Medium Term Presentation

Sub - Nyquist Mimo Radar

By: Ilan Menirom
Yannay Jaffe
Under the guidance of David Cohen
Motivation

- High Resolution
- Low Sampling Rate

Break the link between resolution and sampling rate
MIMO Radar - Setting

- Each target is defined by 4 parameters: $\alpha, \tau, \nu, \theta$
  - Effective reflectivity - $\alpha$
  - Delay - $\tau$
  - Doppler shift - $\nu$
  - Azimuth - $\theta$
- For location and velocity, one must find: $\tau, \nu, \theta$
- We focus on the delay and the azimuth: $\tau, \theta$
Setting (Continued)

Collocated antenna array

\[ M = \text{number of transmitters} \]

\[ N = \text{number of receivers} \]

Virtual antennas concept

MN phased array antennas

\[ \Rightarrow M+N \text{ MIMO antennas} \]
Transmitted Signal

- Each Tx element transmits modulated square pulse (in each PRI).

- The transmissions are orthogonal.

\[
h_m(t) = h_0(t) \cdot e^{j \frac{2\pi}{T_p} (m - \frac{M-1}{2}) t}
\]

\(h_0(t)\) is a square pulse, with support \([0, T_p]\)

![Graph of h(t) and H0(f)]

\[BW = \frac{1}{T_p}\]
Received Signal

- Assuming $L$ targets, $M$ transmitters.

The received signal in the $n$’th receiver:

$$r_n(t) = \sum_{l=1}^{L} \sum_{m=0}^{M-1} \alpha_l \cdot h_m(t - \tau_l) \cdot e^{j2\pi(p_n + q_m)\sin \theta}$$

As we can see the transmissions are nearly orthogonal.
Conventional Radar Processing

Sample at Nyquist frequency in every receiver

Arrange samples from each receiver in a row

Filter each row in M matched filters

Perform DFT on each column

Arrange all the new rows in a matrix

The result is the delay-DOA map
Conventional Radar Processing - Limitations

- **Delay Resolution** - defined by the MF (or by the signal’s BW) ➔ \( T_p = \frac{1}{BW} \)

- **sin \( \theta \) Resolution** - defined by the DFT length - DFT of length \( MN \) ➔ \( MN \) points scattered in \([-1,1)\) ➔ the resolution is \( \frac{2}{MN} \)

Higher resolution in delay ➔ Larger BW ➔ Higher sampling frequency

Higher resolution in the azimuth ➔ Larger \( M,N \) ➔ More antennas
Compressed Sensing Radar Processing

- Use \textbf{xampling} instead of regular time domain sampling.

\textbf{Xample} = Fourier series coefficient.

**Intuition**: each xample contains information of the entire PRI, so with just a “few” xamples we can obtain the wanted information.

- Starting from the received signal in the n’th Rx element, calculate Fourier coefficients:

\[
c_n[k] = \frac{1}{T} \cdot \sum_{l=1}^{L} \sum_{m=0}^{M-1} \alpha_l \cdot e^{j2\pi(p_n+q_m)\sin(\theta_l)} \cdot H_0 \left( \frac{2\pi k}{T} - \frac{2\pi}{T_p} \left( m - \frac{M - 1}{2} \right) \right) e^{-j\frac{2\pi T}{T_p} k}
\]

\[k = -\frac{N' M}{2}, \ldots, \frac{N' M}{2} - 1\] , \(N'\) is \(\frac{T}{T_p}\)
Since the **transmissions are orthogonal**, each Tx element **contributes different k’s** in the Fourier series.

**Rearrange** $c_n[k]$ to a matrix $R_n[k,m]$ containing $M$ columns and $N'$ rows.
Compressed Sensing Radar Processing (Cont.)

- Create new matrix \( \mathbf{D} = [R_0, R_1, R_2, \ldots, R_{N-1}] \):

\[
D[k, m] = \frac{1}{T} \sum_{l=1}^{L} \alpha_l e^{-j\frac{2\pi l N'}{T}(m \mod M-M/2)} \cdot e^{j2\pi(p_n+q_m) \sin(\theta_l)} \cdot H_0 \left( \omega = \frac{2\pi k}{T} - \frac{\pi}{T_p} \right) e^{-j\frac{2\pi \theta_m}{T} k}
\]

- \( p_n \) and \( q_m \) represents the Rx and Tx array elements contribution to the phase, we can replace them with an equivalent vector \( s \).

- For a uniform virtual Rx array \( s = \frac{m + Mn}{2} \) \((s = 0, \frac{1}{2}, \ldots, \frac{MN-1}{2} \rightarrow \text{Nyquist rate})\)
Compressed Sensing Radar Processing (Cont.)

\[ D = M_1 \cdot M_2 \cdot M_3 \cdot M_4 = \frac{1}{T} \cdot \sum_{l=1}^{L} \alpha_l e^{-j\frac{2\pi l}{T}N'(m \mod M)-M/2} \cdot e^{j2\pi(p_n+q_m)\sin(\theta_l)} \cdot H_0 \left( \omega = \frac{2\pi k}{T} - \frac{\pi}{T_p} \right) e^{-j\frac{2\pi l}{T}k} \]

\[
M_{1_{(N'\times N')}} = \frac{1}{T} \begin{pmatrix}
H_0 \left( \frac{2\pi k_0}{T} - \frac{\pi}{T_p} \right) & 0 & 0 & 0 \\
0 & H_0 \left( \frac{2\pi k_1}{T} - \frac{\pi}{T_p} \right) & 0 & 0 \\
0 & 0 & \ddots & 0 \\
0 & 0 & \cdots & H_0 \left( \frac{2\pi k_{N'-1}}{T} - \frac{\pi}{T_p} \right)
\end{pmatrix}
\]

