Homework 2 - Underwater Acoustics

Andreas Raja Goklas Sitorus
Ahmed Abdelgayed

January 22, 2023

Let us consider a water channel, 350 m in depth, and let us assume that the velocity c is constant and equal to 1500 m/s. A signal is transmitted from a point-like source in the water and measured by a set of 9 receivers located along a vertical line at depths $25 \cdot n$ metres, $1 \leq n \leq 9$. When hitting the (flat) boundaries, the acoustic waves are totally reflected.

The file 'Received.mat' contains a 9 x 32000 array which provides the 9 received signals during 6.4 s with sampling path $dt = 2.3e - 04$ s. Line n corresponds to the receiver at depth $25 \cdot n$ metres.

Plot the received signals and explain how you can get a rough estimation of the location of the transmitter from the various time delays. Improve the accuracy of this first estimate thanks to the simulation of the backpropagation of the time-reversed received signals.

Answer:

1 Signal Plotting

To plot the received signals, we load the file ”Received.mat”. We get the following result.

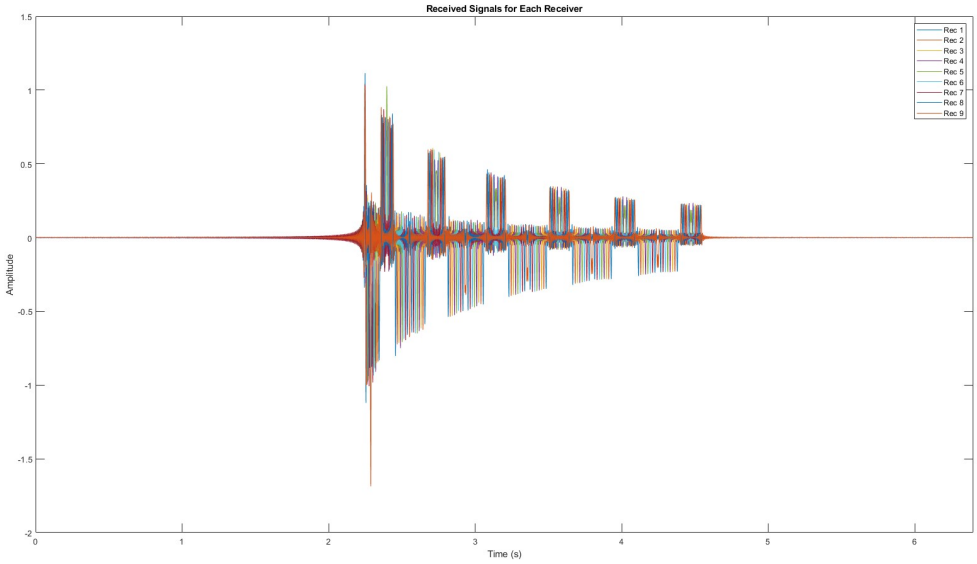

Figure 1: Signal from Every Receiver Plotted

Plotting the nine received signals on the same figure with respect to the same time frame (as shown in Figure 1), we clearly see that there exists a signal that has the highest peaks. This is logical and confirms our simulation since the First signal to arrive is the closest one to this receiver, and hence, suffers less attenuation than others.

The code for calculating the plotting will be shown in the code snippet below:

Initialization

```
clear all
clc
clf

% Environment
Zseabed = 350;
c0 = 1500;

% Receiver Location
xr = 0;
zr = zeros(1,9);
for i = 1 : length(zr);
    zr(i) = 25*i;
end

% Signal Properties
Sigs = load('Received.mat');
n_reflect = 8;
t = 6.4;
dt = 2e-4;
step = t/dt;
t_input = linspace(0,t,step)

% Plot received signal from multiple receiver
for i = 1 : length(zr);
    plot(t_input, Sigs.RecSig(i,:))
    hold on
end
xlim([0 6.4])
ylim([-2 1.5])
legend('Rec 1', 'Rec 2', 'Rec 3','Rec 4','Rec 5','Rec 6','Rec 7','Rec 8', 'Rec 9')
title('Received Signals for Each Receiver')
xlabel('Time (s)')
ylabel('Amplitude')
hold off
```

2 Rough Estimation of the Distance

In order to roughly estimate the distance/ position at which the signal is transmitted (the position of the source), we take the first signal that is being received by the receiver (the nearest one) and the following signal.

