Bachelor project - Channel splicing methods for wireless sensing

Adam Chahed Ouazzani

Telecommunications Circuits Laboratory
Ecole polytechnique fédérale de Lausanne, Switzerland

June 10, 2024

Introduction

Channel Splicing in Telecommunications

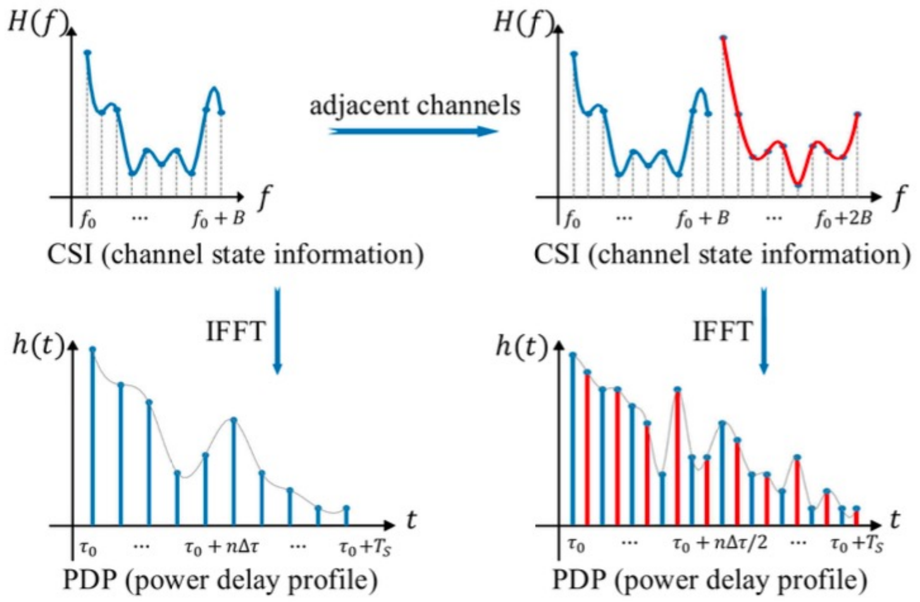
- **Enhances** sensing and localization accuracy
- **Combines** multiple frequency bands for better resolution in the impulse response

Importance

- **Localization**: High-precision positioning
- **Sensing**: Environmental monitoring, motion detection, gesture recognition
- **Hardware :** advantage of using cheap hardware that cannot measure wideband channels directly

Project Objectives

- **Improve Sensing Accuracy**: Using channel splicing
- **Evaluate Performance**: Chronos, MUSIC, and MVDR across scenarios



Project Approach

- **Literature Study :**
  - Understanding existing solution
  - Identifying limitations
- **MATLAB Implementation :**
  - Implementing and testing reconstruction methods
  - Simulating different scenarios
- **Results Analysis :**
  - Evaluating methods performance
  - Identifying the best approaches

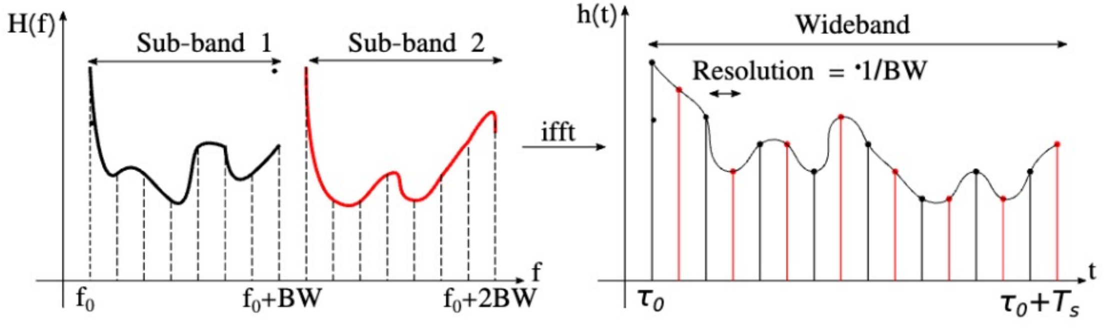
Methodology

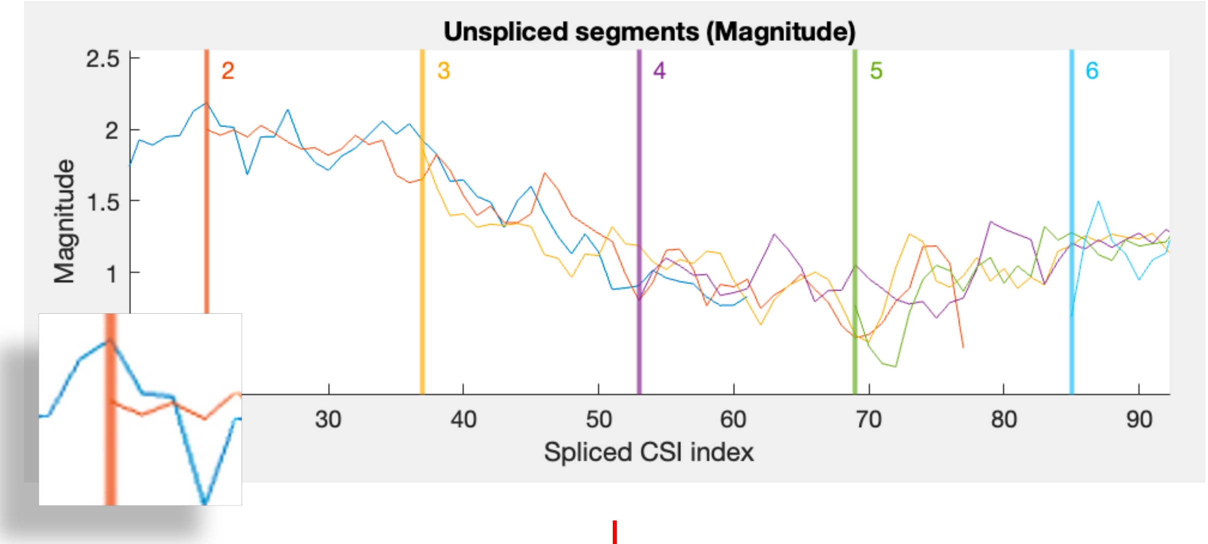
- **Splicing technique**
- **CIR Reconstruction methods**
  - Chronos
  - Music
  - MVDR
- **Data**

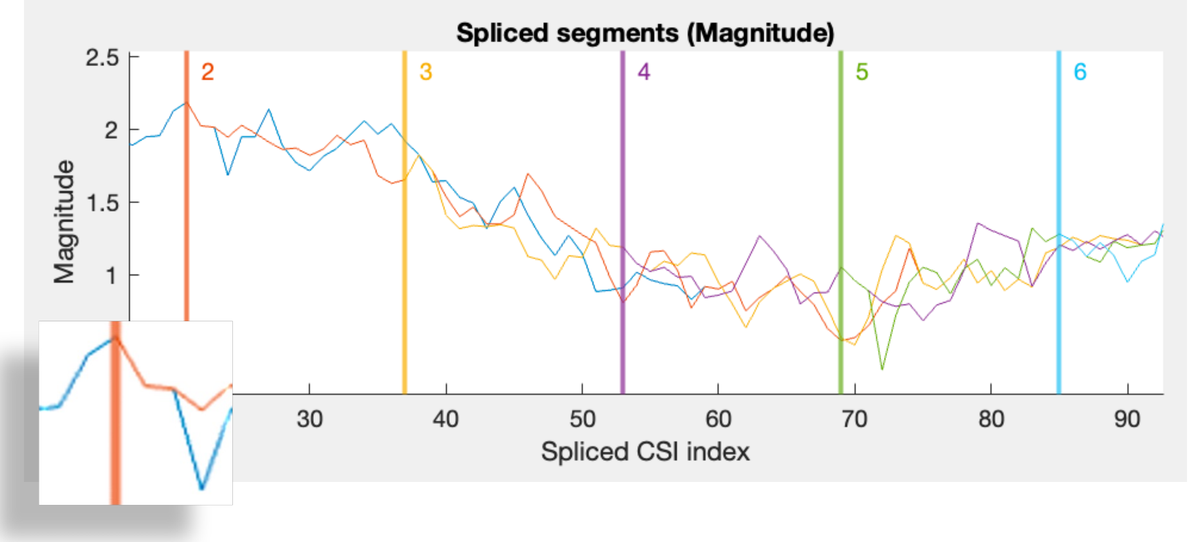
Channel splicing - Splicing technique

- Let $\mathrm{CSI}_i(f)$ be the CSI frequency for the i-th segment.
- **Shifted CSI** : $\mathrm{CSI}_i(f + f_i)$
- **Spliced CSI** : $\mathrm{CSI}_{\mathrm{spliced}}(f) = \sum_{i=1}^{N} \mathrm{CSI}_i(f + f_i)$
- **CIR from spliced CSI :**

$$\mathrm{CIR}(t) = \mathrm{IDFT}(\mathrm{CSI}_{\mathrm{spliced}}(f))$$







Chronos – how does it work ?

- Utilizes **Non-uniform Discrete Fourier Transform** (NDFT).
- Introduces **sparsity constraint** to solve the under-determined problem.
- Minimizes the L-1 norm to favor **dominant paths** in the signal.
- **Iterative algorithm with gradient descent** and sparsification steps.

Chronos – Mathematical formulation and Algorithm

- Let p be the **sampled inverse-NDFT** at discrete times τ.
- **Optimization problem** to solve for inverse NDFT $\min \|\mathbf{p}\|_1$

Subject to : $\|\mathbf{h} - F\mathbf{p}\|_2^2 = 0$

- Reformulated using **Lagrange multipliers** :

$$\min_{\mathbf{p}} \|\mathbf{h} - F\mathbf{p}\|_2^2 + \alpha\|\mathbf{p}\|_1$$

Where :

- F is the Fourier matrix
- h is the measured CSI vector.
- ‖·‖1 promotes sparsity.
- ‖·‖2 ensures the solution fits the measured data.

1 Algorithm to Compute Inverse NDFT

```
▷ Given: Measured Channels, h̃
▷ 𝓕: Non-uniform DFT matrix, such that 𝓕_{i,k} = e^{-j2πf_{i,0}τ_k}
▷ α: Sparsity parameter; ε: Convergence Parameter
▷ Output: Inverse-NDFT, p
▷ Initialize p_0 to a random value, t = 0, γ = 1/||𝓕||_2.
while converged = false do
    p_{t+1} = SPARSIFY(p_t − γ𝓕*(𝓕p_t − h̃), γα)
    if ||p_{t+1} − p_t||_2 < ε then
        converged = true
        p = p_{t+1}
    else
        t = t + 1
    end if
end while
function SPARSIFY(p,t)
    for i = 1, 2, ...length(p) do
        if |p_i| < t then
            p_i = 0
        else
            p_i = p_i (|p_i| − t)/|p_i|
        end if
    end for
end function
```

MUSIC – How does it work ?

- Identifies **subspaces** with **Covariance matrix**
- Separates **signal and noise subspaces** via **eigenvalue decomposition.**
- Uses the **largest** eigenvalues for the **signal** subspace and **smallest** for the **noise** subspace.
- Computes **pseudospectrum** to detect **times of arrival** by identifying peaks.

MUSIC – Mathematical formulation and Algorithm

- **Covariance matrix** $R = E[xx^H]$
- **Eigenvalue decomposition** $R = E\Lambda E^H$
- **Signal and noise subspace** E_s and E_n
- **Pseudospectrum** $P(\theta) = \frac{1}{a^H(\theta)E_n E_n^H a(\theta)}$

Algorithm

1. Construct the covariance matrix R from the measured CSI data.
2. Perform eigenvalue decomposition on R to get the eigenvectors and eigenvalues.
3. Identify the signal and noise subspaces from the eigenvectors.
4. Compute the pseudospectrum using the noise subspace.
5. Identify peaks in the pseudospectrum that correspond to the time of arrival.

MVDR – How does it work ?

- Enhances resolution by **minimizing interference** from other paths.
- Focuses on the direction of the desired signal by using the **covariance matrix**.
- Uses **beamforming techniques to suppress noise** and interference.
- Computes a pseudospectrum to reveal times of arrival.
- **Identifies key signal paths** by locating peaks in the pseudospectrum.

