Second Order Sequential Best Rotation Algorithm with Householder Reduction for Polynomial Matrix Eigenvalue Decomposition
1. Introduction
   Motivation for PEVD
   Polynomial Matrices
   PEVD

2. Proposed Method
   Householder Reduction

3. Simulations and Results
   Experiment Setup
   Single Example
   Results of Monte-Carlo Simulation

4. Conclusion
Introduction
Motivation for PEVD

- EVD of Hermitian matrices is commonly used in
  - subspace decomposition for data compression
  - blind source separation
  - adaptive beamforming

⇒ Assumption: Sources are narrowband

- Broadband signals need to model the correlation between sensor pairs across different time lags
  → Polynomial matrices

- Development of PEVD algorithms and applications in
  - subspace decomposition using polynomial MUSIC [1]
  - blind source separation [2]
  - adaptive beamforming [3]
  - source identification [4]
The data vector at time index $n$ collected from $M$-sensors is

$$x(n) = [x_1(n), x_2(n), \ldots, x_M(n)]^T \in \mathbb{C}^M.$$ 

The space-time covariance matrix for $N$ time snapshots is

$$A(\tau) = \mathbb{E}\{x(n)x^H(n-\tau)\} \approx \frac{1}{N} \sum_{n=0}^{N-1} x(n)x^H(n-\tau) \in \mathbb{C}^{M \times M},$$

and its $z$-transform is a para-Hermitian polynomial matrix,

$$A(z) = \sum_{\tau=-W}^{W} A(\tau)z^{-\tau}.$$
Polynomial Eigenvalue Decomposition

The PEVD of $A(z)$ according to [5] is

$$A(z) \approx U(z) \Lambda(z) U^P(z),$$

where

- $U^P(z) = U^H(z^{-1})$,
- $\Lambda(z)$ is the eigenvalue polynomial matrix and
- $U(z)$ is the eigenvector polynomial matrix, such that
  $$U(z) = U_L(z) \ldots U_2(z) U_1(z),$$

constructed using $L$ para-unitary polynomial matrices.
Comparison between EVD and PEVD

\[
\begin{bmatrix}
9.30 & 5.12 & 4.23 \\
5.12 & 8.61 & 4.50 \\
4.23 & 4.50 & 8.27
\end{bmatrix}
\]

A taken from \( A(\tilde{z}^0) \).

\[ A(\tilde{z}) \] example.
Comparison between EVD and PEVD

\[
\begin{bmatrix}
18.0 & 0 & 0 & 0 \\
0 & 4.53 & 0 & 0 \\
0 & 0 & 3.66 & 0
\end{bmatrix}
\]

\( \Lambda \) using EVD.

\[ \delta \leq \sqrt{\frac{N_1}{3}} \times 10^{-2} \] where \( N_1 \) is the trace-norm of \( \Lambda(z^0) \) [5].

\( \Lambda(z) \) using SBR2 with \( \delta = 0.087 \).
At each iteration, SBR2 will

(i) search for the largest off-diagonal, $|g|$,
(ii) delay and bring $|g|$ to the zero-lag plane,
(iii) zero $|g|$ using a Givens rotation and
(iv) trim negligible high order terms.
Family of PEVD Algorithms

SBR2 provided a framework for extensions based on (i)-(iv).

(i) search: norm-2 instead of inf-norm
   - Householder-like PEVD [6]
   - sequential matrix diagonalisation (SMD) [7]

(ii) delay: multiple-shift (MS) instead of single-shift
   - MS-SBR2 [8]
   - MS-SMD [9]

(iii) zero: one-step diagonalisation of $z^0$ instead of using the Givens rotation
   - SMD [7]
   - Householder-like PEVD [6]
   - approximate PEVD [10].

Proposed Method
Jacobi’s Method for Symmetric EVD

Consider the principal plane of a polynomial matrix, \( A(z^0) \in \mathbb{C}^{M \times M} \).

\[
\begin{bmatrix}
    a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & \cdots & a_{1,M} \\
    a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & \cdots & a_{2,M} \\
    a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & \cdots & a_{3,M} \\
    \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
    a_{M-1,1} & a_{M-1,2} & a_{M-1,3} & \cdots & a_{M-1,M-1} & a_{M-1,M} \\
    a_{M,1} & a_{M,2} & a_{M,3} & \cdots & a_{M,M-1} & a_{M,M}
\end{bmatrix}
\]

\( \Rightarrow \) Cycling through all off-diagonal elements using Jacobi’s algorithm requires \( \frac{M(M-1)}{2} \) Givens rotations.
(M − 1) Householder reflections first reduce the principal plane to tridiagonal form [12].

