HSB
Hochschule Bremen
City University of Applied Sciences

UNDERWATER ACOUSTICS AND SONAR
SIGNAL PROCESSING

SS 2018

ASSIGNMENT 2

SOUND ATTENUATION IN THE OCEAN

Date: 16/04/2018

by

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

under the guidance of

Prof. Zimmer

HSB
Hochschule Bremen
City University of Applied Sciences

Contents

Chapter 1

Introduction

Salt water as a medium is dissipative. Due to viscosity of salt water (above 100 kHz), relaxation (i.e. the conversion of acoustic energy into heat) of magnesium sulphate (above 100 kHz or 10 kHz to 100 kHz) and boric acid (above 1 kHz or $\leq$10 kHz) sound waves are attenuated. There are numerous empirical models available for the prediction of attenuation of underwater sound, e.g. Thorp, Schulkin-Marsh and Francois-Garrison. The absorption of low frequency sound is comparably small. However, the magnitude of absorption increases with increasing frequency. In contrast to the sea surface, sound waves can penetrate into soft sea floors and consequently be absorbed significantly. Since the sea bed is a solid, also shear waves can develop. At long distances, losses in acoustic sound intensity result from attenuation and scattering losses. Sound attenuation in underwater environments is dependent on temperature, salinity, pressure and pH. If sound is propagating into the sea bed, it also is attenuated as a function of sediment type.

This assignment consists of three parts: In the first part we develop a MATLAB program that calculates the sound attenuation in seawater by means of Thorp, Schulkin & Marsh and Francois &

Garrison formula. The results of the three approaches are then compared. The next two parts are done for Francois & Garrison formula. The dependency on the frequency, salinity and temperature for a depth of 50m is investigated and finally a graph for attenuation versus frequency is plotted and frequency regions where the different attenuation processes dominate are specified.

Chapter 2

Theory

2.1 Sound attenuation in water

An acoustic signal underwater experiences attenuation due to spreading and absorption. The acoustic energy of a sound wave propagating in the ocean is partly absorbed, i.e. the energy is transformed into heat and partly lost due to sound scattering by in homogeneities. On the basis of extensive laboratory and field experiments the following empirical formulae for attenuation coefficient in sea water have been derived. Sound absorption in sea water is a function of frequency, temperature, salinity, pH, and pressure. When the acoustic wave propagates in sea water, absorption loss occurs, which is caused by a part of the energy changing into the heat owing to the viscous friction of the water molecule, aside from the spreading loss. The absorption loss is represented as α r, where α is the coefficient in dB/Km and r is the transmission distance.

2.1.1 Thorp formula

The absorption coefficient for frequency range 100 Hz to 3 kHz can be expressed empirically using Thorp?s formula which defines α_w [dB/m] as a function of f [kHz].

$$\alpha_w = \frac{0.11f^2}{1+f^2} + \frac{44f^2}{4100+f^2}$$

2.1.2 Schulkin and Marsh formula

The absorption coefficient for frequency range 3 kHz to 500 kHz can be expressed empirically using Schulkin and Marsh model which defines α_w [dB/m] as a function of f [kHz].

$$\alpha_w = 8.686.10^3(\frac{SAf_tf^2}{f_t^2+f^2} + \frac{Bf^2}{f_t})(1-6.54.10^{-4}P)$$

where $A = 2.34.10^{-6}, B = 3.38.10^{-6}$, S in [ppt], f in [kHz], relaxation frequency $f_t = 21.9.10^{\frac{6-1520}{T+273}}$ with T in [°C] for $0 \leq T \leq 30$ and thy hydrostatic pressure is determined by $P = 1.01(1+z0.1)$ in $[kg/cm^3 = at]$

2.1.3 Francois and Garrison Formula

The absorption coefficient for frequency range 100 Hz to 1 MHz can be expressed empirically using Francois and Garrison model which defines α_w [dB/m] as a function of f [kHz].

$$\alpha_w = \frac{A_1 P_1 f_1 f^2}{f_t 1^2 + f^2} + \frac{A_2 P_2 f_2 f^2}{f_t 2^2 + f^2} + A_3 P_3 f^2$$

