Winter 2016

Evaluating the usage of multi-frequency backscatter data as an additional tool for seafloor characterization

Anderson Barbosa da Cruz Pecanha
University of New Hampshire, Durham

EVALUATING THE USAGE OF MULTI-FREQUENCY BACKSCATTER DATA AS AN ADDITIONAL TOOL FOR SEAFLOOR CHARACTERIZATION

BY

ANDERSON BARBOSA DA CRUZ PEÇANHA

B.S. in Naval Sciences, Brazilian Naval Academy, Rio de Janeiro, Brazil (2003)

THESIS

Submitted to the University of New Hampshire

in partial fulfillment of

the requirements for the degree of

Master of Science

in

Earth Sciences

December, 2016

This thesis has been examined and approved in partial fulfillment of the requirements for the degree of Master of Science in Earth Sciences by:

Thesis Director, Andy Armstrong, Co-Director of the JHC, University of New Hampshire.

Thomas C. Weber, Assistant Professor, University of New Hampshire.

John E. Hughes Clarke, Professor, University of New Hampshire

On September 6th 2016

Original approval signatures are on file with the University of New Hampshire Graduate School.

DEDICATION

I would like to dedicate this work to my love and family, whose motivation and endless support were essential to my success.

ACKNOWLEDGEMENTS

First of all, I would like to acknowledge the Brazilian Navy for trusting and designating me in this mission and also for sponsoring this research.

In addition, I would like to give other special acknowledgements to these special people that have supported me along this journey:

- My advisor, Andrew Armstrong, and also the other committee members, Thomas Weber and John Hughes Clarke, for the invaluable orientation of my work. Your guidance have been crucial and essential to this project. Your suggestions enriched my research and I have no words to describe how thankful I am;

- Vice-Admiral Pontes Lima and Captain Giucemar (Brazilian Navy), for believing in me during the admission process. I recognize that without your support this course would not happen;

- Captain Franswillian (Brazilian Navy) and Professor Cláudia Krueger (Federal University of Paraná, Brazil), for believing in my potential. Thank you for your recommendation letters. Your contribution was essential to make all this happen;

- Captain Alenquer (Brazilian Navy), for all guidance during selection process and support along these two years;

- Commander Adriano Vieira (Brazilian Navy), my co-advisor and friend, for supporting myself since the beginning of the admission process. Thanks for always trusting in me and for helping me with the codes listed in APPENDIX D. I know and I recognize that without your help, I would not be here. Thank you for all. I wish you all the best;

- Lieutenant Commander Freire and Borba (Brazilian Navy) and their family, for our friendship. Thank you for supporting me since my first moments in the United States;

- Onni Irish, John Kidd and Massimo di Stefano (Center for Coastal and Ocean Mapping / Joint Hydrographic Center Grad students), for our friendship and valuable discussions. We started this long ride together and I am truly proud to testify how many challenges we overcame together;

- Hirokazu, Maxlimer, Jaya, Amon, Nilupa and Indra (General Bathymetric Chart of the Oceans students) and Tiziana for our friendship and fruitful discussions;

- Brian Calder, James Gardner, James Irish, Jo Laird, Joel Johnson, John Kelley, Kenneth Baldwin, Larry Mayer, Larry Ward, Lee Alexander, Rochelle Wigley, Schachak Pe'eri, Semme Dijkstra, Steve Wineberg, Thomas Lippman, Val Schimidt and Will Clyde, for your memorable teachings;

- Other faculty and staff members of the Center for Coastal and Ocean Mapping / Joint Hydrographic Center, for your valuable support; and

- To my family, whose endless love and unconditional support made myself focused and confident that I would be able succeed. You've been inspiring me during my whole life. I promise I'll keep doing my best to make you all proud of me.

Finally, I would like to say thank you to my dear love and partner. I know that without your help and support it would not be easy to arrive until here. Your advices helped me to keep myself focused and achieve the main goal. You know all the struggles I've passed along this entire process and you always made myself confident that I would be able to beat all of them. I have no words to express my gratitude. I love you!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

EVALUATING THE USAGE OF MULTI-FREQUENCY BACKSCATTER DATA AS AN ADDITIONAL TOOL FOR SEAFLOOR CHARACTERIZATION

BY

ANDERSON BARBOSA DA CRUZ PEÇANHA
University of New Hampshire, December, 2016

A reliable understanding of seafloor characteristics can have innumerous application in a variety of fields of knowledge, such as ocean mapping and defense. In the last decades, studies associating backscatter intensity to seafloor characterization has increased based on the principle that different types of seabed may provide, a priori, different reflectivity responses patterns. Those differences in intensity can be used to attempt seafloor classification.

This thesis proposes to evaluate the potential usage of multi-frequency backscatter as an additional tool for seafloor characterization. Modern multibeam systems are able to provide high resolution bathymetry and backscatter data. The echosounder used to collect data for this research was a Kongsberg EM 2040, which can transmit using three different center frequencies (200, 300 and 400 kHz). The dataset was collected using the three available frequencies and it was investigated under two different perspectives: The first consists of interpreting how backscattering strength curves may vary when the same frequency is used to ensonify different types of substrates. This approach can be used to establish a connection between acoustic wavelength and intensity levels, and the results can have a huge application in seafloor

characterization. The second consists of verifying the existence of any frequency dependency when the same type of seabed is ensonified with different frequencies.

In addition to the two types of investigation listed in the previous paragraph, some of the corrections that had been applied to the raw data during the data acquisition process were compared to more accurate post-processing models. Those comparisons were made in order to evaluate if the approximations made by the acquisition software could impact the usage of *BS* as a seafloor characterization tool.

INTRODUCTION

Many commercial, biological, geological and military applications depend on the understanding of the characteristics of the seabed, whose comprehension is essential for supporting all type of decisions related to those subjects. In the military field, the knowledge of the seafloor characteristics has two main applications: submarine warfare and seabed mines deployment. In the first case, seafloor characterization is useful because it can provide elements to build submarine nautical charts, which can assure those type of ships to land on the seafloor while minimizing risks of damage. Another application in submarine warfare is related to the fact that a better understanding of seafloor characteristics can be used to predict how different types of seafloor can affect the propagation of the acoustic signal through the ocean, especially in shallow waters. In the second case, knowledge of the type of seafloor can aid in the decision as to where would be the most suitable place to deploy submarine mines.

Modern multibeam echosounders (MBES) are able to provide two types of data: Bathymetry and Echo Intensity. Bathymetry is commonly obtained by the two-way-travel time, depression angle and azimuth of each beam through the water column. In these cases, the bathymetry data are used to locate topographical features on the seafloor and to provide information for nautical charts, which is the primary task of most hydrographic organizations; Echo Intensity is the strength of the return signal for each beam, also referred as backscatter strength, can be used to obtain information about the sediment composition and physical properties of the seabed. Due to difficulties in directly accessing the seafloor, backscatter data could be used to remotely infer information regarding the physical properties of the superficial

sediments, such as porosity and mean grain size. This theory behind the inference is based on the fact that different types of seafloor will provide a particular acoustic signature. Thus the differences in intensity levels among different seabed types can be used to validate the usage of backscatter strength. This result is a potentially powerful remote sensing tool for seafloor characterization purposes, based on the fact that areas with similar acoustic signature have similar seabed composition (Hamilton and Parnum, 2011; Rzhanov et al., 2012). This final product is a huge motivation for the ocean mapping community and that is the reason why this type of technique has quickly evolved in recent years. Due to ambiguities however, this technique still requires ground truth samples, or other previous geological information of the survey area, to derive geologically meaningful seabed maps from backscatter data (Dartnell and Gardner, 2004; Hamilton and Parnum, 2011). The accuracy of predictions of sediment properties using acoustic signals is dependent upon a series of relationship between multiple physical properties and their respective uncertainties (Hamilton, 1974).

Although the main purpose of MBES is to obtain high-resolution bathymetry, it can also provide high-resolution backscatter data with 100% coverage, which makes it an seabed classification approach as it avoids duplicating vessel costs (Preston, 2009). When backscatter imagery is used together with bathymetry data it can enhance the interpretation of fine structures of the seabed that may not necessarily have geomorphic relief, but have distinctive surface texture or roughness (Lamarche et al., 2011). Most modern MBES provide information about the peak average backscattered intensity within each beam. To reduce these values to a measure of the bottom backscatter strength, the system must account for (Hughes Clarke et al., 2008): sonar source levels, pulse lengths and receiver sensitivity; beam patterns of the transmit and receive

arrays; local seabed slopes; applied real-time varying gains; spherical spreading; and attenuation coefficients at the operating frequency.

To better understand what happens with an acoustic wave when it interacts with the seafloor, the first step is to define what backscatter is. In a monostatic system, backscatter is defined as being the part of the scattered energy which is recorded by the receiver. Although backscatter is often very small if compared to the incident intensity, this returned echo is still detectable and measureable and represents the fundamental working principle for MBES and side-scan sonar (SSS). The scattering term is mostly associated with the reflection process where the roughness and the heterogeneity of the seafloor are important. Thus results in a redistribution of acoustic energy over angles other than the incoming sound (Jackson and Richardson, 2007). Regarding seafloor roughness, it is important to highlight that its measurable characteristics are a function of the wavelength of the acoustic signal. At high frequencies, all seafloor have substantial irregularities on the scale of the acoustic wavelength, thus the scattering is omnipresent, but higher frequency signals are rapidly attenuated in water and that attenuation critically compromises the range of the signal. One of the aims of this thesis is to analyze how different wavelengths can be used as an additional classifier when seafloor characterization comes into play. This study is mainly based on the analysis of the backscatter responses from different types of seabed, ensonified with the same sonar system, where all settings configurations were kept the same during the entire acquisition process, except the operating frequency. Based on the fact that only the sonar frequency had changed, the analysis of the data can be made under two different research questions:

i. First is to analyze how backscatter response can change if the same area is ensonified with different frequencies. Depending on the results, one of the

research questions that may be answered is: Is there a most suitable frequency that may be used to ensonify a certain type of substrate or do they all present similar backscatter response?;

ii. The second matter, which is more related to the seafloor characterization perspective, is to analyze how backscatter intensity level can vary depending on the type of the sediment that is being analyzed and on the frequency that is being used. Based on that, another research question that may be answered is: What are the average intensity levels that can be associated to each type of substrate?

CHAPTER 1

BACKGROUND

1.1 The Sonar Equation

In order to interpret the acoustic information that derives from the interaction of an acoustic pressure wave with a certain patch of the seafloor, any artifacts generated by the sonar system or by the environment must be removed from the recorded data. The Sonar Equation (Equation 1) represents the total performance of the system (Augustin and Lurton, 2005; Lurton, 2010) and can be used to understand all the different factors involved in the backscatter data acquisition process. This equation, where all terms are expressed in dB, has been proven to be an important tool to evaluate the contribution of the energy parcels associated to the underwater acoustic wave propagation such as transmission, propagation, target interaction, noise and processing (Weber and Lurton, 2015).

$$SNR = SL - 2TL + BTS + (DI-NL) \text{ or} \qquad 1$$

$$SNR = EL + (DI-NL)$$

Where

SNR: Signal to Noise Ratio

SL: Source Level referenced

TL: Transmission Loss

BTS: Bottom Target Strength

DI: Directivity Index

NL: Noise Level

EL: Echo Level

SL represents transmit power and is referenced to the intensity due to 1 μPa at 1 m from the acoustic source. For the specific case of this work, SL was assumed constant and its real value was not con

TL represents the losses due to spherical spreading and absorption (Equation 2). A monostatic system was used to acquire the data presented in this thesis work. This means that the transmitter (projector) and receiver were mounted together at the same location and that made it necessary to compute *TL* twice, as showed in Equation 1, because it is necessary to consider the losses involved on both paths: from the projector to the seafloor; and from the seafloor to the receiver. *TL* is given by:

$$TL = 20\, log_{10}\, R + \alpha\, R \qquad 2$$

Where

R: Range to target (or seafloor) [m]

α: Absorption coefficient of the water column [dB m^{-1}]

Acoustic signal in the ocean is attenuated by two main mechanisms: chemical relaxation, primarily due to boric acid at low frequency (up to 1 kHz, approximately), and magnesium sulphate (from 1 kHz to 100 kHz, approximately); and viscous absorption, which is significant at frequencies higher than 100 kHz (Ainslie and Mccolm, 1998). According to the Francois-

Garrison absorption algorithm (Francois and Garrison, 1982a, 1982b), seawater temperature is the dominant factor for the α computation, but it will also depends on frequency, salinity, pH and pressure. Those factors will dictate the magnitude of the attenuation mechanisms listed above. Moderns MBES are capable of operating with multiple swath sectors, which helps to improve the system capabilities to meet IHO specifications for depth and positional accuracy (International Hydrographic Organization, 2008). To operate multiple sectors requires that each sector operates at a different center frequency. The usage of high frequency systems is always preferable to get a better resolution of the seafloor macro roughness scales with respect to the beam footprint. Increments in frequency, however, makes α larger and that limits the maximum depth over which higher frequency systems can be used. The absorption coefficient decreases with temperature at frequencies above 500 kHz, where absorption is dominated by water viscosity (Jackson and Richardson, 2007). Figure 1 shows α computed by Francois-Garrison absorption algorithm for seawater temperature of 0º, 10º and 20º C; for frequency varying from 100 to 500 kHz; at 0-meter depth; for pH of 8; and for salinity of 30, 31.5 and 35 ppt. The salinity of 31.5 ppt was chosen because that is the average value for salinity within the survey area.

This model has an associated uncertainty of 5% and its application is limited to parameters showed below (Francois and Garrison, 1982a, 1982b):

i. -2º C< Seawater Temperature < 22º C;

ii. 30 ppt < Salinity < 35 ppt; and

iii. 0 m < Depth < 3.5 km.

An inaccurate usage of the attenuation coefficient has no effect on bathymetry, but it can drastically compromise the utility of the backscatter strength for seafloor characterization,

particularly if the usage of different sonar frequencies is used as an additional classification tool. Thus, for most accurate results in multi-sectors systems, it is advisable to use different attenuations for all of the sectors, based on their center frequencies (Carvalho et al., 2013).

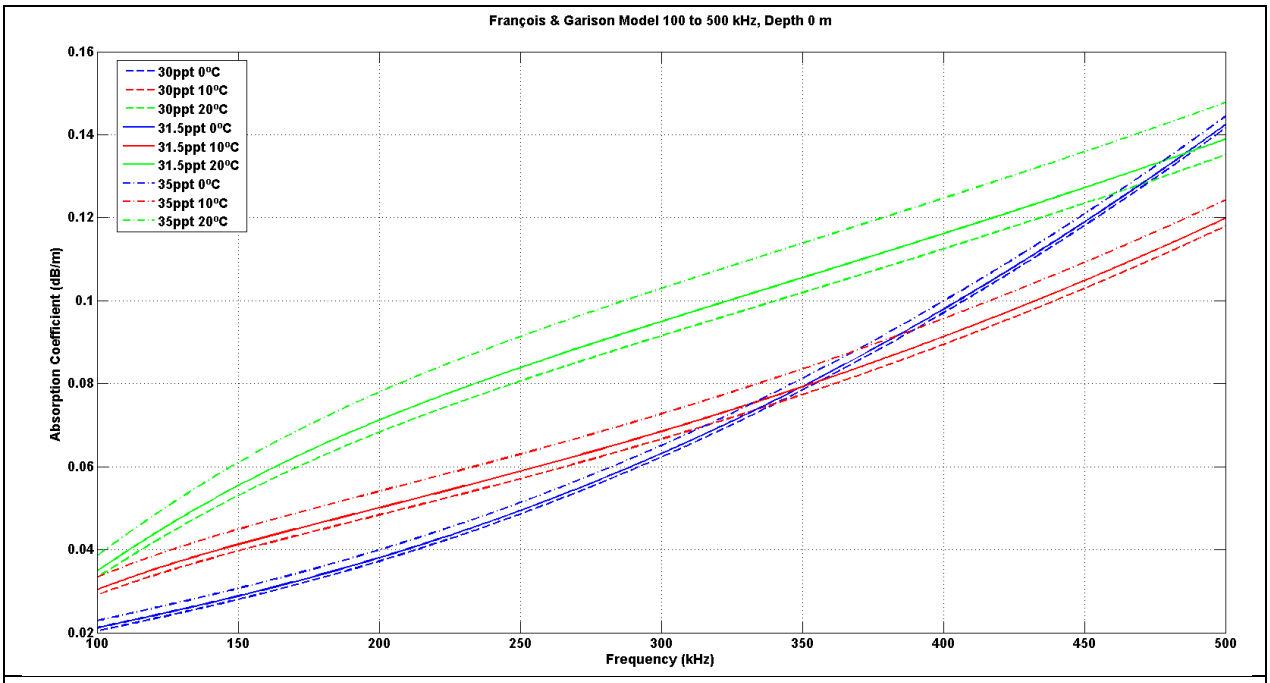


Figure 1. Seawater absorption: for seawater temperature of 0º, 10º and 20º C; for frequency varying from 100 to 500 kHz; at 0-meter depth and pH = 8, resulted from Francois-Garrison absorption algorithm for salinity of 30, 31.5 (salinity at the survey area) and 35 ppt (standard ocean).

BTS will depend on the reflectivity characteristic of the seafloor, which is related to its composition and to its fine-scale geometry (Weber and Lurton, 2015). Additionally, it will also depend on the extent of the ensonified area which contributes to the backscattered signal at any time. Equation 3 shows the *BTS* components (Weber and Lurton, 2015).

$$BTS = 10log(\sigma A) = BS + 10logA \qquad 3$$

Where

σ: Scattering cross section [dimensionless]

A: Ensonified area on the seafloor [m^2]

BS: Bottom scattering strength [dB]

The *BS* term is related to the bottom scattering and reflectivity and these factors make this term the most important for seafloor characterization purposes. It is a combination of effects related to the sonar, e.g. frequency and angular orientation, and to the target, e.g. density, sound speed, interface roughness and heterogeneities within the sediment volume (Rzhanov et al., 2012).

The bottom backscatter strength of the seafloor has a major impact on the strength of the return signal. There are, however, many other factors, such as angle of incidence on the seabed, ensonified area of the beam footprint, pulse length, transmitted power, frequency, beam width, range computation, receiver sensitivity, wavelength of the acoustic wave, absorption coefficient of the water column and ambient noise which can also affect the acoustic level of the recorded signal and must be taken into account.

Backscatter is a physical random process (Jackson and Richardson, 2007; Weber and Lurton, 2015). Acoustic waves are scattered by irregularities in the seafloor, spatial variations in sediment physical properties and also by discrete inclusions, such as shell pieces and bubbles (Jackson and Richardson, 2007). Although backscatter is random, the scattering cross section term, showed in Equation 3, is not. The term σ is defined as a statistical average different from the instant backscatter intensity itself, which is given by ping-to-ping fluctuations. Equation 4 shows that the mean-square pressure fluctuation near nadir is proportional to the ensonified area,

to the squared incident pressure, and to the scattering cross section and inversely proportional to the square of the distance from the ensonified area (Jackson and Richardson, 2007).

$$\langle |P_s|^2 \rangle = |P_i|^2 A\sigma \frac{1}{R^2} \quad (4)$$

Where

$\langle |P_s|^2 \rangle$: Mean square pressure fluctuation

$|P_i|^2$: Squared incident pressure

R^2: Squared distance to the ensonified patch of the seafloor

In a monostatic system, σ will depend on two different angular variables: incident (θ) and azimuth angles (ϕ). For this case, the incident angle for the incident and scattered fields are the same ($\theta_i = \theta_s$) and the azimuth angle for the incident and scattered field are lagged by 180º ($\phi_i = \phi_s + \pi$). ϕ is only needed if the random seafloor has a preferred roughness orientation (Lurton, 2010). So considering an isotropic seafloor, σ will be only a function of θ and then *BS* can be written as:

$$BS(\theta) = 10\log \sigma(\theta) \quad (5)$$

Equations 6 and 7 below are a first good estimation for $BS(\theta)$, which is called Lambert's Law. The Lambertian model is an approximation for the scattering process itself, and can be used to give a general sense of the angular dependence intrinsic to the backscatter data at oblique incident angles. For slightly rough surfaces this model is restricted only for oblique incidences, but it can be used in any incident angle on a very rough surface (Lurton, 2010).

Considering $\theta_i = \theta_s$, σ can be defined as (Lurton, 2010):

$$\sigma(\theta) = A_1 \frac{\eta_0}{\pi} cos^2\theta_i \quad (6)$$

Where

A_1: Unit area (equal to $1m^2$)

η_0 : Ratio between the scattered and the incident acoustic power at normal incidence. Considering a perfectly reflecting interface leads to $\eta_0 = 1$

Then, Equation 5 can be re-written as (Hammerstad, 2000):

$$BS(\theta) = BS_o + 20\log \cos \theta \quad (7)$$

Where

BS_o: Mean backscatter coefficient

In practice, BS_o can vary from a range between -10 dB and -40 dB, according to the sediment type (Lurton, 2010). The typical values for BS_0 are: gravel and rock: -15 dB; sand: -22 dB; and clay-mud: -29 dB (Gensane, 1989).

The ensonified area, *A*, for a flat seafloor, will be a function of the along and across-track beamwidth, at normal incidence, while in other incident angles it will be a function of the along-track beamwidth and to the transmitted pulse length. If the local seafloor slope is considered, the along and across-track slopes need to be added to the model used to compute the ensonified area. Accordingly to backscatter experiments conducted with frequencies from 10-100 kHz (Jackson et al., 1986), in soft sediments, sediment volume scattering is more important than surface roughness, except near normal incidence and for incidence angles smaller than the critical angle. According to the same study, in sand bottom, the surface roughness scattering is more important

than the sediment volume scattering in any case. For the majority of seafloor types, the seafloor roughness dominates near normal incidence (Jackson et al., 1986). The attenuation coefficient in the sediment increases with frequency and values on the order of magnitude of 1-100 dBm^{-1} are typical in the 10-100 kHz frequency range (Jackson et al., 1986). This thesis is based on backscatter studies using frequencies from 200-400 kHz, whose respective acoustic signals are expected to be highly attenuated during their travel through the water column and, intrinsically, have a limited acoustic penetration into the seafloor. A typical penetration depth for a 200 kHz acoustic signal is 0.1 meter (Preston, 2006). Therefore, with respect to the range of frequencies used in this thesis, it is reasonable to consider that the scattering phenomenon is mostly related to the superficial sediments within the ensonified area, because sound attenuation at high frequencies limits penetration of the acoustic wave into the seafloor. This also means that the usage of the roughness scattering instead of the volume scattering strength is appropriate in these cases, as the contribution of sediment volume and underlying rocks for the final *BS* value is small when compared to interface characteristics. Another important aspect that endorses this hypothesis is the fact that the sediment distribution within the thesis' survey area is mainly composed of hard substrate types. Equations 8 and 9 shows how *BTS* is computed as a function of the incident angle θ (Hammerstad, 2000).

$$BTS = BS(\theta) + 10log\psi_x\psi_y R^2, \qquad for\ \theta = 0^o \tag{8}$$

$$BTS = BS(\theta) + 10log\frac{c\tau}{2sin\theta}\,\psi_x R \quad ,\ for\ \theta > 0^o \tag{9}$$

Where

ψ_x: Along-track beamwidth [rad]

ψ_y: Across-track beamwidth [rad]

R: Range to target (or seafloor) [m]

c: Sound speed in water [m/s]

τ: Pulse length [s]

θ: Incident angle [rad]

Beams are considered to be at normal incidence mode if the incident angle is smaller than the limit angle (θ_{lim}), which is largest angle given by Equations 10 and 11. Otherwise, oblique incidence must be considered (Hellequin et al., 2003; Llewellyn, 2006). Note that Equations 8 and 9 do not account for non-normal incidence areas where the ensonified area increases in the across track direction, which is true when the incident angle is $0^o \leq \theta \leq \theta_{lim}$.

Equations 8 and 9 are valid if the seafloor is assumed to be flat and the incidence angle is measured as the angle between the beam vector and the vertical in the across track. Although the true incidence angle will depend on the seafloor inclination, the flat seafloor assumptions are always made by Kongsberg sonar systems (Llewellyn, 2006; Teng, 2012).

$$\theta_{lim} = acos[\frac{D}{D + (c\tau/_2)}] \quad (10)$$

$$\theta_{lim} = asin\left[-\frac{\psi_y D}{c\tau} + \sqrt{1 + \left(\frac{\psi_y D}{c\tau}\right)^2}\right] \quad (11)$$

The default value for the crossover angle adopted by Kongsberg for the EM 2040 MBES is 10º. These systems assume that *BS* will change linearly from incident angles varying from 0º to 10º and change accordingly to Equation 7 for angles bigger than 10º (Gensane, 1989; Hammerstad, 2000). To adjust Lambert's Law to these parameters, *BTS* can be written as shown in Equations 12, 13 and 14 (Hammerstad, 2000):

$$BTS = BS_N + 10log\psi_x\psi_y R^2 \quad for\ R \leq R_I \tag{12}$$

$$BTS = BS_0 - 5\log\left({}^{R}/_{R_I}\right)^2\left[\left({}^{R}/_{R_I}\right)^2 - 1\right] + 10\log\frac{c\tau}{2}\psi_x R \quad for\ R \geq 1.015R_I \tag{13}$$

$$BTS = BS_N + 3.162\sqrt{\frac{R}{R_I - 1}}(BS_o - BS_N) - 5\log\left({}^{R}/_{R_I}\right)^2\left[\left({}^{R}/_{R_I}\right)^2 - 1\right] + 10\log\frac{c\tau}{2}\psi_x R \quad for\ R_I < R \leq 1.015R_I \tag{14}$$

Where

BS_N: *BS* at normal incidence ($\theta = 0$) [dB]

BS_o: *BS* in at oblique incidence [dB]

R: Range to the seafloor [m]

R_I: Range to normal incidence [m]

For an incident angle of 10º, the range to the target *R* is given by: ${}^{R_I}/_{\cos 10^o} = 1.015R_I$. Equation 12 handles the signal recorded before the first arrival. In principal, no seabed signal should be presented at this point, but this accounts for an incorrect estimate of R_I. Equation 13 represents the Lambert zone (incidence angle greater than 10º) and Equation 14 the signal within the linear near-nadir zone (0 to 10º incidence). Although the Kongsberg algorithm assumes a fixed value for the crossover angle, it is important to highlight that the crossover angle may vary according to the sediment type and it can be observed anywhere in the 5-30º zone (Hammerstad, 2000). In order to undo the Kongsberg assumptions, BS_N, BS_O, R_I and crossover angle values must be extracted from Kongsberg raw files (.all format).

The acoustic wave can interact with the seafloor in two different ways, referred to as normal and oblique incidence regimes (Figure 2). It is considered normal incidence regime when the pulse length is long enough that the entire beam footprint can be ensonified at once ($\theta < \theta_{lim}$). In the oblique incidence regime ($\theta > \theta_{lim}$), the pulse length is short enough that only an annulus defined by the pulse length can be ensonified at once (Weber and Lurton, 2015).

The models used to compute the ensonified area for the near-nadir region (Equation 15) and for the oblique regions (Equation 16), taking into account the along and across-track slope of the seafloor (Malik et al., 2015), are the following:

$$Area_{near-nadir} = \frac{\psi_x \psi_y R^2}{\cos(\beta_x) \cos(\beta_y)} \quad 15$$

$$Area_{oblique} = \frac{c\tau}{2\cos(\theta - \beta_y) \cos(\beta_x)} \psi_x R \quad 16$$

Where

β_x: Along-track slope [rad]

β_y: Across-track slope [rad]

The local seafloor slope must always be considered for the grazing angle computation to avoid *BS* variations that are not necessarily related to changes in sediment type.

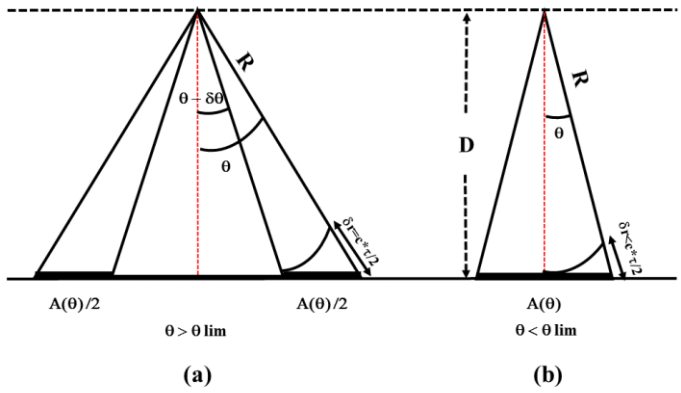


Figure 2. (a) Short pulse regime or oblique transmission case ($\boldsymbol{\theta > \theta_{lim}}$); (b) Long pulse regime or near nadir-case ($\boldsymbol{\theta < \theta_{lim}}$). In this case, the pulse length (τ) is greater than $R(\theta) - D$.

From Equation 1, noise can be defined as being any part of the recorded signal that is undesirable. The sonar performance can be enhanced if the *NL* is reduced. *NL* is considered to be isotropic and can be minimized by using the knowledge of the power spectra of the ambient noise and of the target to select the bandwidth of the system (Kinsler et al., 2000). The ambient noise can be caused by a variety of phenomena. Thermal agitation, bubbles, water droplets, turbulence, oceanic traffic, seismic and biological activities are some of the noise sources (Wenz, 1962). Molecular agitation becomes an important noise source above 50 kHz, when the levels of wind and other sources of surface-generated sound are very low, and the noise spectrum increases at 6 dB/octave (APL-UW, 1994; Kinsler et al., 2000).

It is known that the receiver is tuned to filter out noise outside the nominal band of the pulse. The *NL* that covers the entire receiver band of the echosounder (*W*) is shown in Equation 17, where $NL_{1\ Hz}$ is NL that accounts only for a nominal band of 1 Hz ($NL_{1\ Hz}$).

$$NL = NL_{1Hz} + 10logW \qquad 17$$

Self-noise is the type of noise generated by the receiving platform itself or by the sonar system's own electronics. Self-noise usually tends to increase with increasing platform speed, and at low speeds, self-noise is usually less important than ambient noise (Kinsler et al., 2000; Lurton, 2010). Reverberation is a type of noise which can be observed in monostatic systems that, sometimes, can be loud enough to mask the detection of expected target echoes. Reverberation happens when the acoustic wave scatters from objects other than the target of interest. *SNR* can be limited by ambient noise or reverberation (Weber, 2015). Reverberation and ambient noise will always be part of the signal recorded, but generally, low-power systems are more affected by ambient noise and high-power systems by reverberation.

NL can be minimized if the echosounder is able to be sensitive only to acoustic waves that come from a particular direction. This ability is called *DI* (Equation 1). So, if only the ambient noise that comes from a particular direction will be considered during the data recording process, it means that the impact of the ambient noise in the record data will be lower down if compared with another system whose *DI* is not present (Weber, 2015). For a linear array, *DI* can be numerically defined as the inverse ratio of the directivity pattern integrated over the entire space and the solid angle 4π of an array without directivity (Lurton, 2010). *DI*, in dB, for an unshaded array, is showed in Equation 18:

$$DI = 10\log\left(\frac{2L}{\lambda}\right) \tag{18}$$

Where

L : Length of the acoustic linear array

λ : Wavelength of the transmitted acoustic signal

Although both *NL* and *DI* affect the *BTS* computation (Equation 1), for the specific case of this work, those two variables were not considered during *BTS* computation processes.

1.2 Interactions between the acoustic wave and the seafloor

In a hypothetical situation, where the seafloor is completely flat and homogeneous, the seafloor itself would reflect part of the acoustic wave in the specular direction and the remaining part of the energy would be transmitted into the seabed. If the acoustic system is monostatic, no part of the energy would bounce back to the receiver. The only exception to this is when the beam direction is perpendicular to the seafloor (Weber and Lurton, 2015). Analyzing the geometry involved in the reflectivity process of an underwater acoustic wave, only normal incidence can be considered specular and coherent. In these cases, the wave is reflected by the facets oriented to reflect the specular echo and the recorded energy will be the maximum. The reflected energy that is recorded by the system decreases when the grazing angle gets smaller. At oblique incidence, the backscattered wave comes from continuous sources dominated by microscale roughness (Lurton, 2010).

