Tracking Moving Sources Using The Rank-Revealing QR Factorization

Monique P. Fargues

Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943

Abstract

Subspace decomposition methods are a powerful tool used in different areas of signal processing in which the signal information is usually obtained via eigen-based methods. These techniques are numerically very stable but expensive to update. The rank revealing QR (RRQR) factorization described earlier by Chan provides an attractive alternative to accomplish subspace selection. We propose to add an updating capability to the RRQR factorization, and to apply it to the Direction Of Arrival problem. This technique allows for tracking of moving sources by taking advantage of the simplicity of the regular QR updating scheme and the rank-revealing property of the RRQR factorization.

1. Introduction

Subspace techniques are a powerful tool used in signal processing in which the signal information is obtained via eigen-based methods. Eigen-based updating schemes, which would allow for tracking of time-varying information, have also been proposed \[1,2,3,4\]. Eigendecomposition techniques are numerically stable but are computationally intensive, which is a drawback for real-time applications. Several researchers have proposed various alternatives in an attempt to decrease the computational load associated with the signal information estimation \[5,6,7,8,9\]. The rank-revealing QR algorithm described by Chan \[9\] provides an attractive alternative to accomplish subspace selection, and was applied to the Direction of Arrival (DOA) estimation problem recently \[8\]. In this paper, we investigate an updating scheme for the rank-revealing QR (RRQR) algorithm, and apply it to the DOA problem. Sections 2 and 3 briefly review subspace methods and the RRQR technique. In Section 4 we show how the RRQR algorithm can be used to update signal and noise subspaces. Section 5 presents experimental results, and comparisons with eigen-based signal and noise subspaces. Finally, conclusions are presented in Section 6.

2. Subspace Methods

Assume a linear equi-spaced array of \(N\) sensors receiving \(p\) narrowband signals. Under the assumption of non-dispersive propagation, sensors without distortion, and envelope variations that are slow relative to the carrier frequencies of the narrowband signals, the signal received by the passive array is:

\[ x(t) = M(\theta) s(t) + n(t) \]

where \(s(t)\) represents the narrowband signal, the mode matrix \(M(\theta)\) is defined as: \(M(\theta) = [m(\theta_1), \ldots, m(\theta_p)]\). The mode vector \(m(\theta)\) represents the delay with which the signal impinges on each of the sensors, and \(n(t)\) is additive noise. The mode vector is generally a nonlinear function of the signal arrival angle \(\theta\), center frequency \(\Omega\), array element spacing \(d\), and sensor response. For the above case, \(m(\theta)\) is given by the following vector expression:

\[ m(\theta) = [1, e^{-2\pi j z}, \ldots, e^{-2\pi j (N-1)z}] \]

where the normalized angle \(z\) is defined as \(z = d \sin(\theta)/\Omega\). Under the assumption that the zero mean noise \(n(t)\) is white and uncorrelated with the signals, the theoretical spatial correlation for the received signal is given by:

\[ R = E[x(t)x'(t)] = MSM' + \sigma^2 I \]

The sources are assumed to be uncorrelated, thus the signal correlation matrix is given by \(S = \text{diag}(S_1, \ldots, S_p)\), where \(S_i\) represents the power associated with each signal. The estimated correlation matrix obtained using \(n\) samples of the sensor output \(x(t)\) is given by:

\[ \hat{R} = \frac{1}{n} \sum_{t=1}^{n} x(t)x'(t) \]
where \( X = [x(1), \ldots, x(n)]^T \). Numerous high resolution techniques [15], based on the FVD or SVD have been proposed to identify the DOA information (number and location of sources) from the signal or noise subspace obtained from \( R \). The minimum-norm method [11] is used in this work to do this.