The first term gives the sound absorption due to the Boric Acid and second term gives the sound absorption due to the magnesium sulfate. The contribution of sound absorption due to these chemical ingredients has been found to be small. The third term represents the sound absorption due to pure water. The pressure dependency of above equation is shown by P1, P2 and P3 constants .Frequency dependency is given by f1 and f2 which are the relaxation frequencies of Boric Acid and Magnesium sulfate. f is the frequency of sound. The constants A1, A2 and A3 shown are not purely constants but it has been experimentally proved that their values vary with the water properties, like temperature, salinity and pH of water. The total coefficient of absorption of sea water is calculated by considering separately the absorption due to boric acid, magnesium sulphate and pure water.

$A_1 = \frac{8.686}{c} 10^{0.78ph-5}, f_1 = 2.8\sqrt{\frac{S}{35}} 10^{4-\frac{1245}{T+273}}$

$P_1 = 1, c = 1412 + 3.21T + 1.19S + 0.0167Z_m$

$A_2 = 21.44\frac{S}{c}(1 + 0.0025T), f_2 = \frac{8.17.10^{\frac{8-1990}{T+273}}}{1+0.0018(S-35)}$

$P_2 = 1 - 1.37.10^{-4} z_m + 6.2.10^{-9} z_m^2$

$$A_3 = \begin{cases} 4.937.10^{-4} - 2.59.10^{-5}T + 9.11.10^{-7}T^2 - 1.5.10^{-8}T^3 & \text{for } T \leq 20 \\ 3.964.10^{-4} - 1.146.10^{-5}T + 1.145.10^{-7}T^2 - 6.5.10^{-8}T^3 & \text{for } T \geq 20 \end{cases}$$

$P_3 = 1 - 3.83.10^{-5} z_m + 4.9.10^{-10} z_m^2$

with S in [ppt], f in [kHz], T in [°C]. Furthermore z_m, ph and c denote the water depth in [m], the ph-value and the sound speed in [m/s] respectively.

Chapter 3

Experimental research

By using the formula for velocity of speed in ocean, the graph of velocity of speed in ocean versus depth is plotted for various values of temperature.

