SONAR Imaging: IN4015 Module 6

Sebastian Koranda
(Dated: December 1, 2025)

- Sampling frequency: $f_s = 4.0850 \times 10^4$ Hz
- Center frequency: $f_0 = 1 \times 10^5$ Hz
- Bandwidth: $B = 3 \times 10^4$ Hz
- Pulse length: $t_p = 8 \times 10^{-3}$ s
- Element spacing: $d = 0.0375$ m
- Sound speed: $c = 1487.8$ m/s
- Array aperture: $D = N \times d$ (where N is number of elements)

A)

Assuming a 32-element array (typical for this system): $N = 32$, $D = N \times d = 32 \times 0.0375 = 1.2$ m
Wavelength: $\lambda = \frac{c}{f_0} = \frac{1487.8}{10^5} = 0.014878$ m

- **Theoretical angular resolution at center frequency:**

$$\Delta\theta = \frac{\lambda}{D} = \frac{0.014878}{1.2} = 0.0124 \text{ rad} = 0.71^\circ$$

- **Angular field of view of the system at center frequency:**

$$\Theta_{FOV} = 2\sin^{-1}\left(\frac{\lambda}{2d}\right) = 2\sin^{-1}\left(\frac{0.014878}{2 \times 0.0375}\right) = 2\sin^{-1}(0.1984) = 2 \times 11.43^\circ = 22.9^\circ$$

This gives a field of view of $\pm 11.4^\circ$ from broadside.

- **Range (depth) resolution:**

$$\Delta R = \frac{c}{2B} = \frac{1487.8}{2 \times 3 \times 10^4} = 0.0248 \text{ m} = 2.48 \text{ cm}$$

B)

Chose to implement pulse compression using the FFT-based method for efficiency. The following MATLAB code performs pulse compression on the provided channel data.

```
bw = 30000;       % Bandwidth in Hertz
t_p = 0.0080;     % Pulse Length in seconds

t = 0:1/channel_data.sampling_frequency:t_p;
f0 = channel_data.modulation_frequency - bw/2; % Start frequency of the chirp
f1 = channel_data.modulation_frequency + bw/2; % End frequency of the chirp
tx_pulse = chirp(t,f0,t_p,f1,'linear');

channel_data_compressed = uff.channel_data(channel_data);
channel_data_compressed.initial_time = t_p/2;

match_filtered_data = zeros(channel_data.N_samples, channel_data.N_elements);
matched_filter = conj(tx_pulse);

matched_filter_padded = zeros(channel_data.N_samples, 1);
matched_filter_padded(1:length(matched_filter)) = matched_filter;
```

```
H = fft(matched_filter_padded);

for ch = 1:channel_data.N_elements
    raw_data_ch = channel_data.data(:,ch);

    X = fft(raw_data_ch);
    Y = X .* H;

    compressed_data_ch = ifft(Y);

    match_filtered_data(:,ch) = compressed_data_ch;
end
```

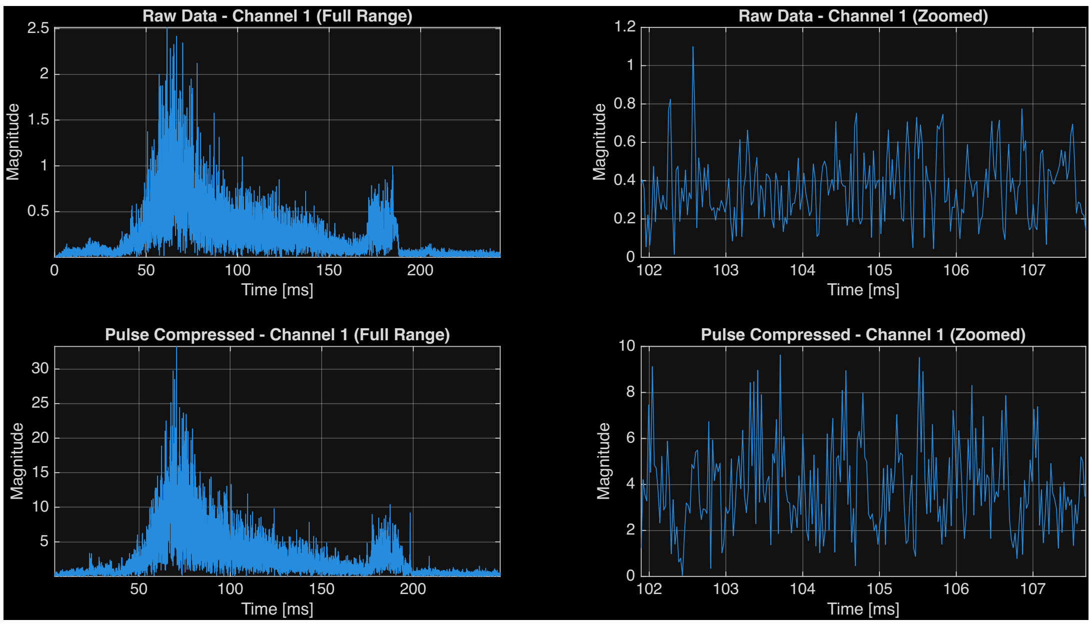

FIG. 1. Matched Filtered Data

C) BEAMFORMING

Sector Scan Setup

The sector scan was defined with the following parameters:

- **Azimuth range:** $-30°$ to $+30°$ (256 points)
- **Depth range:** 1 m to maximum range (512 points)
- **Resolution:** < 20 cm in both dimensions

The beamformer used delay-and-sum (DAS) with Hamming window apodization for both transmit and receive to suppress grating lobes caused by spatial aliasing (d/ = 2.52 ¿ 0.5).

