Direction of Arrival Estimation Using DAS and MUSIC Beamforming

Iman Javaheri Neyestanak

Peyman Javaheri Neyestanak

2024-25

Introduction

This project investigates direction-of-arrival (DoA) estimation of multiple sources using a linear microphone array. Two classical beamforming methods were implemented and compared: Delay-and-Sum (DAS) and MUSIC (Multiple Signal Classification). The objective was to analyze their performance in resolving closely spaced sources and the impact of microphone spacing on resolution and accuracy.

Methodology

- **Array configuration**

To build the linear microphone array, we first need to find the minimum number of microphones needed to achieve a resolution of 2 degrees. For a frequency of 5000Hz, we have:

$$\lambda = \frac{c}{f} = \frac{343}{5000} = 0.0686$$

$$d_x = \frac{\lambda}{2} = 0.0343$$

We want to have a resolution of 2 degrees, so

$$\Delta\theta = 2^\circ = 0.035 rad$$

$$\Delta\theta = \frac{\lambda}{D} \rightarrow D = 1.96$$

And now we can calculate the number of microphones needed as follows:

$$D = (N-1)d_x$$
$$1.96 = (N-1)0.0343 \rightarrow N = 59$$

- **Sources**

Two sources were simulated at different angles.

- **Signal Model**

Each source transmits a narrowband complex sinusoidal signal. The received signals at the array are modeled as sums of steering vectors corresponding to target directions, plus additive complex Gaussian noise.

- **Algorithms Implemented:**

- DAS Beamforming: Simple coherent summation of signals across microphones for each scan angle.
- MUSIC Algorithm: Subspace-based method using covariance matrix eigen decomposition to distinguish signal and noise subspaces, allowing high-resolution DoA estimation.

Code implementation

The project includes two main models for direction-of-arrival estimation: Delay-and-Sum (DAS) Beamforming and the MUSIC Algorithm. Both models use simulated signals at a linear microphone array but differ fundamentally in their processing methods.

Delay-and-Sum (DAS) Beamforming:

- **Signal Model:**

The received signal is modeled as a sum of phase-shifted narrowband signals from all sources plus noise.

- **Implementation Steps:**

1. Compute the steering vector for each scanning angle across the search range.
2. For each scanning angle, calculate the beamformer output as the squared magnitude of the inner product between the steering vector and the received signal vector:

$$P(\theta) = \left| a(\theta)^H x \right|^2$$

3. Normalize the output and find peaks corresponding to source directions.

```
% Loop over scanning angles
for idx = 1:length(theta_scan_rad)
    % Compute steering vector for the scanning angle
    steering_vector = exp(-1j * k * mic_positions * sin(theta_scan_rad(idx)));

    % Compute beamformer output (power)
    beamforming_output(idx) = abs(steering_vector' * received_signal)^2;
end

% Normalize the beamforming output
beamforming_output = beamforming_output / max(beamforming_output);

% Find peaks in the beamforming output
[pks, locs] = findpeaks(beamforming_output, 'MinPeakHeight', 0.5, 'SortStr',
'descend', 'NPeaks', K);
```

MUSIC (Multiple Signal Classification) Algorithm

- **Signal Model:**

Uses multiple snapshots of received signals to estimate the covariance matrix. The eigen-decomposition separates the signal and noise subspaces.

- **Implementation Steps:**

1. Generate multiple snapshots of received signals with noise.
2. Estimate the **covariance matrix** R_{xx} from the snapshots.
3. Perform eigen-decomposition and extract the **noise subspace** (eigenvectors corresponding to smallest eigenvalues).
4. For each scanning angle, compute the MUSIC spectrum using:

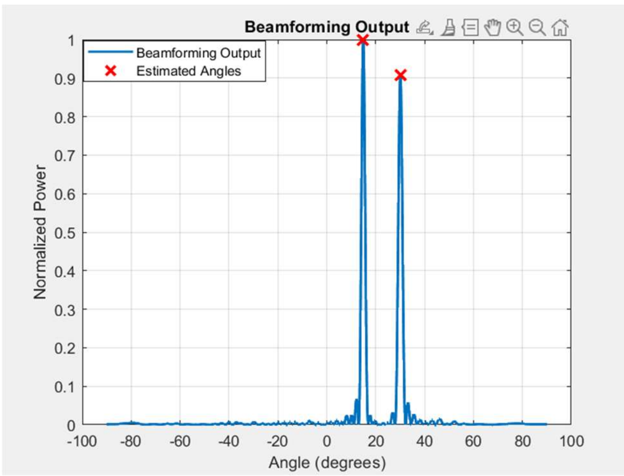
$$P_{MUSIC}(\theta) = \frac{1}{a(\theta)^H E_n E_n^H a(\theta)}$$

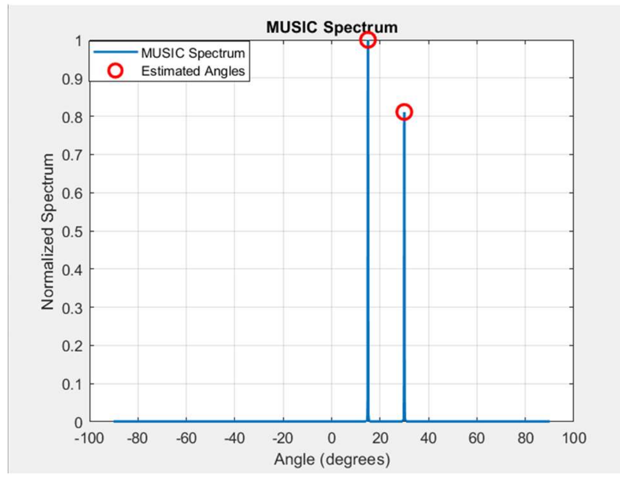
5. Normalize the spectrum and detect peaks indicating source angles.

```
% Estimate covariance matrix (N x N)
Rxx = (received_signals_snapshots * received_signals_snapshots') / L;

% Eigen-decomposition of covariance matrix
[Evec, Eval] = eig(Rxx);
[evals_sorted, idx] = sort(diag(Eval), 'descend');  % Sort eigenvalues
Evec = Evec(:, idx);  % Sort eigenvectors accordingly

% Separate signal and noise subspaces
En = Evec(:, K+1:end);  % Noise subspace (N x (N-K))

% MUSIC spectrum computation
music_spectrum = zeros(size(theta_scan));

for idx = 1:length(theta_scan_rad)
    a_theta = exp(-1j * k * mic_positions * sin(theta_scan_rad(idx)));  % N x 1
    music_spectrum(idx) = 1 / abs(a_theta' * (En * En') * a_theta);
end

% Normalize the MUSIC spectrum
music_spectrum = music_spectrum / max(music_spectrum);
```

Results and Discussion

First, we analyze two separate sources.





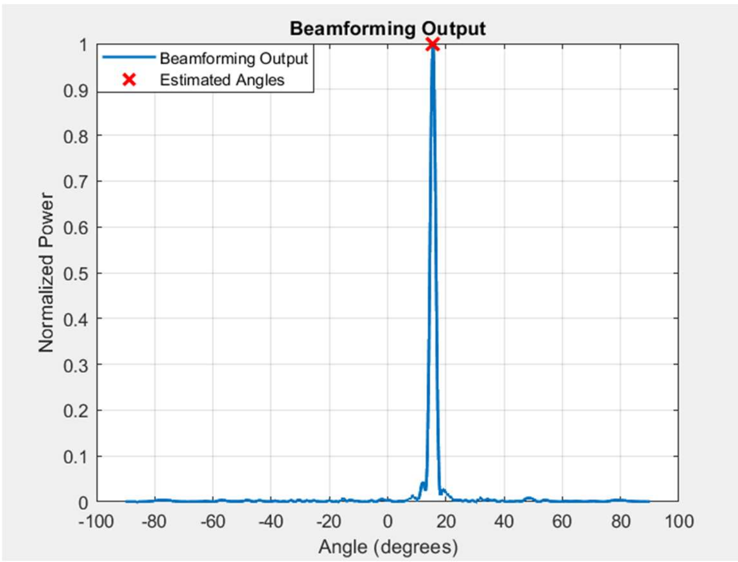

```
True Angles: 15.00 degrees
True Angles: 30.00 degrees

--- DAS Beamforming Results ---
Estimated Angles: 14.90 degrees
Estimated Angles: 30.10 degrees

--- MUSIC Results ---
Estimated Angles: 15.00 degrees
Estimated Angles: 30.00 degrees
```