The intensity of the reflected signal is intrinsically related to the contrast between the characteristic impedance of the water and the seafloor, and also to the grazing angle. The characteristic impedance is given by the product of the density and the speed of sound in a certain medium, and it represents an important property of any material (Hamilton, 1970). As shown in Equation 19, the characteristic impedances of the water and the seafloor are used to compute the reflection coefficient (*RC*), which is the ratio of the reflected to incident pressure (Lurton, 2010; Weber and Lurton, 2015).

$$RC = \frac{\rho_2 c_2 cos\theta_1 - \rho_1 c_1 cos\theta_2}{\rho_2 c_2 cos\theta_1 + \rho_1 c_1 cos\theta_2} \quad 19$$

Where

ρ_1: Density of the water

c_1: Speed of sound in water

ρ_2: Density of the ensonified seafloor

c_2: Speed of sound in the ensonified seafloor

θ_1: Incident angle

θ_2: Transmitted angle

The transmitted wave propagates in a different direction, proportional to the change in sound velocity and follows the Snell-Descartes law:

$$\frac{sin\theta_1}{c_1} = \frac{sin\theta_2}{c_2} \quad 20$$

Commonly θ_2 is greater than θ_1, so there is an angle, θ_c, beyond which transmission is impossible. This angle is called critical angle. *RC* gets larger with increasing θ_1 until it reaches θ_c, at which point no compressional wave can propagate inside the seafloor. All refracted waves become horizontal when the grazing angle is smaller than the θ_c and, beyond that point, penetration of the acoustic field into the seafloor will be insignificant and signal scattered from the volume should be very small (Jackson et al., 1986; Fonseca and Mayer, 2007; Hamilton and Parnum, 2011).

Figure 3 shows some bottom acoustic scattering mechanisms. High porosity sediments, such as silts and clays, have a little acoustic impedance contrast in comparison with the

overlaying water and, as consequence, the seafloor and water interface may not be a strong scatterer of acoustic energy (Hamilton, 1970). Except for very high-frequency systems, this lack of characteristic impedance contrast allows the acoustic energy to penetrate deeper into those soft sediments than into sand-type seafloor, and this means that the backscatter signal from those soft-type bottoms are generally not related to the surface layer material, but to other harder or more compact material below the surface.

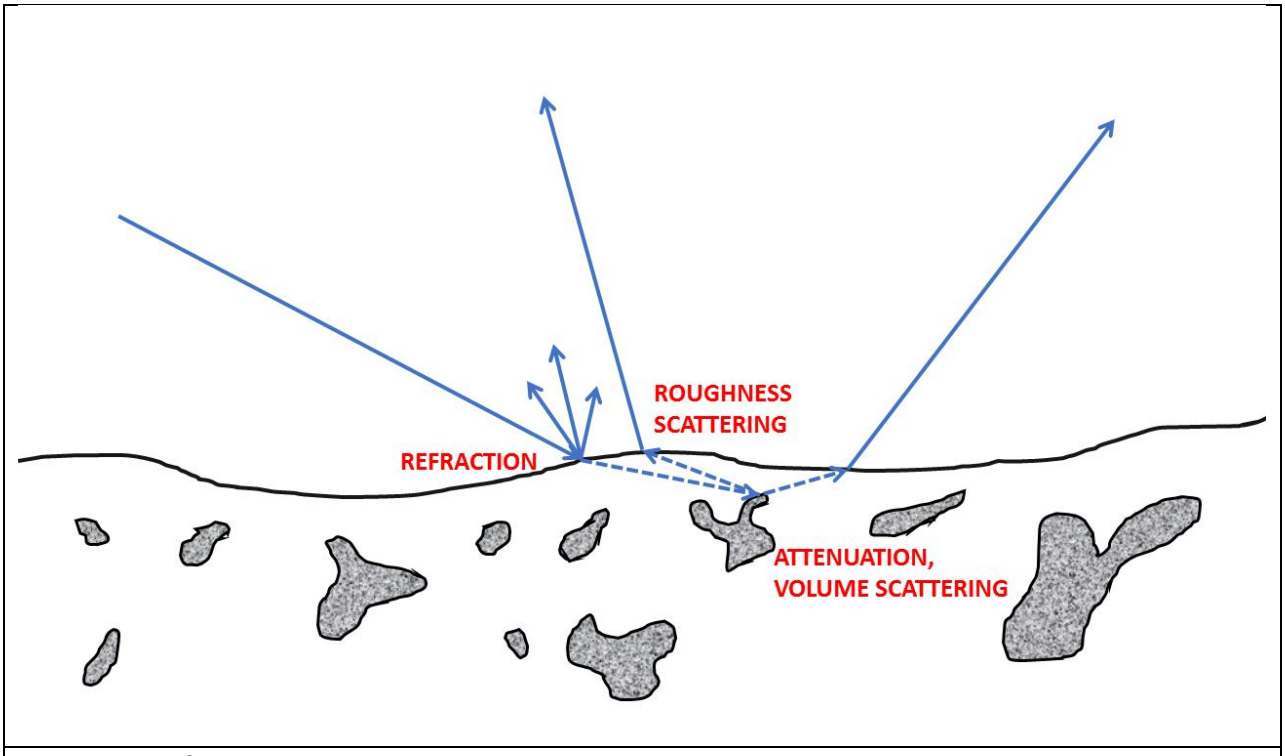


Figure 3. Refraction and scattering at the water-bottom boundary and attenuation and scattering in the sediment.

As the roughness varies with the wavelength, the seabed scattering will be a function of the acoustic wavelength (Hughes Clarke, 2015). In other words, the same patch of the seafloor can be classified as rough or smooth depending on the acoustic frequency considered. Higher frequency signals are more sensitive to shorter scale interface roughness than lower frequency signals, because of their smaller wavelengths. Acoustic backscatter may be interpreted in terms

of acoustic roughness, which is the ratio of the standard deviation *h* of the seabed interface elevation to the acoustic wavelength (Weber and Lurton, 2015). An interface is considered to be smooth if $h < \lambda$ and, in those cases, specular reflection will dominate and the scattered field will be very low. Otherwise, when $h > \lambda$, scattering will dominate and the incident acoustic wave is scattered to all directions (Weber and Lurton, 2015).

1.3 Angle Dependence

The acoustic wavefront usually intersects the seafloor at an angle and the variation of the *BS* with the angle of incidence is an intrinsic property of the seafloor (Beokett and Hussong, 1989; de Moustier and Alexandrou, 1991; Fonseca and Mayer, 2007; Hasan et al., 2012, 2014). If the purpose of a certain survey is to use the backscatter data to estimate the properties of the seafloor, remotely, then the incident angle at the seafloor must be considered during the backscatter analysis. The strongest values for *EL* are expected at the normal incidence region, where the specular reflection phenomenon takes over. When a certain seafloor patch is basically composed of smaller grain-size sediments, such as clay and silt, it tends to have low roughness and, in cases like these, the influence of the specular reflection decreases very quickly with decreasing grazing angle (Weber and Lurton, 2015).

As a general rule, at oblique incidence angle, the angular dependence is small and the *BS* is primarily controlled by impedance contrast and roughness (Weber and Lurton, 2015). At very low grazing angles, the backscatter response of the seafloor disappears. The scattering strength values close to the 0º-grazing angle region are very low and close to one another, which makes it difficult to distinguish different types of sediments at that region. This region is also very

sensitive to noise in the estimates of the grazing angle, particularly if shadows are being cast. The intermediate oblique grazing angles region, where the backscatter response for each type of seafloor sediment-type are more separated and approximately parallel to one another (plateau region), presents the best backscattering strength response separation among different types of seafloor composition. This characteristic makes the plateau region the most indicated region to be used for seafloor characterization purposes, because there different types of sediments can be better distinguished. Errors in interpretation can be minimized if the comparison bears only on the plateau region, instead of comparing data collected over a wide range of grazing angles, but different seabed types may provide similar response on plateau region and only be distinct at higher or lower grazing angles. In theory, for more optimized results in seafloor characterization it is recommended to restrict the analysis to grazing angles from, approximately, 30º to 75º (Weber and Lurton, 2015). In the specific case of this thesis, where multi-frequency signals are going to be used for seafloor characterization purposes, the statistical analysis of the data will be focused on the angular interval limited by grazing angles from 30º to 60º, which represents the plateau region for the data collected for this work.

1.4 The multi-frequency backscatter approach as an additional classifier

This work will analyze the backscatter data under two different perspectives: the first is to compare the backscatter strength behavior when the same frequency is used to ensonify different types of substrates; and the second approach consists of the interpretation of the backscatter strength behavior when the same type of substrate is ensonified by different frequencies. By looking at both perspectives, the thesis will focus on evaluating the potential of using the multi-frequency backscatter in aspects related to seafloor characterization. These

approaches may provide a better comprehension of the way frequency dependence may affect the *BS* values. That understanding can be useful to avoid significant errors in interpretation in analyzing a backscatter mosaic whose content is derived from different mono-frequency sonar systems. If the same patch of the seafloor is ensonified with the same MBES, but with different operating frequencies, each backscatter mosaic will present a particular dB range, based on the principle that seafloor roughness and sediment volume scattering are related to the signal frequency used. For a single frequency backscatter data, some seabed materials type may present similar backscatter strength values, which makes it difficult to discriminate one type of sediment from the other. This issue can potentially be solved when more than one frequency is used to ensonify the same seafloor area, because the backscatter strength values for certain types of seabed material may be better separated at a certain frequency than in others. In this manner one can refine the usage of the backscatter intensity response as a seafloor characterization tool. In previous studies conducted using frequencies between 10-100 kHz, the frequency dependence is little for rough seabed, but large for softer seabed (NDRC, 1946; Jackson et al., 1986; Weber and Lurton, 2015).

The frequency dependency can also be observed when scattering strength versus incident angle is analyzed. The same surface that looks smooth at 30 kHz with a high reflection close to the nadir zone and a low oblique incident scattering can provide a much more uniform backscattering strength distribution at 400 kHz (Weber and Lurton, 2015). For different frequencies, but same incident angle, the values for scattering strength may be different and those differences are usually larger at normal-incidence region (Weber and Lurton, 2015).

Mosaics made with backscatter data from different sonar systems may present different values for backscatter strength in the overlapping areas, but those differences in intensity level

are not only related to the different frequencies used during the survey, but may also be related to different assumptions made during the acquisition and processing steps, such as: absolute source level, pulse length, absolute and time varying gains, attenuation coefficient, sonar transmission and reception sensitivities. Other external factors not related to the sonar, such as different software and hardware used during the data processing, and also how fast seafloor dynamics itself affects the distribution of sediments along the seabed, may also affect the backscatter mosaic (Hughes Clarke et al., 2008). The uncertainty or lack of information about all those factors cited above will certainly contribute to a final mosaic where the differences in backscatter intensity are not actually related to the differences in seabed composition or to the different signal frequencies. Rather the intensity level in the areas where different sonar systems data overlap to one another are often inconsistent, as in the case shown in Figure 4. They are considered inconsistent because at the overlapping area those intensity level were expected to be the same, independently of the acoustic system used during the acquisition, when the backscatter information comes from the same type of sediment.

Note that even after initial adjustments, the backscatter mosaic still presents intensity level contrasts among different surveys, as also showed in Figure 4. The cause for those residual contrasts, which are not related to geological variation, can be attributed to uncertainties associated to the calibration of each system and also to other adjustments made during the processing stage.

To reduce to a minimum level all the unwanted factors that may contribute to an intensity level inconsistency, the dataset used in this research was collected by the same sonar system, where all the acquisition parameters were kept the same during the entire survey (only the signal frequency was switched); the same hardware and software were used during the data acquisition

and processing; and the entire dataset was acquired on the same day during an 5-hour survey, which allows the assumption that the sediments distribution, environmental parameters and meteorological conditions remained essentially the same throughout the survey.

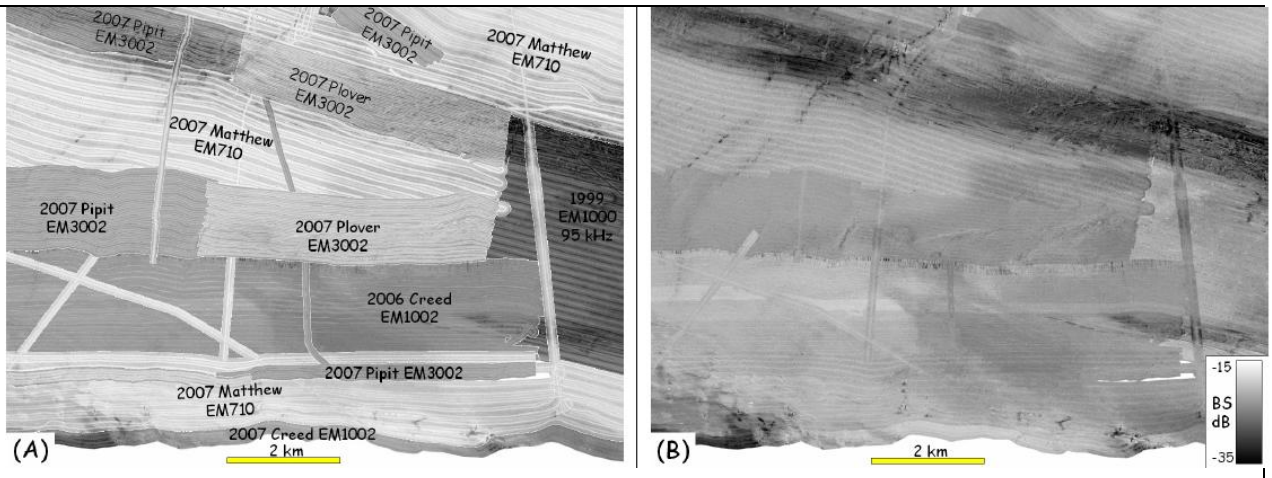


Figure 4. Backscatter mapping in Bay of Fundy, Canada, from 1999 to 2007, using different survey platforms and Kongsberg Maritime MBES. A: Mosaic of backscatter data as acquired. Years, name of the survey platforms and Kongsberg Maritime MBES are indicated over each surveyed area. B: Data adjusted to blend the backscatter data from different sources. Image derived from (Hughes Clarke et al., 2008)

All these precautions reinforce the hypothesis that the differences in *BS* levels in the dataset, when the same patch of the seabed is analyzed, would be driven just by the usage of different frequencies during the acquisition. The analysis of the usage of different frequencies to ensonify the same patch of the seabed might answer some questions related to seafloor characterization, for example: What would be the recommended frequency with which to characterize a certain type of sediment? For a certain frequency, what is the expected *BS* value for a certain type of seabed? Within the survey area, which is the frequency that presents a better *BS* separation among different types of substrates? All methods and hypotheses used to answer those research questions will be presented in the succeeding sections of this thesis.

1.5 Beam Pattern Uncertainty

The algorithm used by Kongsberg MBES, presented by Equations 12, 13 and 14 shows how *BTS* is computed as a function of varying angle. This model is necessary because the angular response curves are different for different types of sediments (Fonseca and Mayer, 2007; Hasan et al., 2012) and, since those curves often change, even if the same MBES is used, the values for BS_O and BS_N, computed by the Kongsberg sonar to best approximate the response curve, are based on the results of preceding pings (Hammerstad, 2000; Llewellyn, 2006). Figure 5 shows a hypothetical result of the application of the angular response model (Equations 12, 13 and 14) applied to response curves, for different types of seabed.

Kongsberg Maritime MBES have a designed array directivity on both transmission and reception. For the MBES model used for acquiring data for this thesis, an EM 2040, those array directivities are sonar-referenced beam patterns. Ideally, each MBES transmission would not vary its *SL* over the full range of pointing angles, but in reality beam pattern oscillations can influence the *BS* values computed by the Kongsberg standard algorithm, cited in the previous paragraph (Hughes Clarke, 2015). The conception of this model is to be able to remove any influence that the grazing angle can have over the *BS* levels, as shown in Figure 5 (A). After this compensation, as a theoretical result, a constant value for *BS* in any grazing angle should be expected, but in reality that does not happen, as shown in Figure 5 (C) where, even after that reduction, the *BS* projected on the seafloor, as a function of angle, may still present some residual variations. Those oscillations will be a contribution of two factors:

i. beam pattern residuals; and
ii. departures of the angular response curves from the Kongsberg model.

The curves in Figure 5 (A) should be derived from an area of homogeneous seabed type, so that, as much as possible, the *BS* response to the incidence angle and the hypothetical residuals showed in Figure 5 (C) would be unique and strictly related to a certain hardware configuration and angular response curve departure from Kongsberg model.

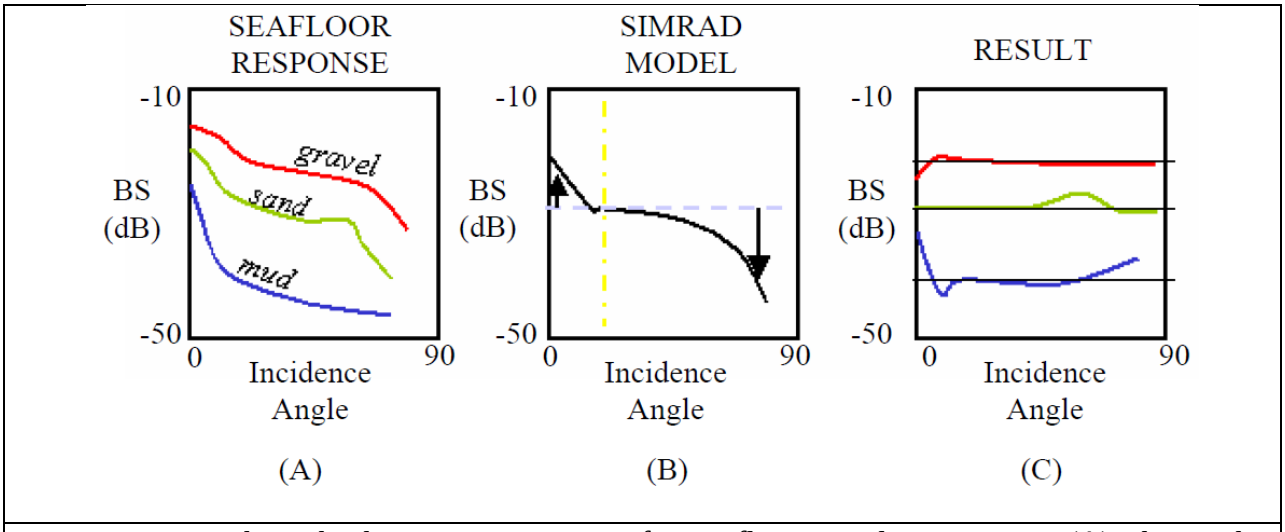


Figure 5. Kongsberg backscatter corrections for seafloor angular response. (A) shows the theoretical angular response curve for gravel, sand and mud. (B) shows the Kongsberg algorithm that is derived from Equations 12, 13 and 14. (C) shows the resulting adjusted angular response curves that highlighted how the true angular response curves depart from the Kongsberg approximation. Image derived from (Hughes Clarke, 2005).

Those fluctuations from the model are usually small, but still represents a significant fluctuation in acoustic intensity for which the signal must be corrected, because even small variations can limit the ability to distinguish different seafloor types. The beam pattern correction is very effective to remove fluctuations of intensity unrelated to seabed type and grazing angle variation (Hughes Clarke, 2015). These effects can be caused by differences in power amplification between each of the roll-stabilized beam forming channels or by changes in physical structure of the transducer elements (Llewellyn, 2006). Modern MBES uses multi-sector and multi-swath techniques to optimize the ensonification of the seafloor. As a result, if the beam pattern correction is not applied, individually to each sector, the apparent seafloor

backscatter can be affected by intensity oscillations that are not necessarily related to changes in seafloor characteristics (Hughes Clarke et al., 2008; Teng, 2012).

The approach that may be used to implement the beam pattern correction can be made by using backscatter data collected from an area where the seabed surface is assumed to be Lambertian (Hughes Clarke, 2015). From those signals, the sonar-relative elevation angle and the associated relative *BS* are computed, for each beam, and the results are compiled in the sonar reference frame as shown in Figure 6.

These reference residual patterns must be removed from the data based on the orientation of each one of the beams with respect to the transmitter at the transmit time (Hughes Clarke, 2015). Once this step is done, the result expected is similar to the one shown in Figure 5 (A), where *BS* as a function of the incident angle will result in the true angular response curve, whose *BS* levels may vary based just on the sediment composition.

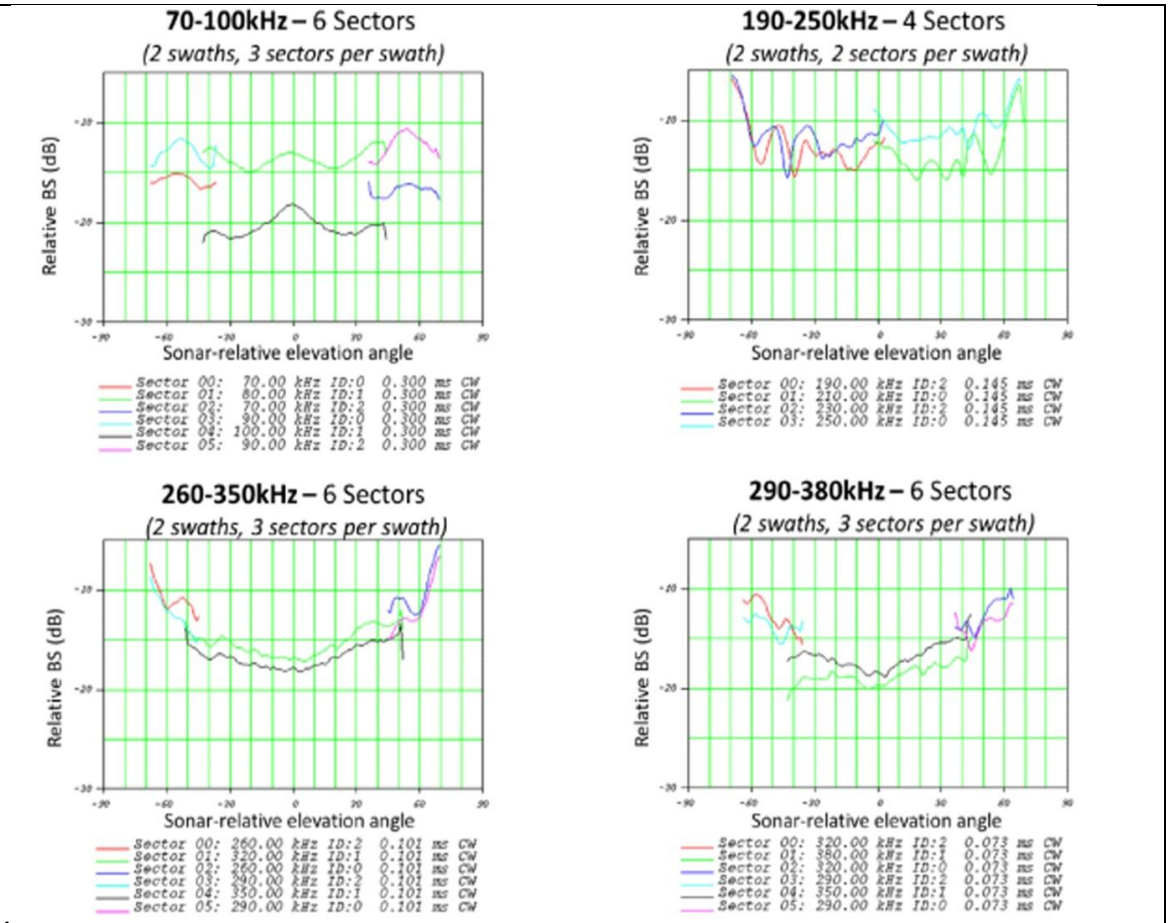


Figure 6. Residual transmit sector beam patterns for each sectors used in different frequency ranges. Image derived from (Hughes Clarke, 2015).

CHAPTER 2

DATA

2.1 Vessel and Equipment

All the equipment used to collect the data to be used in this research were installed onboard the R/V Coastal Surveyor (Figure 7). This vessel is owned and operated by the Center for Coastal and Ocean Mapping/ Joint Hydrographic Center and is ideally designed for near-shore and shallow water operations. The vessel specifications are listed in Table 1.

Figure 7. R/V Coastal Surveyor docked at UNH Judd Gregg Marine Research Complex, in New Castle, New Hampshire.

R/V Specifications	
Flag	United States of America
Dimensions	12.2 m (length) x 3.6 m (beam) x 1.13 m (draft)
Tonnage	16 GRT; 11 DWT
Lab space	2.8 m x 3.4 m; 1.9 m x 3.1 m
Top speed	10 knots
Minimum speed for full roll stabilization	5 knots
Minimum survey speed	2.5 knots
Propulsion	1 x Caterpillar 3116; 200HP Marine Diesel; 2.57:1 reduction
Shipboard Power	12KW Northern Lights Diesel Generator, 240/120v, 60Hz
Roll stabilization	Niad active fins
Table 1. R/V Coastal Surveyor specifications. Credits: http://ccom.unh.edu/facilities/research-vessels/rv-coastal-surveyor. Accessed on January 22nd, 2015.	

As mentioned above, an EM 2040 model was used to acquire the bathymetry and backscatter data to be used in this thesis. The EM 2040 represents the latest generation of MBES which are designed to operate in shallow water surveys, where high resolution data is commonly required. Figure 8 shows the installation of the MBES on board R/V Coastal Surveyor.

This MBES can operate in 3 different frequencies centered in 200 kHz, 300 kHz and 400 kHz. The echosounder can only be operated within a single frequency band at one time. A pulse length of 200 μs was chosen when the data were being logged with the 200 and 300 kHz frequencies. For the data recorded with 400 kHz, the pulse lengths used were 100 and 200 μs. The pulse length parameter were kept fixed while each survey line was being logged, to avoid automatic changes in pulse driven by changes in depth.

For the specific single swath EM2040 model used for this thesis, the 200 kHz mode has two sectors, whose frequencies are centered in 190 and 205 kHz; the 300 kHz has three sectors, whose frequencies are centered in 270, 282.5 and 295 kHz; and the 400 kHz mode has three sectors, whose frequencies are centered in 320 kHz (two lateral sectors) and 380 kHz (one central sector). Table 2 lists some of the EM 2040 technical specifications.

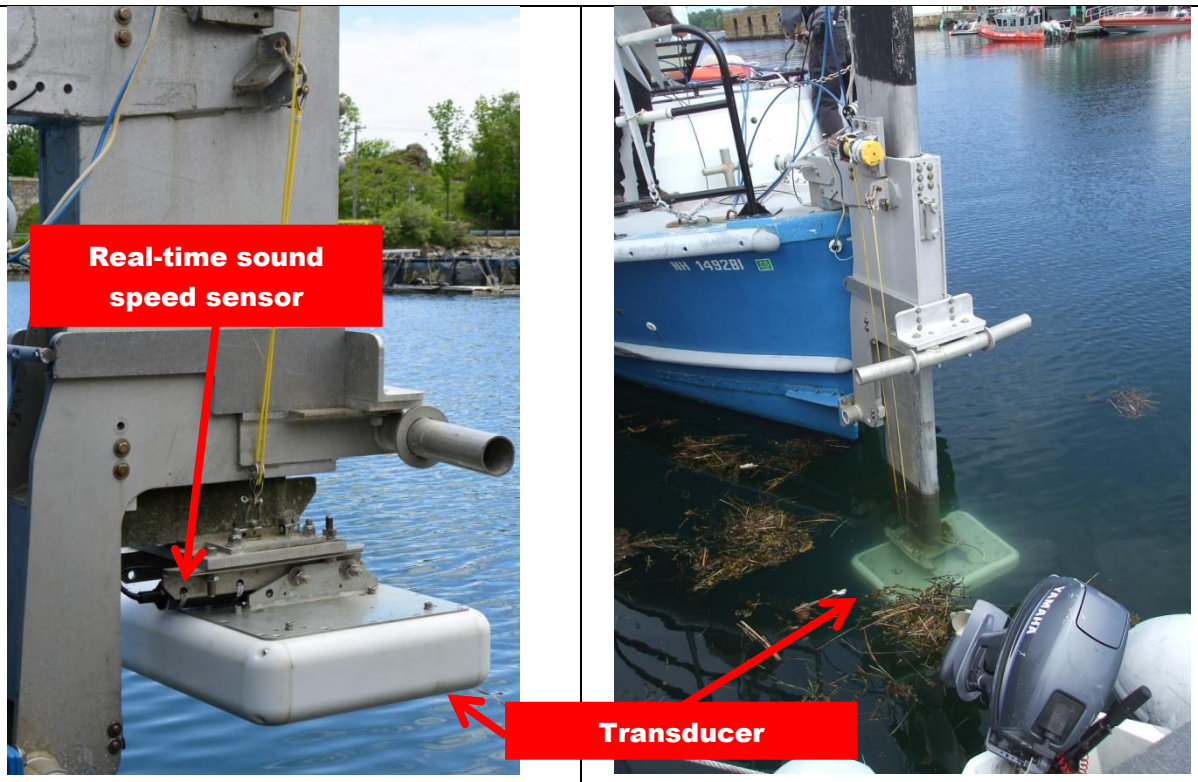


Figure 8. EM 2040 installed onboard R/V Coastal Surveyor. The EM 2040 transducer installed in a pole attached to the bow of the vessel.

Technical Specifications	
Roll stabilized beams	Yes (±15º)
Pitch stabilized beams	Yes (±10º)
Yaw stabilized beams	Yes (±10º)
Swath coverage sector (single receiver)	Up to 140º
Pulse Lengths (CW)	70, 200 and 600 μs for 200 and 300 kHz 50, 100, 200 μs for 400 kHz
Max no. of soundings per ping (single swath, single receiver)	400
Beamwidths (at 300 kHz)	TX: 1º and RX: 1º (300 kHz)
Max ping rate	50 Hz

Table 2. Technical Specifications for EM 2040. Information extracted from (Kongsberg Maritime, 2016).

Table 3 shows a list of hardware and software used during data acquisition. To restrict the difference between different survey lines to changes in frequency and in pulse length parameters

only, the configuration of the hardware installed onboard was kept exactly the same during the entire survey.

Purpose of use	Hardware or Software	Manufacturer
RTK GPS Receiver	5700	Trimble
Positioning and Attitude	PosMV 320 with IMU 200	Applanix
Data acquisition software	Seafloor Information System (SIS)	Kongsberg Maritime
Data processing software	Caris Hips and Sips 9.0	Caris
	FMGeocoder Toolbox – Fledermaus Geocoder Toolbox, version 7.4.1, 64 bit Edition	QPS
	MATLAB R2015a	MathWorks
Sound Speed at transducer head	Smart SV&T	AML
Sound Speed Profiles	Digibar Pro	Odom

Table 3. List of hardware and software used together with the EM 2040 during the data acquisition.

2.2 Survey area

The data were collected close to the New Hampshire's coast line, as showed in Figure 9. The average depth within the survey area is about 19 meters and the size of the survey area is 0.88 Km^2, approximately. A big concern in using shallow water backscatter data is that, in contrast to the deep sea, sediment types and properties may vary significantly over short distances (Hamilton, 1974). However, for this study, a shallow water area was chosen to allow the usage of the EM 2040's available higher frequencies, without compromising the SNR.

The survey lines were planned parallel to one another in a way to guarantee a minimum overlap of 50% of between consecutive lines, considering a swath angle of 120º. In total, twelve survey lines were planned and each one of them were run with a different frequency (200, 300 or

400 kHz). All 400-kHz lines were run twice, each of them with a different pulse length (100 and 200 microseconds). Thus, in total, sixteen regular lines were run, and Figure 10 shows bathymetry surfaces derived from the backscatter data acquired in each different frequency mode.