\[
M_{2_{(N'\times N')}} = \begin{pmatrix}
e^{-j\frac{2\pi l_0 k_0}{T}} & e^{-j\frac{2\pi l_1 k_0}{T}} & \cdots & e^{-j\frac{2\pi l_{N'-1} k_0}{T}} \\
e^{-j\frac{2\pi l_0 k_1}{T}} & e^{-j\frac{2\pi l_1 k_1}{T}} & \cdots & e^{-j\frac{2\pi l_{N'-1} k_1}{T}} \\
\vdots & \vdots & \ddots & \vdots \\
e^{-j\frac{2\pi l_0 k_{N'-1}}{T}} & e^{-j\frac{2\pi l_1 k_{N'-1}}{T}} & \cdots & e^{-j\frac{2\pi l_{N'-1} k_{N'-1}}{T}}
\end{pmatrix}
\]

\[
M_{3_{(N\times MN)}} = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & \alpha_l \cdot e^{-j\frac{2\pi l}{T}N'[m \mod M]-M/2} & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
\]

\[
M_{4_{(MN\times MN)}} = \begin{pmatrix}
e^{j2\pi s_0 \sin(\theta_{l_0})} & e^{j2\pi s_1 \sin(\theta_{l_0})} & \cdots & e^{j2\pi s_{MN-1} \sin(\theta_{l_0})} \\
e^{j2\pi s_0 \sin(\theta_{l_1})} & e^{j2\pi s_1 \sin(\theta_{l_1})} & \cdots & e^{j2\pi s_{MN-1} \sin(\theta_{l_1})} \\
\vdots & \vdots & \ddots & \vdots \\
e^{j2\pi s_0 \sin(\theta_{l_{MN-1}})} & e^{j2\pi s_1 \sin(\theta_{l_{MN-1}})} & \cdots & e^{j2\pi s_{MN-1} \sin(\theta_{l_{MN-1}})}
\end{pmatrix}
\]
Compressed Sensing Radar Processing

Properties

**Decimation**

Delay resolution is defined by the **columns** of \( M_2 \)

DOA resolution is defined by the **rows** of \( M_4 \)

\[ M_1^{-1} \cdot D = M_2 \cdot M_3 \cdot M_4 \]

\[ \rightarrow \text{No physical resolution limitation} \] (resolution is now limited due to solution algorithms).

We can keep only small group of \( k \) values and \( s \) values

\[ \rightarrow \text{decimation in time} (k - \text{less samples per antenna}) \]

\[ \rightarrow \text{decimation in space} (s - \text{less antennas}) \]

\[ \rightarrow \text{overall less samples per second} \]

**CS Solution Algorithms:**

For decimated matrices we solve linear equations for sparse matrix using methods such as **OMP** and **FISTA**.
Simulation Results
Simulation Results

SNR Definition:

$$\text{SNR} = \frac{P_{\text{sig, total}}}{N_0 \cdot BW}$$

This is a strict definition for SNR.

Examples:

- SNR = $-10 \text{dB}$
- No noise

Graphs showing received signal FFT and pulse frequency in the receiver.
Simulation Results

Example 1:
Simulation Results

- 50% Kappa Decimation
- 50% Antennas

\[ 2.5 \cdot 10^6 \frac{\text{Samples}}{\text{Pulse}} \rightarrow 0.3125 \cdot 10^6 \frac{\text{Samples}}{\text{Pulse}} \]

→ 12.5% of the Nyquist rate Samples!

![Graph showing Received Signals vs. Time with SNR of -10dB](image)
Simulation Results

Results & Original Targets Locations by Algorithm Type: OMP
SNR is: -10dB

Did it work???

Success!
Example 2:

Simulation Results

Original Targets Locations

Radius [Km]
Recieved Signals vs. Time
SNR is: -15dB

<table>
<thead>
<tr>
<th>Time [sec]</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.1</td>
<td>10</td>
</tr>
<tr>
<td>0.2</td>
<td>20</td>
</tr>
<tr>
<td>0.3</td>
<td>30</td>
</tr>
<tr>
<td>0.4</td>
<td>40</td>
</tr>
<tr>
<td>0.5</td>
<td>50</td>
</tr>
<tr>
<td>0.6</td>
<td>60</td>
</tr>
<tr>
<td>0.7</td>
<td>70</td>
</tr>
<tr>
<td>0.8</td>
<td>80</td>
</tr>
<tr>
<td>0.9</td>
<td>90</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>

- Noisy Signal
- Clean Signal
Simulation Results

Results & Original Targets Locations by Algorithm Type: OMP
SNR is: -15dB

Did it work???

70% Success...

Radius [Km]

- Estimated Targets
- Original Targets
Detection vs. SNR

- 50% Kappa decimation
  (1000 → 500)
- 50% antennas
  (50 Tx, 50 Rx → 25 Tx, 25 Rx)
Detection Vs. Decimation

- Decimation here is decimation in kappa and in total antennas
- SNR = -15 dB

Example: for 65% decimation, the samples decimation is 25.35%
What's next?

- We noticed that using square pulse isn’t optimal - the side lobes are damaging the detection → we want to try using other pulses such as “sharp” LPF.

- We would like to compare the performance of the conventional method to the performance of the CS method, and decide which method is the best under different conditions (such as SNR, computation abilities, etc...).

- So far we used only OMP algorithm for the discovery of the sparse matrix $M_3$, perhaps other algorithm, such as FISTA will work better.

- So far we found for each target it’s range and azimuth, we would like to expand the model so we can find the velocity.
Questions?