MVDR – Mathematical formulation and Algorithm

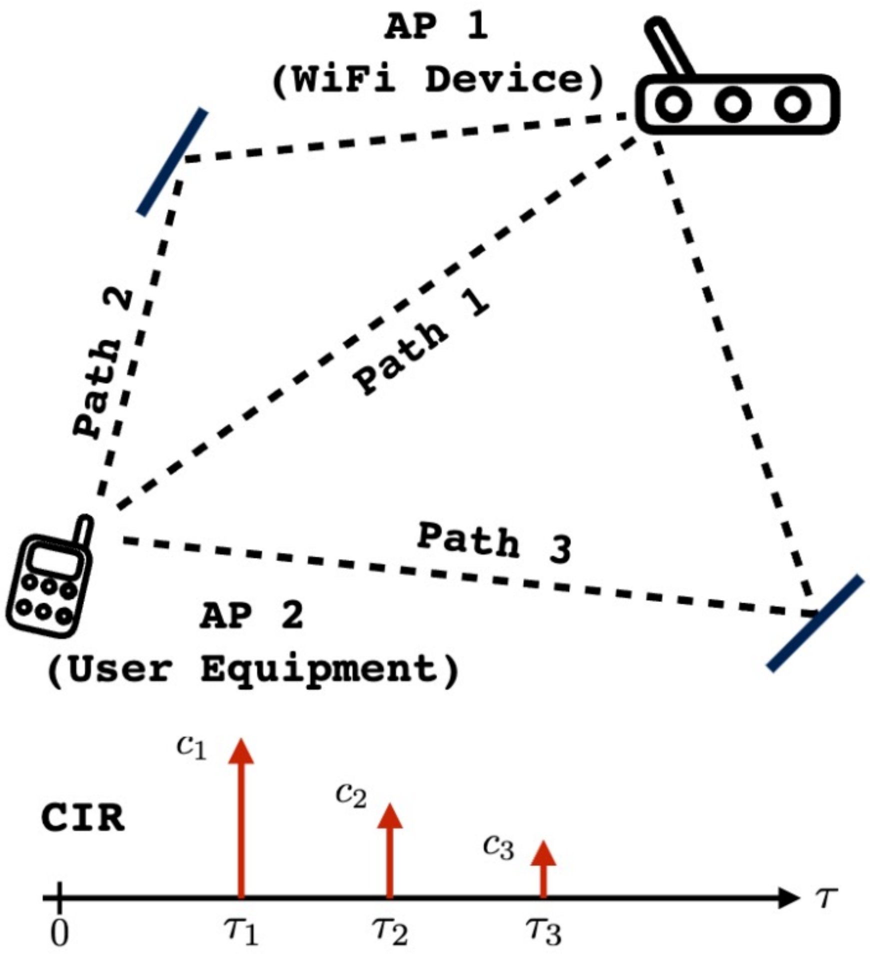
- **Covariance matrix** $R = E[xx^H]$
- **MVDR beamformer** $w = \frac{R^{-1}a(\theta)}{a^H(\theta)R^{-1}a(\theta)}$
- **Pseudospectrum** $P(\theta) = \frac{1}{a^H(\theta)R^{-1}a(\theta)}$

Algorithm

1. Construct the covariance matrix from measured CSI data.
2. Compute the inverse of the covariance matrix
3. Compute the MVDR beamformer
4. Compute the MVDR pseudospectrum
5. Identify peaks in the pseudospectrum that correspond to the time of arrival.

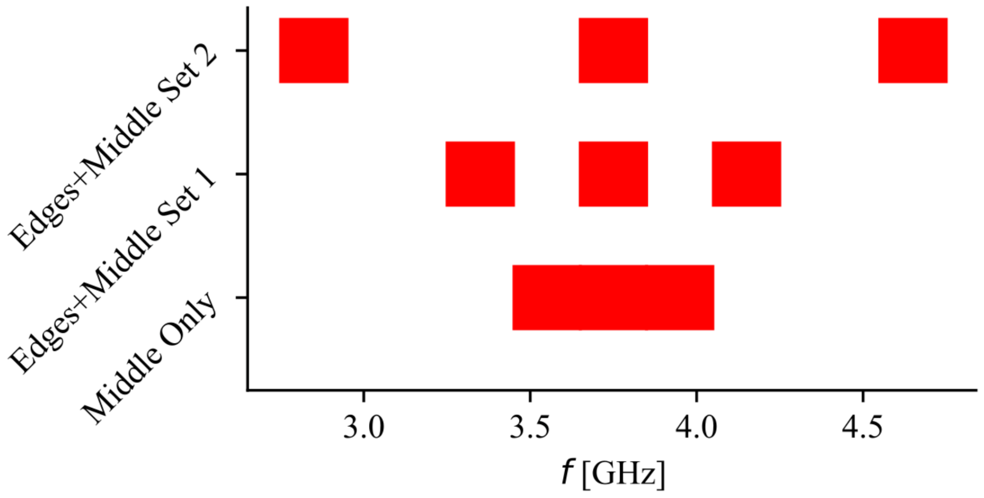
Data generation : Simulated data

- Simulate **two paths** channel with **Rayleigh fading.**
- Generate Channel State Information (CSI) across **multiple carrier frequencies**. (Carrier frequency spacing = 5MHZ, standard for WI-FI)
- Adds **a random phase offset** to the signal to simulate phase variations due to frequency changes.
- **SNR** (Signal to noise ratio) set to 10 DB.
- **Subcarrier spacing** – 312.5khz (standard for OFDM)



Data generation : Gaps

- Introduce **gaps** by removing **specific frequency segments**.
- Test the impact of missing data on the **reconstruction accuracy**.
- **Different gaps for two scenarios**
  - **Scenario 1 :**
    - Alternate on all BW by keeping **62.5**/**31.25** MHZ and keeping **31.25**/**15.6** MHZ each time for **1015** /**220** MHZ

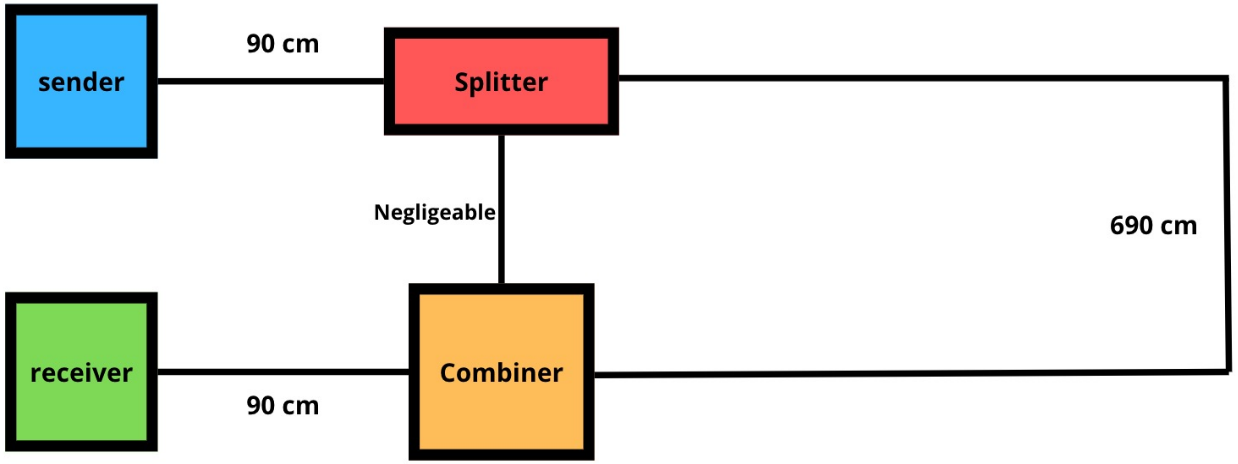


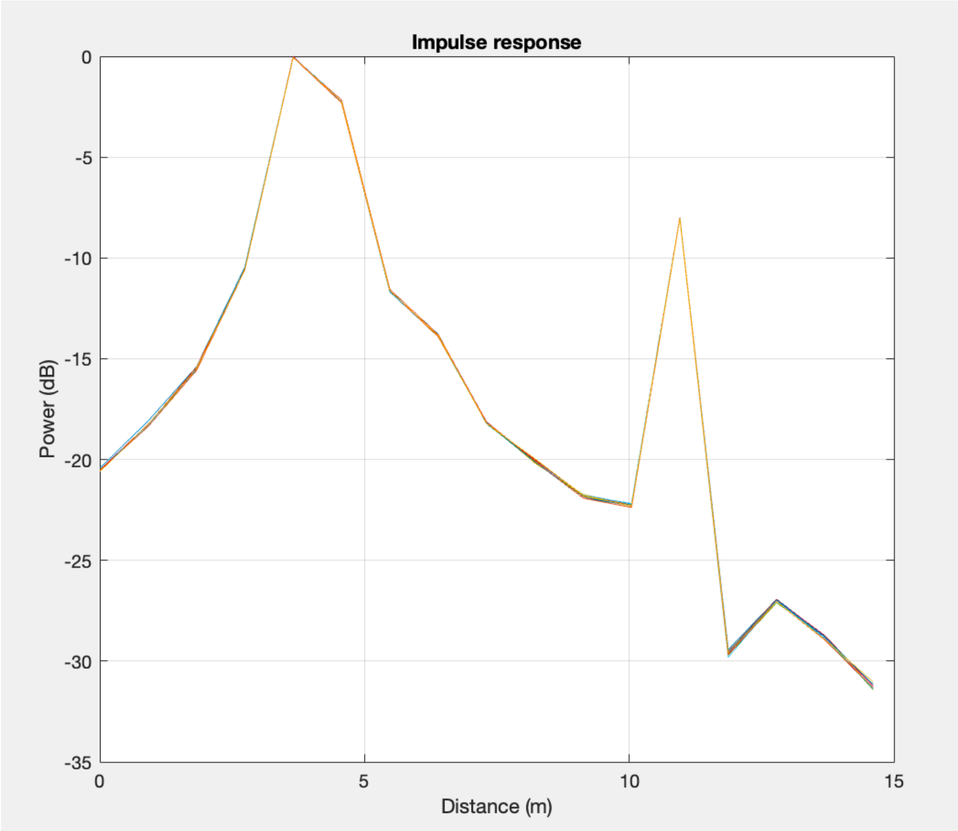
  - **Scenario 2 :**
    - Remove **187.5**/**31.25** MHZ at beginning and end + Alternate on remaining BW by keeping **62.5**/**31.25** MHZ and removing **31.25**/**15.6** MHZ each time for **1015**/**220** MHZ

Data generation : Real Data

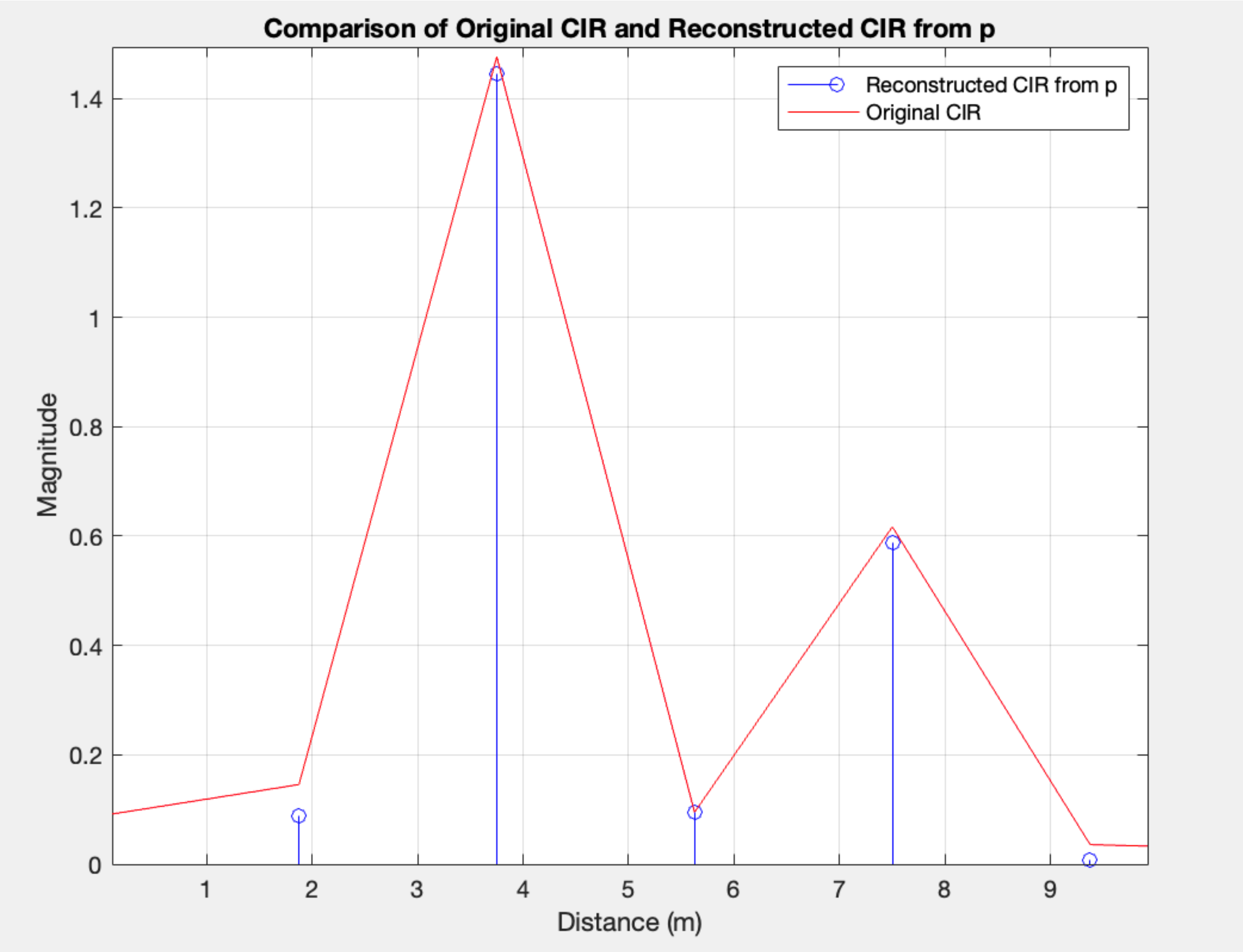
- Collect real **measured** data from a **cable** setup → 2 paths

- **Validate** the reconstruction methods with **real data**.



