HSB
Hochschule Bremen
City University of Applied Sciences

UNDERWATER ACOUSTICS AND SONAR SIGNAL PROCESSING

SS 2018

ASSIGNMENT 6

IMAGE SOURCE APPROACH

Date: 22/05/2018

by

Shinde Mrinal Vinayak (Matriculation No.: 5021349)
Kshisagar Tejashree Jaysinh (Matriculation No.: 5019958)

guided by

M.Sc. Zimmer

Contents

Introduction

The sonar equation is a systematic way of estimating the expected signal-to-noise ratios for sonar systems. The signal-to-noise ratio determines whether or not a sonar will be able to detect a signal in the presence of background noise in the ocean. It takes into account the source level, sound spreading, sound absorption, reflection losses, ambient noise, and receiver characteristics. The sonar equation is used to estimate the expected signal-to-noise ratios for all types of sonar systems.

In this assignment we first develop a MATLAB program for determining the signal to noise ratio. The calculations for various parameters are then carried out. We then discuss the impact of bottom type (mud, sand and gravel) and wind speed on the signal to noise ratio.

Theory

0.1 Image Source Approach

The wave field within a homogeneous waveguide can be interpreted as the superposition of infinitely many spherical waves that are reflected at the boundaries.

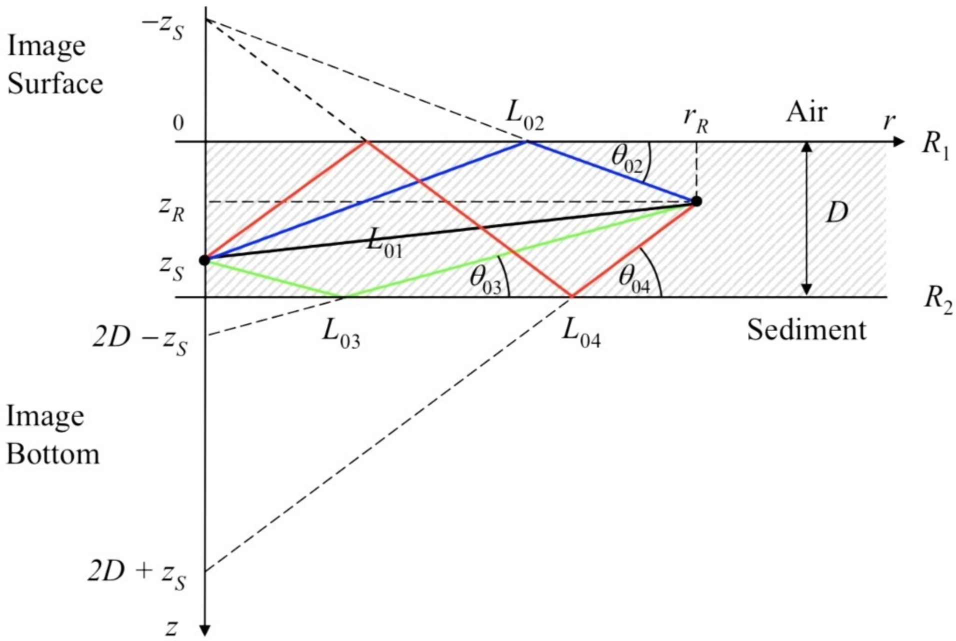


Figure 1: Image source technique

As a first approximation, the sound pressure in the waveguide can be determined by superimposing the four contributions indicated in the Figure 1, i.e.

$$P(r_R, z_R, \omega) = A(\omega)(\frac{e^{-jkL_{01}}}{L_{01}} + R_1(\theta_{02}, \omega)\frac{e^{-jkL_{02}}}{L_{02}} + R_3(\theta_{03}, \omega)\frac{e^{-jkL_{03}}}{L_{03}} + R_4(\theta_{04}, \omega)\frac{e^{-jkL_{04}}}{L_{04}}) \tag{1}$$

with

$$\begin{aligned} L_{01} &= \sqrt{r_R^2 + (z_R - z_S)^2} \\ L_{02} &= \sqrt{r_R^2 + (z_R + z_S)^2} \\ L_{03} &= \sqrt{r_R^2 + (2D - z_S - z_R)^2} \\ L_{04} &= \sqrt{r_R^2 + (2D + z_S - z_R)^2} \end{aligned} \tag{2}$$

and

$$\begin{aligned} \theta_{02} &= arctan((z_S + z_R)/r_R) \\ \theta_{03} &= arctan((2D - z_S - z_R)/r_R) \\ \theta_{04} &= arctan((2D + z_S - z_R)/r_R) \end{aligned} \tag{3}$$