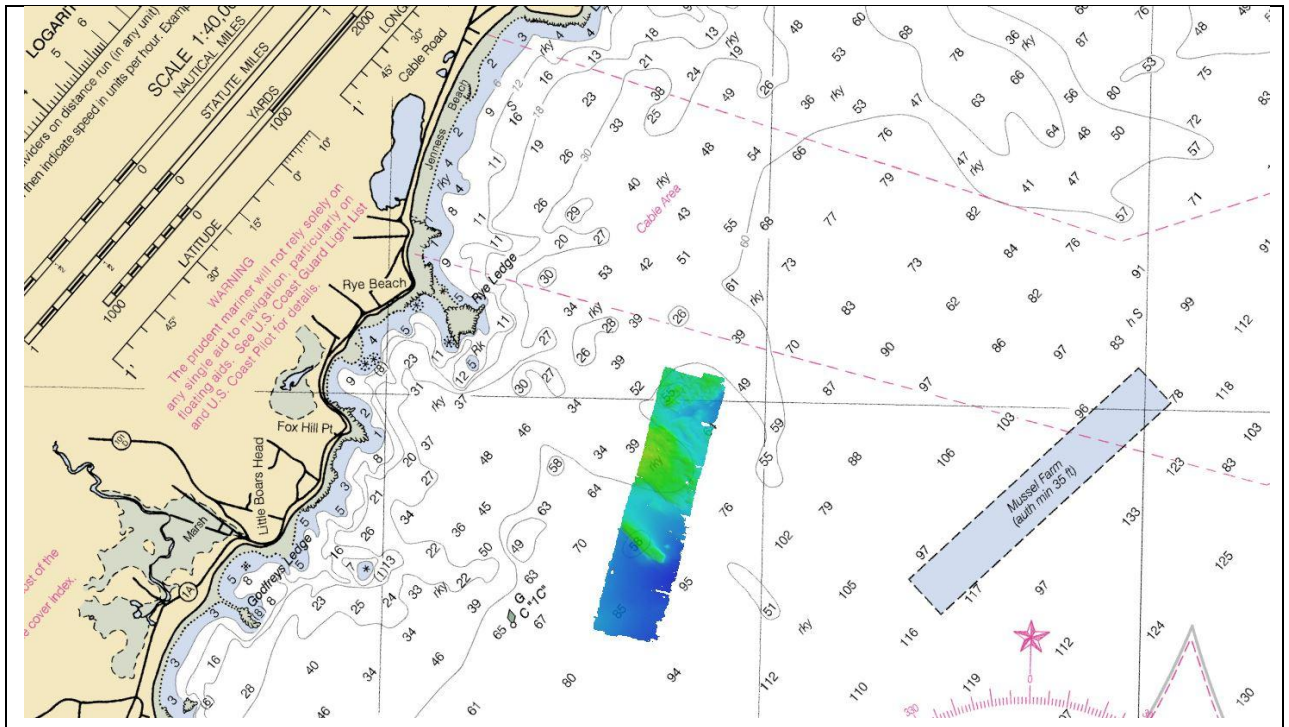


Figure 9. Survey area. Location - NW corner (UTM 19N): Latitude: 42º 58.815'N and Longitude: 070º 44.532'W; and SE corner (UTM 19 N): Latitude: 42º 57.046'N and Longitude: 070º 44.532'W. Area: 0.88 Km^2 (approximately). Average depth: 19 meters. Maximum depth: 35 meters. The center of the area is located less than 3 Km from Fox Hill Point, at New Hampshire's coastline. Nautical Chart used as background: NOAA nº 13278 - Portsmouth to Cape Ann; Hampton Harbor (scale: 1:80,000).

Crosslines were run in a zigzag pattern. The same line were run for 200 (medium pulse), 300 (medium pulse) and 400 kHz (medium and long pulse). Although a swath angle of 120º had been used for line spacing computations, the data was acquired using a swath angle of 150º, except for the 400 kHz mode where, due to system limitations related to the receiver sensitivity across-track, the data was acquired using a swath angle of only 120º.

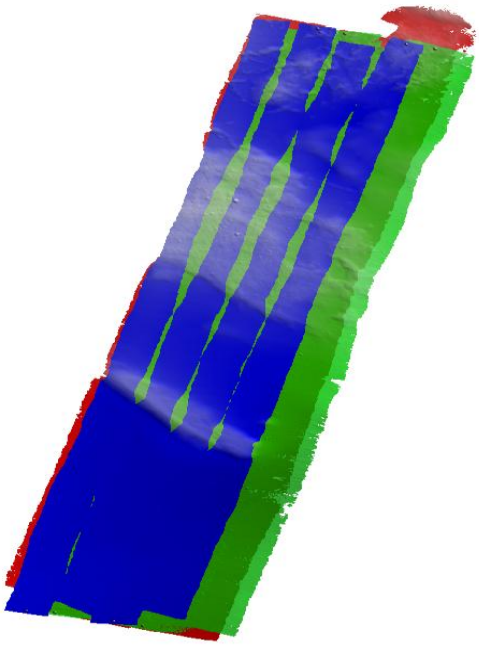


Figure 10. Regular survey lines. 400 kHz (medium and large pulses): blue; 300 kHz (medium pulse): red; and 200 kHz (medium pulse): green. Those lines were planned to achieve, at least, 50% of overlapping, considering a swath angle of 120º.

Figure 11 shows different backscatter mosaics for each one of the four different configurations of frequency and pulse lengths described above. Note that the 400 kHz mosaics, which were derived from a narrower swath angle, present more blank spaces between consecutive lines.

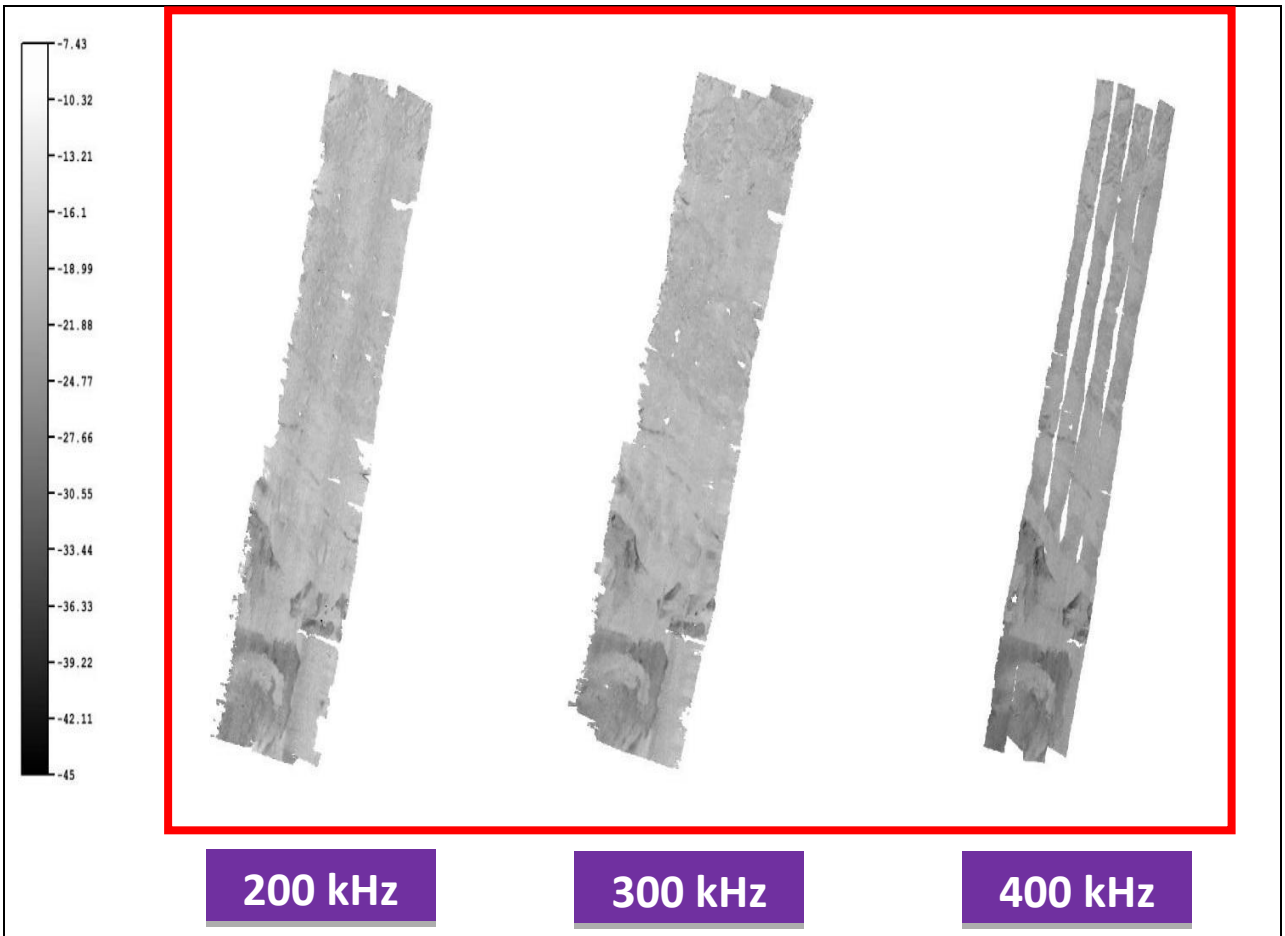


Figure 11. Backscatter mosaic for each frequency used during the acquisition. From left to right the frequency and pulse lengths used were: 200 kHz/ Medium Pulse; 300 kHz/ Medium Pulse; 400 kHz/ Long Pulse.

2.3 Absorption coefficient profiles

Sound speed profiles are a great concern in hydrographic surveys, especially when meeting IHO standards (International Hydrographic Organization, 2008) are required. To ignore this environmental variable or even to use a profile that does not correspond to the real sound speed profile within the survey area, may drastically invalidate the data acquired, making it unreliable and inconsistent. During the survey, which lasted less than five hours, this parameter

was constantly monitored and five sound speed profiles were measured. A DIGIBAR Pro sensor was used to collect those casts, but the issue is that this type of equipment computes the sound speed measuring it directly while it travels along the water column, without registering temperature (*T*, ºC) and salinity (*S*, ppt) values. Only depth and respective sound speed is registered. For charting purposes, this type of sound speed profiler is efficient, but not for backscattering surveys, where temperature and salinity profiles are necessary to compute absorption coefficient profiles.

In the specific case of this thesis, where the acoustic data was collected in shallow waters, the variable most important for sound speed computation is the water temperature, assuming that salinity does not vary considerably either in space or during survey hours. Thus, in order to overcome the lack of information regarding sea water temperature and salinity, the first step taken was to determine the most suitable value for salinity that should be applied in Equation 21 (Leroy et al., 2008). This equation provides a simplified and accurate numerical model to compute the sound speed in all oceans. Considering that sound speed and depth are known, and salinity can be estimated, the temperature profile can be derived from Equation 21.

To determine which salinity value should be used, salinity information at 1-meter depth, measured by Western Maine Shelf B01 (33 Km from the survey area) and Appledore Island C02 (16 Km from the survey area) buoys were analyzed. These buoys belong to the Northeastern Regional Association of Coastal and Ocean Observing Systems (NERACOOS), whose historical data is available for download at http://neracoos.org/. The analysis of the salinity time series recorded by both buoys during the survey hours showed that this environment parameter remained practically steady along the water column. Considering that the maximum depth within the survey area is 35 meters, the usage of a single value (average) for salinity should not impact

the accuracy of the absorption coefficient computation as will be demonstrated below. Based on that, a single value for *S* was chosen and applied to Equation 21, where all variables, but temperature, are now known, making it possible to determine the temperature profile for each one of the DIGIBAR Pro casts.

$$c = 1402.5 + 5T - 5.44\,x\,10^{-2}T^2 + 2.1\,x\,10^{-4}T^3 + 1.33S - 1.23\,x\,10^{-2}ST + 8.7\,x\,10^{-5}ST^2 + 1.56\,x\,10^{-2}D + 2.55\,x\,10^{-7}D^2 - 7.3\,x\,10^{-12}D^3 + 1.2\,x\,10^{-6}D(\varphi - 45) - 9.5\,x\,10^{-13}TD^3 + 3\,x\,10^{-7}T^2D + 1.43\,x\,10^{-5}SD \quad (21)$$

Where

φ: is latitude

After computing the temperature profile for each DIGBAR Pro cast, all elements needed to compute the absorption coefficient profiles were known. The empirical algorithm used to calculate the absorption coefficient from salinity, temperature, depth, sound speed and seawater pH, is shown in Equation 22 (Francois and Garrison, 1982a, 1982b). f_1 is the relaxation for magnesium sulfate, in kHz and f_2 is the relaxation frequency for boric acid, in kHz. The pH of seawater in the ocean is mildly alkaline and it generally ranges between 7.8 and 8.3 (Carvalho et al., 2013; Jackson and Richardson, 2007). The value for pH used in this thesis is 8.0.

$$\alpha = 10^{-3}\left(\frac{A_1P_1f_1f^2}{f^2 + f_1^2} + \frac{A_2P_2f_2f^2}{f^2 + f_2^2} + A_3P_3f^2\right) \quad (22)$$

Where

$A_1 = \frac{8.86}{c}\,10^{(0.78pH-5)}$

$A_2 = 21.44\,\frac{S}{c}\,(1 + 0.025T)$

$P_1 = 1$

$P_2 = 1 - 1.37\,X\,10^{-4}D + 6.2\,x\,10^{-9}D^2$

$$P_3 = 1 - 3.83\ x\ 10^{-5}\ D + 4.9\ x\ 10^{-10} D^2$$

$$f_1 = 2.8\ \left(\frac{S}{35}\right)^{0.5}\ x\ 10^{\left[4 - \frac{1245}{(273+T)}\right]}$$

$$f_2 = \frac{8.17\ x\ 10^{\left[8 - {}^{1990}/_{(273+T)}\right]}}{1 + 0.0018\ (S - 35)}$$

$$c = 1412 + 3.21T + 1.19S + 0.0167D$$

for T ≤ 20ºC:

$$A_3 = 4.937\ x\ 10^{-4} -\ 2.59\ x\ 10^{-5}\ T + 9.11\ x\ 10^{-7} T^2 -\ 1.5\ x\ 10^{-8} T^3$$

and, for T ≥ 20ºC:

$$A_3 = 3.964\ x\ 10^{-4} -\ 1.146\ x\ 10^{-5}\ T + 1.45\ x\ 10^{-7} T^2 -\ 6.5\ x\ 10^{-10} T^3$$

The first two terms in Equation 22 represents attenuation due to boric acid and magnesium sulfate. The last term represents the viscous dissipation in pure water based on this attenuation model. Boric acid relaxation dominates for frequencies up to 1 kHz and its importance decreases between 1 and 5 kHz. Magnesium sulfate absorption dominates between 5 and 100 kHz, but between 100 and 500 kHz its importance decreases. Above 500 kHz absorption is primarily driven by water viscosity alone (Jackson and Richardson, 2007).

Equation 22 gives a single value for α for every point in the water column where *c* was measured by DIGIBAR Pro. At the end, for each sound speed profile, seven different absorption coefficient profiles can be generated for each frequency that the EM 2040 can operate, as follows: 195 and 205 kHz (200 kHz mode); 270, 282.5 and 295 kHz (300 kHz mode); and 320 and 380 kHz (400 kHz mode). Figure 12 shows the temperature profile (top right) derived from the sound speed profile (top left) measured at the survey area, using the model expressed in Equation 21. This figure also shows the absorption coefficient profiles (bottom) computed by Equation 22, for each frequency.

Figure 12 (bottom) shows that the absorption values are higher close to the sea surface and that is because the seawater temperature values decrease with depth. According to the time series data recorded by local buoys, salinity did not change significantly close to the survey area.

This information makes seawater temperature variations the most critical variable that must be considered during seawater attenuation computation. In deep waters, where both temperature and salinity are more stable, attenuation decreases while pressure increases (Carvalho et al., 2013). Table 4 shows the harmonic means value for the absorption coefficient profiles showed in Figure 12. Harmonic mean is a kind of average that should be used when the majority of the values are uniformly distributed, but there are a few outliers in the population.

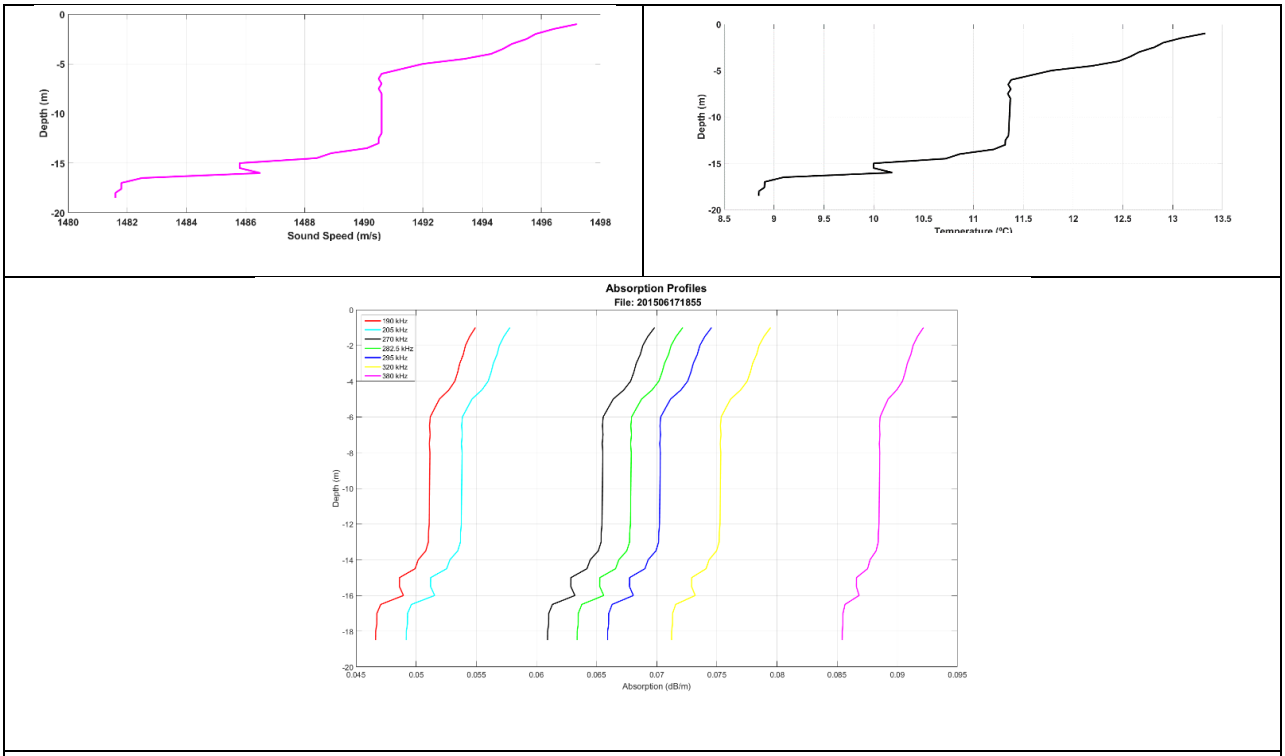


Figure 12. Top left: Sound speed profile. Top right: Temperature profile derived from Equation 21. Bottom: Absorption coefficient profiles derived from Equation 22.

Frequency (kHz)	Harmonic Mean values for absorption coefficients (dB/Km)
190	50.86
205	53.54
270	65.29
282.5	67.66
295	70.10
320	75.16
380	88.47

Table 4. Harmonic mean values for the absorption coefficients profiles showed in Figure 12.

To evaluate this methodology used to derive the absorption coefficient from a sound speed profile measured by a DIGIBAR Pro, absorption coefficient profiles derived from Castaway-CTD measurements were compared to absorption coefficient profiles derived by the method described above, based on Equations 21 and 22. The Castway-CTD is an oceanographic instrument that measures Conductivity (and from that is possible to derive salinity), Temperature and Depth (derived from pressure). Figure 13 shows the absorption profiles computed for a same site at the Piscataqua River, but one during high tide (higher salinity) and the other during low tide (lower salinity).

Salinity profiles in both situations showed in Figure 13 are barely stratified (well-mixed layer), which represents the same pattern of salinity behavior observed within the survey area. The "CTD" profiles showed in Figure 13 are those generated based on the temperature and salinity profiles measure, *in situ*, by the Castaway-CTD equipment used, and the "DIGIBAR" profiles are those generated based on temperature profiles computed by Equation 21 and average salinity.

Table 5 shows the average of the differences and the standard deviation between “CTD” and “DIGIBAR” curves. The values showed in the table, for all frequencies, are small and, to the average depth of the survey area, the usage of the absorption coefficient profiles derived from DIGIBAR Pro would not impact the purpose and the accuracy of this work. This thus validate the usage of this methodology (Equations 21 and 22). Nonetheless, whenever possible, CTD casts are always preferable, because they are believed to be the most reliable representation of the water mass composition in the survey area.

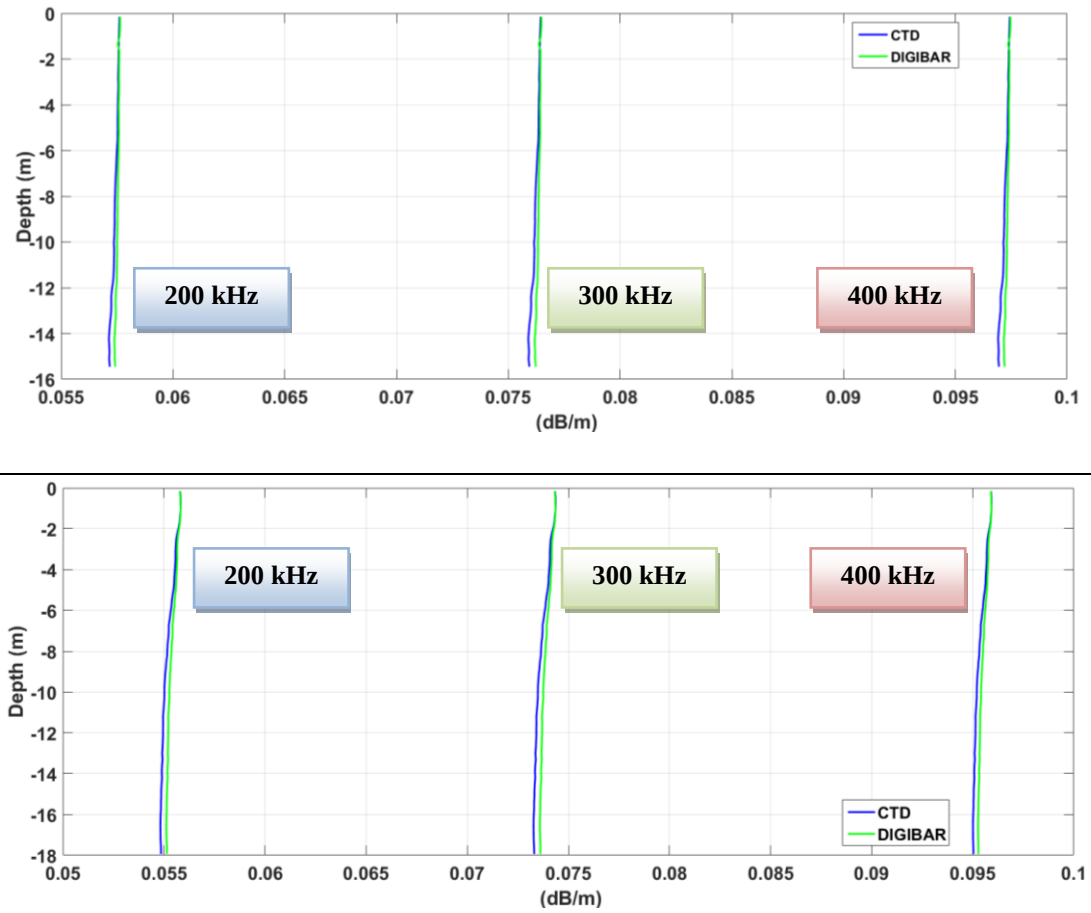


Figure 13. Comparison between absorption coefficient profiles computed based on temperature profiles measured in situ (CTD) and temperature profiles derived from Equation 21 (DIGIBAR). Figure on top represents a cast taken during low tide and figure on the bottom represents a cast taken during high tide cycles. Site Location - Latitude: 43º04'32"N; and Longitude: 070º42'37"W.

Profiles	Frequency (kHz)	Difference between CTD and DIGIBAR methodologies (dB/Km)
Figure 13 (top) Low Tide	200	0.1
	300	0.2
	400	0.1
Figure 13 (top) High Tide	200	0.2
	300	0.2
	400	0.1

Table 5. Statistics between absorption coefficient profiles showed in Figure 13. Values for the differences between CTD and DIGIBAR methodologies are small, which validates the usage of the models presented by Equations 21 and 22.

2.4 Bottom samples

Nine ground truth samples (grab samples and/or video collections) were performed based on the backscatter mosaic made with the data collected for this thesis. Each grab sample was collected using a Stainless Steel Grab, with 9" Steel-Plated Arms (WILDCO Standard Ponar) and the videos were recorded using an underwater video camera (Ocean Systems Delta Vision HD). Due to the rocky nature of the seafloor and also due to the small size of the grab sampler, in four stations it was not possible to collect any grab sample, only video.

Table 6 shows the class terms for each ground truth sample and APPENDIX B provides pictures and more information about each of them. Sediments were classified according to their grain sizes and also according to their aggregate composition (Wentworth, 1922).

Figure 14 shows a map that was colored based on the backscatter mosaic contrast and bottom samples information showed in Table 6. In general, there was a very good correlation between the sediment characteristics, observed in videos and grab samples, and the backscatter mosaic.

Station #	Class Term
1	Gravel
2	Sandy Gravel
3	Sandy Gravel
4	Gravel
5	Sandy Gravel/ Gravel
6	Silty Sand
7	Gravel
8	Gravelly Sand
9	Sandy Silt

Table 6. Class Terms for the sediments collected during the survey.

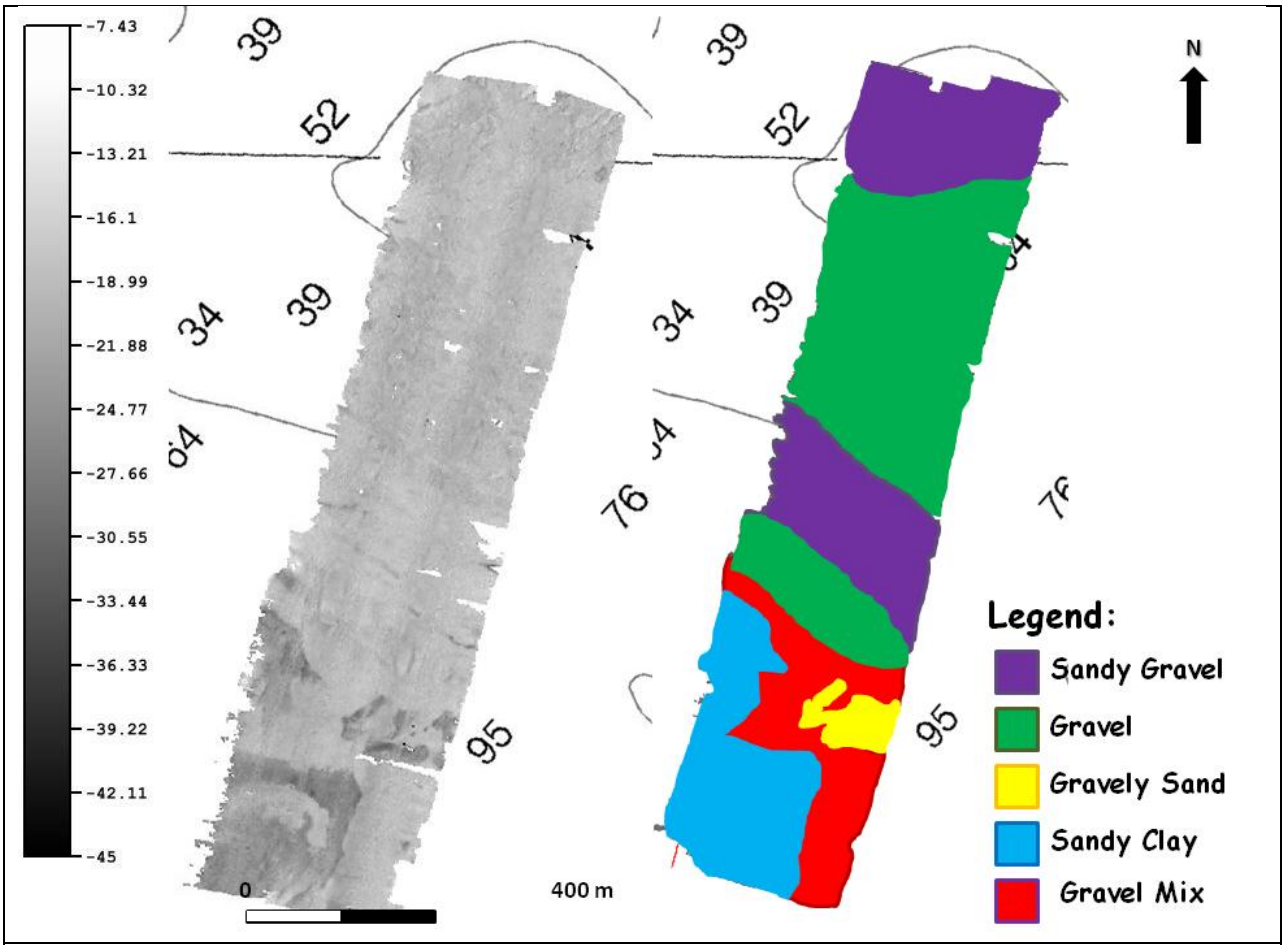


Figure 14. Left: Backscatter mosaic from the survey area. Right: Colored map indicates the seafloor classification based on grab/ video samples and differences in intensity contrast shown in the backscatter mosaic on the left.

The survey area is mostly characterized by hard substrate materials (pebbles and cobbles) and some few areas with soft substrate ranging from very fine sediment to fine sand in some stations. All stations, according to the video images, indicate the presence of shell fragments over the seafloor, in different quantities, where stations 1 and 5 stands out from the others due to the large amount of this type of material. These fragments are characterized by non-subspherical shapes and abrupt scattery edges that can potentially impact the value of the intensity of the signal recorded by the transducer, which may produce divergences from the expected average *BS* for a certain type of substrate.

CHAPTER 3

DATA PROCESSING

3.1 Data processing workflow

To achieve the main goal of this thesis, which is to analyze how different MBES frequencies can be used for seafloor characterization purposes, some corrections applied during real-time acquisition needed to be replaced by more accurate parameters. To do so, some MATLAB codes were written for that purpose, as shown in Figure 15.

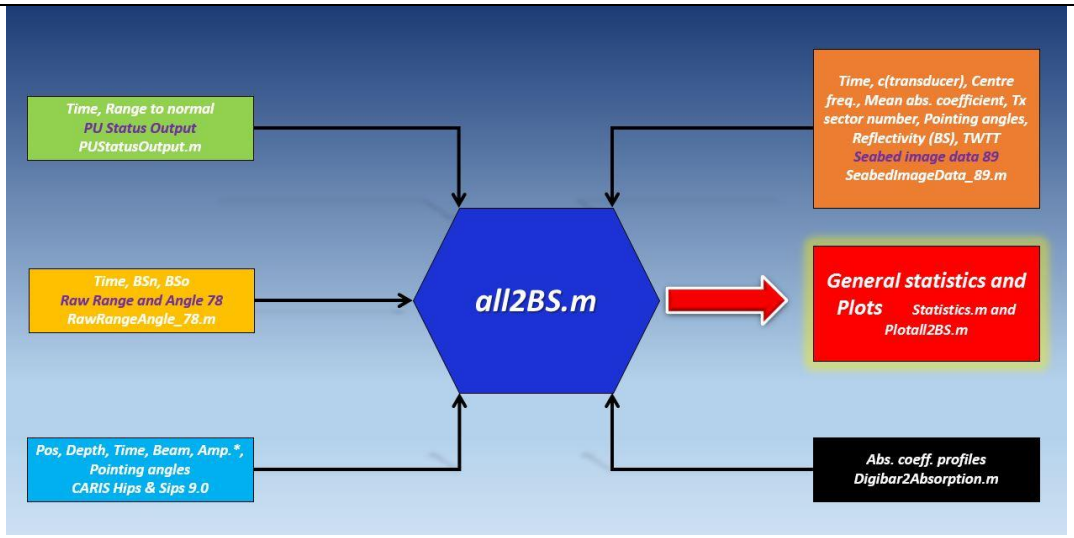


Figure 15. Summary of data required to be loaded into *all2BS.m* (black arrows). The red arrows indicates the products generated by script, which are general statistics and plots, for a survey line only.