Reilly, et al [5] proposed to apply a QR factorization with a pre-selected pivoting scheme to the noise-free (signal) correlation matrix \( R_n = R - \sigma^2 I = MSM^H \) in an attempt to decrease the computational load associated with obtaining the DOA information. Thus, their work can be viewed as a QR factorization with "partial" pivoting. This specific "partial" pivoting could explain the loss of performance they found when comparing their algorithm with column pivoting is not completely rank deficient. They further showed that \( \Pi \) can be used as an approximation to the null space of \( A \). The RRQR algorithm applied to a \( N \) dimensional square matrix \( A \) is summarized below:

1. Compute a QR factorization of \( A \): \( \Pi = QU \)
2. Initialize \( W \) (of dimension \( N \times p \)) to zero
3. for \( i = N, \ldots, p + 1 \)
   a. Partition \( U = \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \)
   b. Compute an approximation to the minimum \( L \) singular value and the right singular vector \( v \) corresponding to \( U_{11} \)
   c. Compute a permutation \( P \) (of dimension \( i \times i \)) such that \( \|Pv\|_2 = \|Pv\|_\infty \)
   d. Assign \( \bar{v} = [v, 0]^T \) (of dimension \( N \times 1 \)) to the \( i \)th column of \( W \).
   e. Compute \( W = \bar{P}W \), where \( \bar{P} = \begin{bmatrix} p' & 0 \\ 0 & 1 \end{bmatrix} \)
   f. Compute the QR factorization: \( U_{11}P = Q_{11} \bar{U}_{11} \)
   g. \( \Pi \leftarrow \Pi \bar{P} \)
   h. \( Q \leftarrow Q_{11} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \)
   i. \( U \leftarrow \begin{bmatrix} U_{11} & Q_{11}U_{12} \\ 0 & U_{22} \end{bmatrix} \)
4. end of the for loop

Note that:
1. The QR factorization may be derived cheaply in step (3.1) using Givens rotations.
2. The matrix \( W \) is used to store the singular vectors corresponding to the smallest singular values of the successive matrices \( U_{11} \). Chan and Hansen [18] showed that the matrix \( W = \Pi W \) can be viewed as an approximation to the null space of \( A \). They further showed that \( W \) can be used as a starting matrix for inverse iteration to compute an accurate estimate of \( \text{Null}(A) \). Such a null space can be used obtain estimates of the source directions.
3. Prasad and Chandana [8] noted that the bounds to the singular values proposed by Chan [9] seem to be quite tight. Thus, these bounds could potentially be used to determine the number of sources with the MDL [16] or AIC [17] tests in lieu of the eigenvalues.

3. The Rank Revealing QR Factorization

The RRQR algorithm is derived in detail in [9]. Basically, the idea is to compute a QR factorization of \( A \) in the following form:

\[
\Pi = QU = \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}
\]

with \( \|U_{22}\| \) small, so that any rank deficiency of \( A \) will be revealed by the smaller lower right block in \( U \). The algorithm derived by Chan computes a permutation matrix \( \Pi \) which guarantees \( U_{22} \) to be small in norm when \( A \) is nearly rank deficient. Furthermore, Chan's derivation gives upper and lower bounds for the singular values of \( A \), and an approximate null space for \( A \). The RRQR algorithm applied to a \( N \) dimensional square matrix \( A \) is summarized below:

1. Compute a QR factorization of \( A \): \( \Pi = QU \)
2. Initialize \( W \) (of dimension \( N \times p \)) to zero
3. for \( i = N, \ldots, p + 1 \)
   a. Partition \( U = \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \)
   b. Compute an approximation to the minimum \( L \) singular value and the right singular vector \( v \) corresponding to \( U_{11} \)
   c. Compute a permutation \( P \) (of dimension \( i \times i \)) such that \( \|Pv\|_2 = \|Pv\|_\infty \)
   d. Assign \( \bar{v} = [v, 0]^T \) (of dimension \( N \times 1 \)) to the \( i \)th column of \( W \).
   e. Compute \( W = \bar{P}W \), where \( \bar{P} = \begin{bmatrix} p' & 0 \\ 0 & 1 \end{bmatrix} \)
   f. Compute the QR factorization: \( U_{11}P = Q_{11} \bar{U}_{11} \)
   g. \( \Pi \leftarrow \Pi \bar{P} \)
   h. \( Q \leftarrow Q_{11} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \)
   i. \( U \leftarrow \begin{bmatrix} U_{11} & Q_{11}U_{12} \\ 0 & U_{22} \end{bmatrix} \)
4. end of the for loop