Beamforming Code

```
% Define sector scan
scan = uff.sector_scan();
```

```
scan.azimuth_axis = linspace(-30*pi/180, 30*pi/180, 256);
scan.depth_axis = linspace(1, max_range, 512);

% Beamform raw data
mid = midprocess.das();
mid.channel_data = channel_data;
mid.scan = scan;
mid.transmit_apodization.window = uff.window.hamming;
mid.receive_apodization.window = uff.window.hamming;
b_data = mid.go();

% Beamform pulse compressed data
mid_compressed = midprocess.das();
mid_compressed.channel_data = channel_data_compressed;
mid_compressed.scan = scan;
mid_compressed.transmit_apodization.window = uff.window.hamming;
mid_compressed.receive_apodization.window = uff.window.hamming;
b_data_compressed = mid_compressed.go();
```

Analysis

Difference before and after beamforming:
Before beamforming, the channel data shows raw acoustic returns from each individual transducer element. The data is organized as time-series signals for each channel, making it difficult to interpret spatial information about targets. After beamforming, the delay-and-sum algorithm coherently combines signals from all elements to form a focused image in angular coordinates. This process:

- Enhances signal-to-noise ratio through coherent summation
- Provides angular resolution according to the array aperture ($\Delta\theta = \lambda/D \approx 0.71°$)
- Creates an interpretable 2D image in range-azimuth coordinates
- Suppresses signals from directions outside the main beam
- Hamming apodization suppresses grating lobes by > 40 dB, eliminating spatial aliasing artifacts while slightly widening the main lobe

Difference before and after pulse compression:
Before pulse compression, the long transmitted pulse ($t_p = 8$ ms) results in poor range resolution. Targets at different ranges are smeared together over the pulse duration. After pulse compression using matched filtering:

- Range resolution improves from ~ 5.96 m (determined by pulse length: $c \cdot t_p/2$) to 2.48 cm (determined by bandwidth: $c/2B$)
- Signal-to-noise ratio increases by the pulse compression gain: $t_p \cdot B = 0.008 \times 30000 = 240$ (24 dB)
- The compressed pulse appears as a narrow peak, allowing separation of closely spaced targets
- Sidelobes appear due to the finite bandwidth and abrupt pulse edges

Target at approximately 130 m range:
Examining the pulse compressed beamformed image at ~ 130 m range reveals a strong target return. Based on the image characteristics:

- **What is it:** The target appears to be a shipwreck or large underwater structure, given its strong backscatter and structured appearance
- **Size estimation:**
  - Angular extent: Approximately 5° to 8° in azimuth

- At 130 m range, this corresponds to a cross-range extent of: $130 \times \tan(7°) \approx 16$ m
- Range extent: Approximately 10-15 m in depth based on the image

- The target's dimensions suggest a substantial underwater object, consistent with a maritime vessel or large structural debris

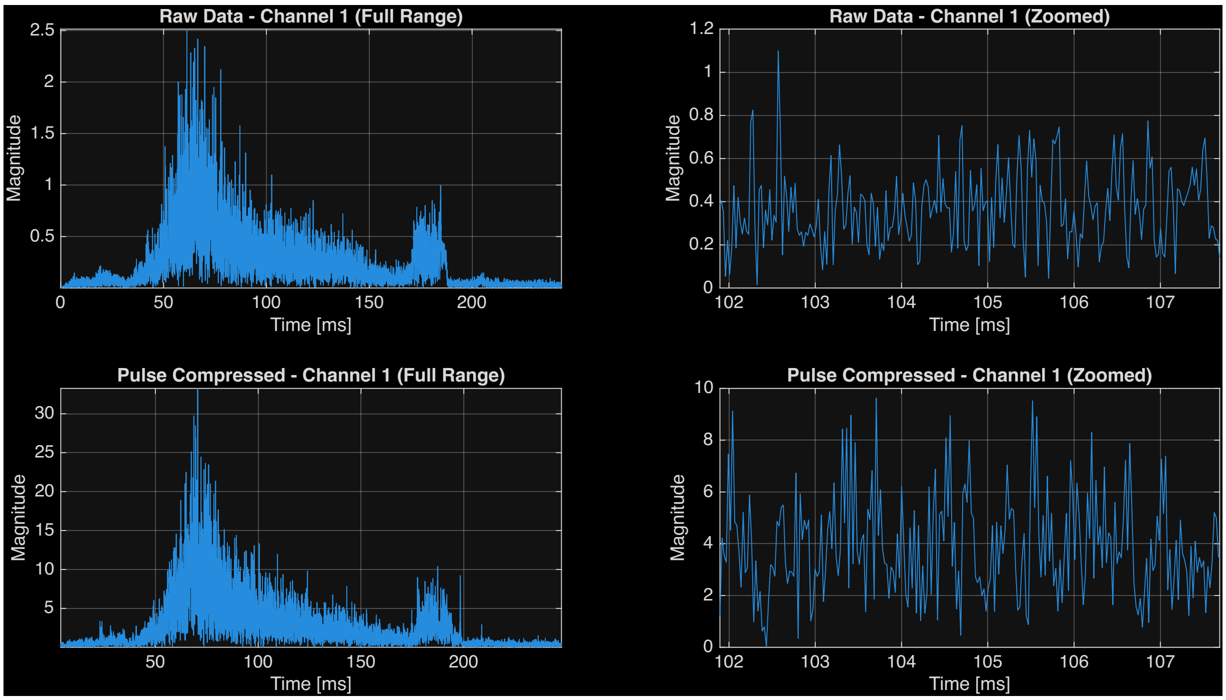

FIG. 2. Comparison of beamformed images: Raw (left) and Pulse Compressed (right)