Continuation of the image source technique in multiples m = 1,2,... of groups of four contributions provides

$$\begin{aligned} P(r_R, z_R, \omega) = A(\omega)\sum_{m=0}^{\infty}\Bigg(& R_1^m(\theta_{m1}, \omega)R_2^m(\theta_{m1}, \omega)\frac{e^{-jkL_{m1}}}{L_{m1}} + \\ & + R_1^{m+1}(\theta_{m2}, \omega)R_2^m(\theta_{m2}, \omega)\frac{e^{-jkL_{m2}}}{L_{m2}} + R_1^m(\theta_{m3}, \omega)R_2^{m+1}(\theta_{m3}, \omega)\frac{e^{-jkL_{m3}}}{L_{m3}} + \\ & + R_1^{m+1}(\theta_{m4}, \omega)R_2^{m+1}(\theta_{m4}, \omega)\frac{e^{-jkL_{m4}}}{L_{m4}}\Bigg) \end{aligned} \tag{4}$$

Experimental Research

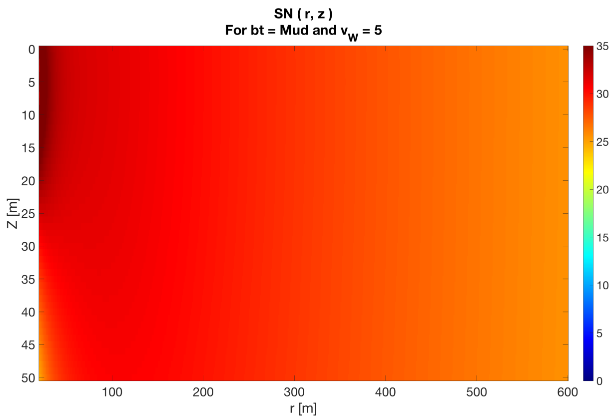
0.2 Signal to Noise Ratio

The MATLAB program for determining the signal to noise ratio was developed. The following parameter values were considered.

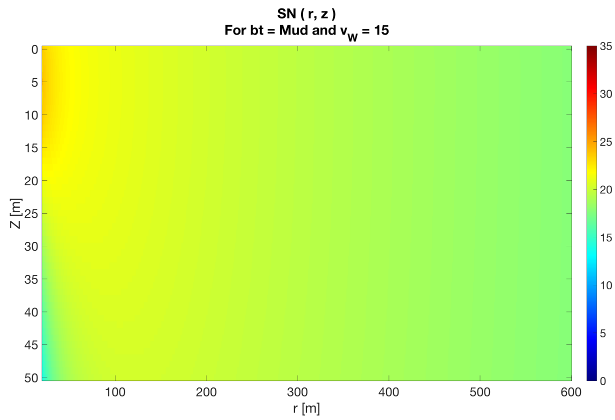
z / r	uptp 50 m / 600 m
bt	mud, sand and gravel
v_W	5, 15 and 25 knots
S	33 ppt
T	15°
c	1480 m/s
SL	220 dB
f	100 kHz
τ	$100\mu s$
DI	30 dB

B	10 kHz
BP_T	0 dB (±90°)
BP_R	0 dB (±90°)
$2\theta_{h,R}$	0.5°
$2\theta_{h,T}$	90°
$2\theta_{v,R}$	180°
$2\theta_{v,T}$	180°
r_S	0 m
z_S	5 m
TS	$-15dB$