Those corrections are mostly associated with the physical processes presented in some parcels of the Sonar Equation, such as range, absorption coefficient and ensonified area.

The main code, named as *all2BS.m*, performs the great majority of the needed corrections. It needs some information as input, represented as black arrows in Figure 15, and gives as output all the information needed for statistics computation and plots. Both input and output files are in ASCII format.

3.1.1 Input files

The data input can be divided in three major groups, as listed below:

a. Group 1 - CARIS file: CARIS HIPS & SIPS 9.0 was used to extract the following information from the raw files (.all format): position of each sounding, depth, time, beam number, amplitude of the signal (dB) and pointing angles.

b. Group 2 - Absorption Coefficient Profiles: Provides the absorption coefficient profiles derived from the sound speed profiles. *Digibar2Absorption.m* scrip is used to generate those profiles. Depending on the time that the line was logged, an absorption coefficient file was chosen based on the nearest in time criteria.

c. Group 3 - Datagram Files: These files are necessary to implement a series of corrections needed. Those information are:

    i. From *PU Status Output* datagram: time and range to normal incidence (MATLAB script: *PUStatusOutput.m*);

    ii. From *Raw Range and Angle 78* datagram: time, BS_N and BS_O (MATLAB script: *RawRangeAngle_78.m*); and

    iii. From *Seabed Image Data 89* datagram: time, *c* at the transducer, centre frequency, mean absorption coefficient, transmit sector number, pointing

angles, reflectivity (dB) and two-way travel time (MATLAB script: *SeabedImageData_89.m*).

The three scripts listed in Group 3 were written because those information could not be extracted from the raw files by any of the commercial software used in this work. Although dB had been extracted from both CARIS and Seabed Image Data 89, only the latter is going to be used during the data processing. From CARIS, only georeferenced depth information is going to be used. That information is useful to compute the local seafloor slope, *TL* and grazing angle at the seafloor.

3.1.2 all2BS.m script

Some computations made by all2BS.m script are made to remove from the raw data some corrections applied during the acquisition process. The result from the manufacturer algorithm (hereafter referred to as "Kongsberg methodology") and the new corrections are herein applied (hereafter referred to as "Thesis methodology"), which are expected to be more refined and accurate, to get values for *BS* that are closer to the reality.

All the input files, listed in the three groups above and shown in Figure 15, were loaded into the *all2BS.m*. Inside this MATLAB script the great majority of the data processing steps were carried out. Figure 16 lists all the computations made by this script, detailed by different steps.

In STEP 1, the data exported from CARIS, .all datagrams and absorption files (ASCII format) were loaded and parsed by the script. Based on the time, for each line, which corresponds to the moment when the research vessel passed through the sample site position, the

script selects, according to the acquisition sampling rate, a time interval that contains at least 200 pings. 200 is considered the minimum amount of pings necessary to average the data to remove part of the noise, which is inherent to the *BS* data. This data delimitation is important because once the time interval is defined, the data is clipped and all information outside of the interval is discharged, improving the script performance.

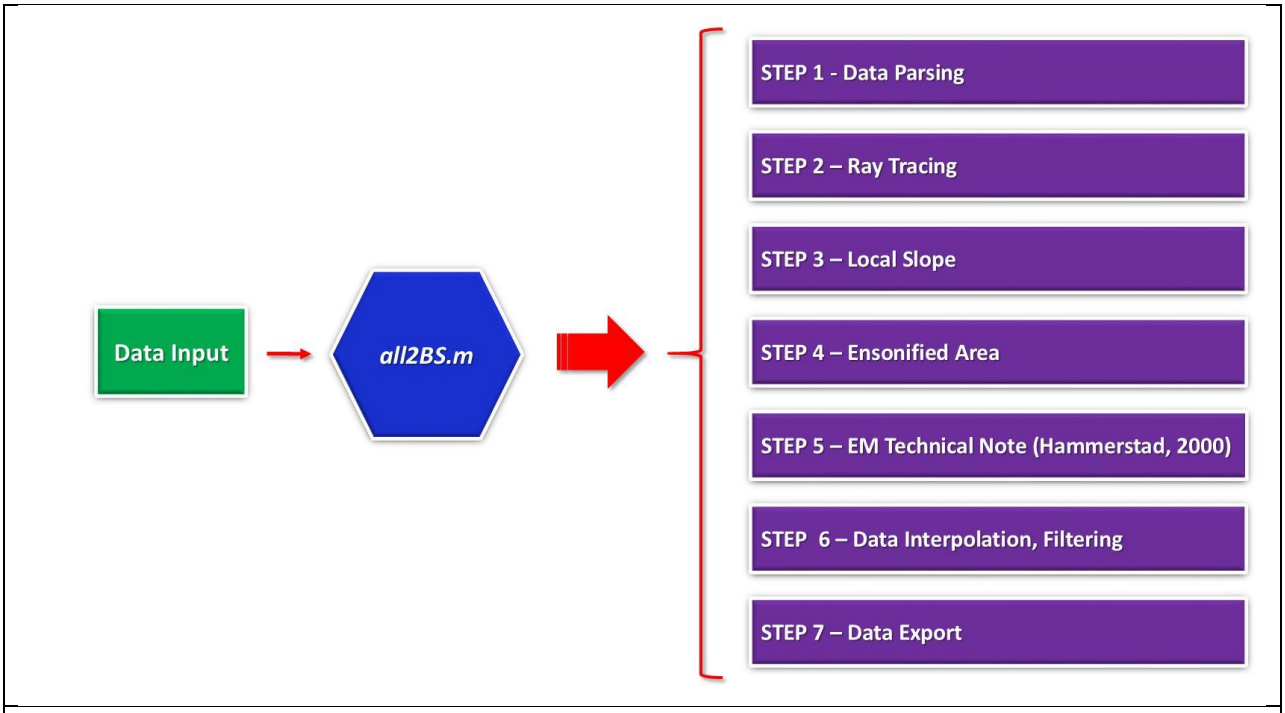


Figure 16. *all2BS.m* script workflow.

In STEP 2, the path of the acoustic signal, including refractions that occur (raytracing) is calculated. This is based on changes in the sound speed profiles (Equation 20) and is computed from the transducer until the point where the acoustic signal reaches the seafloor assuming that the signal travels along the across track plane. Using the ray vector at the seabed depth, the grazing angle at the seafloor can be computed. This is different from the one assumed by the real-time Kongsberg's algorithm, that assumes that the seafloor is flat and the grazing angle is equal to the complement of the pointing angle. Another aspect that the Kongsberg's algorithm

ignores is the extra path length due to the bending effect caused by the signal refraction. It computes the ray path just multiplying out the sound speed at the transducer by the two-way travel time over two.

TL (Equation 2) is also computed in STEP 2, for both Thesis and Kongsberg methodologies. The previous paragraph showed how range is computed in both cases, but it is important to highlight that the value used for absorption coefficient in both cases are also different. The one used in Thesis methodology comes from the nearest in time absorption coefficient profile and the one used in Kongsberg methodology will depend on how SIS was configured for this matter. In the specific case of this work, SIS was configured to apply a preset default value for α during the entire survey (which can be recovered from the Raw Range Datagram).

The local seafloor inclination was computed in STEP 3, for both along and across track planes. To compute the along-track slope for each beam, the position and depth among previous and consecutive pings were used. To compute the across-track slope, the position and depth among across-track neighbor beams were used. To avoid blunders, the computed slope was filtered. For each ping, a local digital terrain map with information about local along and across-track slopes was generated. Now that both along and across track slope are known, it is possible to correct the value of the grazing angles that were previously computed in STEP 2 (Thesis method), which is the complement of the dot product between the ray vector and the surface normal.

In STEP 4, after computing the limit angle, based on the criteria established by Equations 10 and 11, the ensonified areas for each beam were calculated by Equations 15 or 16, which represent two different models to compute near-nadir and oblique incident areas, respectively.

In STEP 5 the Backscattering and Seabed Image Reflectivity Model (Hammerstad, 2000) is applied to the *BS* data recorded in the .all files. At this point, all real-time corrections related to *TL*, ensonified area and the Lambertian correction, made by Backscattering and Seabed Image Reflectivity Model, were removed from the data. More accurate *TL*, which accounts for ray-tracing and absorption coefficient profiles derived from sound speed profiles, and ensonified area, which accounts for along and across-track slopes, were applied to the data. The "New *BS*" data are the reflectivity data from Kongsberg Seabed Image datagram, one per beam, compensated by the corrections listed above in this paragraph.

The reflectivity data was averaged, filtered and interpolated in STEP 6. For the data averaging, as mentioned above, at least 200 pings were used, which considering the echosounder sampling rate, represents 35 seconds of data, on average. This procedure makes the final angular response curve value smoother and works as a pre-filter operation, removing a great part of the unwanted large variation of the backscattering level, as shown in Figure 17. The acoustic wave reflected from a specific patch of the seafloor consists of contributions from many independent scattering areas. Propagation of this acoustic wave to the MBES delays which may vary from several wavelengths, depending on the seabed roughness and the geometry. The interference of these out of phase waves results in the noisy aspect shown in Figure 17, known as speckle incoherent scattering distribution (Goodman, 1976).

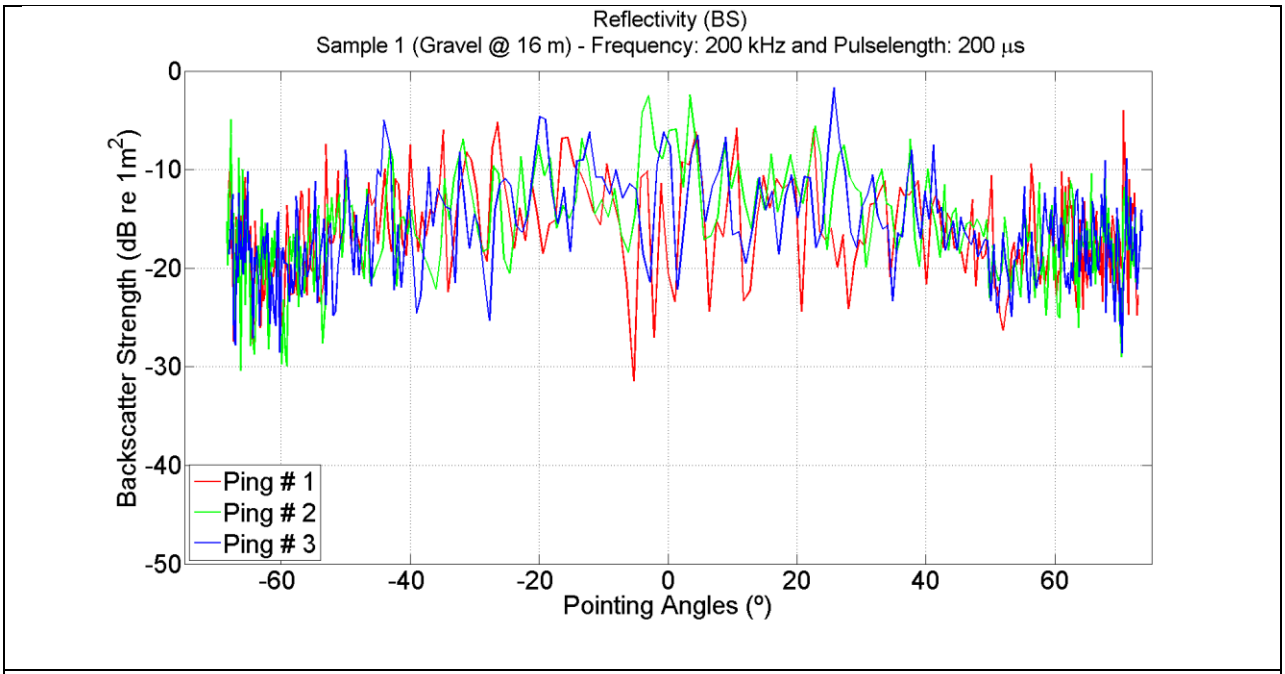


Figure 17. Recorded backscattering levels (3 pings only), extracted from Seabed Image datagram, *versus* pointing angles. This figure shows how noisy the backscatter data can be when a single ping is considered. To overcome this issue, the data was averaged using at least 200 consecutive pings.

Right after the averaging operation, the data was interpolated in 0.05º interval and then filtered using the MATLAB function *filt*, which is a built-in 1-D moving-average digital filter (Equation 23).

$$y(n) = \frac{1}{window\ size}\ (x(n) + x(n-1) + \ ... + x\big(n - (window\ size - 1)\big)) \qquad 23$$

Where

y: filtered data

x: data to be filtered

window size: window size, in degrees, to be used in the moving-average filter

There is no a default value for window size. Different values were tested in order to define which one would be the best to remove the high-frequency noise component from the data, and 5 was the window size that came out with the best filtering result without changing the shape of the backscatter curves. After the filtering process, the backscatter data from both starboard and port side sides were combined and plotted from 25º to 75º (grazing angles), using a 1º interval. Figure 18 shows an example of filtered *BS* (from a .all file - "*Raw Data*"), without the Lambertian correction, assigned as "Kongsberg", versus backscatter data resulted from STEP 6, assigned as "Thesis".

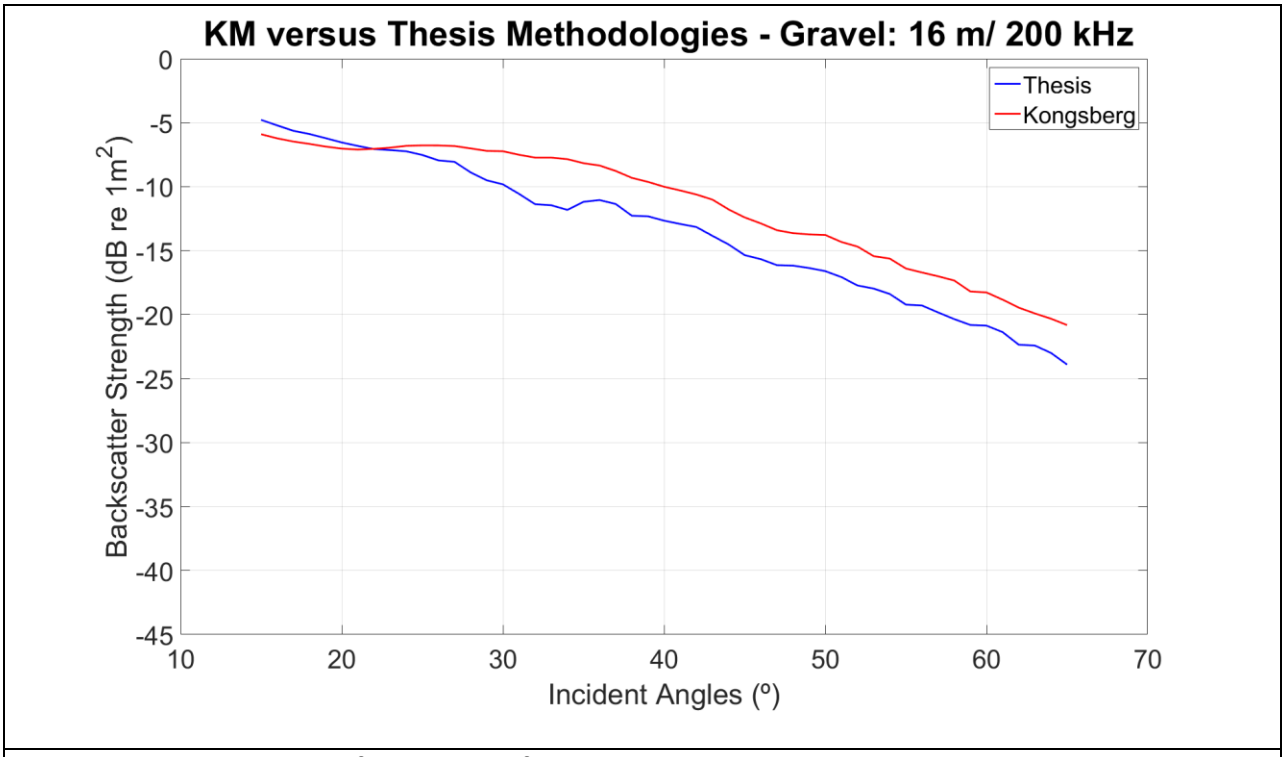


Figure 18. Filtered *BS* (from a .all file - "*Raw Data*"), without the Lambertian correction ("Kongsberg"), versus filtered "*New BS*" ("Thesis").

Finally, in STEP 7, the data export step consists of exporting all the information needed for statistics and plots. The angular range of the exported data extends from 25º to 75º (grazing angle).

All the steps showed in this section were repeated, for each sample, three times, once for each of the EM 2040 frequency modes.

3.2 Preliminary results

All files exported by *all2BS.m* were compiled and a series of different *BS* angular curves were plotted in the following categories:

a) Same frequency, but different samples; and

b) Same sample, but different frequencies.

As a case study to evaluate and validate the general usage of the data reduction presented in this work, the same data processing procedures previously discussed were applied to another dataset, hereafter referred to as "*SH2014*". This extra dataset consists of three different seabed types, as shown in Figure 19, and backscatter data acquired by the same EM 2040 and acquisition software, in 2014, using just the 400 kHz-mode and pulse length of 50 μs.

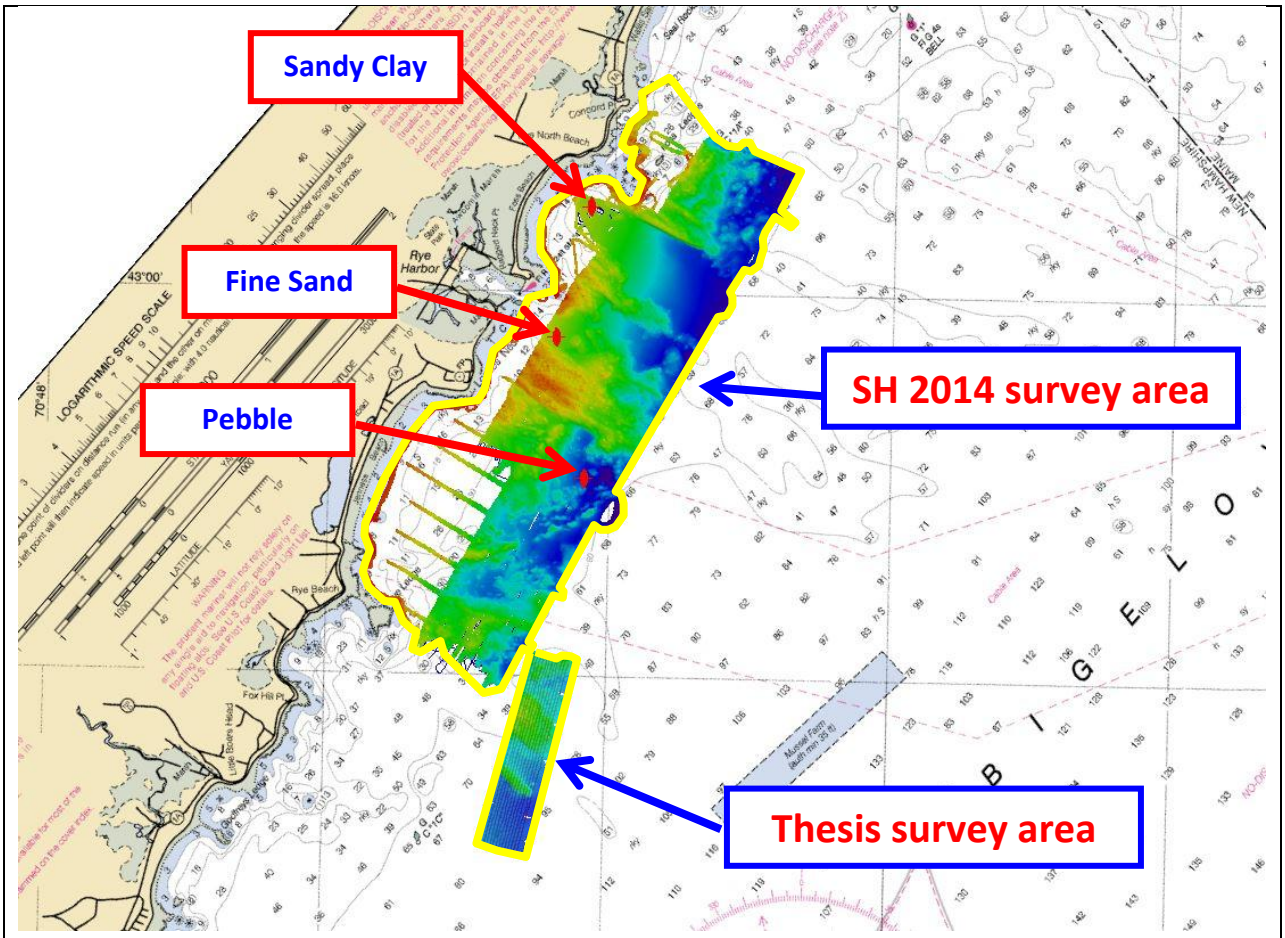


Figure 19. *SH2014* versus Thesis survey areas. *SH2014* Location - NW corner (UTM 19N): Latitude: 43º 01.110'N and Longitude: 070º 43.654'W; and SE corner (UTM 19 N): Latitude: 42º 57.930'N and Longitude: 070º 44.749'W. The three different sample sites to be used as reference are assigned as "Sand Clay", "Fine Sand" and "Pebble". Nautical Chart used as background: NOAA nº 13278 - Portsmouth to Cape Ann; Hampton Harbor (scale: 1:80,000).

The *BS* results from this case study are presented in two different ways: One consists of analyzing and comparing the computed *BS* values from each of the three samples chosen from *SH2014* survey area; and the other consists of adding *BS* plots derived from sandy clay to the computed *BS* from the Thesis survey area plots, 400 kHz mode. This comparison is done as it is softer than any other seabed sample collected specifically for this work.

Figure 20 shows an example of angular response curves, for different samples, derived from 200 kHz/ 200 μs survey lines. Note that sample depths are similar, thus systematic

uncertainties that could be associated to *TL* computation may affect the BS curves in the same way and this would not influence the relative change in shape.

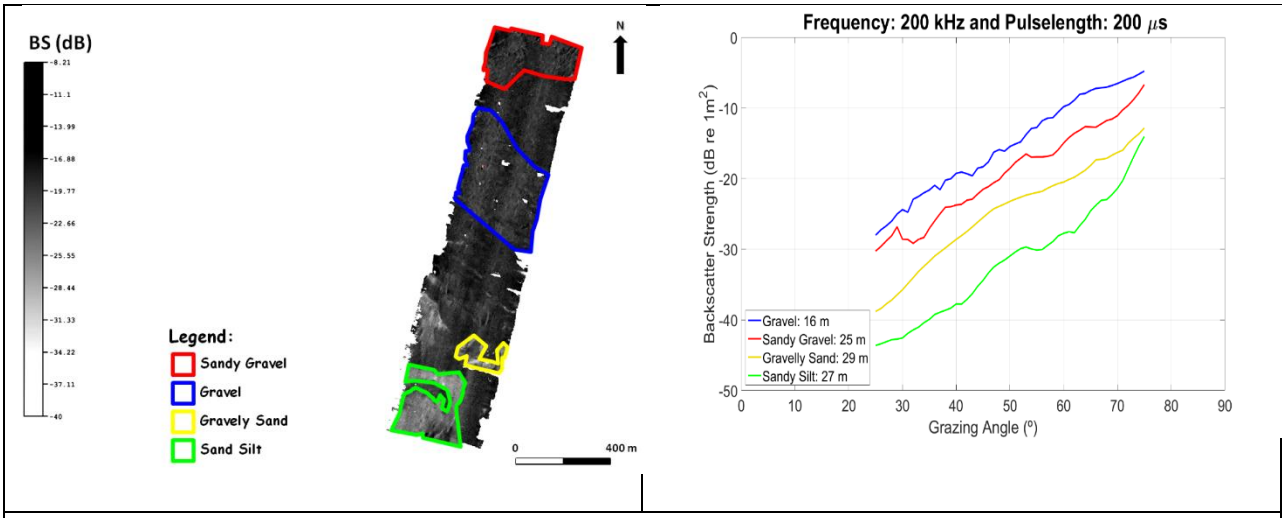


Figure 20. Example of a backscatter mosaic (left) and angular response curves (right). Data shown on the left side are derived from 200 kHz/ 200 μs survey lines.

Figure 21 shows examples of angular response curves, for different frequencies, derived from sample 1 - Gravel (blue polygon in Figure 20), sample 5 - Sandy Gravel (red polygon in Figure 20), sample 8 - Gravelly Sand (yellow polygon in Figure 20) and sample 9 - Sandy Silt (green polygon in Figure 20).

To make the comparison easier among different curves, in both categories a) and b) listed in this section, it is recommended to remove the average angular response effect from the data. To remove both beam pattern uncertainty and the angular response effect from the dataset is an alternative, as mentioned before. In the specific case of this thesis, however, where the entire dataset was collected using the same MBES, a simpler alternative to get rid of that angular effect consists of choosing a certain type of seabed as reference and, for each frequency-mode curve, to compute the difference between the reference angular response curves to their respective averages, as shown in Figure 22. After doing that, for each frequency, a vector was generated

with the dB correction that must be applied to the dataset to get rid of combined beam pattern uncertainty and referenced angular response effect.

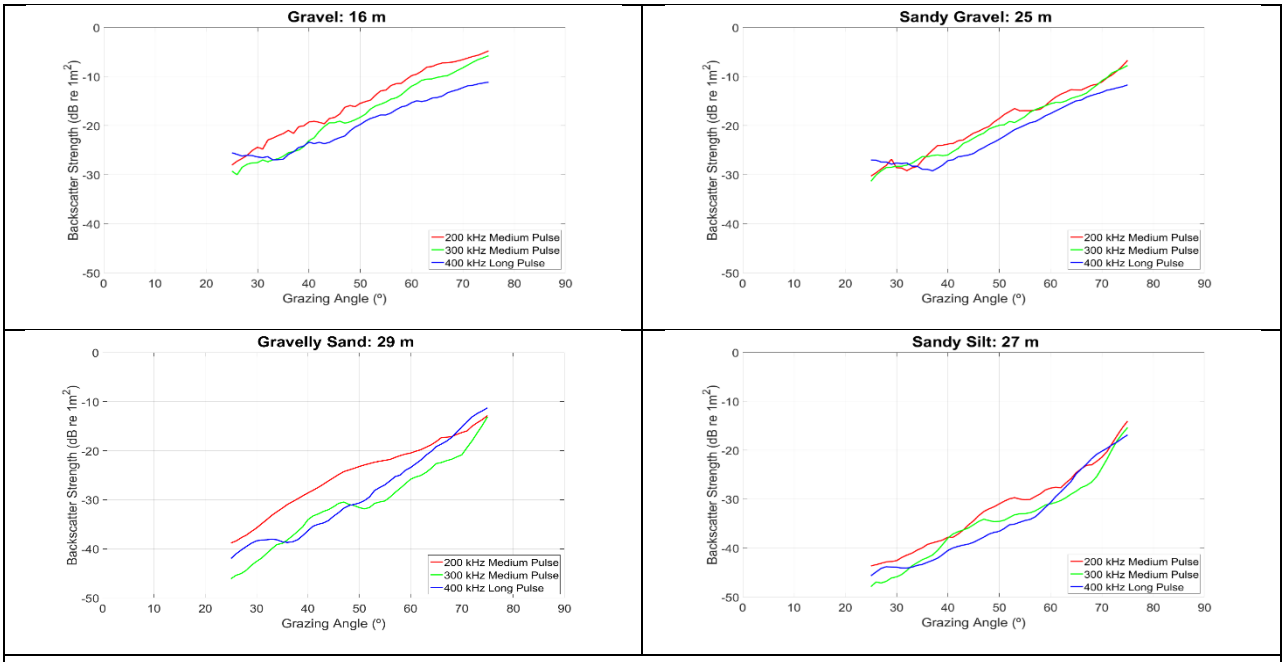


Figure 21. Multi-frequency angular response curves for different seabed types.

Those differences, were assumed to be constant for each frequency mode and then they were applied to the entire dataset. Site 1 (Gravel) was chosen as reference to compute, for each frequency, the correction vectors that are needed to "normalize" the entire dataset. Site 1 was chosen as reference because it was the roughest and therefore assumed to have a response that is closest to an ideal "Lambertian". Figure 23 shows the same plots showed in Figure 21, but now with the frequency specific beam pattern and the angular dependence effect minimized, after executing the procedure explained above.

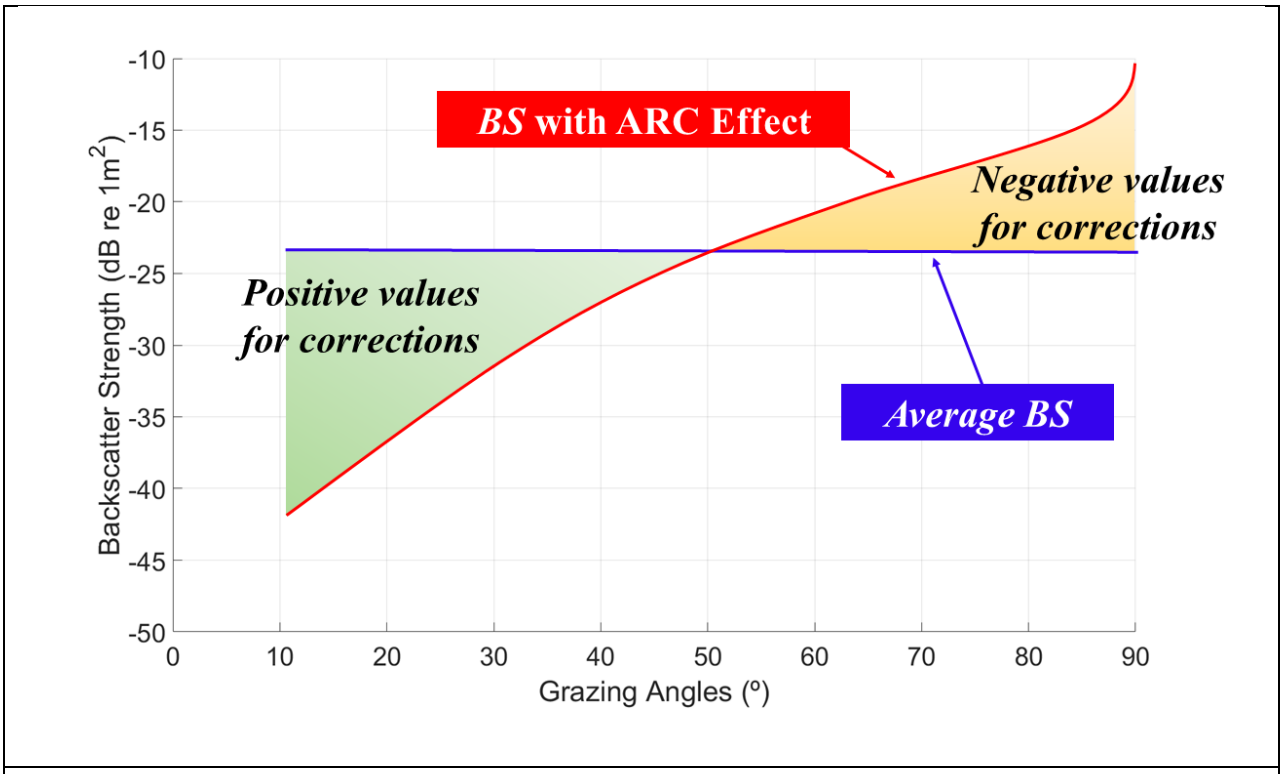


Figure 22. Removing frequency specific beam pattern and the angular response effect from the original *BS* curves. After choosing a certain seabed type as reference, the difference between the *BS* curve and its respective average *BS* will result in corrections that have to be applied to the entire dataset in order to get it rid of the main part of the angular response effect.