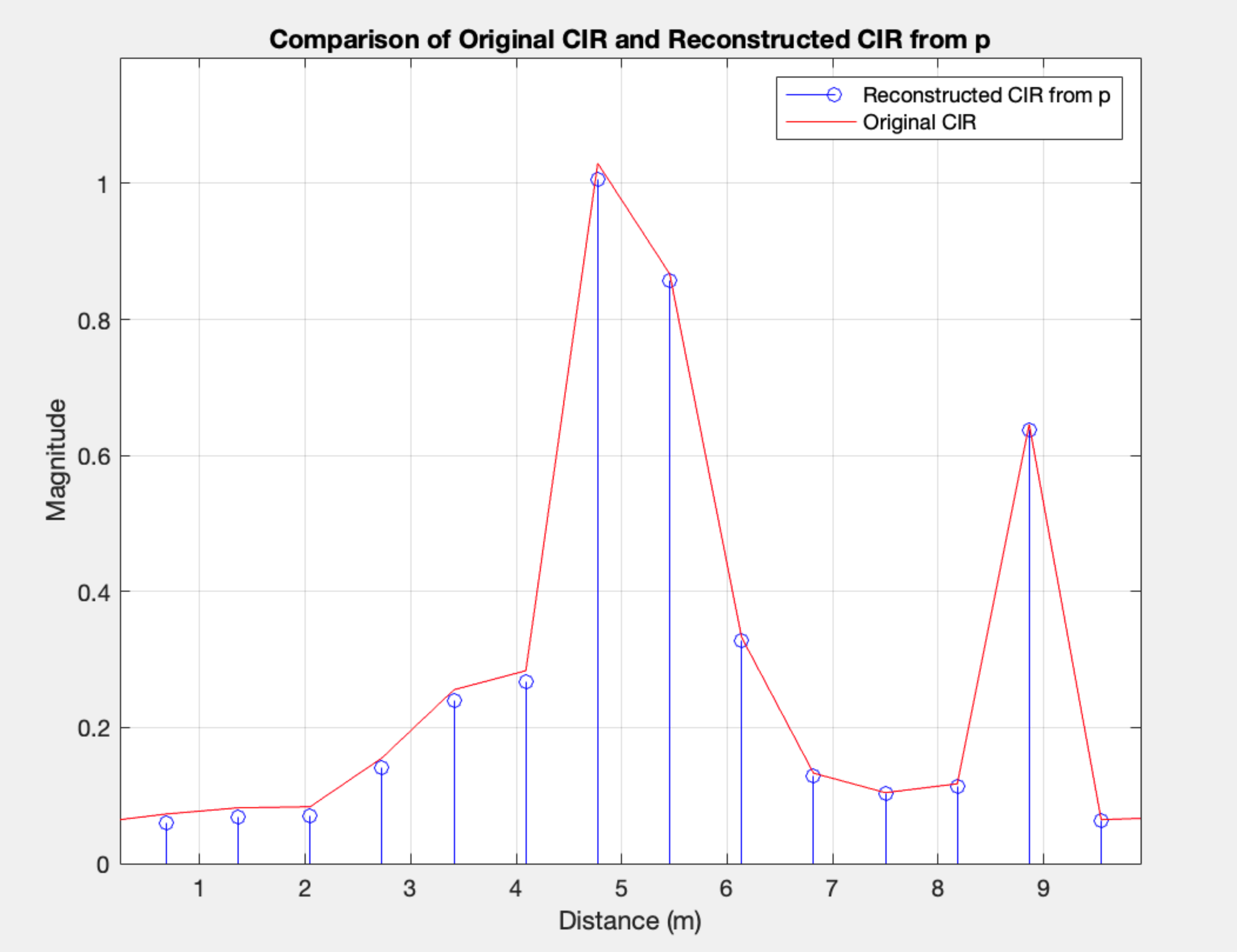


Results – Chronos



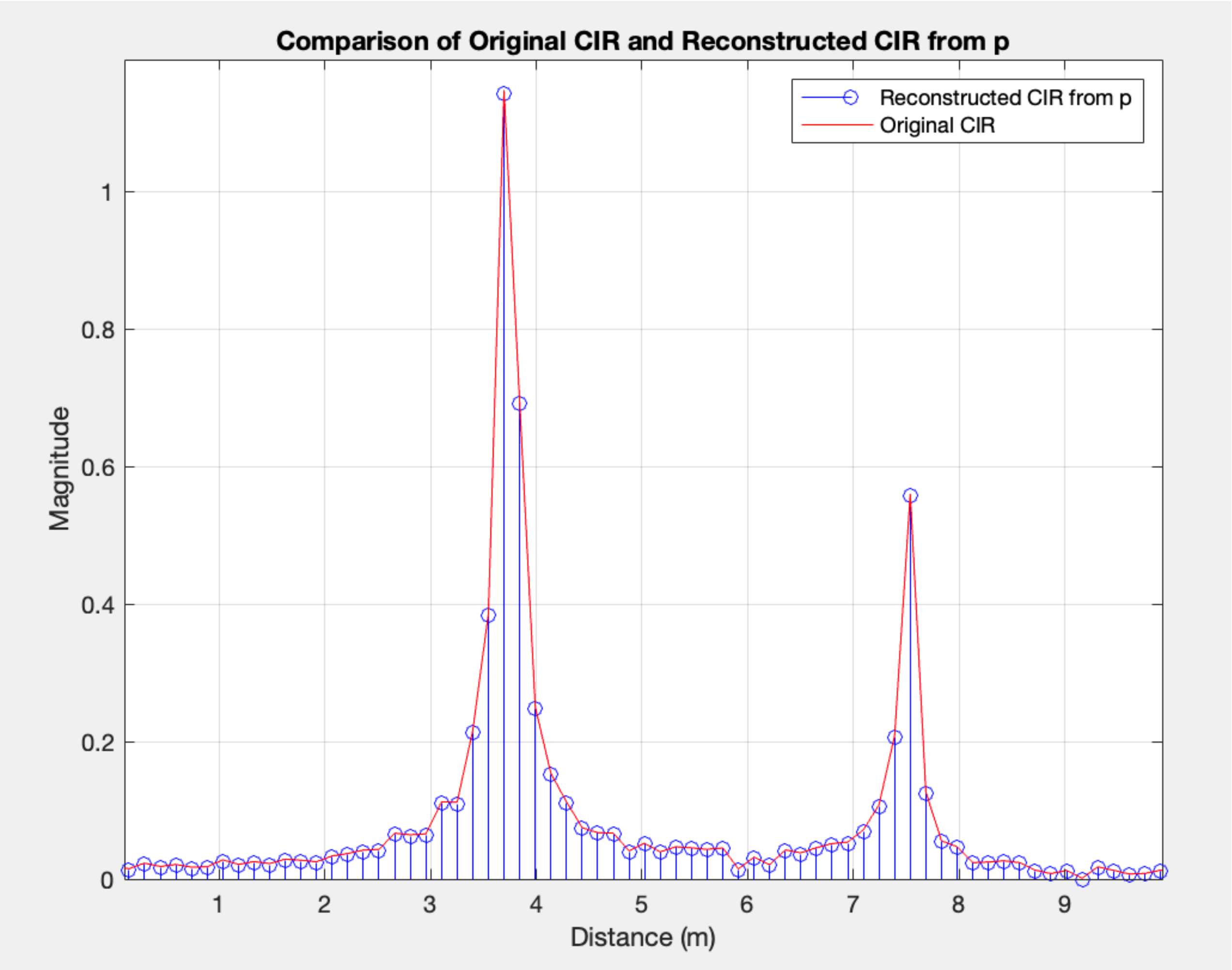

```
Splicing information:
        Original bandwidth = 20 MHz
        New bandwidth = 80 MHz
Selected scenario: 2
        Attenuations: [0 -6]
        Delays: [0 2.5e-08]
        Bandwidths needed for all paths: 40.00 MHz
```

Results – Chronos



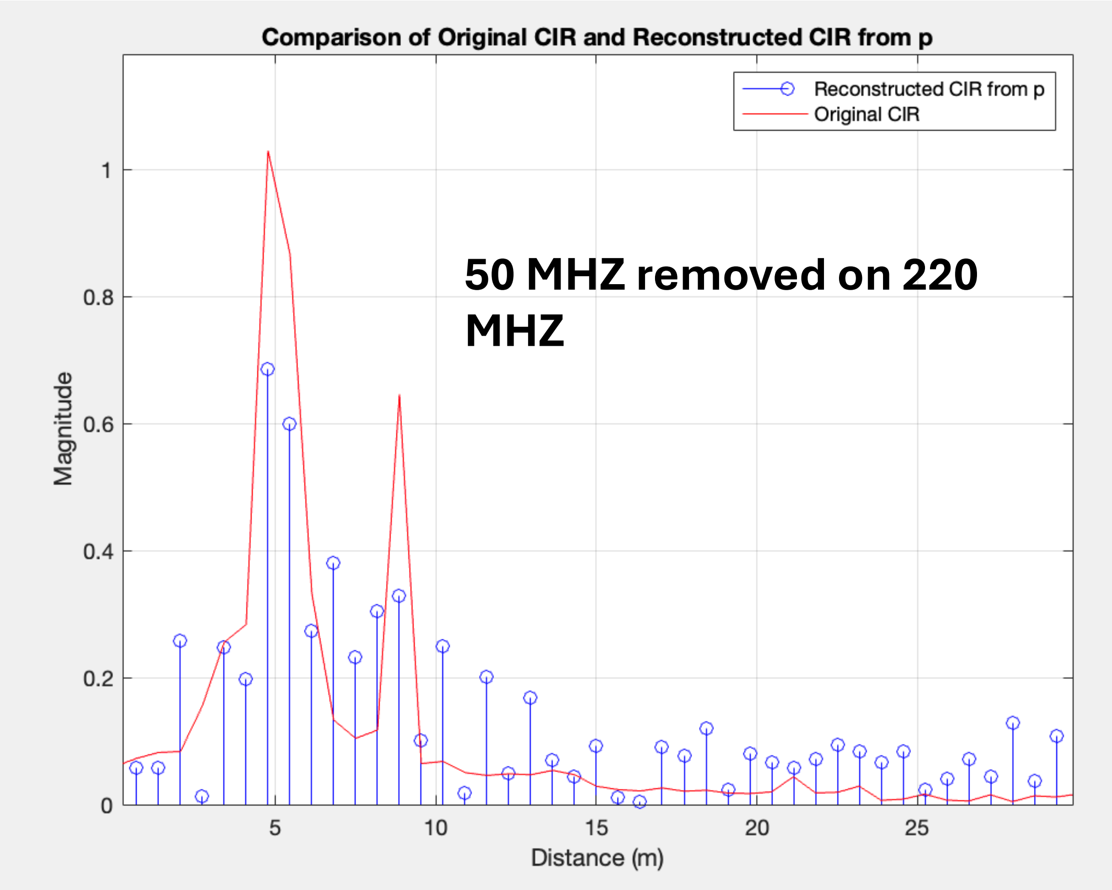

```
Splicing information:
        Original bandwidth = 20 MHz
        New bandwidth = 220 MHz
Selected scenario: 2
        Attenuations: [0 -6]
        Delays: [0 2.5e-08]
        Bandwidths needed for all paths: 40.00 MHz
```

Results – Chronos

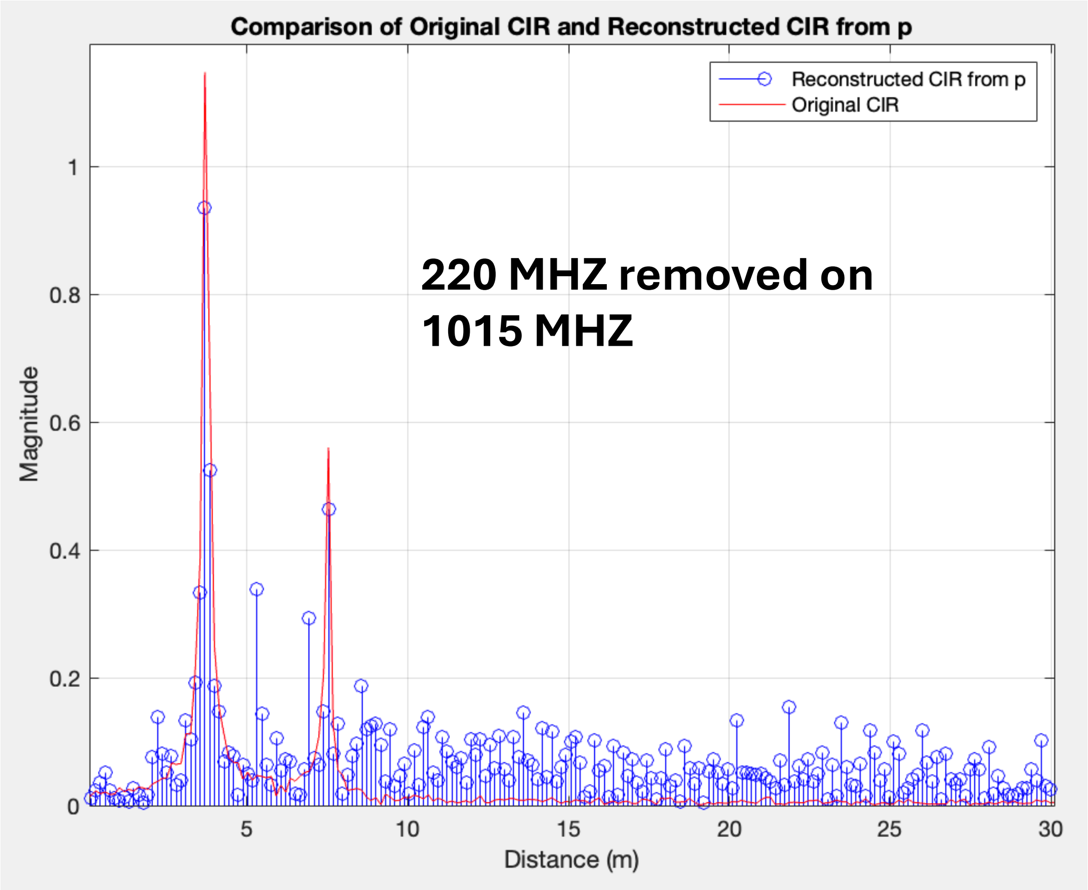



```
Splicing information:
        Original bandwidth = 20 MHz
        New bandwidth = 1015 MHz
Selected scenario: 2
        Attenuations: [0 -6]
        Delays: [0 2.5e-08]
        Bandwidths needed for all paths: 40.00 MHz
```