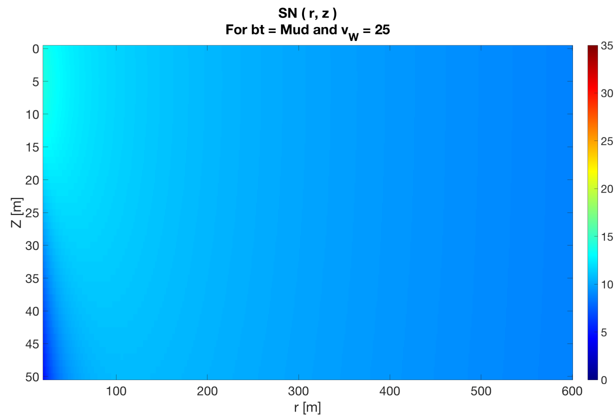
Signal to noise ratio (SNR) is a measure of how strong the signal of interest is with respect to the noise environment. The higher the SNR, the better the detection. The Figure 1, Figure 2 and Figure 3 show SNR as a function of depth and range for this particular sonar/environment scenario. In the images, colour is mapped to SNR where red is high and blue is low.



(a) Wind speed = 5 kn



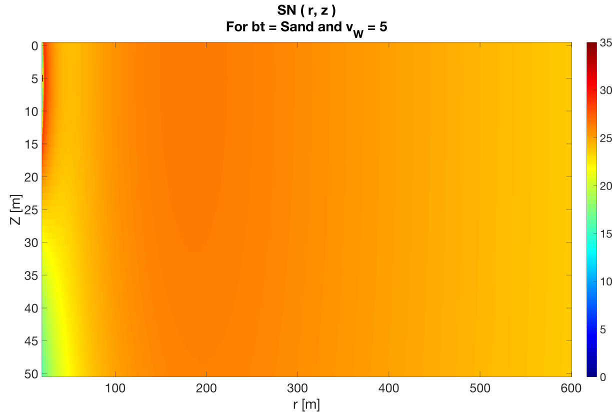
(b) Wind speed = 15 kn



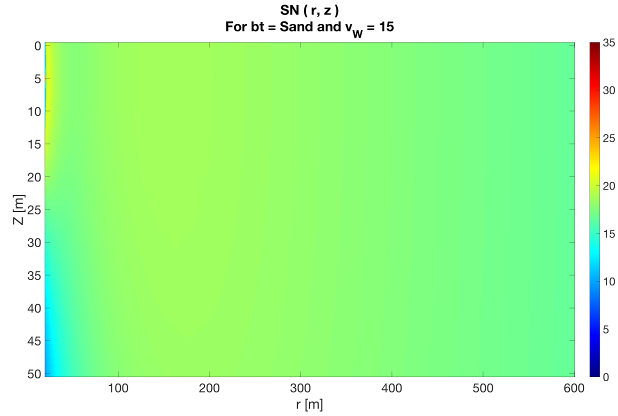
(c) Wind speed = 25 kn

Figure 2: Impact of mud as the bottom type and wind speed of 5, 15 and 25 knots on the signal to noise ratio

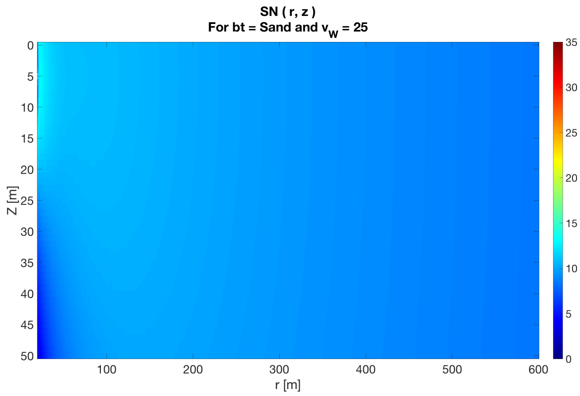
The Figure 1 shows the impact of wind speed on signal to noise ratio in the range of $r(50, 600)$. The figure has been plotted for fixed value of wind speeds (5 kn, 15 kn and 25 kn) and the bottom type is considered to be mud. We can see from the Figure 1 that as the wind speed is increased from 5 kn to 25 kn, the signal to noise ratio decreases.