Figure 24 shows, as example, *BS* curves for different seabed types when ensonified by 200, 300 and 400 kHz frequencies, with and without the frequency specific beam pattern and reference angular dependence effect. After those corrections it was simpler to analyze the multi-frequency data, relative to reference site 1, under two different perspectives:

i. Verify the frequency dependency effect, which is inherent to the backscatter data, when, for the same site, the average dB offset between frequencies are compared to one another; and

ii. Verify the change in shape of the Angular Response Curves (ARC) among the four different substrates types used in this work.

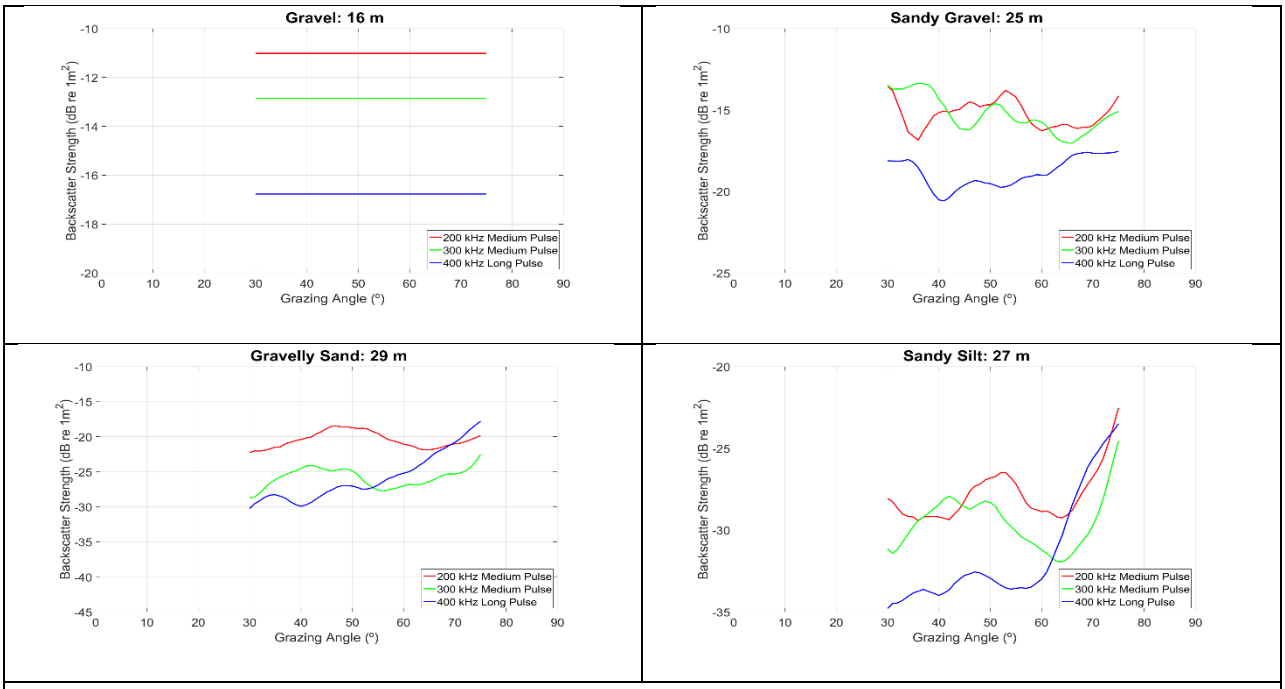


Figure 23. Same plots shown in Figure 21, but without the angular dependence effect.

As long as gravel (sample 1) was chosen to be the reference seabed type to calculate the corrections discussed above, only their corrected *BS* plots are perfectly straight and horizontal, as showed in Figure 23 (top left) and Figure 24 (on the right), on the right side. The reason why it is so difficult to remove the angular response effect from the *BS* data can be attributed to (Hughes Clarke, 2015):

i. the shape of the *BS* curves changes depending on the substrate; and

ii. for a single substrate, the shape of the *BS* curves changes with changes in frequency.

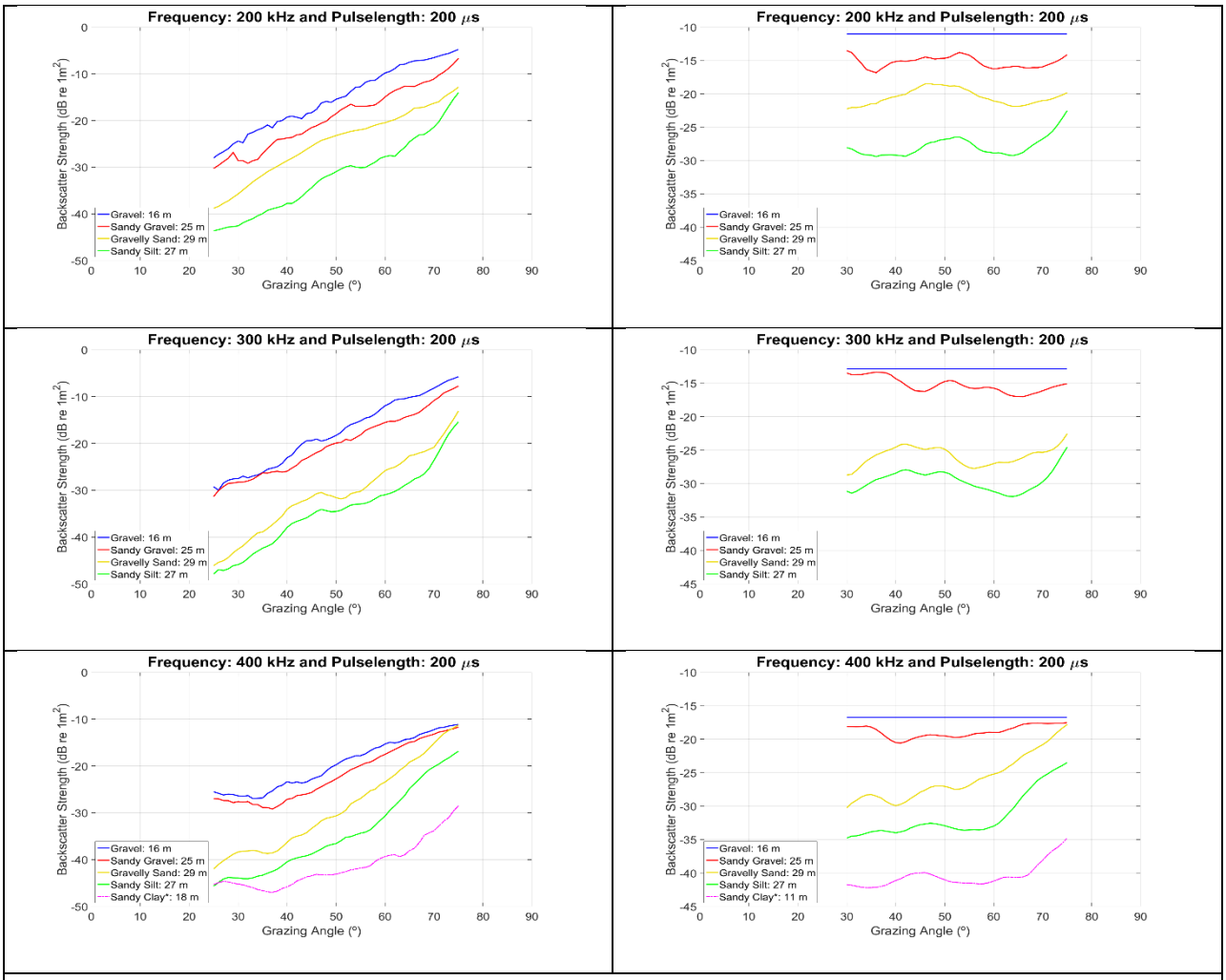


Figure 24. Same frequency used to ensonify different types of seabed, with and without the reference angular response effect. Left: Original angular response curves for different types of seabed ensonified with 200 (top), 300 (middle) and 400 kHz (bottom). Right: Same plots showed on the left, but with the beam pattern and reference angular response curve effect minimized. Legends indicate both seabed type and sample depth.

Note that for the 400 kHz-mode, in Figure 24, an extra *BS* referred to *SH2014* was added (sandy clay). As mentioned before, this is the "softest" seabed sample available in both Thesis and *SH2014* dataset, and as expected, its BS curve presents the lowest reflectivity levels. Figure 25 shows *BS* levels that correspond to *SH2014* samples.

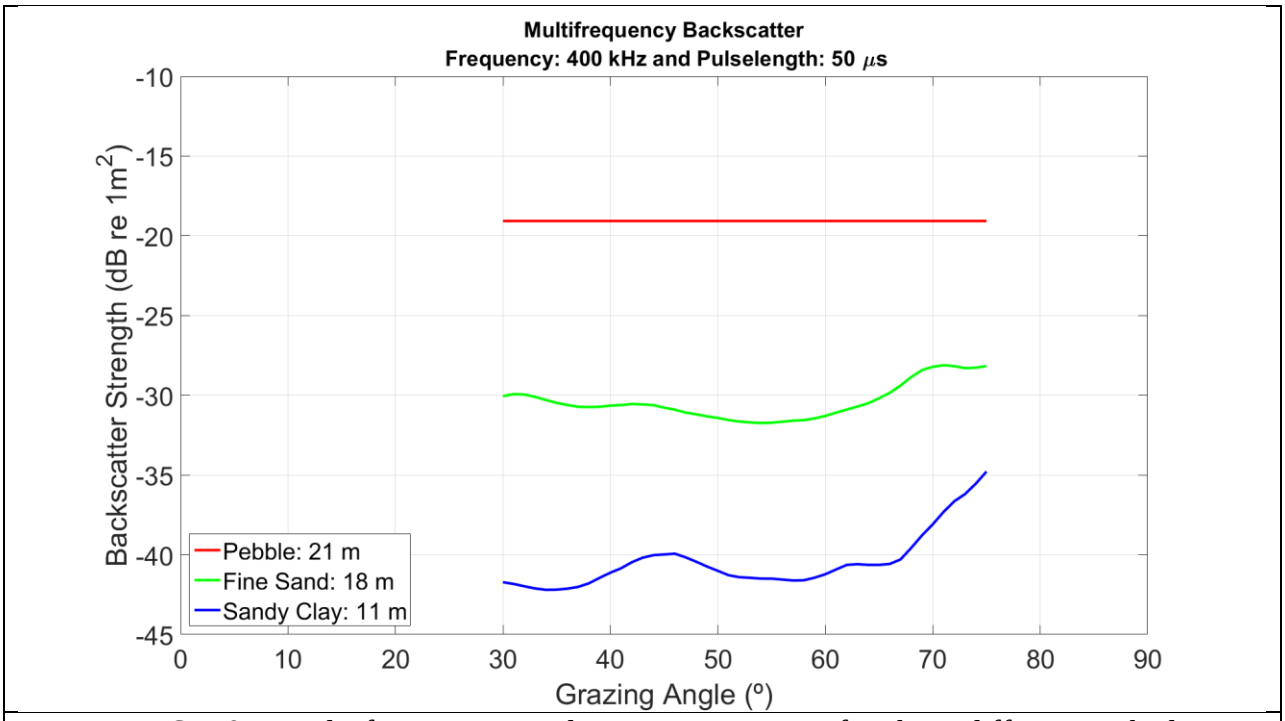


Figure 25. *SH2014* multi-frequency angular response curves for three different seabed types. Legend indicate both seabed type and sample depth.

Intensity levels shown in Figure 25 had their beam pattern and angular response effects minimized according to the same methodology applied to Thesis' dataset. In this case, pebble *BS* curve was assumed to be the most "Lambertian" sediment type and it was used as seabed reference. After getting through all the processing steps discussed until here, some statistics were calculated and results are shown below.

Analyzing all data plots, it can be noticed that the angular interval between 30º and 60º is the simplest region to be used for statistic. In that region, different *BS* curves are relatively parallel to one another (plateau zone). For different types of seafloor and frequency, the average difference between the *BS* computed by Kongsberg and Thesis methodologies are shown in Table 7. As mentioned before, the main improvements proposed by Thesis methodology when compared to the approximations made by Kongsberg algorithm in real-time are:

i. Grazing angle computation that accounts for seafloor slope and ray-tracing;

ii. Transmission Loss computation that accounts for ray-tracing and cumulative absorption coefficient; and

iii. Ensonified area that accounts for local seafloor slope.

Frequency Mode	Gravel Sample 1	Sandy Gravel Sample 5	Gravelly Sand Sample 8	Sandy Silt Sample 9
200 kHz	-2.8 dB	-3.0 dB	-3.4 dB	-3.1 dB
300 kHz	-2.9 dB	-3.0 dB	-3.3 dB	-3.0 dB
400 kHz	-3.0 dB	-2.6 dB	-2.9 dB	-2.7 dB

Table 7. Average difference between the *BS* computed by Kongsberg and Thesis methodologies

Table 8 shows another potential application of this work, which is to verify the average *BS* values for each type of substrates in different frequency modes and Table 9 shows the average *BS* differences between gravel, sand gravel and gravelly sand substrates with respect to sandy silt substrate, which is the softest seabed type in this case and, as consequence, the lowest *BS* levels, for each frequency mode.

Frequency Mode	Gravel Sample 1	Sandy Gravel Sample 5	Gravelly Sand Sample 8	Sandy Silt Sample 9
200 kHz	-11 dB	-15 dB	-20 dB	-28 dB
300 kHz	-13 dB	-15 dB	-26 dB	-29 dB
400 kHz	-17 dB	-19 dB	-27 dB	-33 dB

Table 8. Average *BS* values for each type of substrates in different frequency modes.

Frequency Mode	Gravel *vs* Sandy Silt	Sandy Gravel *vs* Sandy Silt	Gravelly Sand *vs* Sandy Silt
200 kHz	17 dB	13 dB	8 dB
300 kHz	17 dB	15 dB	4 dB
400 kHz	17 dB	14 dB	6 dB

Table 9. Average *BS* differences between gravel, sand gravel and gravelly sand substrates with respect to sandy silt substrate, for each frequency mode.

CHAPTER 4

DATA ANALYSIS

4.1 Uncertainties inherent to the backscatter data

As showed before, various simplifications found in real-time acquisition processing (KM methodology), used by SIS, may render statistical analysis difficult, since they notably affect the *BS* data. An alternative to improve final *BS* results consists of replacing those real-time processing corrections by more accurate models, such as those used in this work (Thesis methodology), in a way that an at-sea or in-tank calibration may not be required. In many cases, the uncertainty associated to those real-time simplifications are greater than 2 dB, which may drastically affect the ability to distinguish two different seabed types by their average *BS* (Hellequin et al., 2003; Malik et al., 2015). In the specific case of this thesis, where the same MBES was used to acquire the data and the system settings, such as installation and runtime parameters, remained the same during the entire acquisition process, the *TL*, seafloor grazing angle and ensonified area could be considered the major sources of *BS* uncertainty, assuming that *SL* uncertainty may affect *BS*, systematically, at any frequency mode. Thus, the next three sections are going to describe the major components that could affect the computation of those three variables whose Kongsberg computations were replaced by more accurate values.

4.1.1 Causes of *TL* uncertainties

TL uncertainties, according to Equation 2, are related to the estimation of absorption coefficient and range. In this work scattering and viscous absorption processes due to the presence of solid particles in the water column were not considered during *TL* computation. Absorption coefficient sources of uncertainty, according to Equation 22, may include uncertainties in the measurement of the physical and chemical properties used to compute it, which are: sonar frequency (for each transmit sector), pH, salinity, depth and temperature. Figure 26 shows how the uncertainties associated to pH, salinity, depth and temperature measurements may impact the absorption coefficient. In the specific case of this work, depth is the only variable that is known. According to the environmental parameters used to compute those plots, which correspond to the average values observed within the survey area, temperature is the variable whose uncertainty could most impact the absorption coefficient computation, because temperature in shallow waters is considered to be the most important variable in the calculation of the sound speed profile. The accuracy of absorption coefficient model (Equation 22) is estimated to be within about 5% (Francois and Garrison, 1982a).

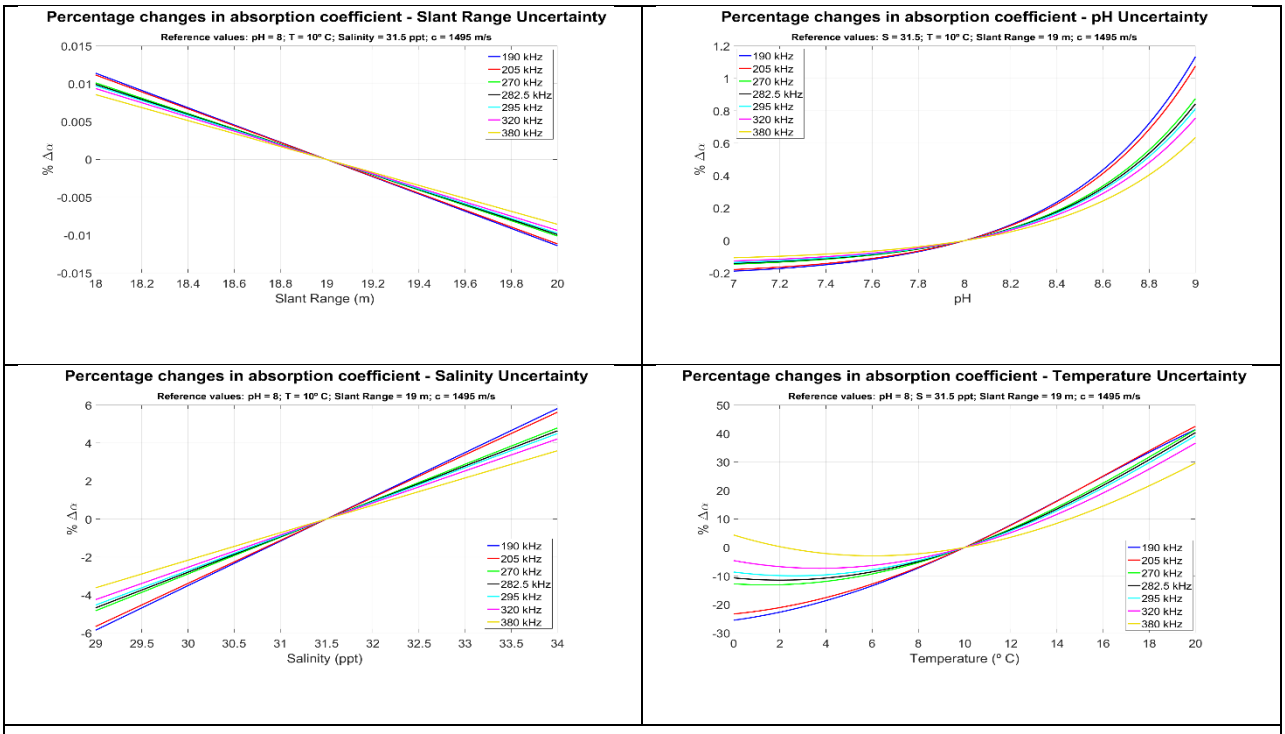


Figure 26 - Percentages changes in absorption coefficient (%Δα) based on changes in slant range (top left), pH (top right), salinity (bottom left) and temperature (bottom right). The reference values assumed are: Temperature: 10º C; c: 1495 m/s; Salinity: 31.5 ppt; pH: 8; and Slant Range: 19 m. Those values correspond to average values observed during the survey hours.

The sources of uncertainty associated to the slant range to the seafloor are related to the way that ship attitude and sound speed profile were measured and, according to studies, these two factors do not appear to be large enough to impact the final *TL* uncertainty (Malik et al., 2015). For both range and absorption computation, the correlation between the measured sound speed profile and the real sound speed profile in every point within the survey area can be a great source of uncertainty, depending on the weather and environmental conditions within the survey area. In the specific case of this thesis, the survey area is small and the entire dataset was collected in less than six hours, the weather conditions were good (calm sea and light winds). In total were taken five sound speed casts, whose profiles did not show large changes in the water column characteristics between two consecutive casts. This is acceptable to assume that the

sources of uncertainties that could be associated to the lack of information about the real sound speed profile are not significant in this case.

The average difference between ray tracing (Thesis methodology) and range paths (Kongsberg methodology), for the entire dataset is 1 meter, which represents 3% of the average ray tracing ranges. This small difference between Thesis and Kongsberg methodologies means that Kongsberg real-time compensation seems to be enough for this data set. The small difference between those two methods can be attributed to:

i. The average depth within the survey area is 19 meters (shallow); and
ii. The sound speed profiles are less stratified (mixed layer), which indicates that the acoustic wave paths, in this case, is roughly a straight line, and that specific characteristic makes the Kongsberg algorithm consistent.

The average difference between the absorption coefficient that takes into account the absorption coefficient derived from each sound speed profile (Thesis methodology) and the absorption coefficient profile used by SIS during the acquisition is 7 dB/Km (200 kHz mode), 6 dB/Km (300 kHz mode) and 2 dB/Km (400 kHz mode). Considering that all *BS* statistics are focused on the grazing angle region within 30º-60º and that the maximum depth of the survey area is 35 meters, the maximum range is expected to be lesser than 100 meters and, for this range, the maximum average difference between the absorption coefficient value applied to the data in real time and the one derived from the sound speed profile is, in the worst case scenario, less than 0.7 dB.

When the *2TL* (Equation 2) that takes into account the absorption coefficient and the range computed by the Thesis methodology versus the same *2TL* computed with Kongsberg parameters are compared, the average difference between them is 1 dB, for any frequency mode.

It is important to highlight that, for the specific case of the dataset used in this work, the average depth is 19 meters and the maximum depth is 35 meters.

In cases where the survey area presents more stratified waters, the difference between Kongsberg and Thesis' ranges will be larger than those computed with the Thesis dataset, because the difference in water density between two consecutive water layers refracts the acoustic signal and, depending on the water depth, the difference between the ray-tracing range and Kongsberg range can be significantly different. A large discrepancy between Kongsberg and Thesis ranges can also be expected in deeper waters, even if the sound speed profile is barely stratified.

Although both Thesis ray-tracing and absorption coefficient methods do not appear to significantly change the corrections applied to the data in real-time, it is important to reinforce that these corrections are more accurate than those corrections made in real-time. Due to the innumerous unknown sources of uncertainties related to backscatter data, it is worth improving every part of the data processing that could contribute for known causes of propagation of uncertainty that may compromise the usage of *BS* data. In this way one can better discriminate similar seabed types, within a same substrate group.

4.1.2 Causes of seafloor grazing angle uncertainties

The seafloor grazing angle is derived when the complement of the angle between the normal to the local seafloor and the beam pointing angle at the seafloor is computed. Uncertainties in the estimation of the grazing angle are related to beam steering uncertainty due to the accuracy of the sound speed at the transducer; and uncertainty related to the mathematical

models used to compute the local seafloor slopes and refractions through the water column (Malik et al., 2015). Uncertainties related to beam steering are assumed to be negligible (Malik et al., 2015). A criterion that can be used to determine if the grazing angle uncertainty is significant or not is to compare its value to the half of the receive beam width (Malik et al., 2015). As mentioned before, the *BS* angular response curves change with respect to changes in grazing angle values. The presence of uncertainties related to the grazing angle will impact the accuracy of the ensonified area, but the most important effect of grazing angle uncertainty is expected to be the distortion of the angular response curves (Jackson et al., 1986). The angular interval where the grazing angle uncertainty is most sensitive is at the steepest part of the angular response curve, which in terms of grazing angle, goes from 90º to the crossover angle. Within this angular interval, even a small angular variation may correspond to a high variation in *BS* value. However, at the plateau zone, where *BS* values do not vary much with respect to changes in angle, uncertainties related to grazing angle computation do not necessarily correspond to significant changes in *BS* values.

4.1.3 Causes of ensonified area uncertainties

The sources of uncertainty for the ensonified area, according to Equations 15 and 16, are: along and across-track local seafloor slope; along and across-track beam widths; bottom sound speed; pulse length; seafloor incident angle; and range. In addition to those factors, the criteria used to determine the limit angle, which in the case of this work are expressed by Equations 10 and 11, may also impact the ensonified area computation. Table 10 shows, in details the uncertainty sources for seafloor ensonified area (Malik et al., 2015). Note that the relevance due to many of the items listed in Table 10 were assigned as "Not known", which makes the

uncertainty associated to the ensonified area computation a big concern. The column on the right specifies if the uncertainty effect can produce bulk shifts in dB or can vary by angle. If the effect produces a common shift in all results, this would be equivalent to a *SL* uncertainty and its effect is less important as it represents just a common bulk shift to all values.

Main component	... depends on (subcomponent)	Subcomponent uncertainty	Significant or not	Bulk shifts or vary by angle?
Area integration bias	Bias between actual area and approximation used	Not negligible for nadir region	Yes for near-nadir	Bulk shifts
Beam width	Frequency	System dependent	Not known	Bulk shifts
	Ship attitude	System dependent	Not known	
	Sound speed	System dependent	YES	
Pulse Length	Pulse length calibration	System dependent	Not known	Bulk shifts
	Band width	System dependent	Not known	
Seafloor incident angle	Discussed in section 4.1.2			Vary by angle
Method errors	Flat seafloor assumption	Case dependent	Case dependent	Vary by angle
Unresolved seafloor slope	Beam footprint	Seafloor topography dependent	Not known	Bulk shifts
	Seafloor topography	XXX	Not known	Vary by angle
Bottom sound speed	XXX	< 0.1 %	NO	Bulk shifts

Table 10. Uncertainty sources for seafloor ensonified area. Table derived from (Malik et al., 2015)

The beam width of a sonar is approximately inversely proportional to the number of wavelengths along the array. As the wavelength varies with sound speed, so does the beam width. Just to illustrate, if the real sound speed is 1448 m/s, but the assumed value, due to uncertainties, is 1504 m/s, the assumed beam width is going to be 3% greater than the real value. Thus transmit directivity is now lower and, a consequence, less *SL* is transmitted. This

percentage can be computed by the difference between the assumed and the real value for sound speed over the real value for sound speed.

4.1.4 Causes of Source Level uncertainties

SL uncertainties sources are generally associated to the age of the transducer, system settings, biofouling on the transducer or transducer element failure. The net uncertainty associated to those parameters is system dependent and also has a lot to do with software versions.

The *SL* uncertainty itself is the biggest single effect in the final value for *BS*, but if the system is not calibrated, there is no way to compensate it in the field.

4.2 Analysis of the data results

As shown in Equations 15 and 16, pulse length has an important role in the ensonified area calculation. As mentioned in section "2.2 Survey area", the 400 kHz lines were run using different pulse lengths (200 and 100 μs) with the purpose of verifying if changes in pulse length could affect the final BS result for 400 kHz mode. In theory, after removing the ensonified area contribution from *BTS*, no variation in *BS* is expected due to changes in pulse length (Gensane, 1989), but, in the case of the dataset used in this thesis, *BS* curves do change with respect to changes in pulse length, as shown in Figure . There are some indications that can explain why these curves are not overlapping one another:

i. The real pulse length does not match with the pulse length information provided by the product description of the MBES (Kongsberg Maritime, 2015). According to the manufacturer, who were asked by electronic email about this topic, the information provided by system manual is accurate, but even if they the nominal values are accurate, Table 10 shows that there are still uncertainties associated to the pulse length, such as pulse length calibration and beamdwidth, whose effects to the ensonified area uncertainties are unknown;

ii. The physical models used to compute the crossover angle (Equations 10 and 11) and the ensonified area (Equations 15 and 16), are not accurate enough;

iii. The physical model used by KM, during real-time processing, to remove the ensonified area from *BTS* that system provides is not exactly as described in Equations 8 and 9;

iv. The real local seafloor slope does not correspond to the local along and across-track slopes computed; and

v. Another point, that has nothing to do with ensonified area computation, is the physical model used to account for *TL* (Kongsberg and Thesis methodologies). Approximations used in Equation 2, like spherical spreading assumption, may not apply to all cases and those could impact the final *BS* values. Although uncertainties associated to ranges and absorption coefficients could influence the final *BS* values, the variable sound speed was constantly monitored during the entire survey and, due to these circumstances, the probability of these types of uncertainty to be responsible for the mismatch shown is very low.

Figure 27 shows that, except for the gravely sand substrate (bottom left), the *BS* curves for both *MP* and *LP* are, on average, less than 1 dB apart.

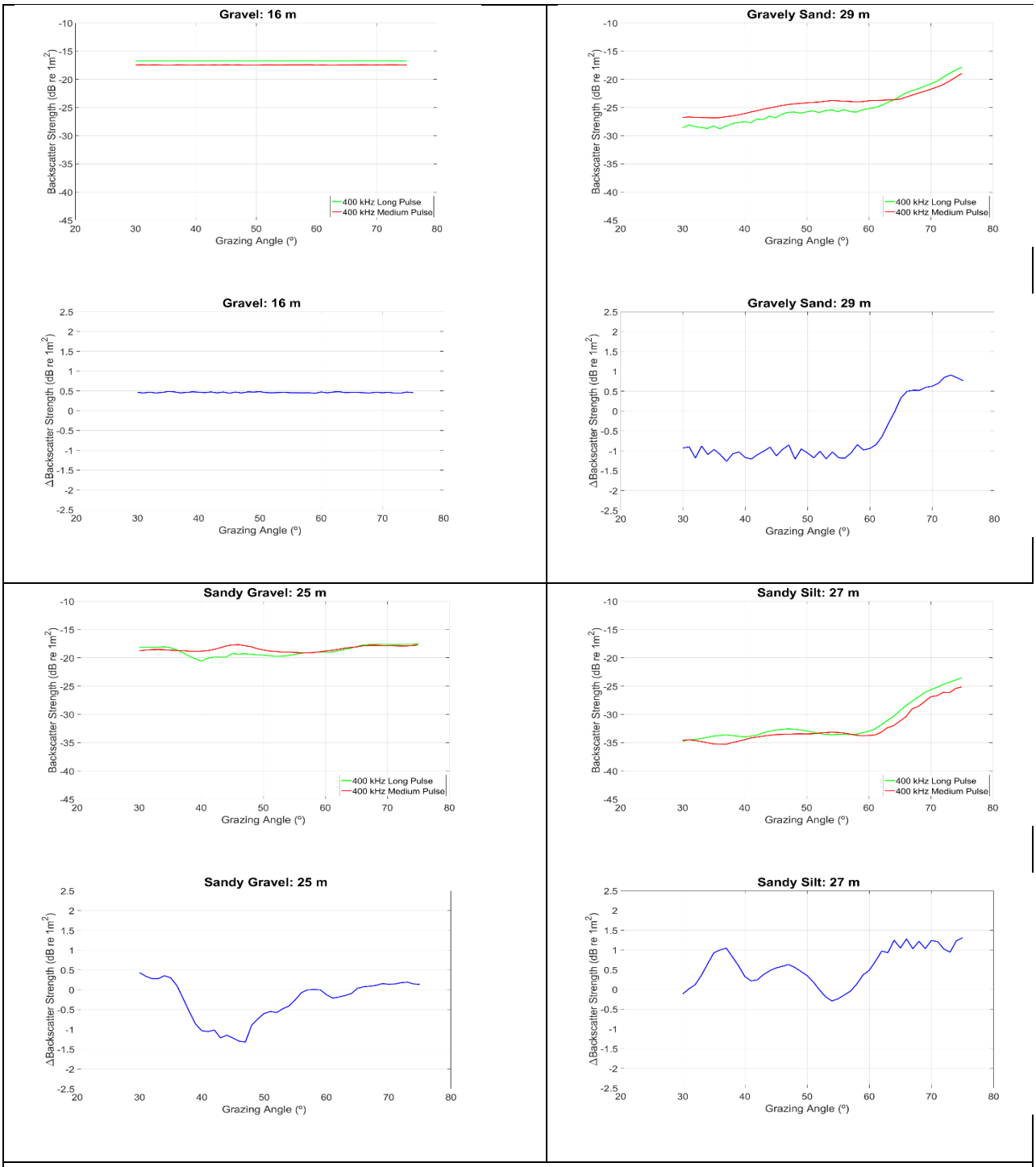


Figure 27. Comparing 400 kHz *BS* where the same seabed patches were ensonified with different pulse lengths. The average difference between 400 kHz *MP* and 400 kHz *LP BS* curves is less than 1 dB for all types of substrates.