Results – Chronos with Alternating gaps

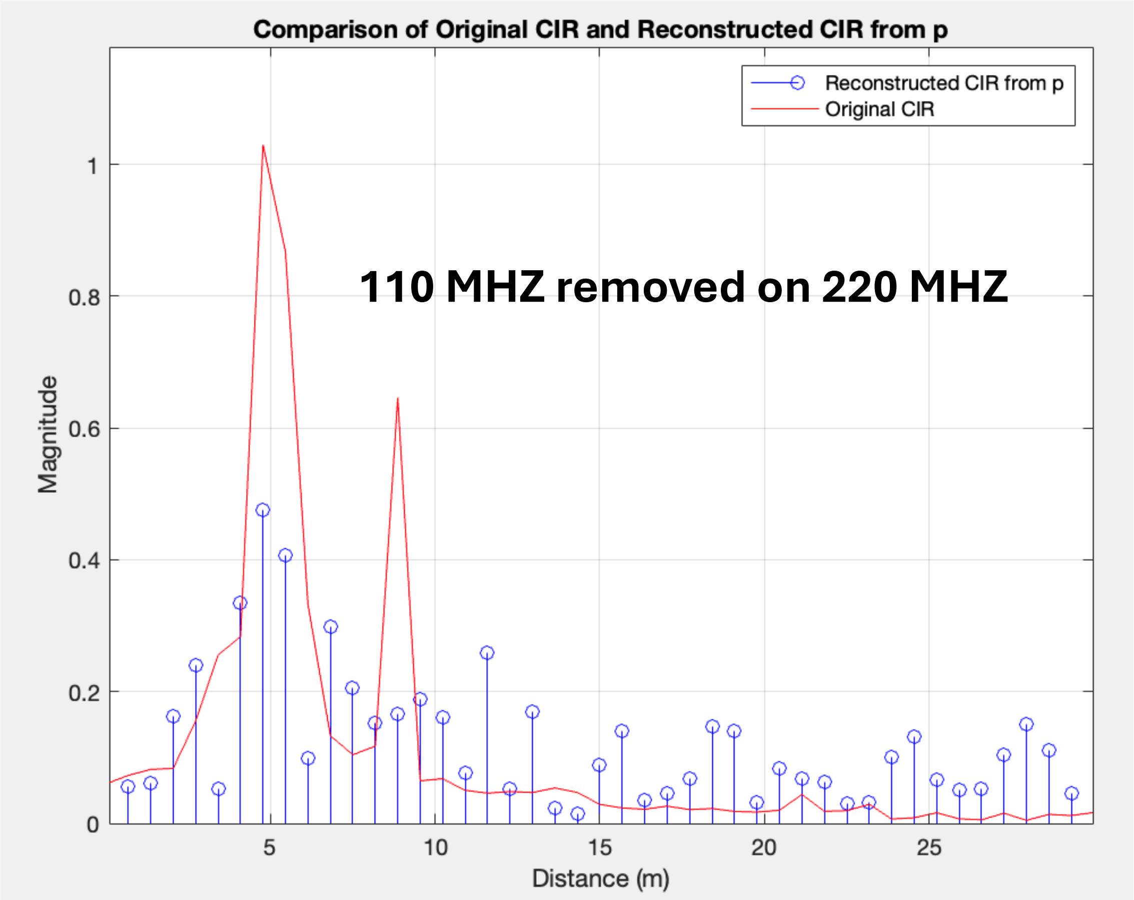


220MHZ

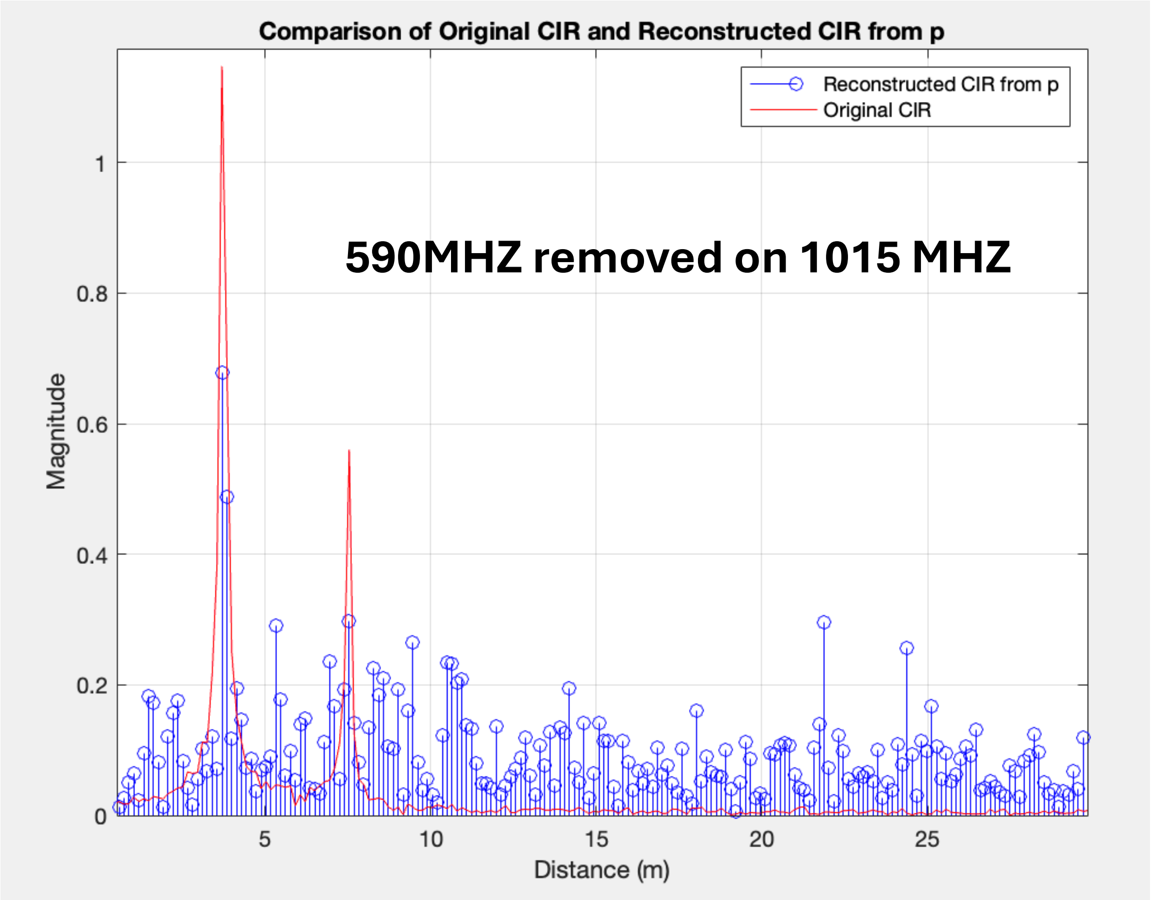


1015 MHZ

Results – Chronos with Alternating gaps and beginning / end removed

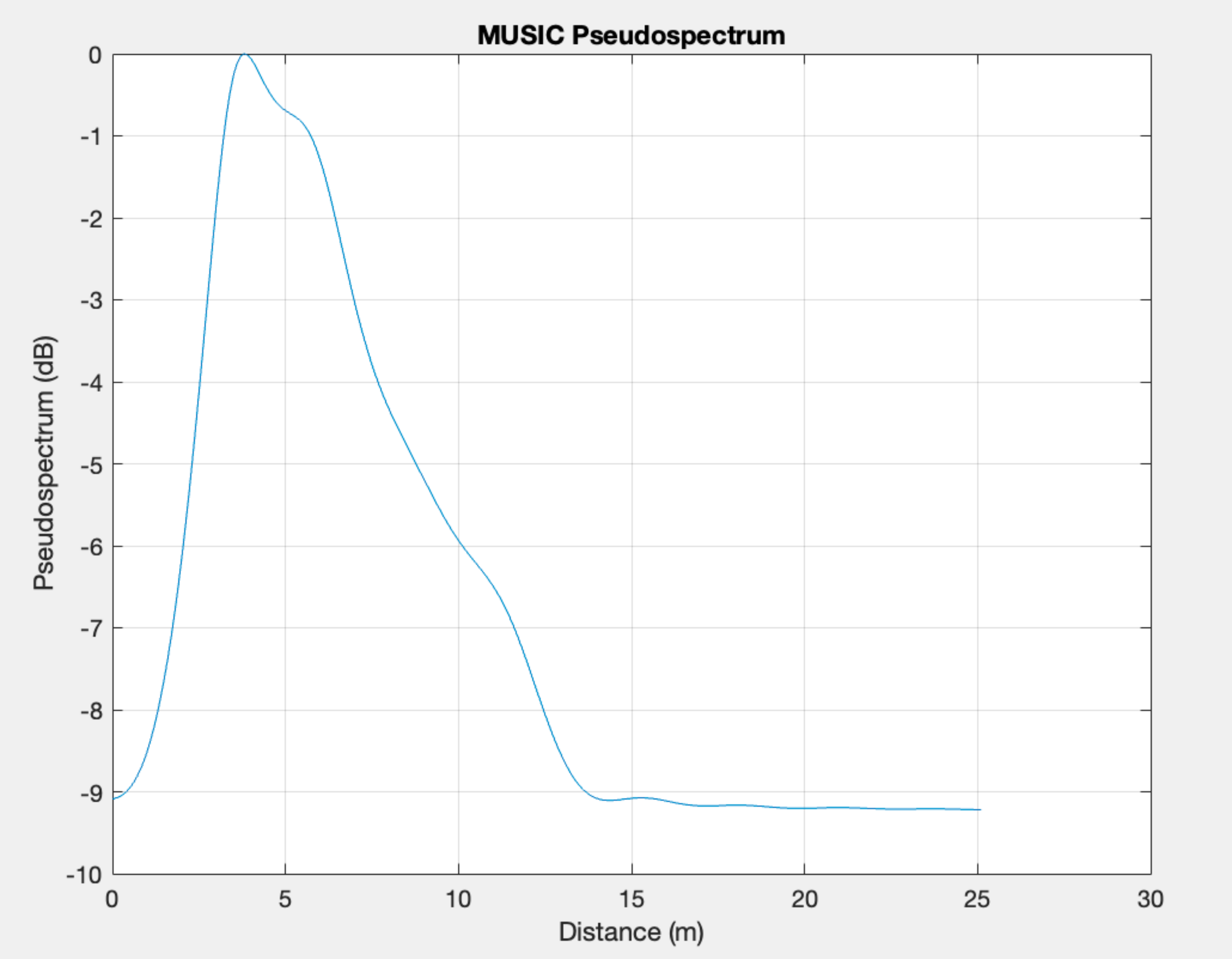


220MHZ with noise



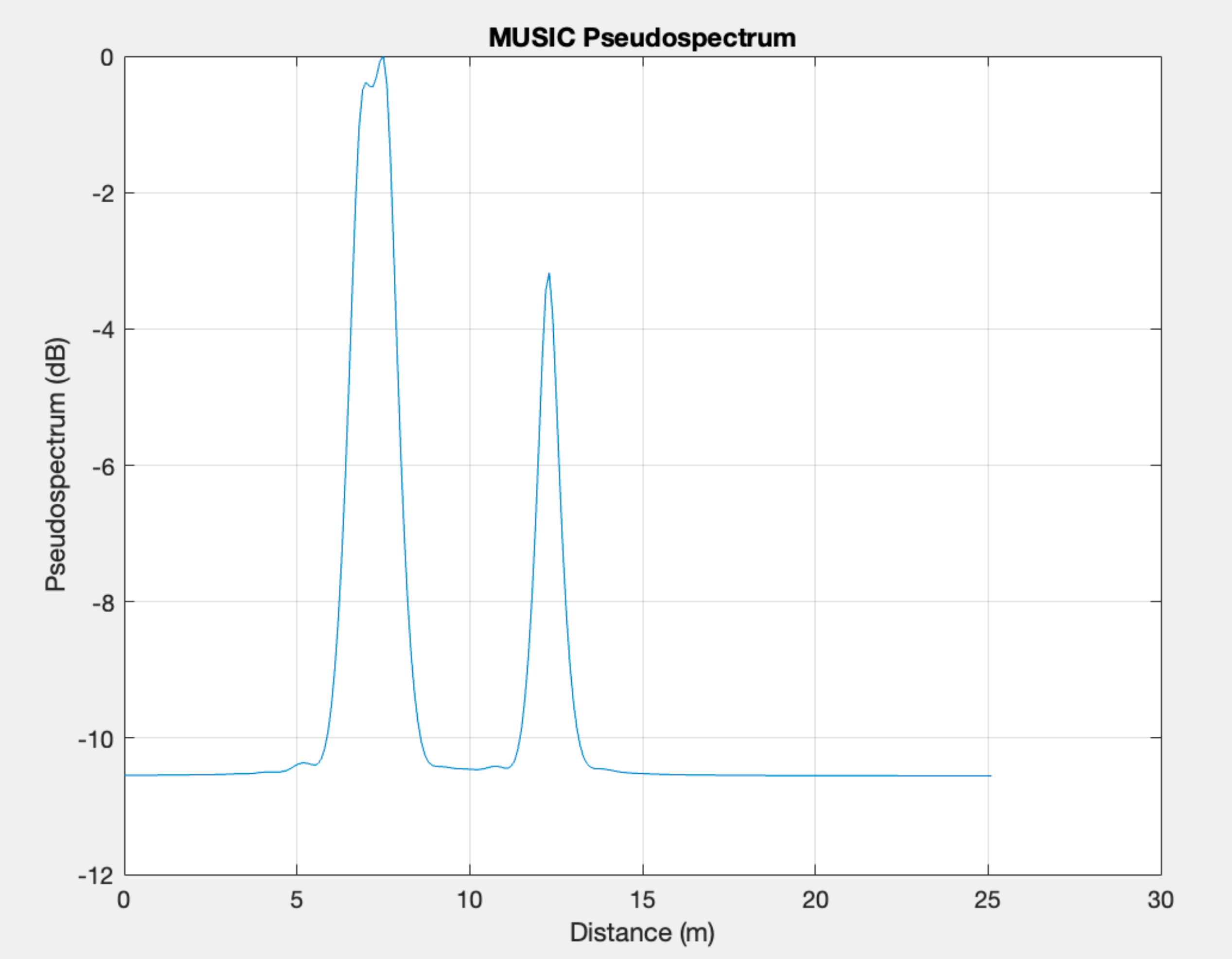
1015 MHZ with noise

Results – Music



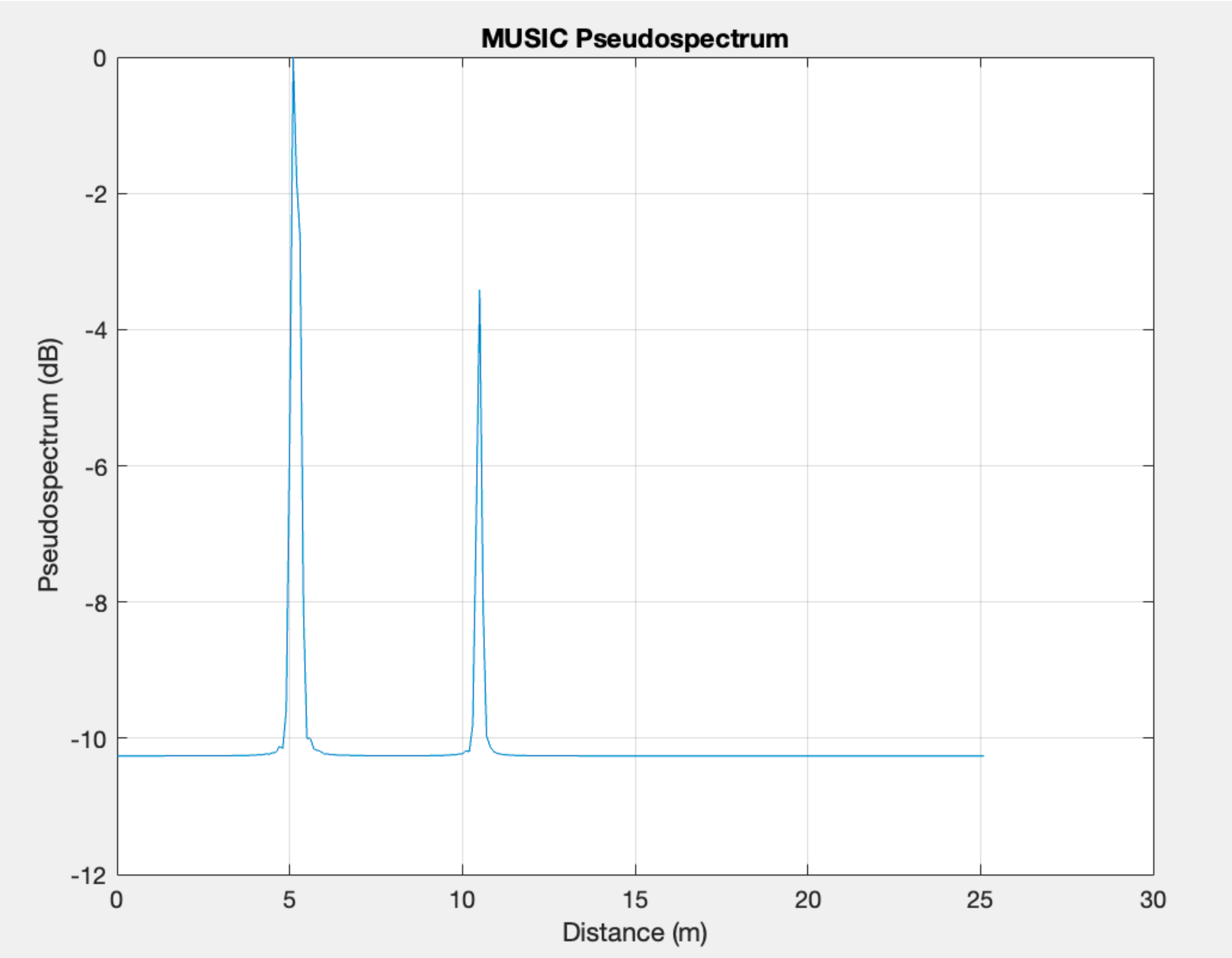

```
Splicing information:
        Original bandwidth = 20 MHz
        New bandwidth = 80 MHz
Selected scenario: 2
        Attenuations: [0 -6]
        Delays: [0 2.5e-08]
        Bandwidths needed for all paths: 40.00 MHz
```

Results – Music



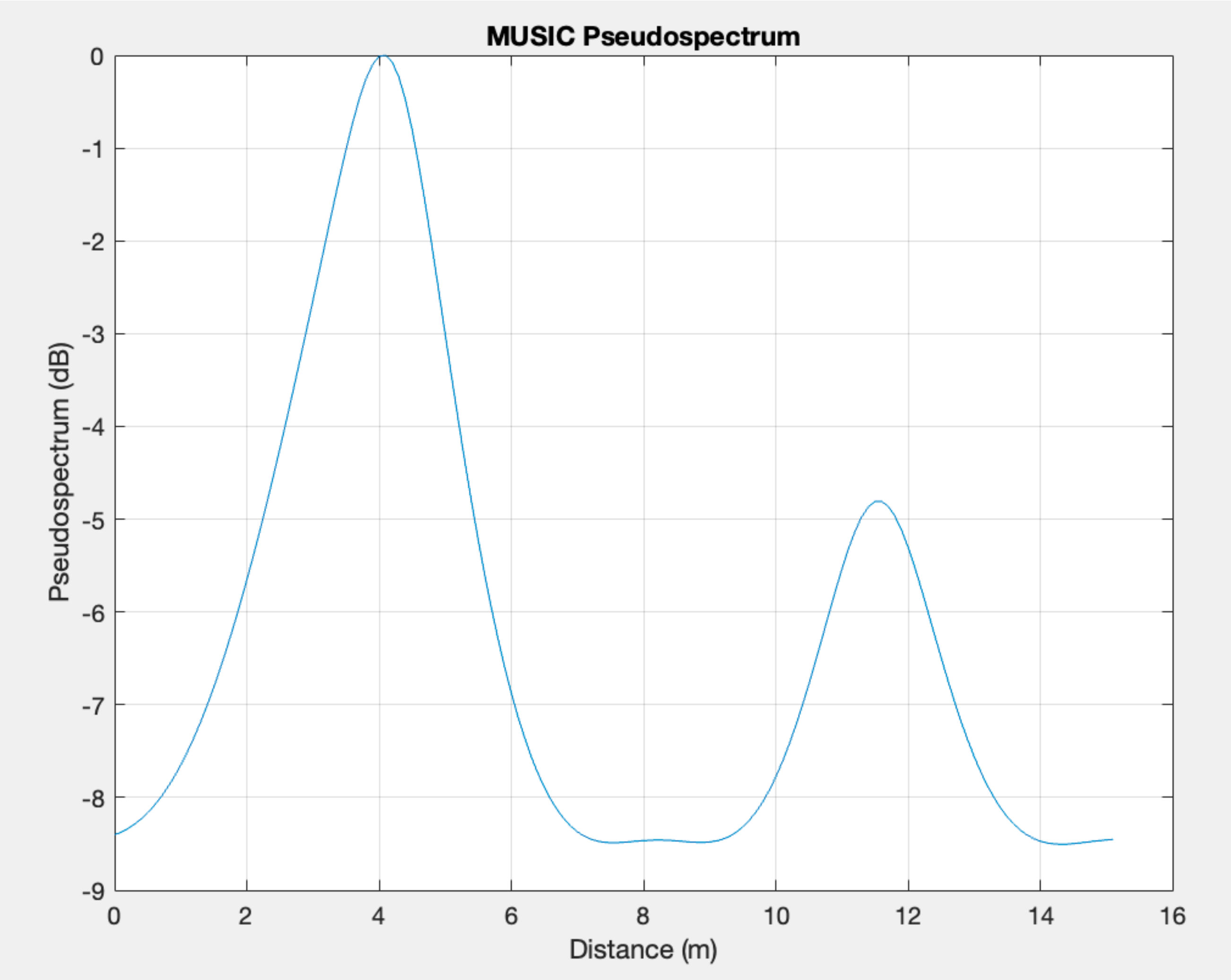

```
Splicing information:
        Original bandwidth = 20 MHz