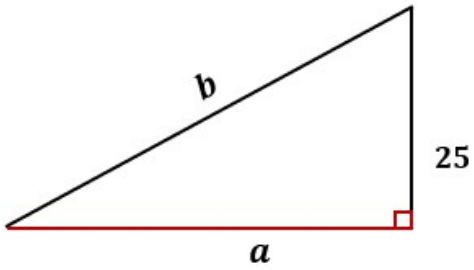

Figure 2: Triangulation Method

Through the investigation of the signals, we get the time at which the first peak of both (signal 1 and signal 2) using the function *findpeaks()* in MATLAB.

Assumptions:

1. Since that we are looking for the closest receiver to the transmitted signal, we assume that both the source and the closest signal are at same depth. Thus, we can assume a right-angle triangle to carry out the calculations.

Considering Figure 2 the following is found:

$$a = c_0 \cdot tb = c_0 \cdot (t + dt)$$

Where **dt** is the delay time between the first peak and the second peak. Using triangulation, we get the following equation:

$$b^2 = a^2 + 25^2$$

We substitute in this equation from **eq1** & **eq2** Using MATLAB's *solve()* function as shown in figure3 we solve for the time t and then we calculate a as the rough estimate of the source's location. We get the following :

$$x_{source} = 1041.5m$$

```
% Rough estimation of the source location
syms tt
[sig1, t1] = findpeaks(Sigs.RecSig(1,:), 'MinPeakHeight',0.7);
[sig2, t2] = findpeaks(Sigs.RecSig(2,:), 'MinPeakHeight',0.7);

delay_time = abs(t2(1)-t1(1))*dt
a = c0 * tt
b = c0 *  (tt+ delay_time)
a = c0 * solve( 25^2 + a^2 -b^2 ==0,tt)
a= vpa(a)
```

3 Simulation of the Back Propagation

Based on the rough estimation we had before, we can try to carry out a brute-force search a grid of possible locations around this estimate. This is only possible due to the time reversal property of the acoustic signals which allows sending back the signal received at the receivers with corresponding time delays, and work out where the source can possibly be by evaluating an energy map. The point with highest energy (with highest focus) is most likely where the signal is transmitted from.

We set the possible x-locations to be (with a step of 5):

$$x = [900, 1200]m$$

and z-locations to be (with step of 5):

$$x = [10, 200]m$$

The energy is evaluated at the every point of this grid, and the results shown in Figure 3 are obtained