We can see that both the methods could successfully distinguish the sources. Also it can be seen that the Music performs slightly better.

Effect of Closely Positioned Sources

- DAS Beamforming:

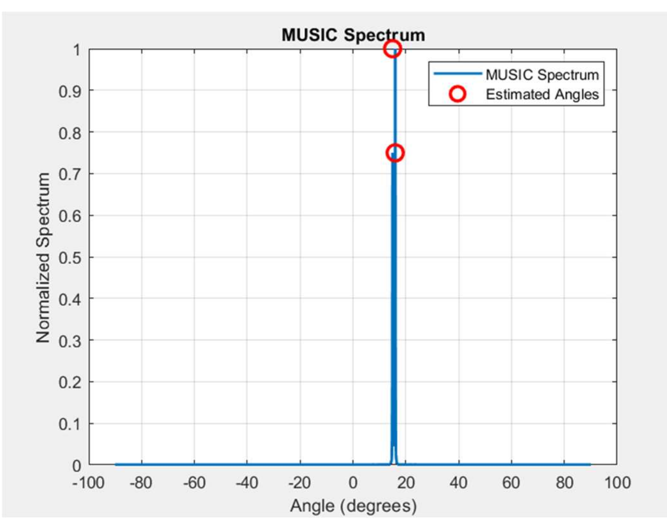
The DAS method showed limited resolution when two sources were very close (1° separation). The beamformer output exhibited a single merged peak rather than two distinct peaks, indicating difficulty in resolving sources with small angular separation.




```
True Angles: 15.00 degrees
True Angles: 16.00 degrees

--- DAS Beamforming Results ---
Estimated Angles: 15.60 degrees
```

- MUSIC Algorithm:

MUSIC demonstrated superior resolution, successfully identifying two distinct peaks even at 1° separation when implemented with appropriate snapshot averaging and noise conditions. This highlights MUSIC's advantage due to its subspace processing, enabling it to resolve sources below the conventional beamwidth limit of DAS.



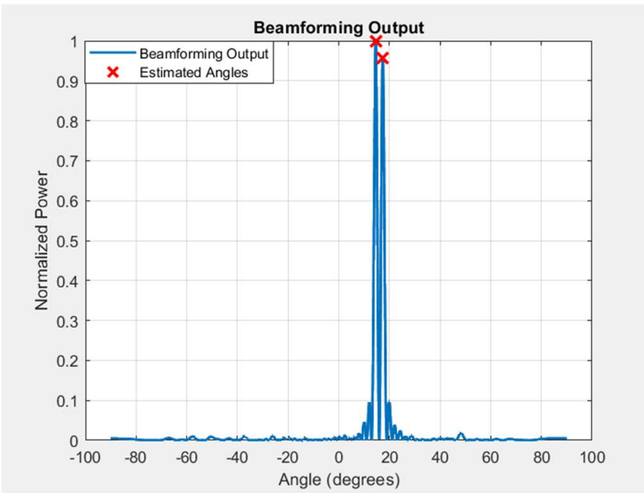

```
--- MUSIC Results ---
Estimated Angles (MUSIC): 15.00 degrees
Estimated Angles (MUSIC): 16.00 degrees
```

