proof of which is given in [1].
- Potential applications (so far) are to broadband MIMO channel equalisation and convolutive blind source separation (strong decorrelation).

[1] J.A. Foster, J.G. McWhirter, M. Davies and J.A Chambers, An Algorithm for Calculating the QR and Singular Value Decompsotion of a Polynomial Matrix, accepted for publication to *IEEE Transactions on Signal Processing*.

# References


- J.G. McWhirter, P.D. Baxter, T. Cooper, S. Redif and J. Foster, An EVD Algorithm for Para-Hermitian Polynomial Matrices, *IEEE Transactions on Signal Processing*, Vol. 55(6), pp. 2158-2169, 2007.
- J. A. Foster, J.G. McWhirter and J. Chambers, A Polynomial Matrix QR Decomposition with Application to MIMO Channel Equalisation, *Proc. 41st Asilomar Conference on Signals, Systems and Computers, Asilomar (Invited Talk)*, California, November, 2007.
- M. Davies, S. Lambotharan, J. Chambers and J. G. McWhirter, Broadband MIMO Beamforming For Frequency Selective Channels Using the Sequential Best Rotation Algorithm, *67th Vehicular Technology Conference*, Singapore, 2008.
- J. A. Foster, Algorithms and Techniques for Polynomial Matrix Decompositions, PhD Thesis, School of Engineering, Cardiff University, UK, 2008.
- M. Davies, S. Lambotharan, J.A. Foster, J. Chambers and J. G. McWhirter, Polynomial Matrix QR Decomposition and Iterative Decoding of Frequency Selective MIMO Channels, *IEEE Wireless Communications and Networking Conference,* Budapest, Hungary, 2009.
- J.A. Foster, J.G. McWhirter and J. Chambers, A Novel Algorithm for Calculating the QR Decomposition of a Polynomial Matrix, *ICASSP,* 2009.
- J.A. Foster, J.G. McWhirter, M. Davies and J.A Chambers, An Algorithm for Calculating the QR and Singular Value Decompsotion of a Polynomial Matrix, accepted for publication to *IEEE Transactions on Signal Processing*.