        New bandwidth = 220 MHz
Selected scenario: 2
        Attenuations: [0 -6]
        Delays: [0 2.5e-08]
        Bandwidths needed for all paths: 40.00 MHz
```

Results – Music



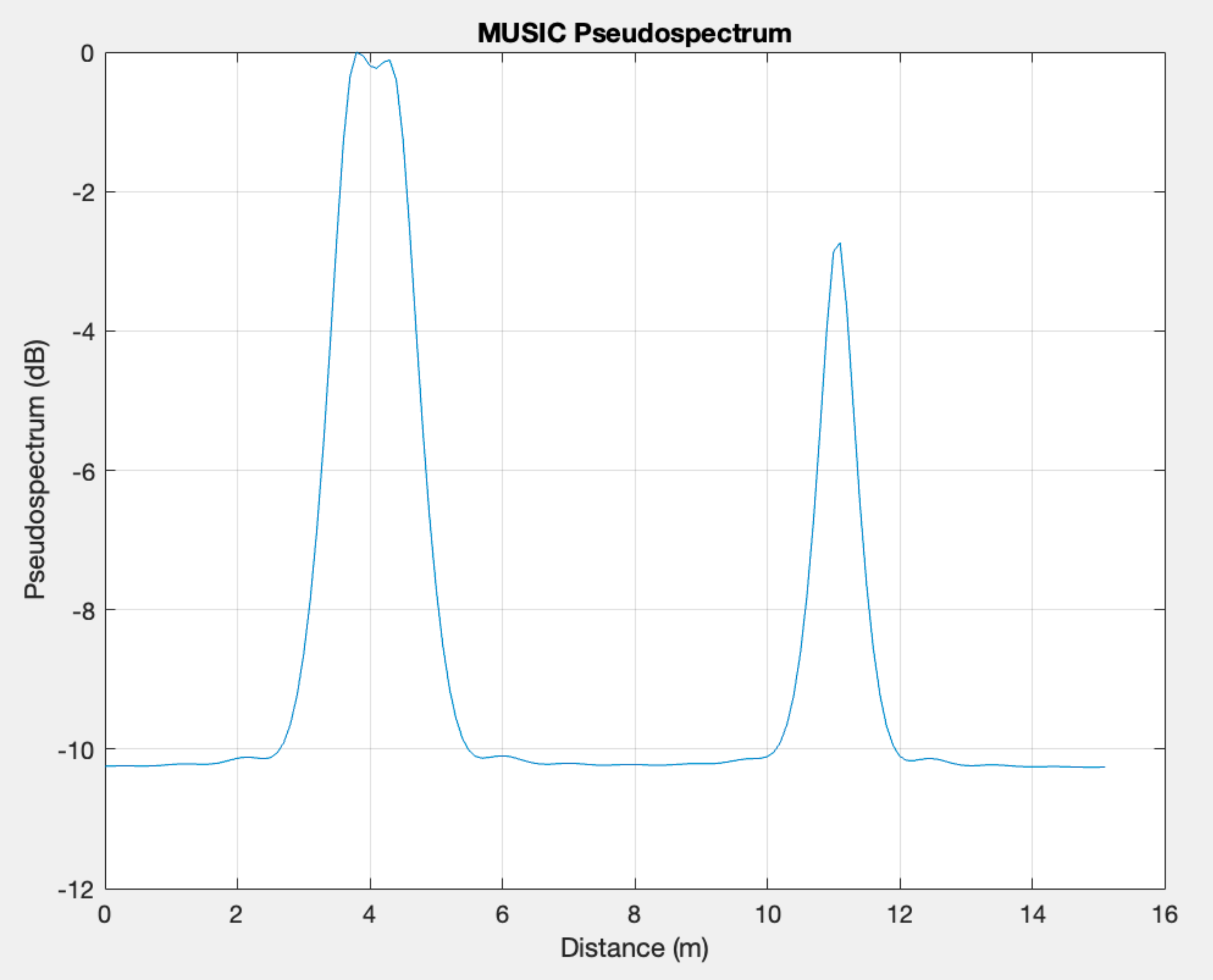

```
Splicing information:
        Original bandwidth = 20 MHz
        New bandwidth = 1015 MHz
Selected scenario: 2
        Attenuations: [0 -6]
        Delays: [0 2.5e-08]
        Bandwidths needed for all paths: 40.00 MHz
```

Results – Music : Real Data on 80 MHZ




```
Impulse Response:
Peak 1: 0.00 dB at 5.25 m
Peak 2: -6.97 dB at 10.50 m
Distance between peaks: 5.25 m

MUSIC Pseudospectrum:
Peak 1: 0.00 dB at 3.40 m
Peak 2: -1.51 dB at 11.30 m
Distance between peaks: 7.90 m
```