Note that:
1. The QR factorization may be derived cheaply in step (3.1) using Givens rotations.
2. The matrix \( W \) is used to store the singular vectors corresponding to the smallest singular values of the successive matrices \( U_{11} \). Chan and Hansen [18] showed that the matrix \( W = \Pi W \) can be viewed as an approximation to the null space of \( A \). They further showed that \( W \) can be used as a starting matrix for inverse iteration to compute an accurate estimate of \( \text{Null}(A) \). Such a null space can be used obtain estimates of the source directions.
3. Prasad and Chandana [8] noted that the bounds to the singular values proposed by Chan [9] seem to be quite tight. Thus, these bounds could potentially be used to determine the number of sources with the MDL [16] or AIC [17] tests in lieu of the eigenvalues.
4. The matrix $Q_t = [q_1, ..., q_r]$ is an approximation of the signal subspace. It can be used to estimate values of the source directions. The null space span($W_n$) and signal subspace span($Q_t$) are used in this work to estimate the source locations and their performances are compared.

4. Updating using the RRQR Factorization

Tracking moving sources requires the correlation matrix to be updated and the signal and noise subspace to be continuously recomputed. The time varying estimated correlation matrix, obtained by adding a $(k+1)^{th}$ snapshot $x(k+1)$ and deleting the old snapshot $x(1)$, is given by:

$$R(k+1) = \frac{1}{n} \left[ x^H(k+1)x(k+1) - x(1)x^H(1) \right]$$

where $X = [x(1), ..., x(n)]^T$.

Several EVD and SVD based algorithms designed to track moving sources have been proposed [1, 2, 3]. Adam, et al [6] recently presented a new rank-revealing orthogonal transformation used to extract the noise subspace information needed for the DOA parameters. Bischof and Shroff [7] applied the RRQR factorization to the data matrix and used hyperbolic Householder transformations to update the noise subspace information. Our approach is different from these as we apply the RRQR factorization to the correlation matrix directly. The rank-1 modification of the QR factorization (without column pivoting) can be done in $O(n^2)$ operations [14]. However, recall that the noise free correlation matrix $R_n$, obtained using the estimated received signal correlation matrix $R$, is nearly rank deficient. Thus, pivoting is needed to assure a correct estimation of the signal and/or noise subspaces.

A basic flowchart of the updating scheme is presented in Figure 1. At each new iteration, we first apply the basic QR updating scheme to $R_n$. The RRQR restricted pivoting scheme [9] is applied next to reveal the numerical rank of $R_n$, and to obtain signal and noise subspace estimates. Note that, in our algorithm the initial rank-1 QR update computed at each time iteration before applying Chan’s pivoting scheme does not involve pivoting, while Chan’s method does use an initial QR decomposition with pivoting. Our simulations indicate no significant difference in the resulting estimated noise and signal subspaces. However, the original QR factorization is used to estimate minimum singular value and vectors needed in Chan’s pivoting scheme. The relationships between the initial QR pivoting, the accuracy of the estimated singular vectors used in the RRQR pivoting section, and approximated noise and signal subspaces are presently under study.

5. Simulation Results

We first study the performance of the RRQR factorization by comparing the signal and noise subspaces obtained by updating the noise free correlation matrix $R$, with those obtained using the eigendecomposition (EVD) of $R$. The largest principle angle between estimated subspaces [14] is used for comparison. Next, the minimum norm algorithm [11] is used to track the estimated source locations obtained using the RRQR factorization and the EVD algorithm.