(a) third



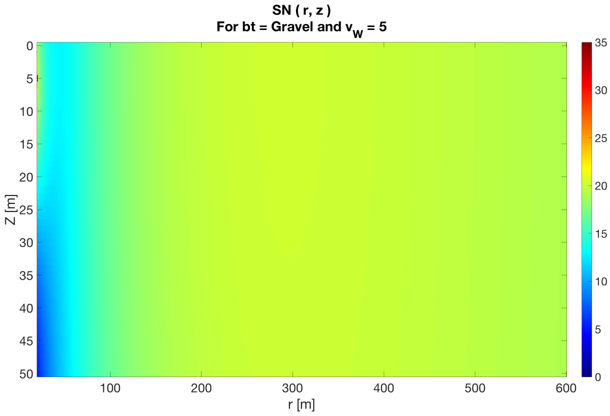
(b) forth



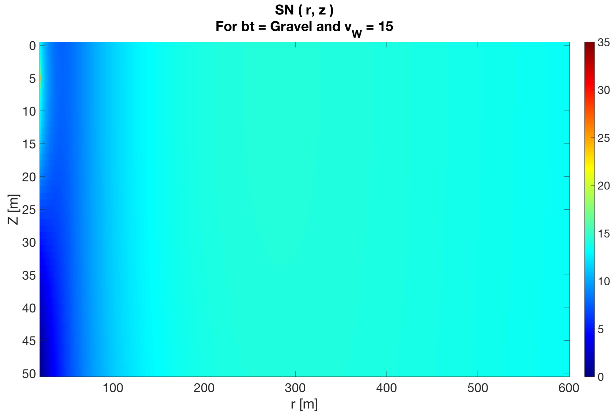
(c) fifth

Figure 3: Impact of sand as the bottom type and wind speed of 5, 15 and 25 knots on the signal to noise ratio

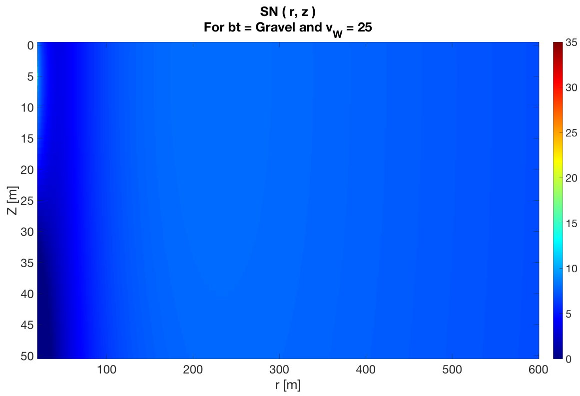
The figure 2 consists of plots with their bottom type as sand. When we compare Figure 2 to Figure 1, we noticed that the signal to noise ratio (SNR) is decreased as the bottom type changed from mud to sand. Also, the dependence if wind speed is similar to that of Figure 1.



(a) first



(b) second



(c) fifth

Figure 4: Impact of gravel as the bottom type and wind speed of 5, 15 and 25 knots on the signal to noise ratio

As the bottom type changes from mud to gravel (soft to hard), signal to noise ratio is decreases gradually. At low wind speed values, the bottom type plays a vital role in determining SNR. For the bottom type of mud, SNR is decreasing with increasing wind speeds. For the hard bottom types with the high wind speeds, SNR will be worst. The SNR has medium values for the highest values of the bottom type mud and sand. For the bottom type gravel, the SNR has the smallest values i.e. SNR is getting worse and we can observe highest noise.

Conclusion

The MATLAB code was written for determining the signal to noise ratio and carry out calculations for the following parameters. Signal to Noise ratio (SNR) due to wind at various speeds ranging from 5 knots to 25 knots and bottom type of mud, sand and gravel is analysed and observed. The signal to noise ratio is higher at lower winds speeds and on soft surfaces (mud) and thus can be well detected.