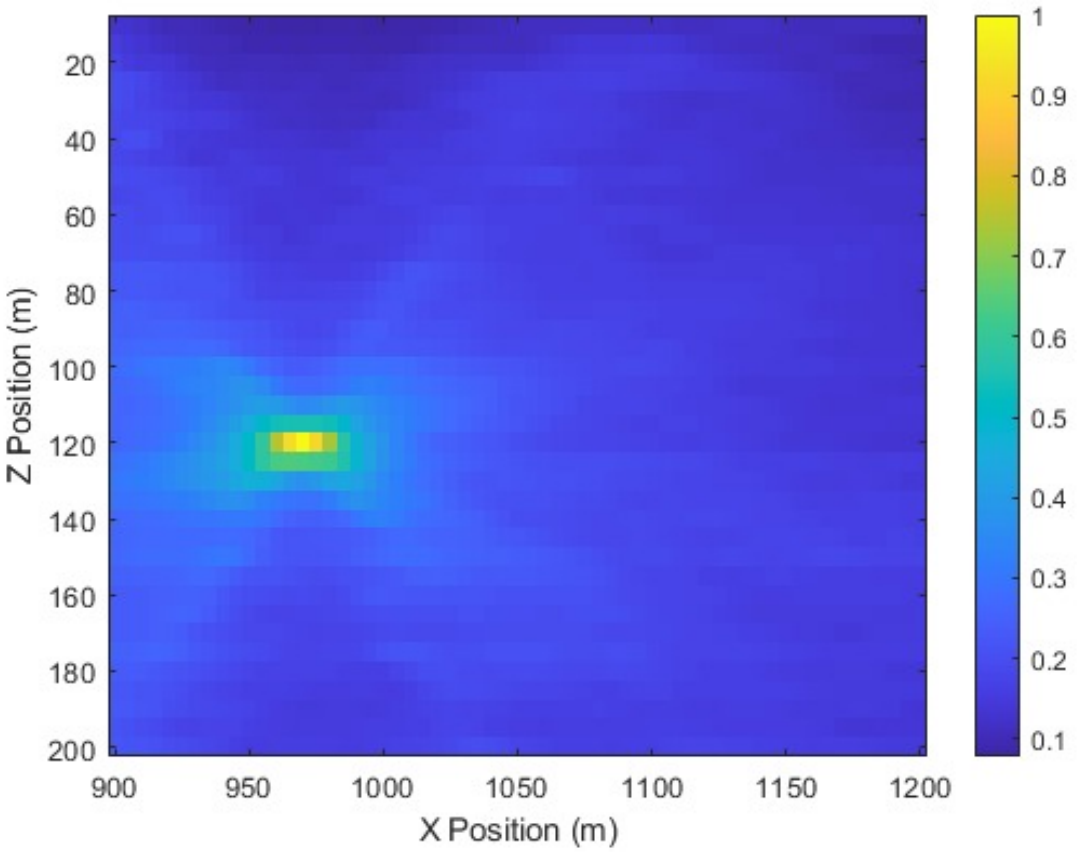

Figure 3: Simulation for Inverse Source Problem with 5 Metres Accuracy

The results show that the source is around the following locations:

$$x_s = 970 \pm 5m$$

$$z_s = 120 \pm 5m$$

Due to the fact that those calculations are carried out over a very large window (grid), it took a lot of time to run the simulation. However, better results can be obtained if the estimation from the last step is taken into account and the grid is more focused. The more focused results are shown in Figure 4

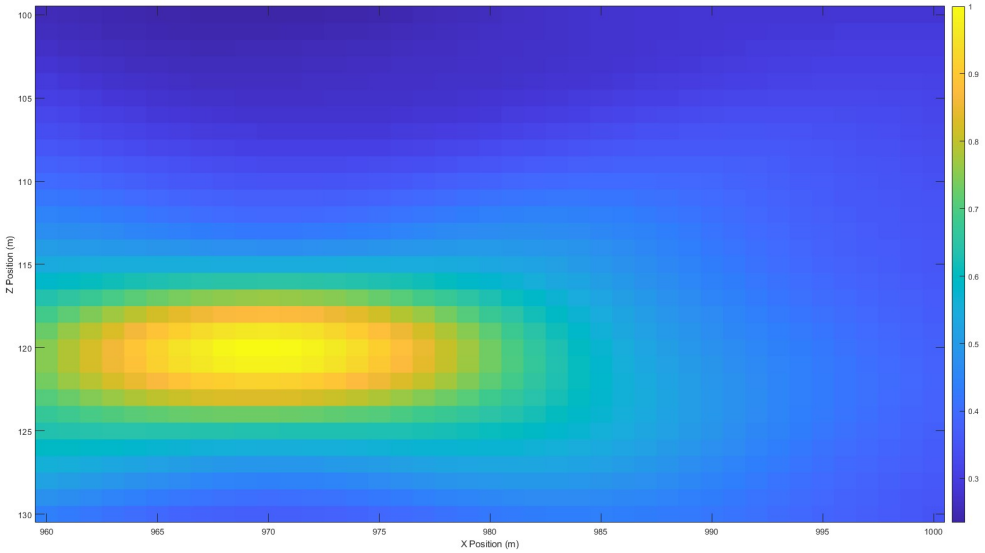

Figure 4: Simulation for Inverse Source Problem with 1 Metres Accuracy

The results show that the source is around the following locations:

$$x_s = 970 \pm 1.5m$$

$$z_s = 120 \pm 1.5m$$

The code for calculating the plotting will be shown in the code snippet below:

Inverse Source Problem

```
% Set the search area grid
xs = 900:5:1200;
zs = 10:5:200; % 0 begin from sea surface
plotgrid = zeros(length(zs),length(xs));

% Find the source
for i = 1:length(xs);
    for j = 1:length(zs);
        % Wave container for the grid
        Superpose_Sigs = zeros(1,length(Sigs.RecSig(1,:)));
        for k = 1:size(zr,2);
            % Inverse the Receiver Signal
            Sig_Inv = flip(Sigs.RecSig(k,:));

            % Send the inverse signal to the search area grid (Receiver
            % become source, and vice versa
            Sig_to_Src = Green_Function(Sig_Inv,xs(i),zs(j),xr,zr(k),dt,n_reflect,Zseabed,c0);

            % Superpose signals from each receiver to the container
            Superpose_Sigs = Superpose_Sigs + Sig_to_Src;
        end
        % Find the maximum energy of the superpose wave
        Max_Energy = sum(Superpose_Sigs.^2, 'all');

        % Map the maximum energy to the grid
        plotgrid(j,i) =  Max_Energy;
    end
end
```