A ten element array is used in the simulations to track two sources. The noise is assumed to be zero-mean gaussian and uncorrelated from sensor to sensor. We consider the case of two sources impinging on the array. The first source is assumed to be fixed at a normalized angle $0_1 = 40^\circ$, the second source location $0_2$ is linear time-varying. The two sources are assumed to have a SNR equal to 20dB, 10dB, and 6dB successively. 150 points are used to estimate the correlation matrix. Table 1 presents the average distance (for 200 updates) between the signal subspaces obtained using the RRQR updating technique, the QR with and without column pivoting (all QR-based techniques are obtained using $R_n$) and the signal subspace obtained from the eigendecomposition of $R$. Approximations to the QR-based noise subspace span($W_n$), where $W_n = \Pi W$, and span($W_{q_1}$), where $W_{q_1}$ is obtained after applying two inverse iteration steps to $W_n$, are considered also. Table 1 presents the distance between the estimated QR-based noise subspaces and the noise subspace obtained from the eigendecomposition of $R$. Table 1 indicates that the span($W_n$) can be considered as an approximation to the EVD-based noise subspace. Note that applying 2 inverse iteration steps to $W_n$ leads to a noise estimation identical as that obtained with the eigendecomposition of $R$. Table 1 also shows that the RRQR updating scheme leads to a better approximation of the signal subspace than the QR without column pivoting even when the SNR is low.

The minimum norm algorithm is applied to compute the estimated source locations next. Figure 2 presents the tracking of the source with linear time varying position $\theta_2$ obtained for the various signal and noise subspaces presented above. Figure 2 shows that the estimated $\theta_2$ computed using span($W_n$) has large variance for medium to small SNR, even though the average distance was found to be small in Table 1. Better tracking is obtained using $W_{q_1}$, as expected from the results given in Table 1. Furthermore, Figure 2 shows that a reasonably good approximation of the signal subspace can be obtained using the RRQR updating technique. Note however, that the source
estimated using the RRQR technique has a higher variance than that obtained using eigen-based techniques.

6. Conclusions

We have presented a simple technique for updating signal and noise subspaces from the noise-free correlation matrix. This method takes advantage of the simplicity of the regular QR updating technique and the rank-revealing QR factorization proposed recently. Simulations show that the RRQR updating algorithm approximates signal and noise subspaces which can be used to track moving sources.

Initialization
\[ R = XX' - n o^{2}I \]
RRQR factorization
\[ R, \Pi = QU \]

Start updating
\[ R,(k + 1) = R,(k) + x(k + 1)x'^{(k + 1)} - x(1)x'^{(1)} \]
- Update the above QR factorization
- Apply 2 successive rank-one modifications to \( R,(k) \)
- Apply Chan’s pivoting scheme to \( \Pi, Q, U \)

(steps 2,3 section 3)
- Form: \( \Pi, Q, U \)
- Set \( \Pi = \Pi_{2} \)
- \( Q = Q_{2} \)
- \( U = U_{2} \)
- Identify signal subspace \( Q \)
- Identify approximated noise subspace \( W' = \Pi \)
- Option to improve \( W' \) by applying inverse iteration [18]

\[ k = k + 1 \]

Figure 1. RRQR updating algorithm

References

Figure 2. Linear time-varying frequency tracking for QR-based and eigen-based signal and noise subspaces: SNR = 20dB, 10dB, 6dB

Table 1. Distance between signal (noise) QR-based and eigen-based subspaces - 200 updates ($\theta_1 = 40^\circ$, $\theta_2$ linear time-varying)

<table>
<thead>
<tr>
<th></th>
<th>SNR 20dB</th>
<th>SNR 10dB</th>
<th>SNR 6dB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Signal subspace</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RRQR</td>
<td>0.164, 0.054</td>
<td>1.721, 0.536</td>
<td>4.2684, 1.193</td>
</tr>
<tr>
<td>QR with pivoting</td>
<td>0.3770, 0.094</td>
<td>2.716, 0.448</td>
<td>5.5736, 1.673</td>
</tr>
<tr>
<td>QR w/o pivoting</td>
<td>1.4502, 0.352</td>
<td>12.064, 2.216</td>
<td>26.6544, 3.644</td>
</tr>
<tr>
<td><strong>Noise subspace</strong></td>
<td>mean, standard deviation (in degrees)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$W_n = \Pi W$</td>
<td>0.0023, 0.0022</td>
<td>0.2859, 0.4912</td>
<td>1.585, 1.975</td>
</tr>
<tr>
<td>$W_n$ (2 inv. iteration steps applied to $W_n$)</td>
<td>0, 0</td>
<td>0, 0</td>
<td>0, 0001, 2.10^-4</td>
</tr>
</tbody>
</table>