Appendix

0.3 MATLAB functions required to plot dependence of bottom type and wind speed on signal to noise ratio

0.3.1 MATLAB function to compute Transmission Loss

```
% Function file saved as 'Atten_Schulkin_Marsh.m'
function [A] = Atten_Schulkin_Marsh(f,S,T,Zmax)
A = 2.34e-6;
B = 3.38e-6;
f = f./1000;
ft = 21.9.*10.^(6-(1520./(T+273)));
P = 1.01.*(1+Zmax.*0.1);
A = 8.686e3.*((S.*A.*ft.*f.^2)./(ft^2+f.^2)+(B.*f.^2)./(ft)) ...
    *(1-6.54e-4.*P);
end
```

0.3.2 MATLAB function to compute the noise level

```
% Function file saved as 'Noise_Level.m'
function NL = Noise_Level(f,Vw)
f = f./1000;
NLTraffic = 10.*log10(3e8./(1+1e4.*f.^4)); % Shipping noise (traffic)
NLTurb = 30-30.*log10(f);  % Turbuleance noise
NLVessel = -999; % Self noise of sonar platform (vessel)
NLBio = -999; % Biological noise (fishes, scrimps, etc.)
NLSS = 40+10.*log10(Vw.^2./(1+f.^(5/3))); % Sea state noise
NLThermal = -15+20.*log10(f); % Thermal noise
```

```
% Total isotropic noise level
NL = 10.*log10(10.^(0.1.*NLTraffic)+10.^(0.1.*NLTurb)+10.^ ...
    (0.1.*NLVessel)+10.^(0.1.*NLBio)+10.^(0.1.*NLSS)+10.^ ...
    (0.1.* NLThermal));
end
```

0.3.3 MATLAB function to compute the bottom reverberation

```
% Function file saved as 'Bottom_Reverberation.m'
function [Sb] = Bottom_Reverberation(f,bt,g)
f = f./1000;
k = 1+125.*exp(-2.64.*(bt-1.75).^2-50./bt.*(cot(g)).^2);
b = k.*(sin(g)+0.19).^(bt.*(cos(g)).^16);
Sb = 10.*log10(3.03.*b.*(f.^(3.2-0.8.*bt).* ...
    10.^(2.8.*bt-12)+10.^(-4.42)));
end
```

0.3.4 MATLAB function to compute the surface reverberation

```
% Function file saved as 'Surface_Reverberation.m'
function [Ss] = Surface_Reverberation(f,Vw,g)
f = f./1000;
b = (4.*(Vw+2)./(Vw+1))+(2.5.*(f+0.1).^(-1./3)-4).* ...
    (abs(cos(g))).^(1/8);
Ss = 10.*log10(10.^(-5.05).*(1+Vw).^2.*(f+0.1).^(Vw./150).* ...
    (tan(g)).^(b));
end
```

0.3.5 MATLAB function to compute the volume reverberation

```
% Function file saved as 'Volume_Reverberation.m'
function [Sv] = Volume_Reverberation(f,Pd)
f = f./1000;
if Pd == '1'
    SP = -50;
else
    if Pd == '0.5'
        SP = -70;
else
SP = -90;
    end
end
Sv = SP+7.*log10(f);
end
```