This indicates that the usage of different pulse lengths, taking into account the precision of the measurements showed in this work, even with all causes of uncertainties listed above, might not impact *BS* curves in a way to invalidate the usage of either pulse length curves with seafloor characterization purposes (Hellequin et al., 2003; Malik et al., 2015).

To validate the usage of remote sensing techniques for seafloor characterization purposes it is expected that sites with the same seabed composition present similar *BS* levels, especially if those sites were ensonified with the same MBES. According to bottom samples that had been collected in the survey area (Figure 14), sites 1 and 4; and 3 and 5, have the same characteristics and that allows their usage to evaluate uncertainties associated to the usage of multi-frequency backscatter in seafloor characterization. Table 11 shows the average *BS* dB offset for those two seabed types, using as reference sites 3 and 4. Note that the dB offsets are greater than 2 dB, in any frequency mode and substrate. The causes that could be attributed to those differences are:

i. Bottom samples positioning. The position used to geo-reference the samples are based on the horizontal position of the R/V, which is not necessarily the position of the sampler when it touched the seafloor. Underwater current was not monitored during the bottom samples acquisition process, but it was noticed that the R/V used to drift tens of meters during each sampling operation. To minimize this positioning issue, it is recommended to use a ship-relative sampling methodology (e.g. Ultra Short Baseline) device and launch the bottom sampler at the center of the area of interest;
ii. The fact that the survey area is small and most of its bottom types are basically composed by hard material, makes difficult to delimitate the boundaries among different types of substrates. This suggests that better results could be reached when substrate patches are large and their borders are well delimited (higher dB contrast); and

iii. The grab sampler used was small and seemed to be inappropriate to collect an adequate representation of the bottom sample composition. In some places, where the seabed type was composed of very fine materials, it was noticed that the amount of sample that were recovered was not enough to allow seafloor classification. In those sites, in the absence of physical sampling, only digital photographic images were used with that purpose. Poor quality of the bottom samples may drastically affect the seafloor classification process.

iv. Another cause that may significantly impact the usage of grab sampler or digital cameras to classify the seafloor is the spatial heterogeneity distribution of the sediments along the survey area. The sampler grabs just a few square decimeters of the seafloor and the video indicates that the seafloor is varying at this scale. As a result, the ground truth samples are not a realistic representative of seafloor patchiness.

Frequency Mode	Sample 1 versus 4 (Gravel)	Sample 3 versus 5 (Sandy Gravel)
200 kHz	5 dB	2 dB
300 kHz	4 dB	4 dB
400 kHz	3 dB	4 dB

Table 11. Comparing dB offsets between different sites with the same seafloor characteristics.

By this point, most of the considerations are going to be related to the usage of the multi-frequency backscatter data as a tool for seafloor characterization. Although the nominal values of the frequencies that a EM2040 system can operate were 200, 300 and 400 kHz, in reality, the range of frequency goes from 190/205 kHz (200 kHz) to 320/380 (400 kHz), passing by 270/282.5/295 (300 kHz). This means that, in most of the cases, the data that are being compared here were derived from acoustic frequencies that are less than one octave apart. Table 7 shows that the average difference between Thesis and Kongsberg methodologies is 3 dB and does not show any frequency or sediment type dependency. The main improvements proposed by Thesis methodology, when compared to real-time corrections are:

i. grazing angle that accounts for the local seafloor slope;

ii. range that accounts for ray tracing path;

iii. cumulative absorption coefficient that were derived from sound speed profiles;

iv. ensonified area that accounts for the local seafloor slope; and

v. limit angle that accounts for local depth; sound speed at the seafloor; pulse length; and across-track beam width.

The average difference between ray tracing (Thesis methodology) and range paths (Kongsberg methodology), for the entire dataset, is 1 meter, which represents 3% of the average ray tracing ranges. This difference is small due to the fact that the survey area is shallow and also because due to unstratified sound speed profile, which minimizes the refraction of the acoustic signal while it travels through the water column. The average difference between the absorption coefficient derived from the sound speed profile and the one applied during real-time data processing is only 3 dB/Km, and with an average range of less than 100 meters, this is only 0.3 dB. The average difference between the difference between *2TL* (Equation 2) that takes into account the absorption coefficient and the range computed by the Thesis methodology versus the same *2TL* computed with Kongsberg parameters is only 1 dB.

Comparing *BS* results derived from Kongsberg and Thesis methodologies (Table 7), the average differences between these two methods is 2.8 dB. Figure 18 shows that the difference between Kongsberg and Thesis' methodologies seems to increase with incident angle. The reason why that happen can be attributed to different values for range and absorption coefficient applied in both methodologies.

The average *BS* values for those different substrates (Figure 25) are: -19 dB for pebble, -31 dB for fine sand and -41 dB for sandy clay. The statistics and *BS* values seem to be coherent

to those calculated using the thesis' dataset and that can be used to assume the MATLAB script used during the data processing is coherent and can be applied to different .all files. For the SH2014 dataset the multi-frequency analysis cannot be done, because only one frequency was used during the data acquisition.

4.2.2 Interpretation of physical controls on observed angular response curves

As shown in Figure 20, the analysis presented here will focus primarily on four different seabed types, which are gravel, sandy gravel, gravelly sand and sandy silt. For gravel and sandy gravel substrates (Figure 23) the *B*S curves as a function of grazing angle are quite flat, which suggests that the seabed surface is rough at all frequencies (Lambertian areas). On other hand, the same figure shows that gravely sand and sandy silt plots present a much steeper roll off with grazing angle. As expected, softer substrates present lower *BS* values than harder substrates, in any frequency (Figure 24). Table 8 and Figure 24 show that at 200 kHz the different types of substrates subgroups (Federal Geographic Data Committee, 2012) are more separated to one another, which indicates that the usage of that frequency seems to be more effective to seabed classification than 300 kHz and 400 kHz data, because that frequency mode shows more contrast among different types of substrates. On the other hand, harder (gravel and sandy gravel) and softer (gravelly sand and sandy silt) substrates are more clearly separated in both 300 and 400 kHz mode than in 200 kHz, which could validate the usage of these two frequencies in cases where information about substrate groups (Federal Geographic Data Committee, 2012) are more important than the seafloor classification itself. The *BS* variation from sandy gravel to gravelly sand is larger than 10 dB for 300 and 400 kHz and only 5 dB for 200 kHz.

Table 8 shows that, for the range of grazing angles used to compute the statistics results, average *BS* does change accordingly to the change in frequency (assuming that the *SL* for the three different frequencies are correct) and to the change in substrate group. In studies that were conducted for frequencies between 10-100 kHz, the frequency dependence is little for rough seabed, but large for softer seabed (Jackson et al., 1986; Weber and Lurton, 2015). Comparing 200 and 400 kHz results, whose frequencies can be considered one octave apart, approximately, the average *BS* difference is somewhere between 4 (sandy gravel) and 7 (gravelly sand) dB. Those results are larger than previous studies which predict a 1.5 to 3 dB changes in average *BS*, per octave (Jackson et al., 1986). Greater changes in *BS* due to changes in frequency are only expected within the normal incident zone (Gensane, 1989).

CHAPTER 5

CONCLUSION

Regarding the results presented by this work, there are pros and cons associated to the fact that the entire dataset shown here has been collected by the same MBES.

Pros:

i. system failures related to the acquisition or data processing stages may affect the entire dataset in the same way;

ii. large dB offsets, as shown in Figure 4, was not observed when the dataset was merged; and

iii. as shown above, a simplified alternative to remove the frequency specific beam pattern and the angular response effect from the original *BS* curves (Figure 22) could be used to get the data rid of the angular dependence, which is inherent to *BS* data.

Cons:

i. the range of available frequencies is reduced and less than an octave away, in most of the cases; and

ii. the usage of the same sonar might add artifacts to the entire dataset, introducing systematic uncertainties to the final results, which may not be easy to figure out and remove.

All statistical analysis shown in this thesis are related to *BS* levels along the plateau region, which in this work address the grazing angular interval from 30º to 60º.

The average difference between the *2TL* computed using Kongsberg and Thesis methodologies was 1 dB. This small figure can be attributed to the fact that the data was collected in shallow waters and, at least during survey hours, the sound speed profiles were barely stratified. For deeper waters or more stratified sound speed profiles, the difference between Kongsberg and Thesis' *2TL* may increase, either because of the difference between ranges or the difference between in-situ and preset absorption coefficient values. As shown in Figure 12, the absorption coefficient changes drastically from 200 kHz to 400 kHz signals (40 dB/Km on average) and that must limit the maximum depth over which the MBES system can be operated using the highest frequency. Due to the importance of the absorption coefficient to the final *BS* values, this variable must be constantly monitored during the entire survey, preferably using sensors that are able to measure temperature, salinity and pressure, such as CTD. Nevertheless studies presented in this thesis has shown that, in some specific situations, absorption coefficient profiles derived directly from sound speed profiles, with assumed values for temperature and salinity, might be accurate enough.

The ensonified area computation models used in this study (Equations 15 and 16) took into account local seafloor slopes (along and across-track) contrary to the ensonified area model used in real-time processing, which assumes that the seafloor is planar. To avoid erroneous slope measurements due to outlier soundings, the local slope was filtered. There are many unknown parameters (Table 10) that may contribute to uncertainties related to ensonified area computation what reinforces the necessity of accounting for local seafloor slopes to compute *BS* (Equation 3).

If the true geometry (slopes and areas) is accounted for, *BS* is not expected to change with respect to changes in pulse length (Gensane, 1989), because its contribution to the reflectivity data is supposed to be normalized when the ensonified area is removed from *BTS*. To confirm that, *BS* values derived from identical 400 kHz lines ran with different pulse lengths were compared. The average differences between 400 kHz *MP* and 400 kHz *LP BS* curves was less than 1 dB for any substrate site. These differences show that changes in pulse length, as expected, might not impact seafloor classification (Hellequin et al., 2003; Malik et al., 2015). However, additional combinations of frequencies and pulse lengths should be investigated to provide more conclusive results.

Similar seabed types, but located in different sites, were compared to verify the uncertainty related to the usage of seafloor backscatter as a reliable remote sense technique that could be used for seafloor characterization purposes. These comparisons end up with an average *BS* difference of 4 dB. This relevant difference can be attributed to: inaccurate bottom samples positioning; imperfect bottom sampling methodology; and small substrates patches within the survey area. As a future direction, the same type of comparison using different types of substrates should be investigated to achieve improved results.

The aim of this thesis is to evaluate how powerful the usage of multi-frequency backscatter data in seafloor characterization can be. To do so, a *BS* dataset was investigated under two different perspectives: The first consists of interpreting how *BS* curves can vary when the same frequency is used to ensonify different types of substrates; and the second consists of verifying the existence of any frequency dependency when the same type of seabed is ensonified with different sonar frequencies. Regarding the studies presented here, to achieve more

consolidated results, a suggestion for a future work is to amplify the spectrum of frequencies and to increase the number of substrates types used.

Under the first perspective, the frequency mode that presented *BS* curves with more contrast between different substrate subgroups is the 200 kHz (Table 8) and the frequency mode with less contrast is the 400 kHz (Figure 24). Both 300 and 400 kHz modes seem to be more efficient than 200 kHz mode to discriminate harder from softer substrate groups (Figure 24). This reinforces that the usage of *BS* can provide enough information to discriminate different substrate groups, although bottom samples are still needed if substrate subgroups classification is desired. Without physical samples, any attempt with the objective of associate changes in *BS* to seafloor types is rather speculative. This analysis shows that the usage multi-frequency backscatter signal as an additional tool to classify the seafloor can have a huge application in the ocean mapping community. Some of the Brazilian Navy's hydrographic survey ships are equipped with different models of Kongsberg MBES. The great majority of those echosounders can operate in frequencies higher than 100 kHz. The ability to decide which of the available operating frequency should be used, based on the desired final product (i.e. seafloor characterization, limitation of an anchor area, submarine nautical charts, etc.) is an efficient information that should be taken into account during survey planning stages.

Under the second perspective, the data analysis shown a frequency dependency component when the same substrate is ensonified with different frequencies. Figure 4 shows that the frequency dependency is one of the factors that can cause the dB offset when mosaics from different MBES are merged. To minimize those dB offsets, the causes of this frequency dependency should be more tested for different frequencies and seabed types.

BS extracted from Kongsberg's datagrams were compared to new *BS* values that accounted for more accurate physical. After removing all real-time corrections implemented during the data acquisition and adding new corrections proposed in this work, the average *BS* difference between Kongsberg and Thesis methodologies was 3 dB, as shown in Table 7. This high difference in *BS* suggests that the replacement of real-time corrections by more accurate physical models should be the best decision to make, although more expensive. Doing this, some sources of uncertainties due to non-realistic assumptions are partially eliminated. Improvements in data processing suggested by this thesis are: grazing angle that accounts for the local seafloor slope; range to the seafloor that accounts for refraction of the signal on its path through the water column; cumulative absorption coefficients; and ensonified area model that accounts for along and across-track slopes.

LIST OF REFERENCES

Ainslie, M. A., and Mccolm, J. G. (1998). "A simplified formula for viscous and chemical absorption in sea water," J. Acoust. Soc. Am., 103, 1671. doi:10.1121/1.421258

APL-UW (1994). High frequency ocean environmental acoustic models handbook (APL-UW TR 9407), Applied Physics Laboratory, University of Washington, Seattle, WA. doi:10.1017/CBO9781107415324.004

Augustin, J. M., and Lurton, X. (2005). "Image amplitude calibration and processing for seafloor mapping sonars," Ocean. 2005, IEEE, Brest, France, 698–701 (Vol. 1).

Beokett, T., and Hussong, D. (1989). "Digital image processing techniques for enha , cement and classification of SeaMARC II side scan sonar imagery," 94, 7469–7490.

Carvalho, R., Oliveira Junior, A. M., and Hughes Clarke, J. E. (2013). "Proper environmental reduction for attenuation in multi-sector sonars," IEEE/ OES RIO Acoust. 2013,.

Dartnell, P., and Gardner, J. V. (2004). "Predicting seafloor facies from multibeam bathymetry and backscatter data," Photogramm. Eng. Remote Sens., 70, 1081–1091. doi:10.14358/PERS.70.9.1081

Federal Geographic Data Committee (2012). FGDC-STD-018-2012 Coast. Mar. Ecol. Classif. Stand.,343 pages.

Fonseca, L., and Mayer, L. (2007). "Remote estimation of surficial seafloor properties through the application Angular Range Analysis to multibeam sonar data," Mar. Geophys. Res., 28, 119–126. doi:10.1007/s11001-007-9019-4

Francois, R. E., and Garrison, G. R. (1982). "Sound absorption based on ocean measurements Part I: Pure Water and magnesium sulfate contributions," J. Acoust. Soc. Am., 72, 896–907. doi:10.1121/1.388673

Francois, R. E., and Garrison, G. R. (1982). "Sound absorption based on ocean measurements Part II: Boric acid contribution and equation for total absorption," J. Acoust. Soc. Am., 72, 1879–1890. doi:10.1121/1.388673

Gensane, M. (1989). "A statistical study of acoustic signals backscattered from the sea," IEEE J. Ocean. Eng., 14, 84–93. doi:10.1109/48.16818

Goodman, J. W. (1976). "Some fundamental properties of speckle," J. Opt. Soc. Am., 66, 1145–1150. doi:10.1364/JOSA.66.001145

Hamilton, E. L. (1970). "Reflection coefficients and bottom losses at normal incidence computed

from pacific sediment properties," Geophysics, 35, 995–1004.

Hamilton, E. L. (1974). "Prediction of deep-sea sediment properties: State-of-the-Art," Nav. Oversea Cent.,.

Hamilton, L. J., and Parnum, I. (2011). "Acoustic seabed segmentation from direct statistical clustering of entire multibeam sonar backscatter curves," Cont. Shelf Res., 31, 138–148. doi:10.1016/j.csr.2010.12.002

Hammerstad, E. (2000). "EM techinical note: Backscattering and seabed image reflectivity.,"

Hasan, R. C., Ierodiaconou, D., and Laurenson, L. (2012). "Combining angular response classification and backscatter imagery segmentation for benthic biological habitat mapping," Estuar. Coast. Shelf Sci., 97, 1–9. doi:10.1016/j.ecss.2011.10.004

Hasan, R. C., Ierodiaconou, D., Laurenson, L., and Schimel, A. (2014). "Integrating multibeam backscatter angular response, mosaic and bathymetry data for benthic habitat mapping," PLoS One, 9, 1–14. doi:10.1371/journal.pone.0097339

Hellequin, L., Boucher, J., and Lurton, X. (2003). "Processing of high-frequency multibeam echo sounder data for seafloor characterization," IEEE J. Ocean. Eng., 28, 78–89. doi:10.1109/JOE.2002.808205

Hughes Clarke, J. E. (2005). "Multibeam Training Course Notes," Swath sonar Train. course given as part Can. Hydrogr. Conf., Ottawa.

Hughes Clarke, J. E. (2015). "Multispectral acoustic backscatter from multibeam, improved classification potential," U.S. Hydro 2015 Conf., 1, 19.

Hughes Clarke, J. E., Iwanowska, K, K., Parrott, R., Duffy, G., Lamplugh, M., and Griffin, J. (2008). "Inter-calibrating multi-source, multi-platform backscatter data sets to assist in compiling regional sediment type maps : Bay of Fundy," Can. Hydrogr. Conf. Natl. Surv. Conf. 2008, Paper 8-2, 22.

International Hydrographic Organization (2008). IHO Standards for Hydrographic Surveys - Special Publication No. 44, S-44, International Hydrographic Bureau, Monaco, 5th ed.

Jackson, D. R., and Richardson, M. D. (2007). High-frequency seafloor acoustics, Springer, N.Y., 616 pages.

Jackson, D. R., Winebrenner, D. P., and Ishimaru, A. (1986). "Application of the composite roughness model to high-frequency bottom backscattering," J. Acoust. Soc. Am., 79, 1410. doi:10.1121/1.393669

Kinsler, L. E., Frey, A. R., Coppens, A. B., and J.V., S. (2000). Fundamentals of acoustics, John

Wiley & Sons, Inc., Hoboken, Fourth.

Kongsberg Maritime (2015). “Kongsberg EM 2040 Multibeam echo sounder - Product description.,”

Kongsberg Maritime (2016). Technical Specifications, EM 2040 Multibeam Echo Sounder Tech. Specif., Available: http://www.km.kongsberg.com/ks/web/nokbg0397.nsf/AllWeb/248996D7F1021D46C12575E500285652/$file/332644_em2040_product_specification.pdf?OpenElement, (date last viewed: 22-Jan-16). Retrieved January 22, 2016, from http://www.km.kongsberg.com/ks/web/nokbg0397.nsf/AllWeb/248996D7F1021D46C12575E500285652/$file/332644_em2040_product_specification.pdf?OpenElement

Lamarche, G., Lurton, X., Verdier, A. L., and Augustin, J. M. (2011). “Quantitative characterisation of seafloor substrate and bedforms using advanced processing of multibeam backscatter—Application to Cook Strait, New Zealand,” Cont. Shelf Res., 31, S93–S109. doi:10.1016/j.csr.2010.06.001

Leroy, C. C., Robinson, S. P., and Goldsmith, M. J. (2008). “A new equation for the accurate calculation of sound speed in all oceans,” J. Acoust. Soc. Am., 124, 2774. doi:10.1121/1.2988296

Llewellyn, K. C. (2006). Corrections for beam pattern residuals in backscatter imagery from the Kongsberg EM300 multibeam echosounder University of New Brunswick, 105 pages.

Lurton, X. (2010). An introduction to underwater acoustics - principles and applications, Springer Heidelberg Dordrecht, London, New York, Second., 680 pages.

Malik, M., Lurton, X., and Mayer, L. (2015). “Evaluation of uncertainty in multibeam echo sounders derived seafloor backscatter data,” US Hydro 2015,.

de Moustier, C., and Alexandrou, D. (1991). “Angular dependence of 12-kHz seafloor acoustic backscatter,” J. Acoust. Soc. Am., 90, 522–531. doi:10.1121/1.401278

NDRC (1946). “National Defence Research Committee Summary Technical Reports,” Phys. Sound Sea, Part II Reverberation, Washington D.C., pp. 308–323.

Preston, J. (2009). “Automated acoustic seabed classification of multibeam images of Stanton Banks,” Appl. Acoust., 70, 1277–1287. doi:10.1016/j.apacoust.2008.07.011

Preston, J. M. (2006). “Acoustic classification of seaweed and sediment with depth-compensated vertical echoes,” Ocean. 2006, IEEE, Boston, USA, 1–5.

Rzhanov, Y., Fonseca, L., and Mayer, L. (2012). “Construction of seafloor thematic maps from multibeam acoustic backscatter angular response data,” Comput. Geosci., 41, 181–187.

doi:10.1016/j.cageo.2011.09.001

Teng, Y. (2012). “Sector-specific beam pattern compensation for multi-sector and multi-swath multibeam Sonars.,”

Weber, T. C. (2015). The Sonar Equation. OE 865 Underw. Acoust. - Lect. Notes,Durham: School of Marine Science and Ocean Engineering, University of New Hampshire.

Weber, T. C., and Lurton, X. (2015). “Background and fundamentals,” In X. Lurton and G. Lamarche (Eds.), Backscatter Meas. by seafloor-mapping sonars. Guidel. Recomm., 1st ed., pp. 25–52. Retrieved from http://geohab.org/wp-content/uploads/2013/02/BWSG-REPORT-MAY2015.pdf

Wentworth, C. K. (1922). “A scale of grade and class terms for clastic sediments,” J. Geol., 30, 377–392.

Wenz, G. M. (1962). “Acoustic ambient noise in the ocean: spectra and sources,” J. Acoust. Soc. Am., 34, 1936. doi:10.1121/1.1909155

APPENDICES

APPENDIX A - TEMPERATURE AND ABSORPTION COEFFICIENT PROFILES

The following is the entirety of the code for deriving absorption coefficient profiles from DIGIBAR Pro sound speed profiles. Temperature profiles were derived from sound speed profiles measured by a DIGIBAR Pro sensor (Equation 21) and absorption coefficient profiles for each of the frequencies operated by Kongsberg EM 2040, which are 190, 205, 270, 282.5, 295, 320 and 380 kHz were computed using Equation 22. This script may present consistent results only if the salinity profile within the survey area is slightly stratified (well mixed).

```
% CCOM / JHC
% Anderson Pecanha
% Created: Oct/10/2015
% Last Update: Mar/22/2016

% To adjust this code for your data, please change: '201506171613' by the name of the sound speed file you want to load.
%'201506171613' is a SSP file formatted as a *.m file (nx2 matrix), where column 1 is depth and column 2 is the related sound
%speed.

% '201506171613' data format (example):
%   D     c
% 1.00 1492.90
% 1.50 1492.60
% 2.00 1492.20
% 2.50 1492.10
% 3.10 1491.80
% 3.50 1491.60
% 4.00 1491.50
% 4.50 1491.40
% 5.00 1491.40
% 5.50 1491.20
% 6.10 1491.10
% 6.50 1491.00
% 7.00 1490.70 ...

clc
clear

load 201506171613.m
[lines,~] = size (X201506171613);

%%Computing Temperature Profiles
S = input('What is the Salinity, in ppt, that is going to be used for this project?? ==> ');

lat = input('What is the latitude of the place where the cast was taken? (Use DD.ddddd format, with 5 decimal places)? ==> ');

%The temperature computations consists of a cubic function, such as aT3 + bT2 + cT + d.
a=zeros(1,lines);
b=zeros(1,lines);
c=zeros(1,lines);
d=zeros(1,lines);
Temperature=zeros(lines,3);
for i = 1:lines
a(i) = 2.1*10^-4;
```

```
b(i) = -5.44*10^-2 + 8.7*10^-5*S + 3*10^-7*X201506171613(i,1);
c(i) = 5 - 1.23*10^-2*S - 9.5*10^-13*X201506171613(i,1)^3;
d(i) = -1*(X201506171613(i,2)) + 1402.5 + 1.33*S + 1.56*10^-2*(X201506171613(i,1))+2.55*10^-7*(X201506171613(i,1))^2-
7.3*10^-12*(X201506171613(i,1))^3+1.2*10^-6*(X201506171613(i,1))*(lat-45)+1.43*10^-5*S*(X201506171613(i,1));
p = [a(i) b(i) c(i) d(i)];
T = roots(p);
Temperature (i,1) = T(1);
Temperature (i,2) = T(2);
Temperature (i,3) = T(3);
end;
T_real = real(Temperature);
D_c_T = [X201506171613(:,1), X201506171613(:,2), T_real(:,3)]; %Depth, sound speed and Temperature - respectively
%columns 1, 2 and 3

Plot_Temperature = input ('Do you want to plot c and T profiles? Yes = 1; No = 0 ==> ');
if Plot_Temperature ==1
figure('Name','Sound Speed Profile')
plot (D_c_T(:,2), -1*D_c_T(:,1), 'r', 'LineWidth', 3.0);
title({'\fontsize{20}Sound Speed Profile'; ...
    ('\fontsize{18} File: 201506171613')});
xlabel('Sound Speed (m/s)', 'Fontsize', 20);
ylabel('Depth (m)', 'Fontsize', 20);
set(gca,'fontsize',20)
grid;

figure('Name','Temperature Profile')
plot (D_c_T(:,3), -1*D_c_T(:,1), 'b', 'LineWidth', 3.0);
title({'\fontsize{20}Temperature Profile'; ...
     ('\fontsize{18} File: 201506171613')});
xlabel('Temperature (ºC)', 'Fontsize', 20);
ylabel('Depth (m)', 'Fontsize', 20);
set(gca,'fontsize',20)
grid;
end;

%% Computing Seawater Absorption and Attenuation
%For this part, Francois and Garrison (1982a) model is going to be used. It is an empirical algorithm to calculate
%compressional wave attenuation (α in dB m^-1) from S (ppt), T (ºC), D (m) and Seawater pH. f1(i) and f2(i) are, respectively, the
%relaxation frequencies (kHz) for magnesium sulfate (MgSO4) and boric acid (H3BO3). The first two terms represent the
%chemical relaxation processes due to boric acid and magnesium sulfate and the last term represents the viscous dissipation in
%pure water. Francois and Garrison estimate their model to be accurate to within about 5%.
%For frequencies of 10-500 kHz (where the MgSO4 contribution dominates), the limits of reliability are:
%-2 < T < 22 °C
%30 < S < 35 ppt
%0 < D < 3.5 km
P1 = 1;
freq = [190 205 270 295 282.5 320 380 320]; %in kHz
size_freq = length(freq);
pH = input('What is the pH of the water (Default Ocean: 7.8 - 8.3)? ==> ') ;

A1 = zeros(lines);
A2 = zeros(lines);
A3 = zeros(lines);
P2 = zeros(lines);
P3 = zeros(lines);
f1 = zeros(lines);
f2 = zeros(lines);
alfa = zeros(lines);
Absorption = zeros(lines,size_freq);

for k = 1:size_freq %1 to the number of frequencies used...
   f = freq(k);
```

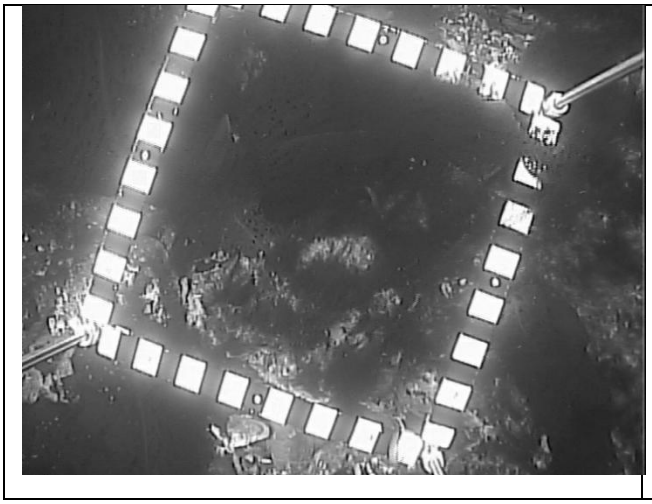
```
for i = 1:lines
   if i > i
   A1(i) = (8.86/(D_c_T(i,2)) * 10^(0.78*pH-5));
   A2(i) = 21.44*S*(1+0.025*D_c_T(i,3))/(D_c_T(i,2));
   P2(i) = 1-1.37*10^-4*(D_c_T(i,1)) + 6.2*10^-9 * (D_c_T(i,1))^2;
   P3(i) = 1-3.83*10^-5*(D_c_T(i,1)) + 4.9*10^-10*(D_c_T(i,1))^2;
   f1(i)= 2.8*((S/35)^0.5) * 10^(4-1245/(273+D_c_T(i,3)));
   f2(i) = (8.17*10^(8-1990/(273+D_c_T(i,3))))/(1+0.0018*(S-35));
      if D_c_T(i,3)<20
      A3(i) = 4.937*10^-4-2.59*10^-5*D_c_T(i,3) + 9.11*10^-7*(D_c_T(i,3))^2 - 1.5*10^-8*(D_c_T(i,3))^3;
      else
      A3(i) = 3.964*10^-4 - 1.146*10^-5*D_c_T(i,3) + 1.45*10^-7*(D_c_T(i,3))^2 - 6.5*10^-10*(D_c_T(i,3))^3;
      end;
alfa(i) = 10^-3*((A1(i)*P1*f1(i)*f^2)/(f^2+f1(i)^2) + (A2(i)*P2(i)*f2(i)*f^2)/(f^2+f2(i)^2) + A3(i)*P3(i)*f^2);
  Absorption(i,k) = alfa(i);
   else
   A1(i) = (8.86/(D_c_T(i,2)) * 10^(0.78*pH-5));
   A2(i) = 21.44*S*(1+0.025*D_c_T(i,3))/(D_c_T(i,2));
   P2(i) = 1-1.37*10^-4*(D_c_T(i,1)) + 6.2*10^-9 * (D_c_T(i,1))^2;
   P3(i) = 1-3.83*10^-5*(D_c_T(i,1)) + 4.9*10^-10*(D_c_T(i,1))^2;
   f1(i)= 2.8*((S/35)^0.5) * 10^(4-1245/(273+D_c_T(i,3)));
   f2(i) = (8.17*10^(8-1990/(273+D_c_T(i,3))))/(1+0.0018*(S-35));
      if D_c_T(i,3)<20
      A3(i) = 4.937*10^-4-2.59*10^-5*D_c_T(i,3) + 9.11*10^-7*(D_c_T(i,3))^2 - 1.5*10^-8*(D_c_T(i,3))^3;
      else
      A3(i) = 3.964*10^-4 - 1.146*10^-5*D_c_T(i,3) + 1.45*10^-7*(D_c_T(i,3))^2 - 6.5*10^-10*(D_c_T(i,3))^3;
      end;
alfa(i) = 10^-3*((A1(i)*P1*f1(i)*f^2)/(f^2+f1(i)^2) + (A2(i)*P2(i)*f2(i)*f^2)/(f^2+f2(i)^2) + A3(i)*P3(i)*f^2);
Absorption(i,k) = alfa(i);
   end;
end;
end;
Absorption = [D_c_T(:,1), D_c_T(:,2), Absorption];

save_absorption = input('Do you want to save the absorption coefficients as a .txt file? Yes = 1; No = 0 ==> ');

if save_absorption ==1
   save(['Absorption_201506171613_Salinity_' num2str(S) '_ppt.txt'], 'Absorption', '-ascii');
   disp('File saved as Absorption_201506171613.txt');
end;

Plot_Absorption = input ('Do you want to plot the absorption profiles? Yes = 1; No = 0 ==> ');
if Plot_Absorption ==1
figure('Name','Absorption Profiles')
plot (Absorption(:,3), -1*Absorption(:,1), 'r','LineWidth', 2.0);
hold on
plot (Absorption(:,4), -1*Absorption(:,1), 'c','LineWidth', 2.0);
hold on
plot (Absorption(:,5), -1*Absorption(:,1), 'k','LineWidth', 2.0);
hold on
plot (Absorption(:,7), -1*Absorption(:,1), 'g','LineWidth', 2.0);
hold on
plot (Absorption(:,6), -1*Absorption(:,1), 'b','LineWidth', 2.0);
hold on
plot (Absorption(:,8), -1*Absorption(:,1), 'y','LineWidth', 2.0);
hold on
plot (Absorption(:,9), -1*Absorption(:,1), 'm','LineWidth', 2.0);
hold on

legend('190 kHz', '205 kHz', '270 kHz', '282.5 kHz', '295 kHz', '320 kHz', '380 kHz', 'Location','northwest')
title({'\fontsize{20}Absorption Profiles'; ...
      ('\fontsize{18} File: 201506171613')});
```