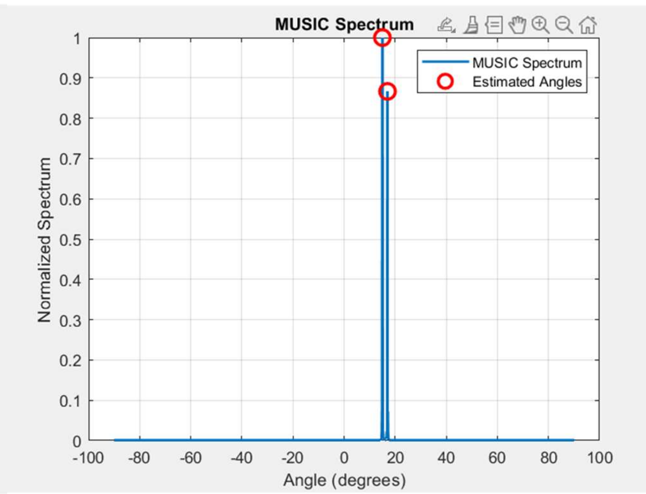
Effect of Changing Microphone Spacing

In this part we are analyzing the effect of changing the distance between the microphones.

- **Spacing at d=λ/2**

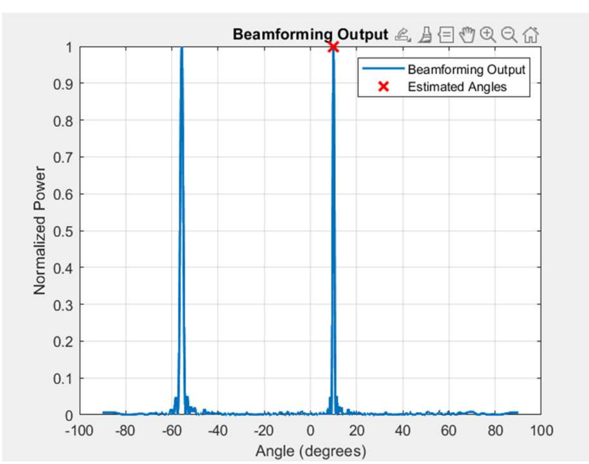
Both methods performed optimally with no spatial aliasing (grating lobes) and good resolution. MUSIC showed sharper peaks due to its higher resolution capability.

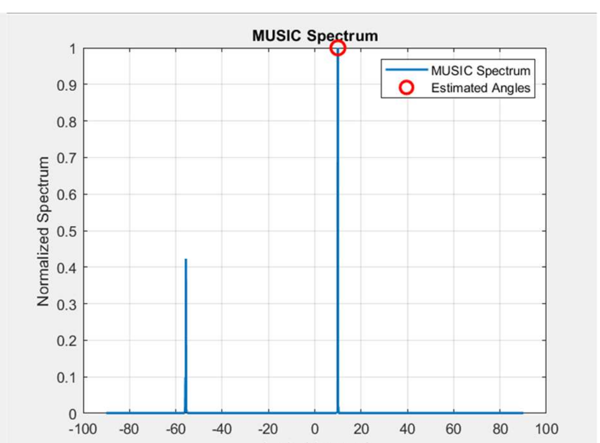




- **Spacing at d>λ/2**

DAS exhibited grating lobes (false peaks) due to spatial aliasing, degrading the reliability of DoA estimates. MUSIC showed better robustness against grating lobes but could still experience performance degradation for very large spacing. Note that here we used just one source.





Conclusion

The project successfully demonstrated the strengths and limitations of DAS and MUSIC for DoA estimation with a linear microphone array. The MUSIC algorithm outperforms DAS in resolving closely spaced sources due to its ability to exploit signal subspace structure. The microphone spacing critically affects resolution and accuracy: half-wavelength spacing provides the best trade-off between avoiding aliasing and maximizing angular resolution.