0.4 MATLAB program to plot dependence of bottom type and wind speed on signal to noise ratio

```
% main program saved as 'USP5.m'
bt = 1:1:3;
v_W = 5:10:25;
Pd = 'low';
S = 33;
T = 15;
C = 1480;
SL = 220;
TS = -15;
r_S = 0;
z_S = 5;
Zmax = 25;
f = 100000;
tau = 100e-6;
B = 10e3;
Thetah = deg2rad(0.5);
Thetav = deg2rad(60);
% Area
r1 = 20:1:600;
z1 = 0:1:50;
[r,z,Vw,bt] = ndgrid(r1,z1,v_W,bt);
% Distance
RR = sqrt((r-r_S).^2+(z-z_S).^2);
% Transmission loss
alpha = Atten_Schulkin_Marsh(f,S,T,Zmax).*(RR-1)./1000;
TL = 20.*log10(RR) + alpha;
% for isovelocity
TLe = TL;
TLb = TL;
TLs = TL;
TLv = TL;
% Isotropic Noise Level
NL = Noise_Level(f,Vw);
NLb = NL + 10.*log10(B);
% Angles
Thetab = atan((Zmax-z_S)./sqrt(RR.^2-(Zmax-z_S).^2));
Thetas = atan(z_S./sqrt(RR.^2-(z_S).^2));
Thetae = atan((z-z_S)./(r-r_S));
% Bempattern values
BPtb = 10*log10(cos(Thetab));
BPrb = 10*log10(cos(Thetab));
BPts = 10*log10(cos(Thetas));
BPrs = 10*log10(cos(Thetas));
BPte = 10*log10(cos(Thetae));
```

```
BPre = 10*log10(cos(Thetae));
% Bottom reverberation strength
Ab = Thetah.*RR.*C.*tau./cos(Thetab);
Sb = Bottom_Reverberation(f,bt,Thetab);
RSb = Sb + 10.*log10(Ab);
% Surface reverberation strength
As = Thetah.*RR.*C.*tau./cos(Thetas);
Ss = Surface_Reverberation(f,Vw,Thetas);
RSs = Ss + 10.*log10(As);
% Volume reverberation strength
V = 2.*Thetah.*Thetav.*RR.^2.*C.*tau;
Sv = Volume_Reverberation(f,Pd);
RSv = Sv + 10.*log10(V);
% Directivity index
DI = 40 - 10.*log10(Thetav*Thetah*180.^2./(pi.^2));
% SNR
Rlb = SL+BPtb+BPrb-2.*TLb+RSb;
Rls = SL+BPts+BPrs-2.*TLs+RSs;
Rlv = SL-2.*TLv+RSv;
TIL = 10.*log10(10.^(0.1.*(NL-DI))+10.^(0.1.*Rlb)+ ...
    10.^(0.1.*Rls)+10.^(0.1.*Rlv));
SNR = SL+BPte+BPre-2.*TL+TS-(NL-DI)-TIL;
% Plot
set(0,'DefaultAxesFontSize',25)
figure(1);
CLIM = [0 35];
imagesc(r1,z1,SNR(:,:,1,1)',CLIM);
colormap(jet(256));
colorbar('vert');
title({'SN ( r, z )'; 'For bt = Mud and v_W = 5'});
xlabel('r [m]');
ylabel('Z [m]');
figure(2);
CLIM = [0 35];
imagesc(r1,z1,SNR(:,:,2,1)',CLIM);
colormap(jet(256));
colorbar('vert');
title({'SN ( r, z )'; 'For bt = Mud and v_W = 15'});
xlabel('r [m]');
ylabel('Z [m]');
figure(3);
CLIM = [0 35];
imagesc(r1,z1,SNR(:,:,3,1)',CLIM);
colormap(jet(256));
colorbar('vert');
title({'SN ( r, z )'; 'For bt = Mud and v_W = 25'});
xlabel('r [m]');
ylabel('Z [m]');
figure(4);
```

```
CLIM = [0 35];
imagesc(r1,z1,SNR(:,:,1,2)',CLIM);
colormap(jet(256));
colorbar('vert');
title({'SN ( r, z )'; 'For bt = Sand and v_W = 5'});
xlabel('r [m]');
ylabel('Z [m]');
figure(5);
CLIM = [0 35];
imagesc(r1,z1,SNR(:,:,2,2)',CLIM);
colormap(jet(256));
colorbar('vert');
title({'SN ( r, z )'; 'For bt = Sand and v_W = 15'});
xlabel('r [m]');
ylabel('Z [m]');
figure(6);
CLIM = [0 35];
imagesc(r1,z1,SNR(:,:,3,2)',CLIM);
colormap(jet(256));
colorbar('vert');
title({'SN ( r, z )'; 'For bt = Sand and v_W = 25'});
xlabel('r [m]');
ylabel('Z [m]');
figure(7);
CLIM = [0 35];
imagesc(r1,z1,SNR(:,:,1,3)',CLIM);
colormap(jet(256));
colorbar('vert');
title({'SN ( r, z )'; 'For bt = Gravel and v_W = 5'});
xlabel('r [m]');
ylabel('Z [m]');
figure(8);
CLIM = [0 35];
imagesc(r1,z1,SNR(:,:,2,3)',CLIM);
colormap(jet(256));
colorbar('vert');
title({'SN ( r, z )'; 'For bt = Gravel and v_W = 15'});
xlabel('r [m]');
ylabel('Z [m]');
figure(9);
CLIM = [0 35];
imagesc(r1,z1,SNR(:,:,3,3)',CLIM);
colormap(jet(256));
colorbar('vert');
title({'SN ( r, z )'; 'For bt = Gravel and v_W = 25'});
xlabel('r [m]');
ylabel('Z [m]');
```