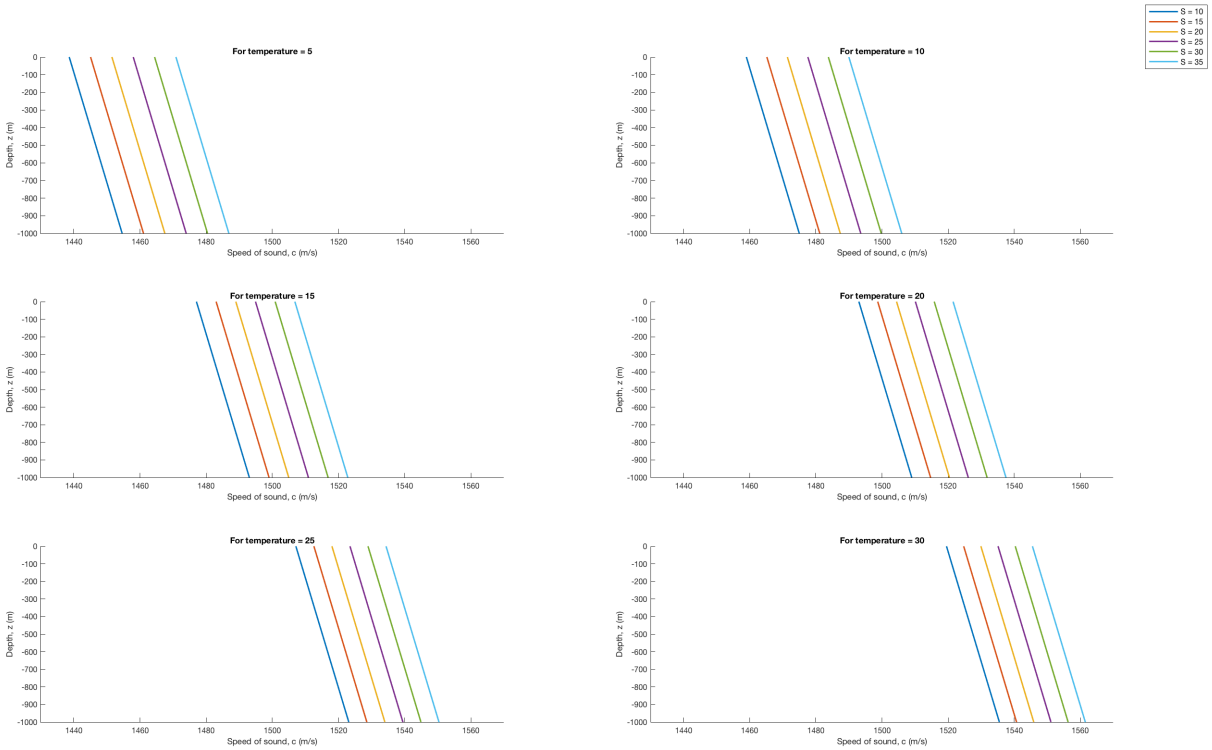

Figure 3.1: c versus z for various sets of T

By using the formula for velocity of speed in ocean, the graph of velocity of speed in ocean versus depth for various values of salinity is obtained.