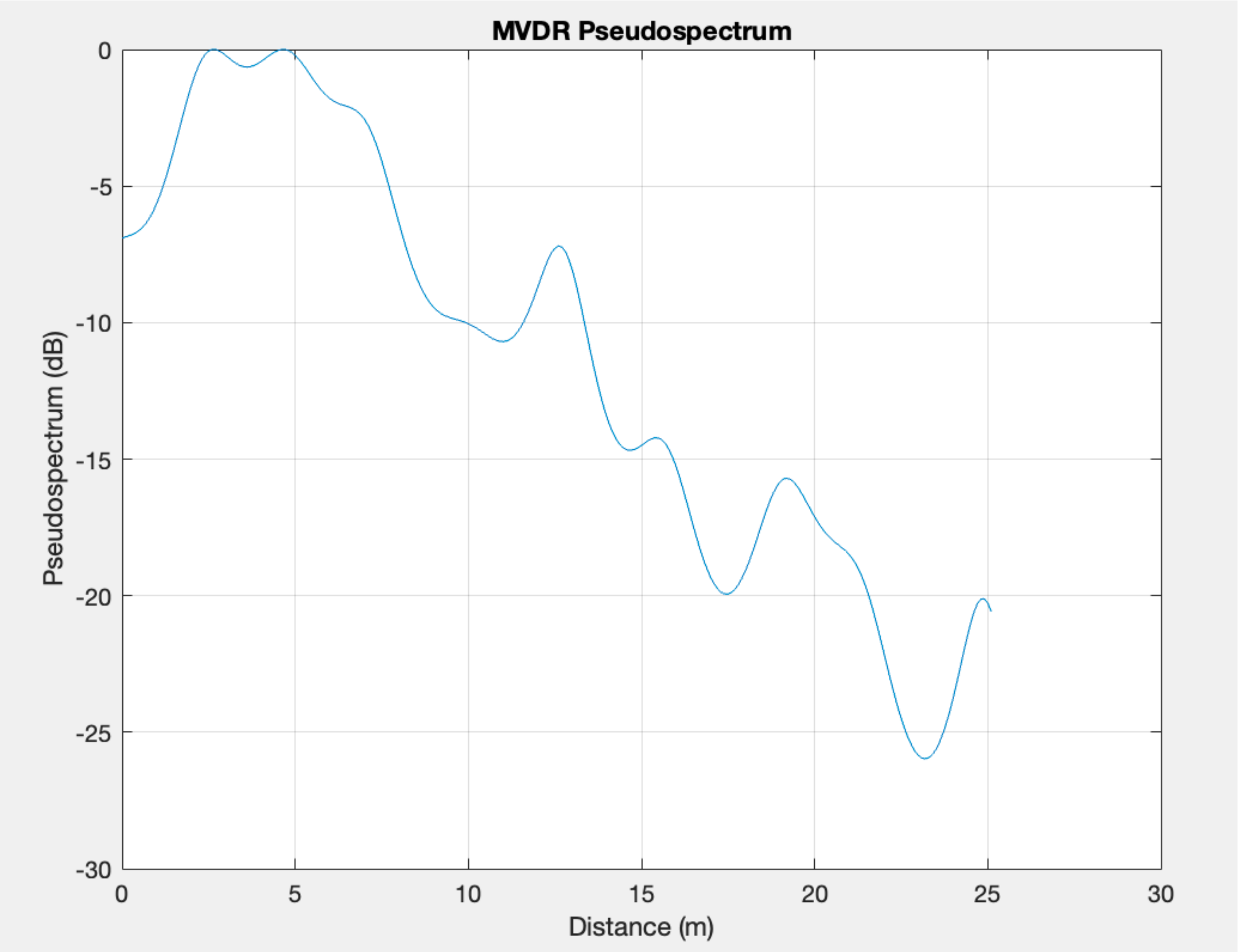
Results – Music : Real Data on 230 MHZ




```
Impulse Response:
Peak 1: 0.00 dB at 3.65 m
Peak 2: -7.98 dB at 10.96 m
Distance between peaks: 7.30 m
```

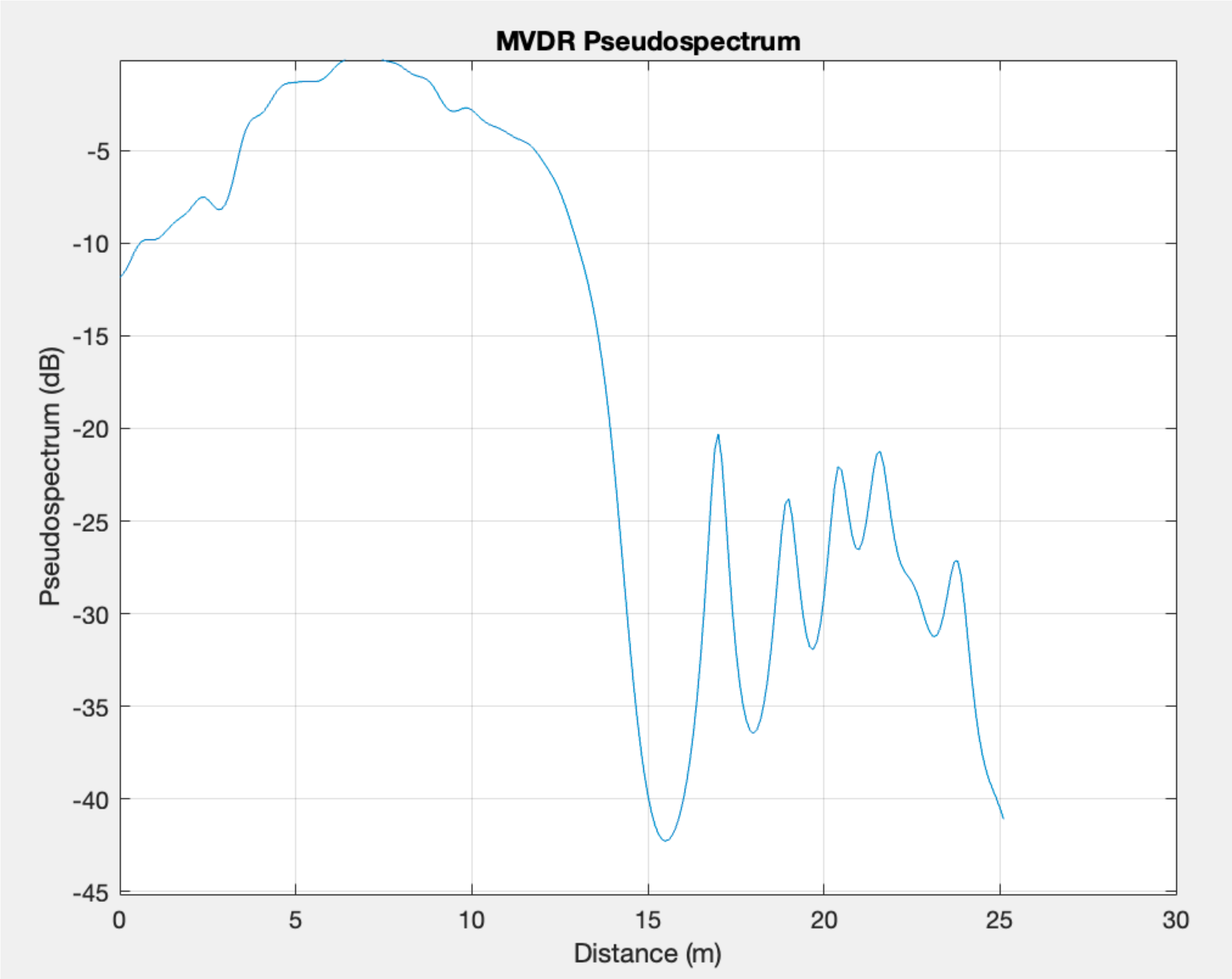
```
MUSIC Pseudospectrum:
Peak 1: 0.00 dB at 3.80 m
Peak 2: -2.73 dB at 11.10 m
Distance between peaks: 7.30 m
```

Results – MVDR



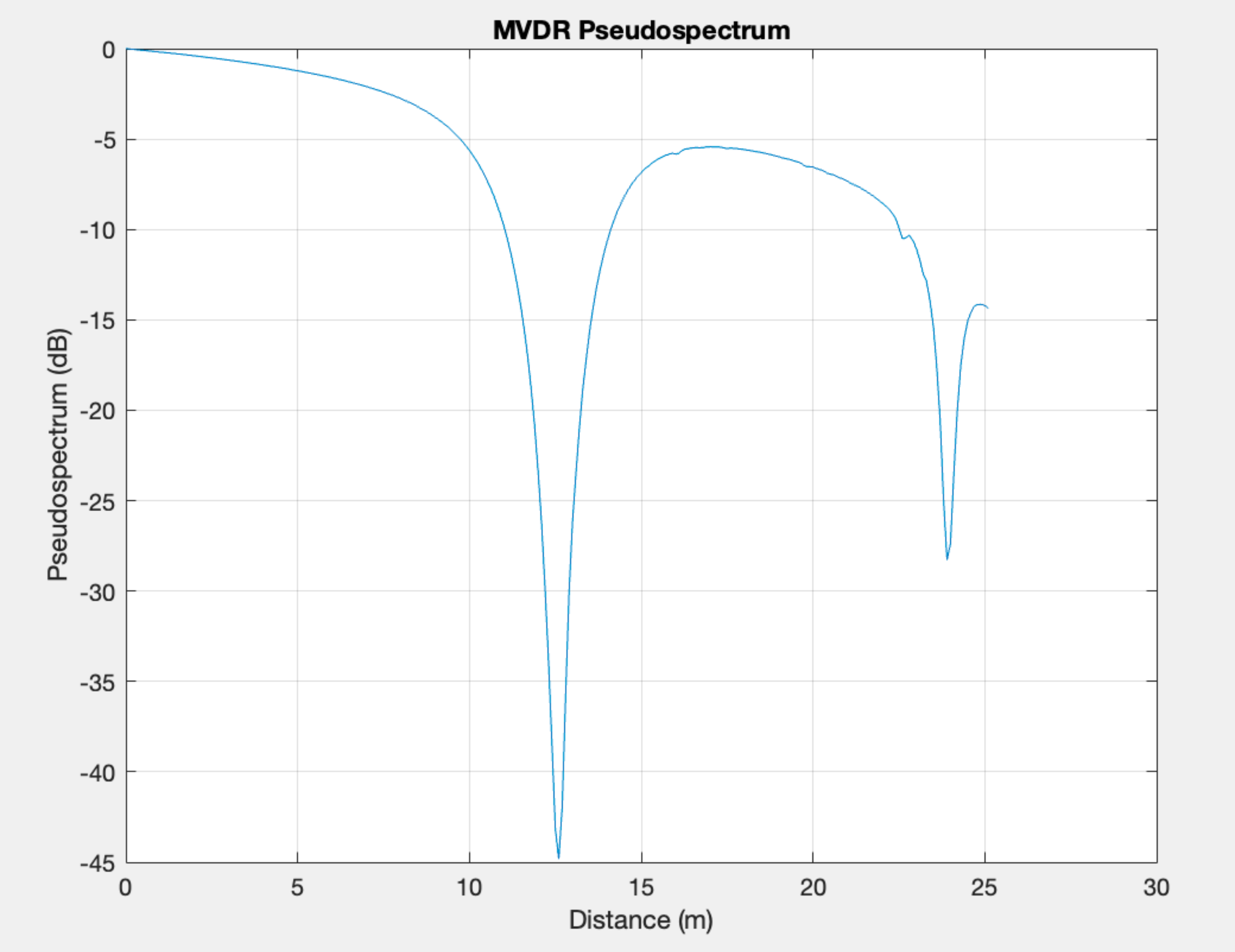

```
Splicing information:
        Original bandwidth = 20 MHz
        New bandwidth = 80 MHz
Selected scenario: 2
        Attenuations: [0 -6]
        Delays: [0 2.5e-08]
        Bandwidths needed for all paths: 40.00 MHz
```

Results – MVDR



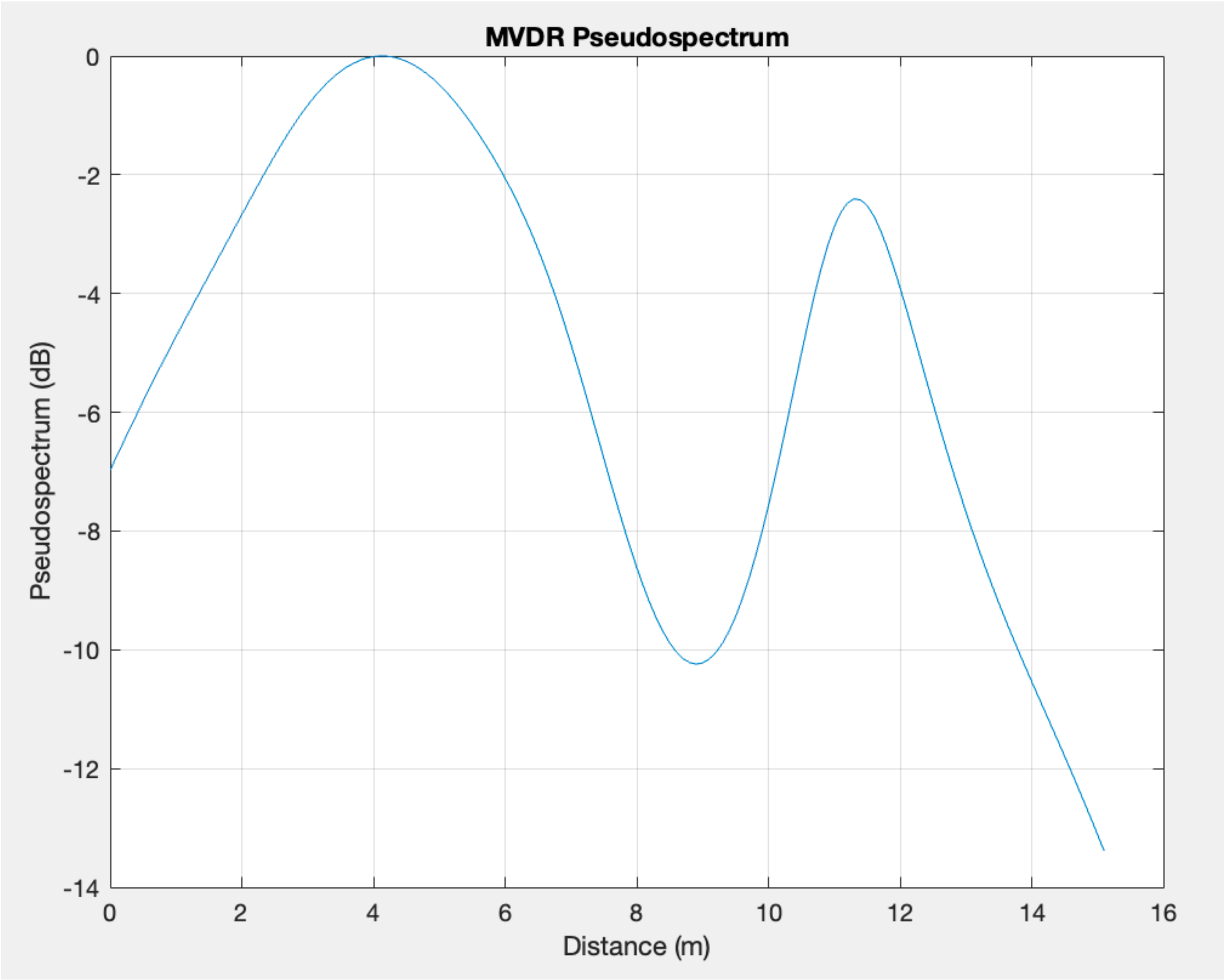

```
Splicing information:
        Original bandwidth = 20 MHz
        New bandwidth = 220 MHz
Selected scenario: 2
        Attenuations: [0 -6]
        Delays: [0 2.5e-08]
        Bandwidths needed for all paths: 40.00 MHz
```