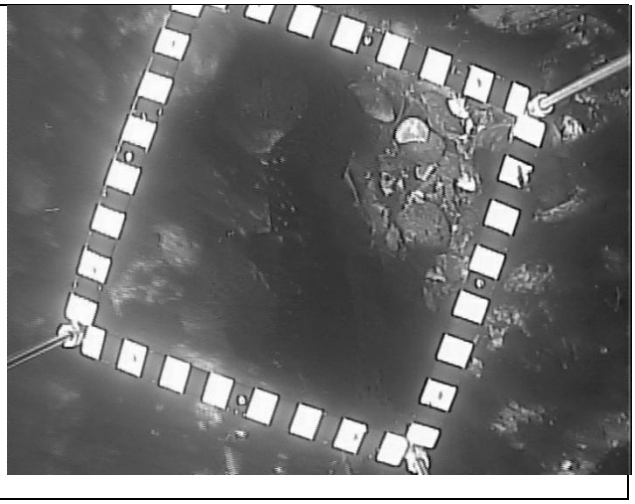
```
xlabel('Absorption (dB/m)', 'Fontsize', 14);
ylabel('Depth (m)', 'Fontsize', 14);
set(gca,'fontsize',14)
grid;
end;

%Absorption Variable Format (columns):
%Depth      c       190 kHz      205 kHz      270 kHz      295 kHz      282.5 kHz      320 kHz      380 kHz      320 kHz

%% Harmonic Mean for the Absorption profiles:
% First step is to convert the absorption value from dB to a real number:

Absorption_real_number = zeros(lines, size_freq);

for i = 1:size_freq
   for k=1:lines
      Absorption_real_number(k,i) = 10^((Absorption(k,i+2)/10));
   end;
end;

HM = [190 0 10*log10(mean(Absorption_real_number(:,1))); 205 0 10*log10(mean(Absorption_real_number(:,1))); 270 0
10*log10(mean(Absorption_real_number(:,1))); 295 0 10*log10(mean(Absorption_real_number(:,1))); 282.5 0
10*log10(mean(Absorption_real_number(:,1))); 320 0 10*log10(mean(Absorption_real_number(:,1))); 380 0
10*log10(mean(Absorption_real_number(:,1))); 320 0 10*log10(mean(Absorption_real_number(:,1)))];
for i = 1: lines
HM(1,2) = HM(1,2) + 1/(Absorption_real_number(i,1));
HM(2,2) = HM(2,2) + 1/(Absorption_real_number(i,2));
HM(3,2) = HM(3,2) + 1/(Absorption_real_number(i,3));
HM(4,2) = HM(4,2) + 1/(Absorption_real_number(i,4));
HM(5,2) = HM(5,2) + 1/(Absorption_real_number(i,5));
HM(6,2) = HM(6,2) + 1/(Absorption_real_number(i,6));
HM(7,2) = HM(7,2) + 1/(Absorption_real_number(i,7));
HM(8,2) = HM(8,2) + 1/(Absorption_real_number(i,8));
end;

HM(1,2) = 10*log10(lines/HM(1,2));
HM(2,2) = 10*log10(lines/HM(2,2));
HM(3,2) = 10*log10(lines/HM(3,2));
HM(4,2) = 10*log10(lines/HM(4,2));
HM(5,2) = 10*log10(lines/HM(5,2));
HM(6,2) = 10*log10(lines/HM(6,2));
HM(7,2) = 10*log10(lines/HM(7,2));
HM(8,2) = 10*log10(lines/HM(8,2));

save_hamonic_mean = input('Do you want to save the Harmonic Mean and the Mean for the absorption as a .txt file? Yes = 1;
No = 0 ==> ');

%Hamonic Mean Variable Format (columns):
%       Freq.       Harmonic Mean       Mean

if save_hamonic_mean ==1
   save(['Harmonic_Mean_and_Mean_201506171613_Salinity_' num2str(S) 'ppt.txt'], 'HM', '-ascii');
   disp('File saved as Harmonic_Mean_and_Mean_201506171613.txt');
end;

clearvars a i k save_absorption save_harmonic_mean
```

APPENDIX B - BOTTOM SAMPLES

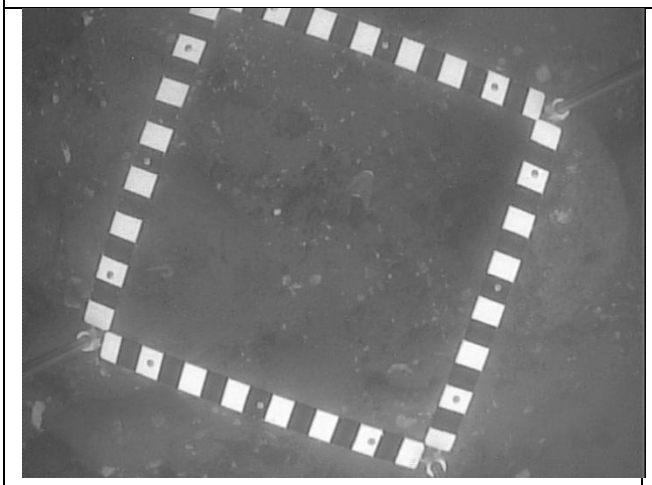
Samples presented in this section were classified according to the size their sediments, based on video (black and white pictures) or grab samples (color pictures) (Wentworth, 1922). Although the grab sampler had been deployed in every stations, in some of them it did not work properly and, as result, there is no grab sample collected from those sites.

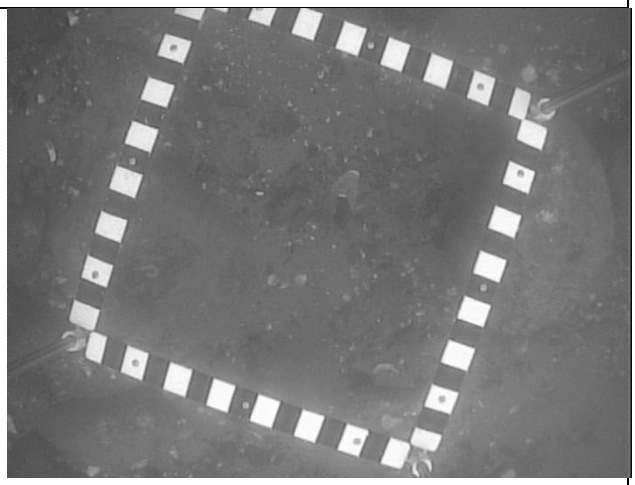
STATION # 1	
Latitude (N)	42º 57.773'
Longitude (W)	70º 44.567'
Depth (m)	16
Size Terms	Pebble (very coarse, coarse, medium) cobble, shell fragments
Class Terms	Gravel
Observation	No grab sample



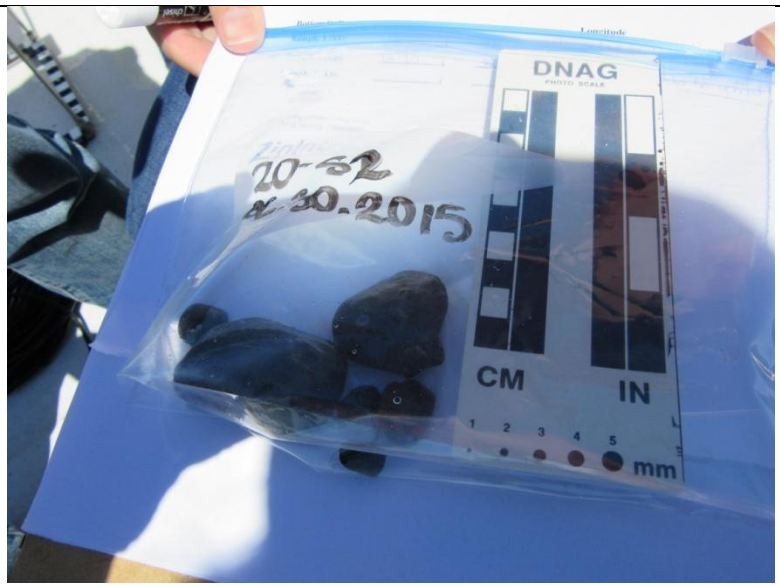


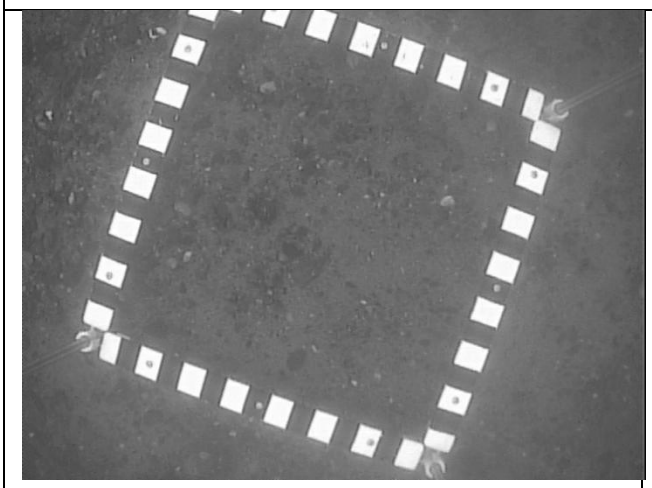
STATION # 2	
Latitude (N)	42º 57.965'
Longitude (W)	70º 44.384'
Depth (m)	26
Size Terms	Pebble (medium, fine), sand, cobble (some), shell fragment (some)
Class Terms	Sandy gravel
Observation	XXX

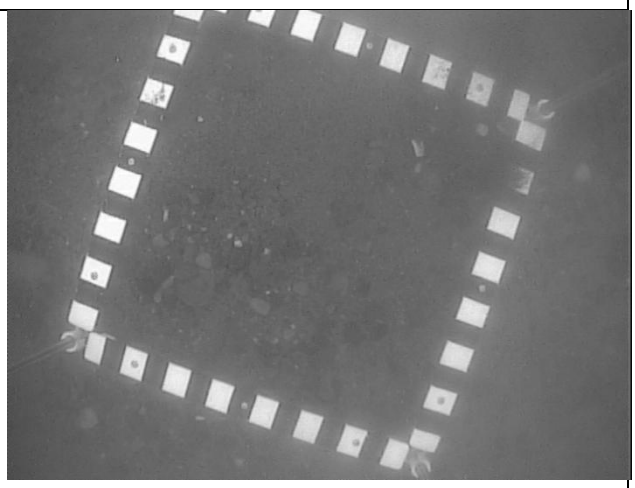




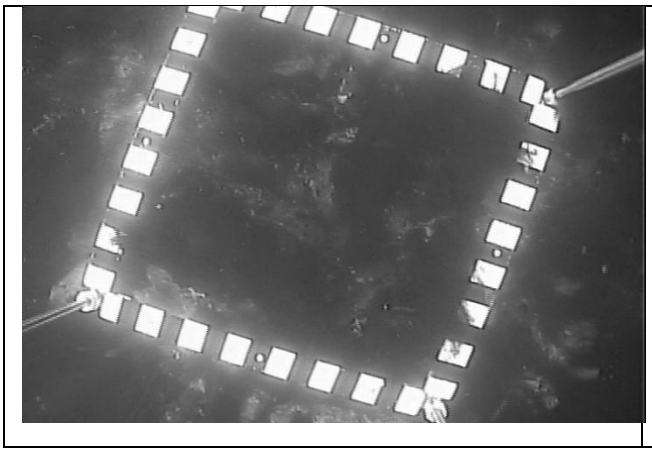
STATION # 3	
Latitude (N)	42º 57.495'
Longitude (W)	70º 44.534'
Depth (m)	26
Size Terms	Pebble (all types), sand, cobble (some), shell fragment (some)
Class Terms	Sandy gravel
Observation	XXX

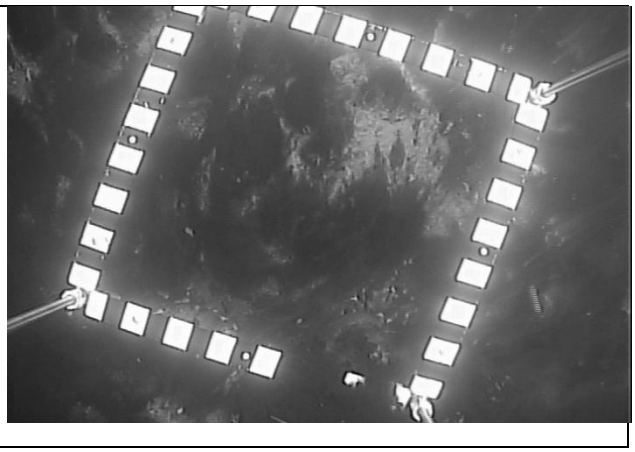




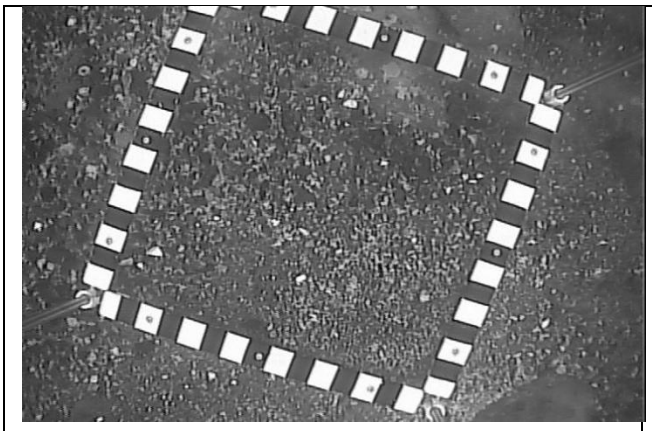


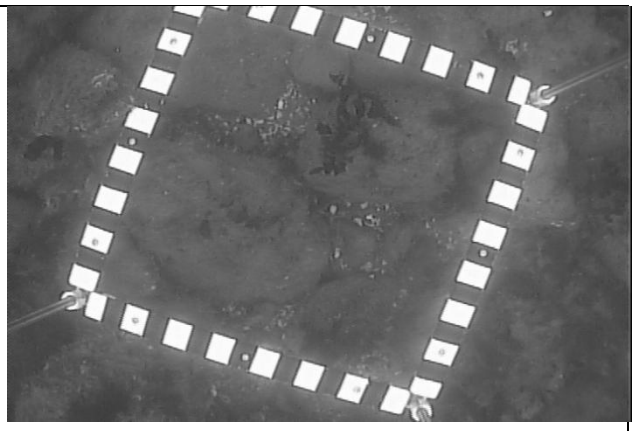
STATION # 4	
Latitude (N)	42º 57.435'
Longitude (W)	70º 44.607'
Depth (m)	20
Size Terms	Cobbles, shell fragment
Class Terms	Gravel
Observation	No grab sample



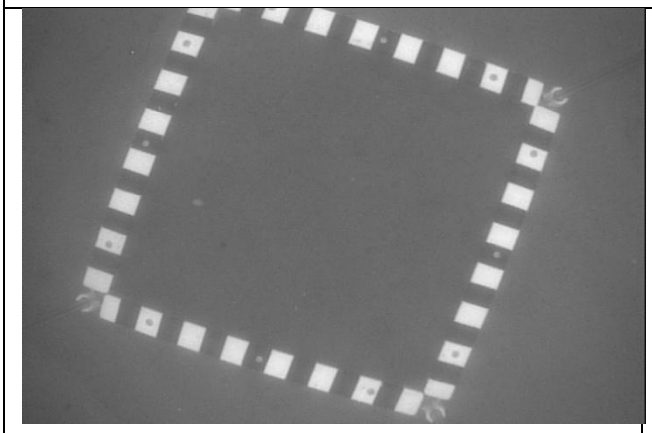


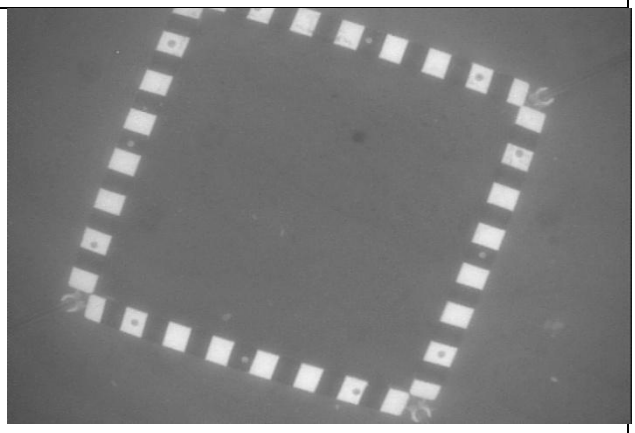
STATION # 5	
Latitude (N)	42º 58.020'
Longitude (W)	70º 44.388'
Depth (m)	25
Size Terms	Cobbles, shell fragment, pebble (some; all types)
Class Terms	Gravel/ sandy gravel
Observation	No grab sample



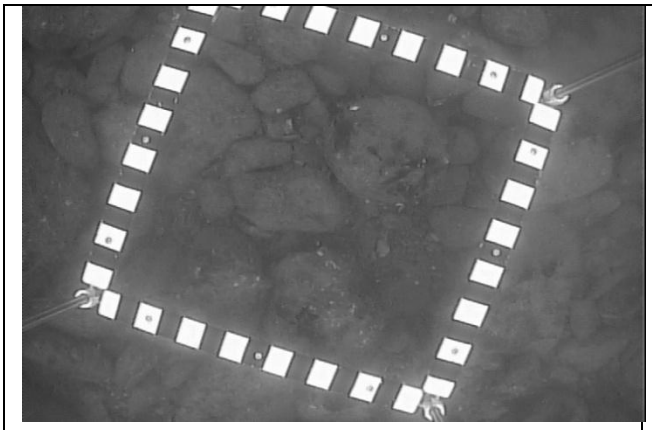


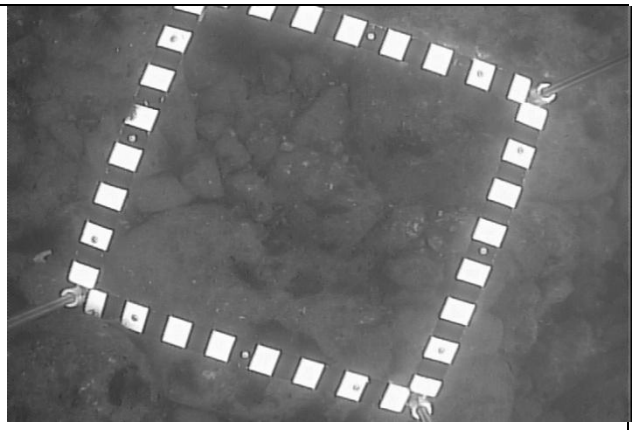
STATION # 6	
Latitude (N)	42º 57.237'
Longitude (W)	70º 44.663'
Depth (m)	28
Size Terms	Sand (very fine), pebble (medium, fine), shell fragment (rare)
Class Terms	Silty sand
Observation	XXX



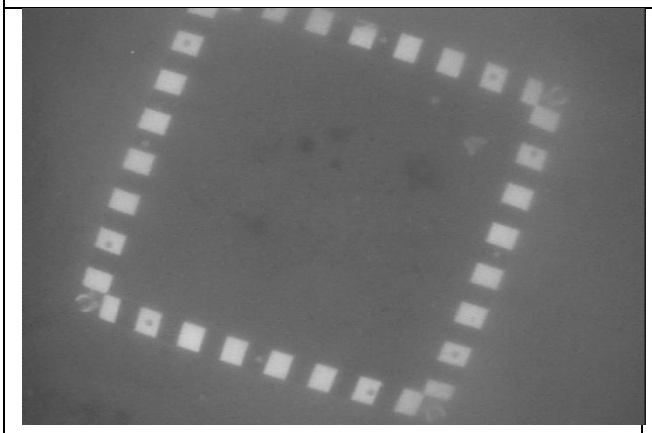


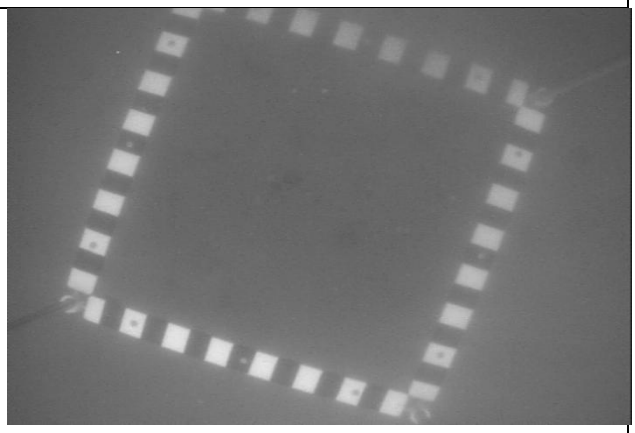
STATION # 7	
Latitude (N)	42º 57.924'
Longitude (W)	70º 44.381'
Depth (m)	25
Size Terms	Pebble (very coarse, coarse), cobble
Class Terms	Gravel
Observation	No grab sample



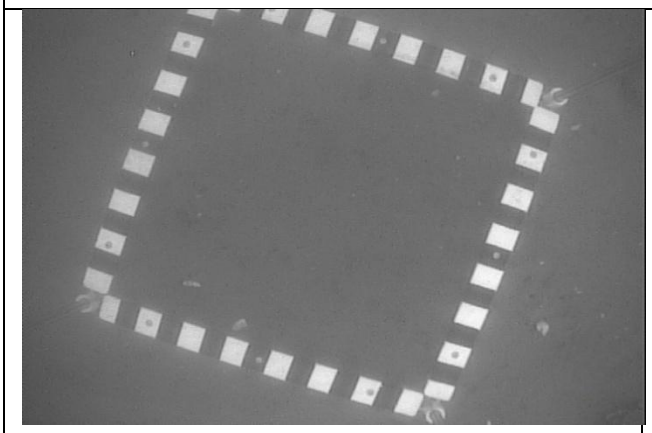


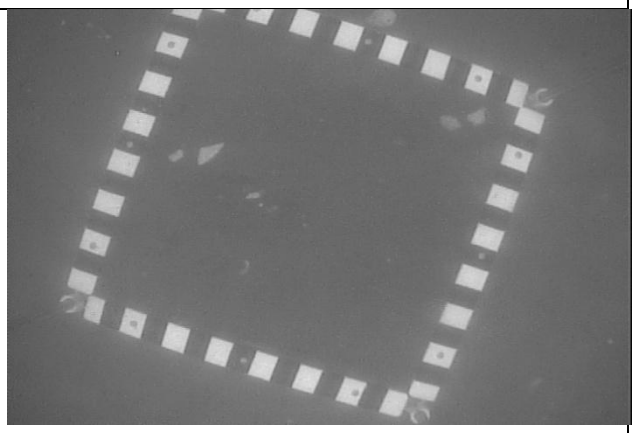
STATION # 8	
Latitude (N)	42º 57.297'
Longitude (W)	70º 44.578'
Depth (m)	29
Size Terms	Sand (very coarse, coarse), pebbles (rare; fine, very fine)
Class Terms	Gravelly sand
Observation	XXX





STATION # 9	
Latitude (N)	42º 57.167'
Longitude (W)	70º 44.764'
Depth (m)	27
Size Terms	Sand (very fine), silt (coarse, medium), pebble (fine, very fine), shell fragment (rare)
Class Terms	Sandy silt
Observation	XXX





APPENDIX C - *all2BS.m* SCRIPT

The following is the entirety of the code for deriving *BS* plots and statistics from Kongsberg raw files (.all format). The data processing can be divided in the following steps:
1) Data parsing;
2) Ray tracing;
3) Local slope;
4) Ensonified area;
5) EM Technical Note (Hammerstad, 2000);
6) Data interpolation and Filtering; and
7) Data export.