\[
\begin{bmatrix}
    a_{1,1} & a_{1,2} & 0 & \ldots & \ldots & \ldots & 0 \\
    a_{2,1} & a_{2,2} & a_{2,3} & 0 & \ldots & \ldots & \vdots \\
    0 & a_{3,2} & a_{3,3} & a_{3,4} & \ddots & \ddots & \vdots \\
    \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
    \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
    0 & \ldots & \ldots & a_{M−1,M−1} & a_{M−1,M} \\
    0 & \ldots & \ldots & \ldots & a_{M,M−1} & a_{M,M}
\end{bmatrix}
\]

⇒ In this reduced form, there are fewer elements to zero.
⇒ Cycling through all off-diagonal elements uses (M − 2) Householder reflections followed by (M − 1) Givens rotations.
Householder Reduction in EVD

Comparison of diagonalisation using Householder + Givens (HG) and Givens-only (G) using 1000 randomly generated symmetric matrices for every $M$ with $\delta \leq \sqrt{N_1/3} \times 10^{-2}$.

$\Rightarrow$ The reduction in $L$ achieved by Householder + Givens over Givens-only method scales with matrix dimension, $M$. 

- Proposed Method
- SBR2 with Householder Reduction for PEVD
Inputs: $A(z) \in \mathbb{C}^{M \times M}$, $\delta$, maxIter, $\mu$.
initialise: $l \leftarrow 0$, $g \leftarrow 1 + \delta$, $\tilde{\Lambda}(z) = A(z), \tilde{U}(z) = I$.
for $l < $ maxIter and $g > \delta$ do
  $g \leftarrow \max |r_{jk}(z^t)|, k > j, \forall t$.
  if ($g > \delta$) then
    $l \leftarrow l + 1$.
    $\tilde{\Lambda}(z) \leftarrow D_j(z)\tilde{\Lambda}(z)D_j^P(z)$,
    $\tilde{U}(z) \leftarrow D_j(z)\tilde{U}(z) // delay$
    $\tilde{\Lambda}(z) \leftarrow H\tilde{\Lambda}(z)H^H$
    $\tilde{U}(z) \leftarrow H\tilde{U}(z) // reflect$
    $\tilde{\Lambda}(z) \leftarrow G(\theta, \phi)\tilde{\Lambda}(z)G^H(\theta, \phi)$,
    $\tilde{U}(z) \leftarrow G(\theta, \phi)\tilde{U}(z) // rotate$
    $\tilde{\Lambda}(z) \leftarrow \text{trim} (\tilde{\Lambda}(z), \mu)$,
    $\tilde{U}(z) \leftarrow \text{trim} (\tilde{U}(z), \mu) // trim$.
  end if
end while
return $\tilde{U}(z), \tilde{\Lambda}(z)$. 
Simulations and Results
Experiment Setup

The setup was based on the 3 sensors, 2 sources decorrelation simulation in [5] which used

- i.i.d. source signals of 1000 samples each and each sample was assigned $\pm 1$ with equal probability
- each channel was modelled as a 5-th order FIR filter and each coefficient was drawn from $U[-1, 1]$
- additive white Gaussian noise with $\sigma = 1.8$
- PEVD parameters: $W = 10, \mu = 10^{-4}$, $\delta \leq \sqrt{N_1/3} \times 10^{-2}$

This was repeated 1000 times for the Monte-Carlo simulation.
Evaluation Measures

For each algorithm, we computed the

- Number of iterations, \( L \)
- Reconstruction error, \( \epsilon \triangleq \sum_{\forall z} \| \tilde{A}(z) - A(z) \|_F \)

For comparisons of both algorithms, we used

- Relative \( L \) difference, \( \Delta L(\%) = \frac{L_{\text{Proposed}} - L_{\text{SBR2}}}{L_{\text{SBR2}}} \times 100\% \)
- Relative \( \epsilon \) difference, \( \Delta \epsilon(\%) = \frac{\epsilon_{\text{Proposed}} - \epsilon_{\text{SBR2}}}{\sum_{\forall z} \| A(z) \|_F} \times 100\% \)
diagonalisation target: Maximum off-diagonal $|g| \leq 0.087$

SBR2 took 169 iterations.

Our method took 101 iterations.

⇒ Tridiagonal reduction prior to applying the Givens rotations reduces the number of iterations for PEVD.
Our method achieved an average of 12% reduction in $L$ over SBR2.
Reduction in $L$ was achieved in 82% of the trials.
⇒ Our method achieved an average of 0.1% reduction in $\epsilon$.
⇒ Both methods were consistent to $\pm 1\%$ in $\epsilon$. 
Conclusion
Conclusion

- Proposed the use of Householder reduction before applying the Givens rotations at the zeroing step in SBR2.
- An average of 12% reduction in iteration counts is achievable.
- An average of 0.1% improvement in reconstruction error is achievable.
- Further reduction in iteration counts is expected as the matrix dimension increases.
References


References