Results – MVDR




```
Splicing information:
        Original bandwidth = 20 MHz
        New bandwidth = 1015 MHz
Selected scenario: 2
        Attenuations: [0 -6]
        Delays: [0 2.5e-08]
        Bandwidths needed for all paths: 40.00 MHz
```

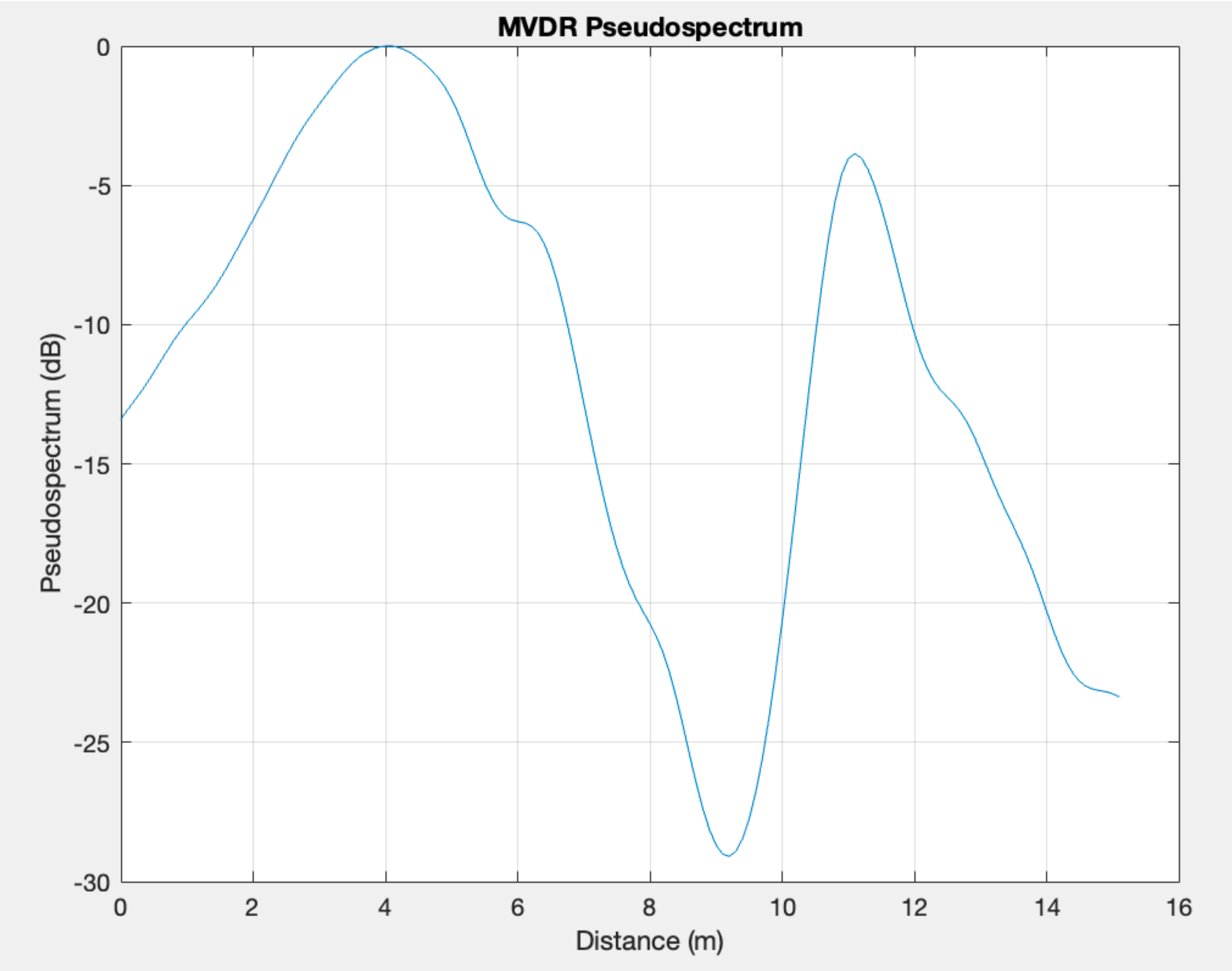
Results – MVDR : Real Data on 80 MHZ




```
Impulse Response:
Peak 1: 0.00 dB at 5.25 m
Peak 2: −6.97 dB at 10.50 m
Distance between peaks: 5.25 m
```

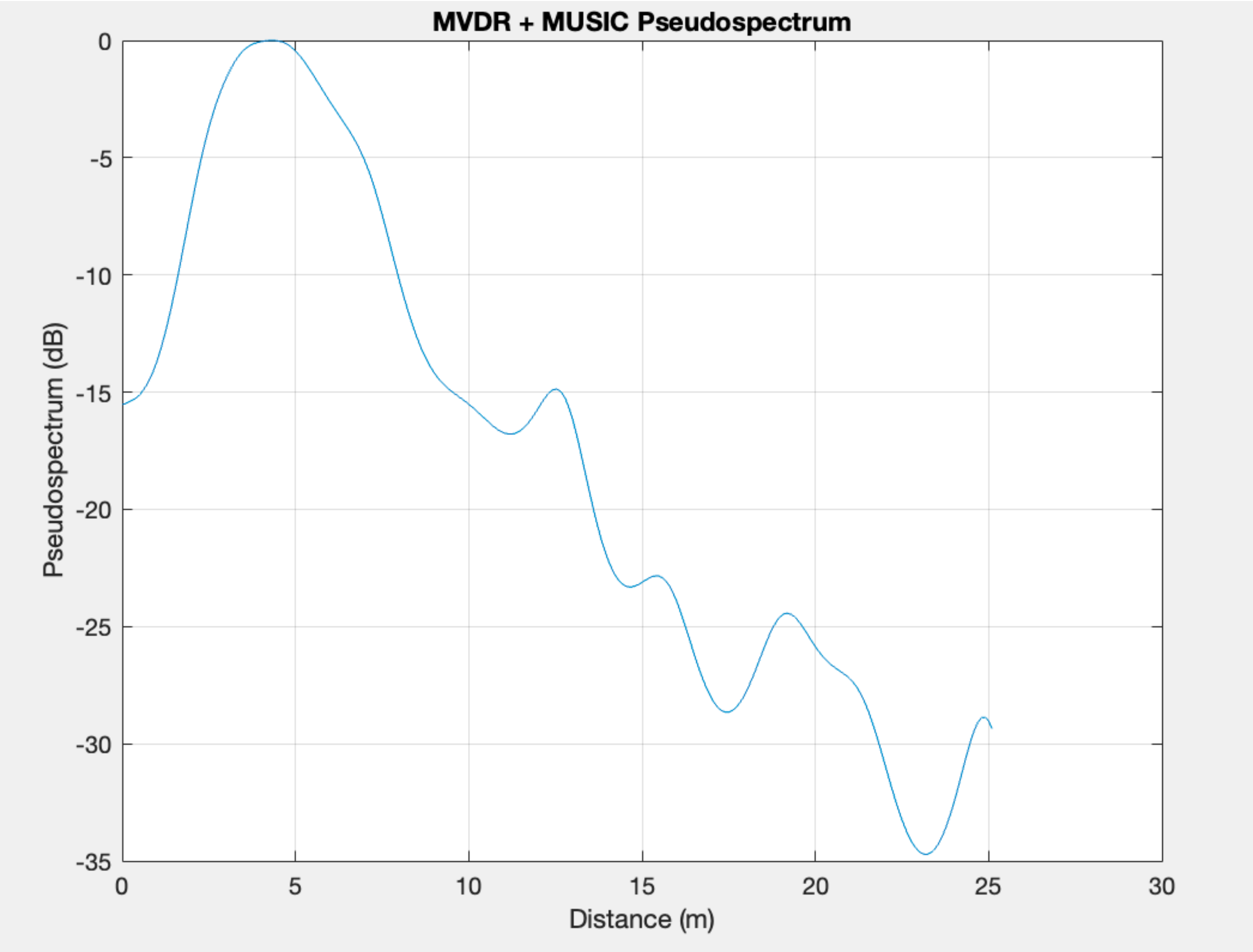
```
MVDR Pseudospectrum:
Peak 1: 0.00 dB at 4.10 m
Peak 2: −2.41 dB at 11.30 m
Distance between peaks: 7.20 m
```

Results – MVDR : Real Data on 230 MHZ



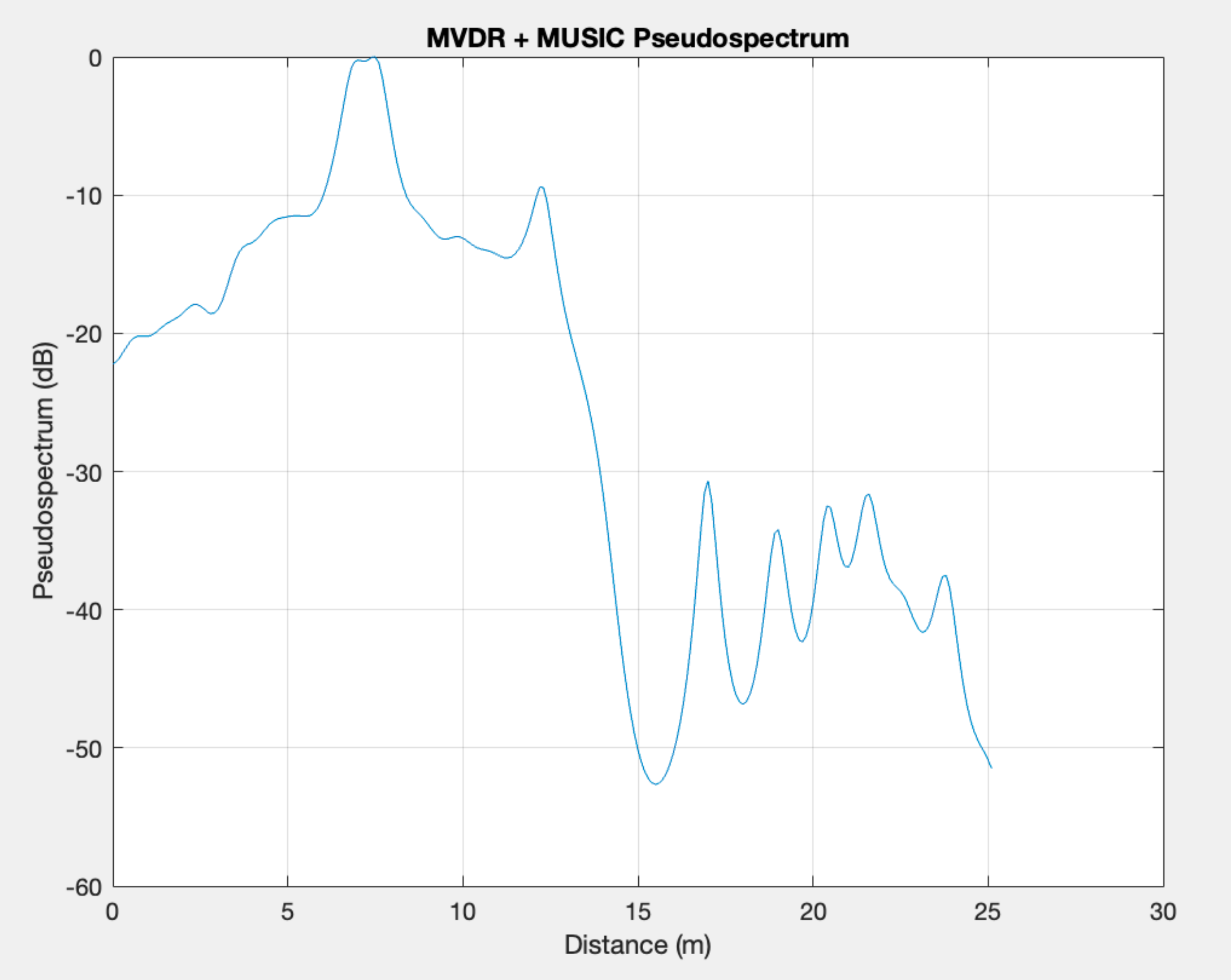

```
Impulse Response:
Peak 1: 0.00 dB at 3.65 m
Peak 2: -7.98 dB at 10.96 m
Distance between peaks: 7.30 m

MVDR Pseudospectrum:
Peak 1: 0.00 dB at 4.00 m
Peak 2: -3.87 dB at 11.10 m
Distance between peaks: 7.10 m
```