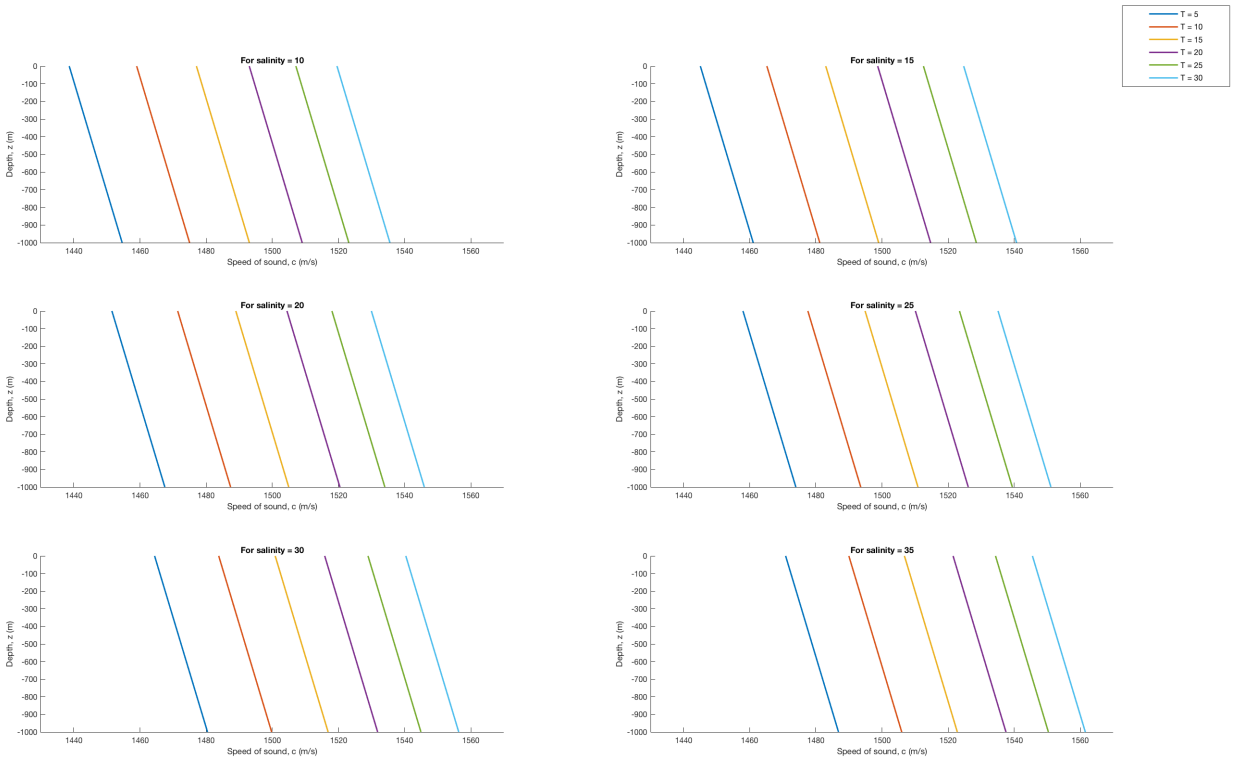

Figure 3.2: c versus z for various sets of S

The dependency of non-linear temperature profile and depth is plotted using an exponential function for temperature. The plot contains of three curves: temperature versus depth, sound velocity (having constant salinity and temperature) versus depth and sound velocity (having expoenential temperature and constant salinity) versus depth.

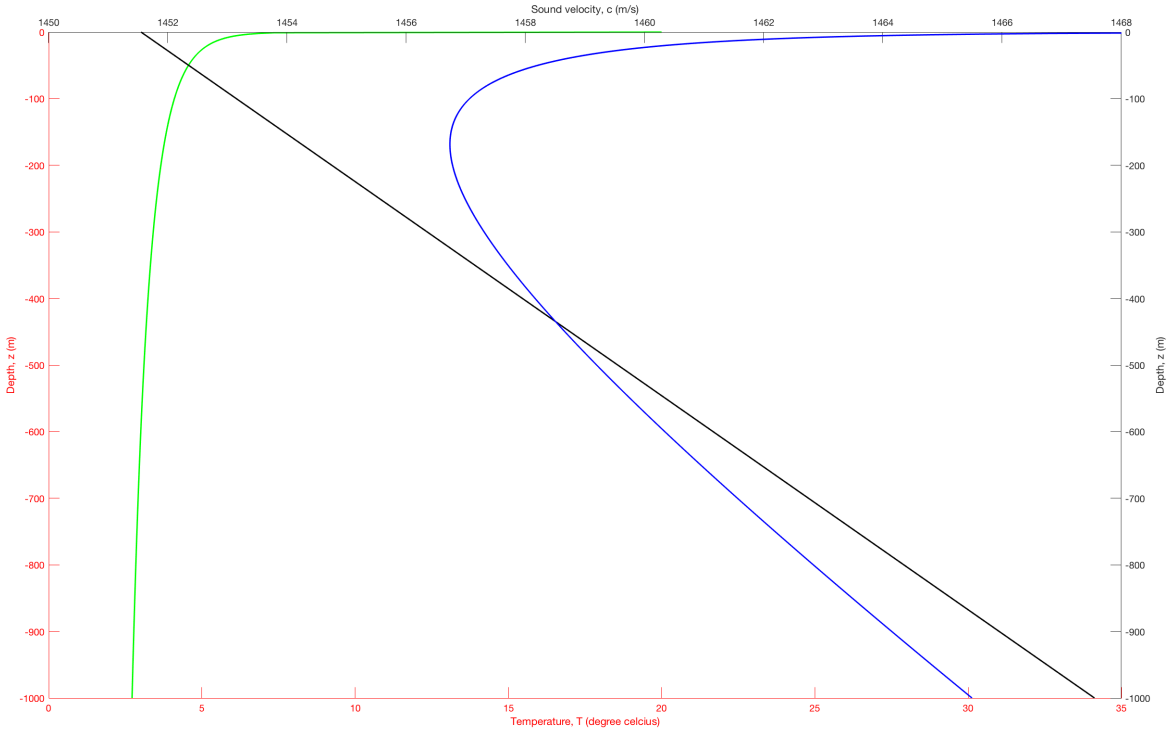

Figure 3.3: Dependency of non-linear temperature profile and depth

Chapter 4

Conclusion

It can be concluded that depth and salinity of the ocean are linearly dependent on sound velocity in the ocean while temperature has non linear dependance on speed of sound in the ocean

Chapter 5

Appendix

5.1 MATLAB code for dependence of c on T, S and z

5.1.1 Function to compute velocity of sound

```
% Function saved as speedOfSound.m and is called from other MATLAB files
function c = speedOfSound(S, T, Z)
c = 1449.2 + 4.6*T - 0.055*T.^2 + 0.00029*T.^3 ...
                + 1.34*S - 46.9 - 0.01*T.*S + 0.35*T + 0.016*Z;
end
```