Results : Music + MVDR



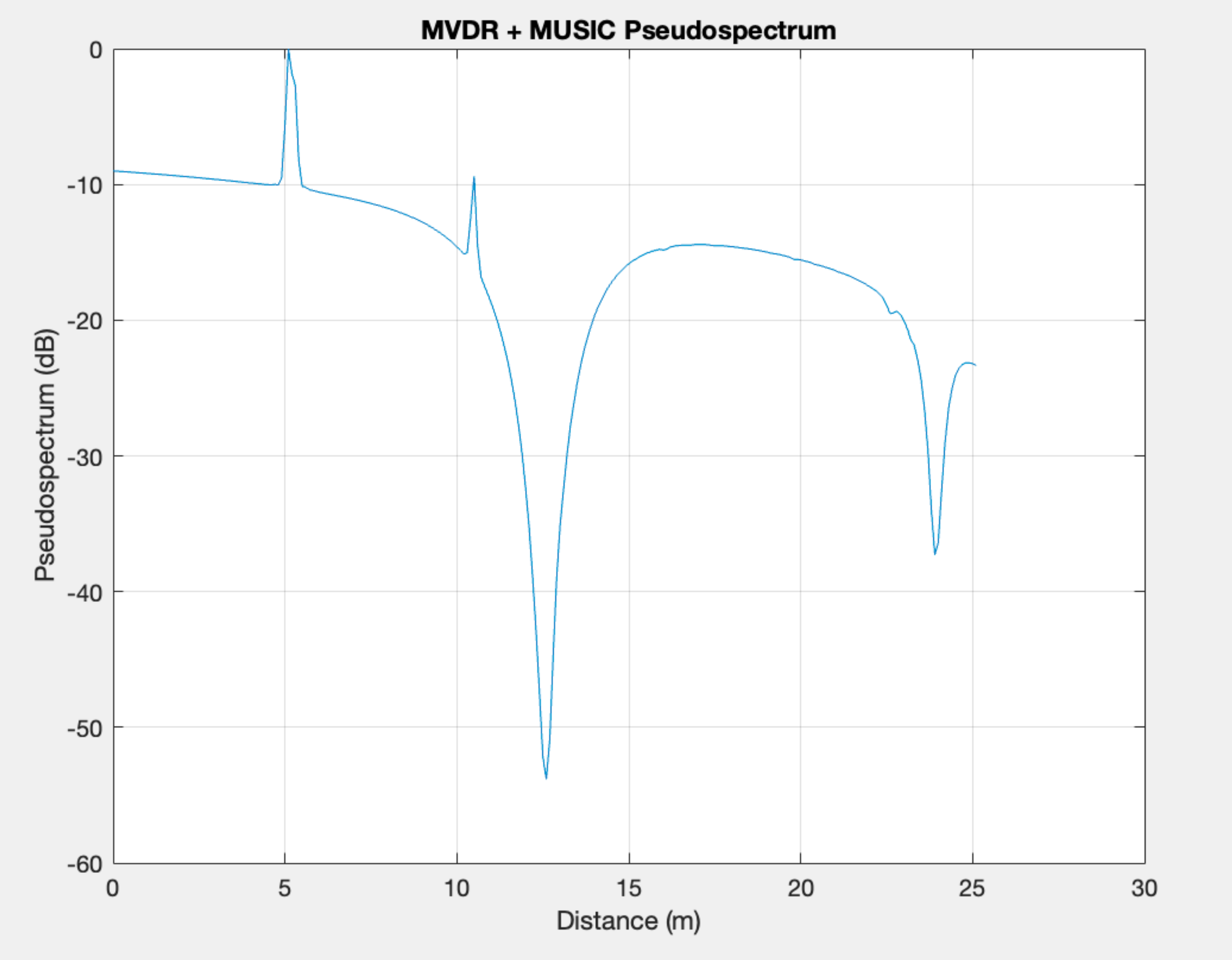

```
Splicing information:
        Original bandwidth = 20 MHz
        New bandwidth = 80 MHz
Selected scenario: 2
        Attenuations: [0 -6]
        Delays: [0 2.5e-08]
        Bandwidths needed for all paths: 40.00 MHz
```

Results : Music + MVDR



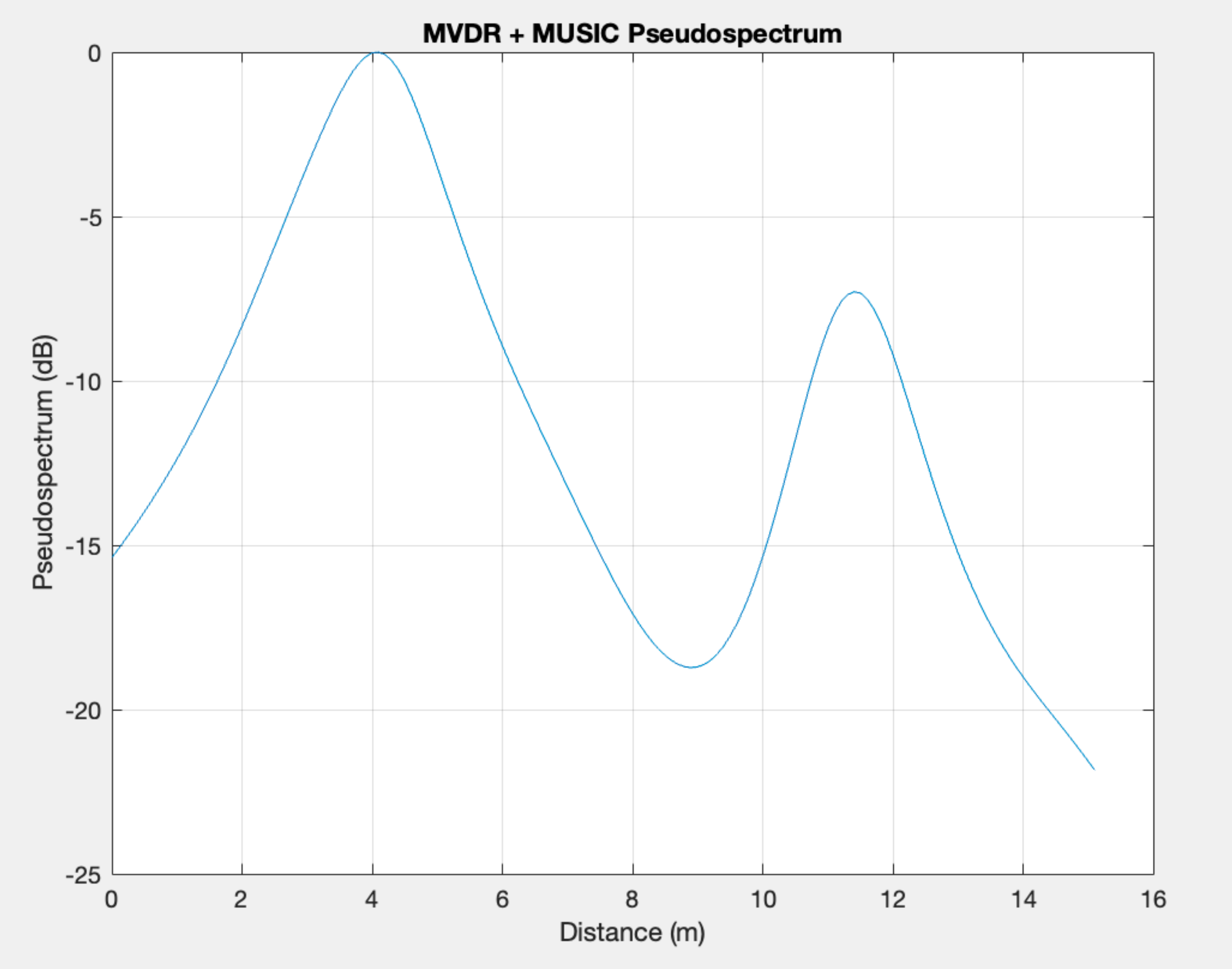

```
Splicing information:
        Original bandwidth = 20 MHz
        New bandwidth = 220 MHz
Selected scenario: 2
        Attenuations: [0 -6]
        Delays: [0 2.5e-08]
        Bandwidths needed for all paths: 40.00 MHz
```

Results : Music + MVDR




```
Splicing information:
	Original bandwidth = 20 MHz
	New bandwidth = 1015 MHz
Selected scenario: 2
	Attenuations: [0 -6]
	Delays: [0 2.5e-08]
	Bandwidths needed for all paths: 40.00 MHz
```

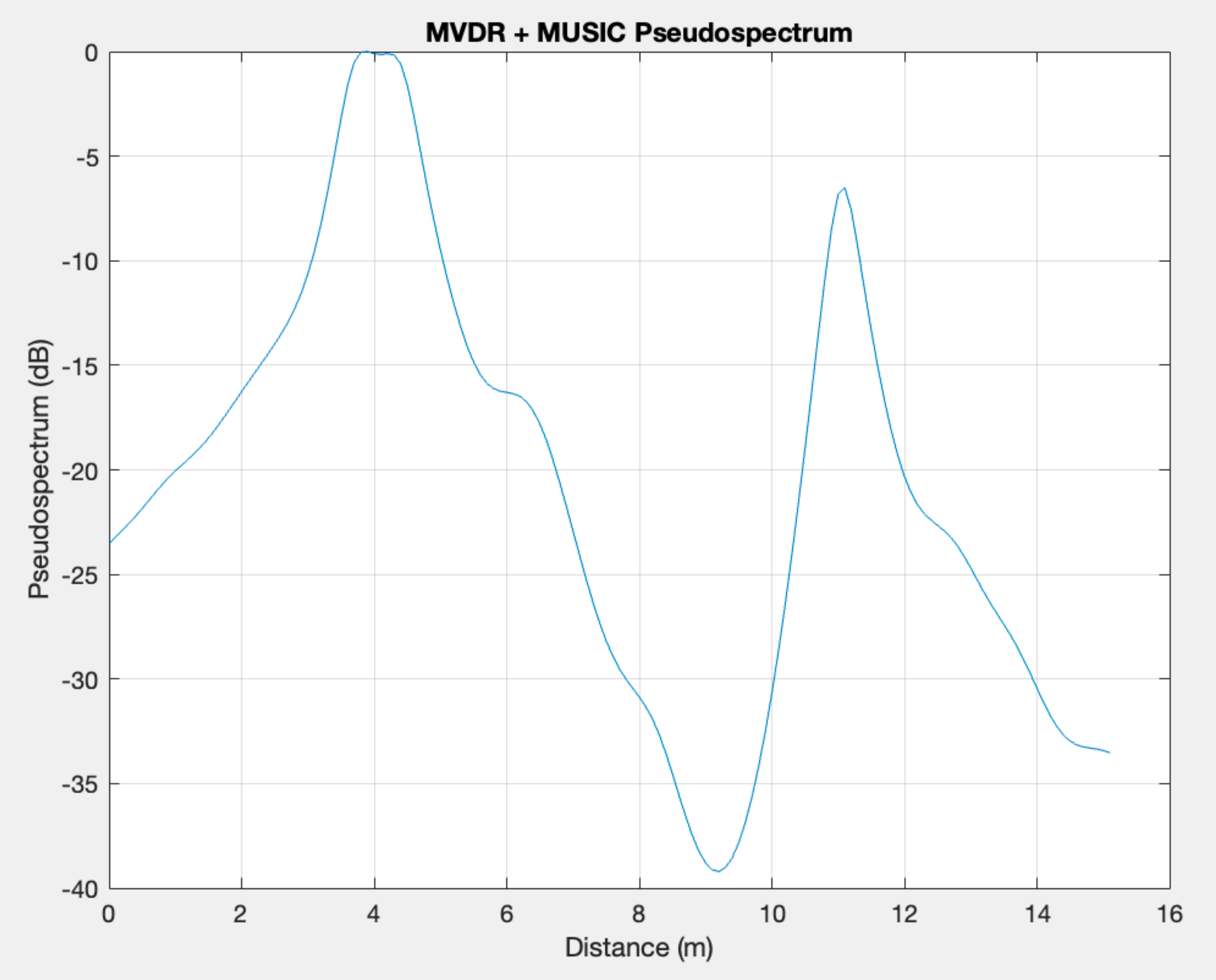
Results : Music + MVDR Real Data on 80 MHZ




```
Impulse Response:
Peak 1: 0.00 dB at 5.25 m
Peak 2: −6.97 dB at 10.50 m
Distance between peaks: 5.25 m
```

```
MVDR + MUSIC Pseudospectrum:
Peak 1: 0.00 dB at 4.50 m
Peak 2: −3.69 dB at 11.30 m
Distance between peaks: 6.80 m
```

Results : Music + MVDR Real Data on 230 MHZ




```
Impulse Response:
Peak 1: 0.00 dB at 3.65 m
Peak 2: -7.98 dB at 10.96 m
Distance between peaks: 7.30 m
```

```
MVDR + MUSIC Pseudospectrum:
Peak 1: 0.00 dB at 3.90 m
Peak 2: -6.50 dB at 11.10 m
Distance between peaks: 7.20 m
```

Future work

- **Extend Testing Scenarios**
  - Introduce moving targets to simulate dynamic environments.
- **Real-world Implementation**
  - Apply methods to live Wi-Fi environments for practical validation.
- **Optimize Computational Efficiency**
  - Develop faster algorithms to reduce processing time