5.1.2 For various values of temperature

```
%file saved as UCP1_1.m
temperature = 5:5:30; % temperature varies in steps on 5 from 5 to 30
```

```
3  salinity = 10:5:35; % salinity varies in steps of 5 from 10 to 35
4  depth = 0:200:1000; % depth varies in steps of 200 from 0 to 1000
5  for i = 1:6
6      u = temperature(i);
7      for j = 1:6
8          v = salinity(j);
9          c = speedOfSound(v, u, depth); % calling the function
10         subplot(3,2,i);
11         hold on;
12         plot(c,-depth,'LineWidth',1.5);
13         title(['For temperature = ',num2str(temperature(i))])
14         ax = gca; % current axes
15         ax.FontSize = 8;
16         ax.XLim = [1430 1570];
17
18        % labelling of axes
19         ylabel('Depth, z (m)')
20         xlabel('Speed of sound, c (m/s)')
21     end
22 end
23 % labelling of waveforms
24 hL = legend('S = 10', 'S = 15', 'S = 20', 'S = 25','S = 30', 'S = 35');
25 newPosition = [0.85 0.85 0.2 0.2];
```

```
26 newUnits = 'normalized';
27 set(hL,'Position', newPosition,'Units', newUnits);
```

5.1.3 For various values of salinity

```
1  % file saved as UCP1_2.m
2  temperature = 5:5:30; % temperature varies in steps on 5 from 5 to 30
3  salinity = 10:5:35; % salinity varies in steps of 5 from 10 to 35
4  depth = 0:200:1000; % depth varies in steps of 200 from 0 to 1000
5  for i = 1:6
6      u = salinity(i);
7      for j = 1:6
8          v = temperature(j);
9          c = speedOfSound(u, v, depth); % calling the function
10         subplot(3,2,i);
11         hold on;
12         plot(c,-depth,'LineWidth',1.5);
13         title(['For salinity = ',num2str(salinity(i))])
14         ax = gca; % current axes
15         ax.FontSize = 8;
16         ax.XLim = [1430 1570];
17         ylabel('Depth, z (m)')
18         xlabel('Speed of sound, c (m/s)')
19     end
```

```
20 end
21 hL = legend('T = 5', 'T = 10', 'T = 15', 'T = 20','T = 25', 'T = 30');
22 newPosition = [0.85 0.85 0.2 0.2];
23 newUnits = 'normalized';
24 set(hL,'Position', newPosition,'Units', newUnits);
```

5.2 MATLAB code for dependence of non linear-temperature profile and depth

```
1 temperature = 5:5:30; % temperature varies in steps on 5 from 5 to 30
2 salinity = 10:5:35; % salinity varies in steps of 5 from 10 to 35
3 depth = 0:1000; % depth varies in steps of 1 from 0 to 1000
4 u = 4; % salinity
5 v = 10; % temperature
6 expo_temp = 20*exp(-depth.^.1);
7 line(expo_temp,-depth,'color','g','LineWidth',1.5);
8 ax1 = gca; %current axis
9 ax1.XColor = 'r';
10 ax1.YColor = 'r';
11 ax1.XLim = [0 35];
12 ylabel('Depth, z (m)')
13 xlabel('Temperature, T (degree celcius)')
14 ax1_pos = ax1.Position; % position of first axes
```

```
15 ax2 = axes('Position',ax1_pos,...
16     'XAxisLocation','top',...
17     'YAxisLocation','right',...
18     'Color','none');
19 ax2.XLim = [1450  1468];
20 ax2.YLim = [-1000  0];
21 ylabel('Depth, z (m)')
22 xlabel('Sound velocity, c (m/s)')
23 c = speedOfSound(u, v, depth); % calling the function
24 line(c,-depth,'Parent',ax2,'Color','k','LineWidth',1.5)
25 c = speedOfSound(26, expo_temp, depth); % calling the function
26 line(c,-depth,'Parent',ax2,'Color','b','LineWidth',1.5)
```