In the code below, "Line_254.txt" is the ASCII file exported from CARIS; "Datagram_PU_Line_254.txt", "Datagram_78_Line_254.txt" and "Datagram_89_Line_254.txt" are files exported from .all datagram *PU Status output*, *Raw range and angle 78* and *Seabed image data 89* datagram, respectively (see APPENDIX D); and "Absorption_201406231438.txt" is an example absorption coefficient profile file derived from script presented in APPENDIX A. This script must be run once for every survey line and sample.

```
% CCOM / JHC
% Anderson Pecanha
% Created: Feb/02/2016
% Last Update: Apr/30/2016


% Line_254 Columns (data format of the ASCII file exported from CARIS)
% #1 - Northing (m)
% #2 - Easting (m)
% #3 - Processed Depth (m)
% #4 - Observed Depth (m)
% #5 - Time (hhmmss.sss)
% #6 - Profile from Caris,
% #7 - Beam
% #8 - Amp (db)
% #9 - Alongtrack angle (degrees)
% #10 - Acrosstrack angle (degrees) - Portside angles are negative!

% Example:
% 4758260.113,357635.740,14.517,16.961,17:42:23.440,2,1,-12,-0.21,61.78
% 4758260.027,357635.589,14.483,16.927,17:42:23.440,2,2,-10,-0.21,61.66
%...
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Datagram_PU_Line_254 (data format):
% #1 - Date (yyyymmdd)
% #2 - Easting (hhmmss.sss)
% #3 - Depth to Normal Incidence (m)
% #4 - Range to normal incidence (m)

% This file is generated by dump_PU.m.

% Example:
```

```
% 20150617 184749.752000 0.00 16.00
% 20150617 184750.748000 0.00 16.00
%...
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Datagram_78_Line_254 Columns (data format):
% #1 - Date (yyyymmdd)
% #2 - Time (hhmmss.sss)
% #3 - Ping counter
% #4 - Sound speed at transducer (0.1 m/s)
% #5 - Centre frequency (Hz)
% #6 - Mean absorption coefficient (0.01dB/Km)
% #7 - Transmit sector number/ TX array index
% #8 - Centre frequency (Hz)
% #9 - Mean absorption coefficient (0.01dB/Km)
% #10 - Transmit sector number/ TX array index
% #11 - Centre frequency (Hz) ***
% #12 - Mean absorption coefficient (0.01dB/Km) ***
% #13 - Transmit sector number/ TX array index ***

% From column 11 (200kHz) or column 14 (300kHz and 400 kHz), start two different repeat cycles:
% 1st: Beam pointing angle / Transmit sector number / Reflectivity (BS) in 0.1 dB resolution/ ...
% 2nd: 0 / 0 / Two way travel time, in seconds/...
% *** Those columns does not exist for 200 kHz, because at this frequency the system operates with only 2 sectors instead of 3.

% This file is generated by dump_RawRange_Angle78.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Datagram_89_Line_254 Columns:
% #1 - Date (yyyymmdd)
% #2 - Time (hhmmss.sss)
% #3 - BSn (0.1 dB)
% #4 - BSo (0.1 dB)
% #5 - Range to normal incidence (samples)

% This file is generated by SeabedImageData89Datagram.m

% Example:
% 20150617 184748.759000 -106.00 -135.00 326
% 20150617 184748.872000 -111.00 -128.00 319
% ...
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% 1 - Loading Files

% IMPORTANT NOTICE 1: Replace "Line_254" by the name of the sonar file name that you want input (sonar data) and also
%"Absorption_201406231438" by the name of the sound speed file name you want to input. Both has to be .txt file. Last
%replacement which is necessary is to replace the old file number, in order to adequate figure header format. Only the line
%number, for example: replace all "Line 254" by "Line 305", if the new file name is "Line_305".

% IMPORTANT NOTICE 2: Replace, in all input files, ":" by "" and "/" by "". That will affect date and time columns.

clc
clear

Line_254 = csvread('Line_254.txt'); %In this case, Line 254 is referred to
Datagram_PU_Line_254 = dlmread('Datagram_PU_Line_254.txt');
Datagram_78_Line_254 = dlmread('Datagram_78_Line_254.txt');
Datagram_89_Line_254 = dlmread('Datagram_89_Line_254.txt');

[size_Line_254, ~] = size(Line_254);
max_observed_depth = max(Line_254(:,4));
```

```
[size_Datagram_78_Line_254, ~] = size(Datagram_78_Line_254);
[size_Datagram_89_Line_254, ~] = size(Datagram_89_Line_254);
[size_Datagram_PU_Line_254, ~] = size(Datagram_PU_Line_254);

transducer_depth = 0.54; %transducer depth

% File Statistics:

real_frequency = Datagram_78_Line_254(1,5)/100000;

if real_frequency >=3.2;
   frequency = 4;
   disp('The frequency used during the acquisition was 400 kHz. Which was the pulse length used during ')
   pulselength = input('the acquisition? Please type "1" for 100 microseconds; or "2" for 200 microseconds => ');
   pulselength = pulselength*100*10^-6;
elseif real_frequency >=2.7;
   frequency=3;
   pulselength=2*100*10^-6;
   disp('The frequency used during the acquisition was 300 kHz.')
else
   frequency=2;
   pulselength=2*100*10^-6;
   disp('The frequency used during the acquisition was 200 kHz.')
end;
disp(' ')
pings = 1; % gives, at the end, the amount of pings in the file
Time = Line_254(1,5);

for i = 1:(size_Line_254)-1
   if Line_254(i+1,5) == Time
   else
   pings = pings + 1;
   Time = Line_254(i+1,5);
   end;
end;

initial_time = min(Line_254(:,5));
end_time = max(Line_254(:,5));
num_beams = max(Line_254(:,7)) - min(Line_254(:,7)) + 1;
initial_time_sec = fix(initial_time/10^4)*3600 + (fix((initial_time - 10^4*fix(initial_time/10^4))/100))*60 + initial_time -
(fix(initial_time/10^4)*10^4 + (fix((initial_time - 10^4*fix(initial_time/10^4))/100))*10^2);
end_time_sec = fix(end_time/10^4)*3600 + (fix((end_time - 10^4*fix(end_time/10^4))/100))*60 + end_time -
(fix(end_time/10^4)*10^4 + (fix((end_time - 10^4*fix(end_time/10^4))/100))*10^2);
average_sampling_rate = pings/((end_time_sec-initial_time_sec));
disp(['This MBES file has ' num2str(num_beams) ' beams per swath; ' num2str(pings) ' pings; ' num2str(size_Line_254) ' rows ']);
disp (['and starts at ' num2str(initial_time) ' and ends at ' num2str(end_time) ' (hhmmss.sss).']);
disp(' ')
disp('Would you like to clip the data and analyze just a specific part of it?');
clip_line = input ('Type "0" for NO and "1" for YES => ');

if clip_line ==0
   disp(' ')
   disp('***You have chosen NO, so the input data remained the same!***');
  for aux = 1: size_Line_254;
   Line_254(aux,6) = 1;
   end;
end;

   if clip_line ==1
disp(' ')
sample_number = input('What is the sample number? => ');
```

```
disp(' ')
disp('You have chosen YES, please indicate the start and end time. You can use an external');
disp('software to query the data to decide which is time interval of interest.');
disp('  ')
disp('Please note that your sample has to have at least 200 pings to you be able to get rid of')
disp('part of the unwanted noise. According to the average sampling rate of the file that you have')
sample_size_seconds = fix(250/average_sampling_rate); %250 is to make sure that at least 200 is going to be picked.
disp(['just loaded, the minimum time interval that you are going to define below should be of: ' num2str(sample_size_seconds) '
s.']);
disp(' ');
time_of_interest = input('Which is the time of interest in hhmmss.sss format? => ');
tic %This will show the elapsed time to execute this part of the script.

%Converting the time of interest to seconds:
time_of_interest_sec = fix(time_of_interest/10^4)*3600 + (fix((time_of_interest - 10^4*fix(time_of_interest/10^4))/100))*60 +
time_of_interest - (fix(time_of_interest/10^4)*10^4 + (fix((time_of_interest - 10^4*fix(time_of_interest/10^4))/100))*10^2);
whenstart_mod_sec = time_of_interest_sec - (sample_size_seconds)/2;
whenend_mod_sec = time_of_interest_sec + (sample_size_seconds)/2;
whenstart= fix((whenstart_mod_sec)/3600)*10^4 +  ...
(fix((((whenstart_mod_sec)/3600) - fix((whenstart_mod_sec)/3600))*60))*100 + ...
(((fix((((whenstart_mod_sec)/3600)  -  fix((whenstart_mod_sec)/3600))*60))  -  ((((whenstart_mod_sec)/3600)  -
fix((whenstart_mod_sec)/3600))*60))*60)*-1;
whenend= fix((whenend_mod_sec)/3600)*10^4 +  ...
(fix((((whenend_mod_sec)/3600) - fix((whenend_mod_sec)/3600))*60))*100 + ...
(((fix((((whenend_mod_sec)/3600)  -  fix((whenend_mod_sec)/3600))*60))  -  ((((whenend_mod_sec)/3600)  -
fix((whenend_mod_sec)/3600))*60))*60)*-1;
clearvars whenstart_mod_sec whenend_mod_sec sample_size_seconds
disp(' ');

%Cliping files based on the time of interest selected by the user.

for aux = 1: size_Line_254;
   if (Line_254(aux,5) >=whenstart) && (Line_254(aux,5) <=whenend)
   Line_254(aux,6) = 1;
   else
    Line_254(aux,6) = 0;
   end;
end;

for aux = 1: size_Datagram_78_Line_254;
   if (Datagram_78_Line_254(aux,2) >=whenstart) && (Datagram_78_Line_254(aux,2) <=whenend)
   Datagram_78_Line_254(aux,1) = 1;
   else
   Datagram_78_Line_254(aux,1) = 0;
   end;
end;

 for aux = 1: size_Datagram_89_Line_254;
   if (Datagram_89_Line_254(aux,2) >=whenstart) && (Datagram_89_Line_254(aux,2) <=whenend)
   Datagram_89_Line_254(aux,1) = 1;
   else
   Datagram_89_Line_254(aux,1) = 0;
   end;
 end;

for aux = 1: size_Datagram_PU_Line_254;
   if (Datagram_PU_Line_254(aux,2) >=whenstart) && (Datagram_PU_Line_254(aux,2) <=whenend)
   Datagram_PU_Line_254(aux,1) = 1;
   else
   Datagram_PU_Line_254(aux,1) = 0;
   end;
end;
```

```
TF1 = Line_254(:,6)==0; %logical variable all rows whose column 6th element equals to zero
Line_254(TF1,:) = []; %Line_254 with only the rows that are going to be used according to the time interval set by the user.

TF78 = Datagram_78_Line_254 (:,1) ==0;
Datagram_78_Line_254(TF78,:) = [];

TF89 = Datagram_89_Line_254 (:,1) ==0;
Datagram_89_Line_254(TF89,:) = [];

TFPU = Datagram_PU_Line_254 (:,1) ==0;
Datagram_PU_Line_254(TFPU,:) = [];

clearvars TF1 TF78 TF89 TFPU
   end;

    if clip_line ~=1 && clip_line~=0
error('The number you have just typed does not match with any of the alternatives!');
    end;

[size_Line_254, ~] = size(Line_254);
% [size_Datagram_78_Line_254, columns_Datagram_78_Line_254] = size(Datagram_78_Line_254);
[size_Datagram_89_Line_254, ~] = size(Datagram_89_Line_254);
[size_Datagram_PU_Line_254, ~] = size(Datagram_PU_Line_254);

% Copying the Range to Normal Incidence information from
% Datagra_PU_Line_254 to Line_254:

for i=1:size_Line_254
Line_254(i,26) = fix(Line_254(i,5));
end;

for k=1:size_Datagram_PU_Line_254
Datagram_PU_Line_254(k,2) = fix(Datagram_PU_Line_254(k,2));
end;

for i=1:size_Line_254
   for k = 1:size_Datagram_PU_Line_254
      if Line_254(i,26) ~= Datagram_PU_Line_254(k,2)
      else
         Line_254(i,27) = Datagram_PU_Line_254(k,4);
      end;
   end;
end;

clearvars size_Datagram_PU_Line_254 Datagram_PU_Line_254 i k clip_line end_time_sec initial_time_sec
time_of_interest_sec time_of_interest end_time initial_time

% Copying the BSo and BSn information from
% Datagra_89_Line_254 to Line_254:

% Assigning pings:
pings = 1;
Time = Line_254(1,5);
for i = 1:(size_Line_254)-1
   if Line_254(i+1,5) == Time
     Line_254(i+1,6) = pings; %Here column 6 is filled with the ping # information. That will be useful for many things,
including the alongtrack slope computation.
  else
   pings = pings + 1;
   Line_254(i+1,6) = pings;
```

```
   Time = Line_254(i+1,5);
   end;
end;
Line_254(i,6) = pings;

for i = 1:pings
   Datagram_89_Line_254(i,1) = i;
end;

aux=1;
for i=1:size_Line_254
   if Line_254(i,6) == Datagram_89_Line_254(aux,1)
      Line_254(i,28) = Datagram_89_Line_254(aux,3)/10;
      Line_254(i,29) = Datagram_89_Line_254(aux,4)/10;
       Line_254(i,30) =  Line_254(i,5);
       Line_254(i,31) =  Line_254(i,6);
   else
      Line_254(i,28) = Datagram_89_Line_254(aux+1,3)/10;
      Line_254(i,29) = Datagram_89_Line_254(aux+1,4)/10;
      Line_254(i,30) =  Line_254(i,5);
      Line_254(i,31) =  Line_254(i,6);
      aux=aux+1;
   end;
end;
 clearvars Datagram_89_Line_254

% Now is the most complex part, which comprehends to get information from Datagram_78_Line_328 and copy then into Line_254 data. Time is going to be used as an index to do that:
%%First part: Copy sound speed at the transducer:
for i = 1:pings
   Datagram_78_Line_254(i,1) = i;
end;
aux=1;
for i=1:size_Line_254
   if Line_254(i,6) == Datagram_78_Line_254(aux,1)
      Line_254(i,30) = Datagram_78_Line_254(aux,4)/10;
   else
      Line_254(i,30) = Datagram_78_Line_254(aux+1,4)/10;
      aux=aux+1;
   end;
end;

%Second Part: Get the information about different transmit sectors, frequencies and mean absorption coefficient applied by %Kongsberg during the acquisition.

if frequency ==2
   Freq_MeanAbs_Sector = zeros(3,3);
   Freq_MeanAbs_Sector(1,1) = Datagram_78_Line_254(1,5)/1000;
   Freq_MeanAbs_Sector(1,2) = -1*Datagram_78_Line_254(1,6)/100;
   Freq_MeanAbs_Sector(1,3) = Datagram_78_Line_254(1,7);
   Freq_MeanAbs_Sector(2,1) = Datagram_78_Line_254(1,8)/1000;
   Freq_MeanAbs_Sector(2,2) = -1*Datagram_78_Line_254(1,9)/100;
   Freq_MeanAbs_Sector(2,3) = Datagram_78_Line_254(1,10)-1;
   Freq_MeanAbs_Sector(3,1) = Datagram_78_Line_254(1,8)/1000;
   Freq_MeanAbs_Sector(3,2) = -1*Datagram_78_Line_254(1,9)/100;
   Freq_MeanAbs_Sector(3,3) = Datagram_78_Line_254(1,10);
%Although the system operates only with two different frequencies while operating with 200kHz, and according to Datagram %78 the datagrams numbers are 0 and 2, it was noticed that when the datagram list the TX sector number, per beam, the TX %sector is listed as 1 instead of 2 and, because of this, Freq_MeanAbs_Sector(2,:) was added to the matrix above.
else
   Freq_MeanAbs_Sector = zeros(3,3);
   Freq_MeanAbs_Sector(1,1) = Datagram_78_Line_254(1,5)/1000;
```

```
   Freq_MeanAbs_Sector(1,2) = -1*Datagram_78_Line_254(1,6)/100;
   Freq_MeanAbs_Sector(1,3) = Datagram_78_Line_254(1,7);
   Freq_MeanAbs_Sector(2,1) = Datagram_78_Line_254(1,8)/1000;
   Freq_MeanAbs_Sector(2,2) = -1*Datagram_78_Line_254(1,9)/100;
   Freq_MeanAbs_Sector(2,3) = Datagram_78_Line_254(1,10);
   Freq_MeanAbs_Sector(3,1) = Datagram_78_Line_254(1,11)/1000;
   Freq_MeanAbs_Sector(3,2) = -1*Datagram_78_Line_254(1,12)/100;
   Freq_MeanAbs_Sector(3,3) = Datagram_78_Line_254(1,13);
end;

% Variable Freq_MeanAbs_Sector format:
% 200 kHz (two Tx Sectors):
% Freq    Mean Abs  Tx sector
% 190     -41.88       0
% 205     -44.52       2

% 300 or 400 kHz (three Tx Sectors):
% Freq    Mean Abs  Tx sector
% 320     -70.44       0
% 380     -87.94       1
% 320     -70.44       2

% Third part: Defining Tx sectors:

if frequency==2
   Datagram_78_Line_254(:,1:10) = [];
   [size_Datagram_78_Line_254, columns_Datagram_78_Line_254] = size(Datagram_78_Line_254);
   AUX_Line_254 = zeros(size_Datagram_78_Line_254,4);
   aux=1;
   for i=1:size_Datagram_78_Line_254
      for k = 1:columns_Datagram_78_Line_254/6

         AUX_Line_254(aux,1) = -1*Datagram_78_Line_254(i,k*3-2)/100; %Pointing angle
         AUX_Line_254(aux,2) = Datagram_78_Line_254(i,k*3-1); %Sector number
         AUX_Line_254(aux,3) = Datagram_78_Line_254(i,k*3)/100; %BS from datagram
         aux=aux+1 ;

      end;

   end;
   aux=1;
   for i=1:size_Datagram_78_Line_254
      for k = 1:columns_Datagram_78_Line_254/6
         Line_254(aux,34) = Datagram_78_Line_254(i,k*3+1200); %TWTT
         AUX_Line_254(aux,4)= Datagram_78_Line_254(i,k*3+1200); %TWTT
         aux=aux+1;
     end;
   end;
else
Datagram_78_Line_254(:,1:13) = [];
   [size_Datagram_78_Line_254, columns_Datagram_78_Line_254] = size(Datagram_78_Line_254);
   aux=1;
   AUX_Line_254 = zeros(size_Datagram_78_Line_254,4);
   for i=1:size_Datagram_78_Line_254
      for k = 1:columns_Datagram_78_Line_254/6

         AUX_Line_254(aux,1) = -1*Datagram_78_Line_254(i,k*3-2)/100; %Pointing angle
         AUX_Line_254(aux,2) = Datagram_78_Line_254(i,k*3-1); %Sector number
         AUX_Line_254(aux,3) = Datagram_78_Line_254(i,k*3)/100; %BS from datagram
         aux=aux+1 ;

      end;
```

```
   end;
   aux=1;
   for i=1:size_Datagram_78_Line_254
      for k = 1:columns_Datagram_78_Line_254/6
         AUX_Line_254(aux,4)= Datagram_78_Line_254(i,k*3+1200); %TWTT
         aux=aux+1;
      end;

   end;
end;

aux = 1;
for i=1:length(AUX_Line_254)
   if aux<size_Line_254
         if AUX_Line_254(i,1) == Line_254(aux,10)
         Line_254(aux,31) = AUX_Line_254(i,1);
         Line_254(aux,32) = AUX_Line_254(i,2);
         Line_254(aux,33) = AUX_Line_254(i,3)*10;
         Line_254(aux,34) = AUX_Line_254(i,4);
         aux=aux+1;
         else
         end;
   end;
end;

% Computing R (range) = 1/2*c*TWTT

for i = 1:size_Line_254
   Line_254(i,35) = 0.5*Line_254(i,30)*Line_254(i,34);
end;
pings=pings-1;
disp(['The new file has ' num2str(pings) ' pings.']);
pings=pings+1;
clearvars AUX_Line_254 i k real_frequency Datagram_78_Line_254 columns_Datagram_78_Line_254
size_Datagram_78_Line_254 size_Datagram_89_Line_254

% Now the sound speed and absorption coefficient profiles that are going to compute the ray tracing and the alfa*R components
%for each segment of ray-tracing has to be loaded. Segments here are referred to the path run by the acoustic signal which is
%between two different layers. The SSP file can be obtained by the file: Digibar2Absorption.m, where the absorption coefficient
%profiles are computed from a DIGIBAR Pro sound speed profile.

% Note: The average depth of the transducer below the sea surface, for this survey, is 0.54 meter and that MUST BE the first
%depth in the SSP profile. To guarantee that the SSP will cover the entire water column, the max_observed_depth and
%max_SSP_depth are compared and, if the latest is bigger additional lines are added to the SSP file to allow the ray-tracing
%computation.

Absorption_201406231438 = dlmread ('Absorption_201406231438_Salinity_31.5_ppt.txt');
max_SSP_depth = max(Absorption_201406231438(:,1));
[size_Absorption_201406231438, ~] = size(Absorption_201406231438);

Diff_SSP_MBES_Depths =  abs(round (3.9*max_observed_depth - max_SSP_depth)/0.5); % Here it is considered 3.9 times the
%maximum observed depth because the swath angle considered is +-75º and depth/cos(75º) = 3.9*depth gives a precision of, at
%least, 0.5 m to the ray tracing computation

   for i = 1: (Diff_SSP_MBES_Depths + 20)
   Absorption_201406231438(size_Absorption_201406231438 + i,1) =
Absorption_201406231438(size_Absorption_201406231438 + i - 1,1) + 0.5;
   Absorption_201406231438(size_Absorption_201406231438 + i,2) =
Absorption_201406231438(size_Absorption_201406231438 + i - 1,2);
   Absorption_201406231438(size_Absorption_201406231438 + i,3) =
Absorption_201406231438(size_Absorption_201406231438 + i - 1,3);
```

```
   Absorption_201406231438(size_Absorption_201406231438 + i,4) =
Absorption_201406231438(size_Absorption_201406231438 + i - 1,4);
   Absorption_201406231438(size_Absorption_201406231438 + i,5) =
Absorption_201406231438(size_Absorption_201406231438 + i - 1,5);
   Absorption_201406231438(size_Absorption_201406231438 + i,6) =
Absorption_201406231438(size_Absorption_201406231438 + i - 1,6);
   Absorption_201406231438(size_Absorption_201406231438 + i,7) =
Absorption_201406231438(size_Absorption_201406231438 + i - 1,7);
   Absorption_201406231438(size_Absorption_201406231438 + i,8) =
Absorption_201406231438(size_Absorption_201406231438 + i - 1,8);
   Absorption_201406231438(size_Absorption_201406231438 + i,9) =
Absorption_201406231438(size_Absorption_201406231438 + i - 1,9);
   Absorption_201406231438(size_Absorption_201406231438 + i,10) =
Absorption_201406231438(size_Absorption_201406231438 + i - 1,10);
  end;

% The specifications listed below were gotten from the EM2040-07 Technical Specifications.

clearvars i

if frequency == 4
   beamwidth_along = 0.7*pi/180; %radians
   beamwidth_across = 0.7*pi/180; %radians
elseif frequency == 3
   beamwidth_along = 1*pi/180; %radians
   beamwidth_across = 1*pi/180; %radians
elseif frequency == 2
   beamwidth_along = 1.5*pi/180; %radians
   beamwidth_across = 1.5*pi/180; %radians
end;

%% 2 - Computing the Grazing Angle and alfa*R
% The grazing angle is computed taking into account refractions of the acoustic signal which are caused by different sound speed
%values (different water densities) through the water column. To address this, Snell's Law was used. The alfa*R factor of the TL
%is also computed at this stage. As an % approximation, the ray tracing is assumed to be within the acrosstrack angle.
% The provided absorption coefficient profile is applied to each part of the ray-tracing between two consecutive layers. The
%observed depth is used as a parameter to interrupt the loop.
% Defining which are the absorption coefficient values according to the Tx sector.

path = zeros(1,size_Line_254);
grazing = zeros(1,size_Line_254);
theta_t = zeros(1,size_Line_254);
alfaR = zeros(3,size_Line_254); % Line 1: sector 0 / Line 2: sector 1 / Line 3: sector 2
old_alfa = zeros(1,size_Line_254);

if frequency ==2
   abs_TX_0 = 3;
   abs_TX_2 = 4;

   for line = 1:(size_Line_254-1)
   aux = 1;
%First layer:
   path(line) = (Absorption_201406231438(aux+1,1) - Absorption_201406231438(aux,1))/cos(Line_254(aux, 10)*pi/180); %first
path segment considering that the initial trigger angle is known
   theta_t (line) = asin((sin(Line_254(line, 10)*pi/180))*
Absorption_201406231438(aux+1,2)/Absorption_201406231438(aux,2)); %theta2
   alfaR (1,line) = Absorption_201406231438(aux,abs_TX_0) * path(line);
   alfaR (3,line) = Absorption_201406231438(aux,abs_TX_2) * path(line);
   alfaR (2,line) = alfaR (3,line);

   %Other layers, except the last:
     while Line_254 (line,4) >  Absorption_201406231438(aux+2,1)
```

```
      path (line) = path (line) + (Absorption_201406231438(aux+2,1) - Absorption_201406231438(aux+1,1))/cos(theta_t(line));
      alfaR (1,line) = alfaR (1,line) + Absorption_201406231438(aux+1,abs_TX_0)*(Absorption_201406231438(aux+2,1) -
Absorption_201406231438(aux+1,1))/cos(theta_t(line));
      alfaR (3,line) = alfaR (3,line) + Absorption_201406231438(aux+1,abs_TX_2)*(Absorption_201406231438(aux+2,1) -
Absorption_201406231438(aux+1,1))/cos(theta_t(line));
      alfaR (2,line) = alfaR (3,line);
      theta_t (line) = asin((sin(theta_t(line))* Absorption_201406231438(aux+1,2)/Absorption_201406231438(aux,2)));
      aux = aux + 1;
      end;

    %Last Line
      path (line) = path (line) + (Line_254 (line,4) - Absorption_201406231438(aux+1,1))/cos(theta_t(line));
      alfaR (1,line) = alfaR (1,line) + Absorption_201406231438(aux+1,abs_TX_0)*(Line_254 (line,4) -
Absorption_201406231438(aux+1,1))/cos(theta_t(line));
      alfaR (3,line) = alfaR (3,line) + Absorption_201406231438(aux+1,abs_TX_2)*(Line_254 (line,4) -
Absorption_201406231438(aux+1,1))/cos(theta_t(line));
      alfaR (2,line) = alfaR (3,line);
      theta_t(line) = asin((sin(theta_t(line))* Absorption_201406231438(aux+1,2)/Absorption_201406231438(aux,2)));
      grazing(line) = pi * 0.5 - theta_t(line);  %the grazing angle is 90º - incident angle
        if theta_t(line)>=0
       grazing(line) = pi * 0.5 - theta_t(line);  %the grazing angle is 90º - incident angle
       else
       grazing(line) = -pi * 0.5 - theta_t(line);  %the grazing angle is 90º - incident angle
       end;
    end;
else
       if frequency==3;
       abs_TX_0 = 5;
       abs_TX_1 = 6;
       abs_TX_2 = 7;
       elseif frequency==4
       abs_TX_0 = 8;
       abs_TX_1 = 9;
       abs_TX_2 = 10;
       end;

    for line = 1:(size_Line_254-1)
    aux = 1;
    %First layer:
    path(line) = (Absorption_201406231438(aux+1,1) - Absorption_201406231438(aux,1))/cos(Line_254(aux, 10)*pi/180); %first
path segment considering that the initial trigger angle is known
    theta_t (line) = asin((sin(Line_254(line, 10)*pi/180))*
Absorption_201406231438(aux+1,2)/Absorption_201406231438(aux,2)); %theta2
    alfaR (1,line) = Absorption_201406231438(aux,abs_TX_0) * path(line);
    alfaR (2,line) = Absorption_201406231438(aux,abs_TX_1) * path(line);
    alfaR (3,line) = Absorption_201406231438(aux,abs_TX_2) * path(line);

  %Other layers, except the last:
       while Line_254 (line,4) >  Absorption_201406231438(aux+2,1)
       path (line) = path (line) + (Absorption_201406231438(aux+2,1) - Absorption_201406231438(aux+1,1))/cos(theta_t(line));
       alfaR (1,line) = alfaR (1,line) + Absorption_201406231438(aux+1,abs_TX_0)*(Absorption_201406231438(aux+2,1) -
Absorption_201406231438(aux+1,1))/cos(theta_t(line));
       alfaR (2,line) = alfaR (2,line) + Absorption_201406231438(aux+1,abs_TX_1)*(Absorption_201406231438(aux+2,1) -
Absorption_201406231438(aux+1,1))/cos(theta_t(line));
       alfaR (3,line) = alfaR (3,line) + Absorption_201406231438(aux+1,abs_TX_2)*(Absorption_201406231438(aux+2,1) -
Absorption_201406231438(aux+1,1))/cos(theta_t(line));
       theta_t (line) = asin((sin(theta_t(line))* Absorption_201406231438(aux+1,2)/Absorption_201406231438(aux,2)));
       aux = aux + 1;
       end;
    path (line) = path (line) + (Line_254 (line,4) - Absorption_201406231438(aux+1,1))/cos(theta_t(line));
    alfaR (1,line) = alfaR (1,line) + Absorption_201406231438(aux+1,abs_TX_0)*(Line_254 (line,4) -
Absorption_201406231438(aux+1,1))/cos(theta_t(line));
```

```
   alfaR (2,line) = alfaR (2,line) + Absorption_201406231438(aux+1,abs_TX_1)*(Line_254 (line,4) -
Absorption_201406231438(aux+1,1))/cos(theta_t(line));
   alfaR (3,line) = alfaR (3,line) + Absorption_201406231438(aux+1,abs_TX_2)*(Line_254 (line,4) -
Absorption_201406231438(aux+1,1))/cos(theta_t(line));
   theta_t(line) = asin((sin(theta_t(line))* Absorption_201406231438(aux+1,2)/Absorption_201406231438(aux,2)));
   if theta_t(line)>=0
      grazing(line) = pi * 0.5 - theta_t(line);  %the grazing angle is 90º - incident angle
   else
      grazing(line) = -pi * 0.5 - theta_t(line);  %the grazing angle is 90º - incident angle
   end;
   end;
end;
Line_254(size_Line_254+1:end,:) = [];
Line_254(:,11) = grazing';
Line_254(:,12) = path';
Line_254(:,13) = old_alfa';
Line_254(:,23) =  alfaR (1,:)'; %Tx sector 0
Line_254(:,24) =  alfaR (2,:)'; %Tx sector 1
Line_254(:,25) =  alfaR (3,:)'; %Tx sector 2
clearvars alfaR grazing path line old_alfa theta_t

%% 3 - Computing TL

%TL is given by: 20 * log10 R + alfa * R. The second parcel was computed in the previous section, and corresponds to the
%column 13th of the Line_254 variable. At this section the first parcel is computed and added up to the alfa*R to get the total
%value for TL. The MBES system used is a monostatic, thus TL need to be computed twice in order to account for the two-way
%travel.

% First: To correlate sector number with alfa*R values computed in columns 23, 24, 25 - This information if going to be storage
% in column 13
for i = 1:size_Line_254

   if Line_254(i,32)==0
   Line_254(i,13)=23;
   elseif Line_254(i,32)==1
   Line_254(i,13)=24;
   elseif Line_254(i,32)==2
   Line_254(i,13)=25;
   end;
end;

% Determining which value for alfa*R is going to be used, for each beam.
% This information is going to be assigned in Column 13 (Ray tracing Method)

for i = 1:size_Line_254
   Line_254(i,13)=Line_254(i,Line_254(i,13));
end;

% Second: To correlate sector number with alfa*R values used by SIS,
% according to information available in .all datagram

for i = 1:size_Line_254
   if Line_254(i,32)==0
   Line_254(i,36)=(Freq_MeanAbs_Sector(1,2)/-1000)*Line_254(i,35);
   elseif Line_254(i,32)==1
   Line_254(i,36)=(Freq_MeanAbs_Sector(2,2)/-1000)*Line_254(i,35);
   elseif Line_254(i,32)==2
   Line_254(i,36)=(Freq_MeanAbs_Sector(3,2)/-1000)*Line_254(i,35);
   end;
end;
```

```
%Now, computing 2TL using the ray tracing and absorption profiles (Column 14) versus 2TL using SIS abs coefficient and
%Range = 1/2*c*TWTT (Column 37):

for i=1:size_Line_254
   Line_254(i,14) = 2 * (20*log10(Line_254(i,12)) + Line_254(i,13));
   Line_254(i,37) = 2 * (20*log10(Line_254(i,35)) + Line_254(i,36));
   Line_254(i,38) = Line_254(i,37) - Line_254(i,14); %2TL (Kongsberg methodology) - 2TL (Thesis methodology)
end;
clearvars  abs_TX_0 abs_TX_1 abs_TX_2

%% 4 - Computing Across-Track Slope and the grazing angle compensated by slope values

% To compute the acrosstrack seafloor inclination and use it to better estimate a value for grazing angle, the observed depth and
%the distance between two consecutive beams are going to be used. There are 3 situations about the seafloor inclination that has
%to be considered:
% a) Flat Seafloor: In this case, the grazing angle is going to the one computed in section #2;
% b) Seafloor gets deeper from starboard to port: In this case, the seafloor inclination need to be added up to the grazing angle
%computed in #2; and
% c) Seafloor gets shallower from starboard to port: In this case, the seafloor inclination will be a negative number and need to be
%added up to the grazing angle computed in #2.
%   b) and c) assumptions can be applied to both starboard and port sides. The initial acrosstrack angles (pointing angles),
%exported by CARIS (column 10) are positive for starboard side angles and they are negative for portside.

% First Step is to distinguish different pings using the TIME variable as parameter; Second Step is to compute the distance
%between two consecutive pings and, with the difference in processed depth between them, the seafloor inclination angle can
% be computed; Third and last step is to compute the final grazing angle, which will be the one computed in #2 added up with
%the seafloor inclination angle. Port and Starboard sides beams are taking into account in this script.

%The final value for the grazing angle is:

Time= Line_254 (1,5);
Line_254(size_Line_254,:) = [];
size_Line_254 = size_Line_254-1;
for i = 1:(size_Line_254)-1
   if Line_254(i+1,5) == Time
   Line_254(i+1,16) = atan ((Line_254(i+1,3)-Line_254(i,3))/ (sqrt ((Line_254(i+1,1) - Line_254(i,1))^2 + (Line_254(i+1,2) -
Line_254(i,2))^2)));
   else
   Time = Line_254(i+1,5);
   end;
end;

% As long as the seafloor slope is computed from starboard to port, for every ping, the value for slope inclination for the first
%starboard ping will always be equal to zero. To fix that, it will be assumed that the seafloor inclination in those cases are the
%same computed for the consecutive ping. The loop below is to implement that correction:

Line_254(1,16) = Line_254(2,16); %The first row value is assumed to be equal to the second row value.

aux=1;
for i=2:size_Line_254-2
   if Line_254(i,6)-Line_254(i+1,6)~=0
      Line_254(i+1,16) = Line_254(i+2,16);
      aux=aux+1;
   end;
end;

% Another situation that need to be fixed are those cases where the grazing angle greater than 90º, which is impossible, because
%the grazing angle can only vary from 0º to 90º. Errors like these may occur if the sum the grazing angle computed by ray
%tracing (column 11) and the seafloor across-track slope is greater than 90º. In cases like this, the grazing angle is going to be
%assumed as the complement angle (180º - angle).

avg_acrosstrackslope=mean(Line_254(:,16));
```

```
std_acrosstrackslope=std(Line_254(:,16))*3; % 99.7% of CL

for i=1:size_Line_254
   if Line_254(i,16)<(avg_acrosstrackslope-std_acrosstrackslope)
   Line_254(i,16)=avg_acrosstrackslope-std_acrosstrackslope;
   elseif Line_254(i,16)>(avg_acrosstrackslope+std_acrosstrackslope)
   Line_254(i,16)=avg_acrosstrackslope+std_acrosstrackslope;
   end;
end;

for i=1:size_Line_254
   Line_254(i,15) = Line_254(i,11) + Line_254(i,16);
end;

for i=1:size_Line_254
   if Line_254(i,15)>90*pi/180
      Line_254(i,15) = Line_254(i,15) - pi;
   elseif Line_254(i,15)<-90*pi/180
      Line_254(i,15) = Line_254(i,15) + pi;
   end;
end;
clearvars i aux

%% 5 - Computing Seafloor inclination - Along-Track Slope

% Computing some statistics to clean blunders

ping = 1;
aux_file = Line_254;
amount =1;
n_per_beam = zeros(1,pings);


for i = 1:(size_Line_254)-1
   if aux_file(i+1,6) == ping
      aux_file(i+1,:) = aux_file(i+1,:) + aux_file(i,:);
      aux_file(i+1,6) = ping;
      amount = amount + 1;
   else
   n_per_beam(ping) = amount;
   ping = ping + 1;
   amount= 1;
   end;
end;

sum_n_per_beam = zeros(1,pings);
n_per_beam(ping) = amount;
sum_n_per_beam(1) = n_per_beam(1);
averages = zeros(pings,38);

for i = 1:pings
   if i == 1
   averages(i,:) = aux_file(n_per_beam(i),:)./n_per_beam(i);

   else
   sum_n_per_beam(i) = n_per_beam(i) + sum_n_per_beam(i-1);
   averages(i,:) = aux_file(sum_n_per_beam(i),:)./n_per_beam(i);
   end;
end;

% Average Columns (each line represent a different ping):
```

```
% Column 1: Northing
% Column 2: Easting
% Column 3: Processed Depth
% Column 4: Alongtrack Slope (to be assigned below)

%Average distance between consecutive pings:
distances = zeros (1,pings-1);
for i = 1:pings-1;
   distances (i) = ((averages(i+1,1)-averages(i,1))^2 + (averages(i+1,2)-averages(i,2))^2)^0.5;
end;

avg_distance = sum(distances)/(pings-1); %in meters
clearvars distances

% Alongtrack slope (taking into account just the average depth and position
% between two consecutive pings)
for i = 1:pings-1
   averages(i+1,4) = atan((averages(i,3) - averages(i+1,3))/ sqrt((averages(i,1)-averages(i+1,1))^2 + (averages(i,2)-
averages(i+1,2))^2));
end;
averages(1,4) = averages(2,4);

aux = 1;
for i = 1:pings-1
   while Line_254(aux,6) ==i
      Line_254(aux,17) = averages(i,4);
      aux = aux+1;
   end;
end;

for i = aux:size_Line_254
   Line_254(i,17) = averages(pings,4);
end;

% Now computing the along track slope, ping-by-ping and beam-by-beam:
for i = 1:pings

   if i==1
      n_per_beam(2,1) = n_per_beam(1,1);
   else
      n_per_beam(2,i) = n_per_beam(1,i) + n_per_beam(2,i-1);
   end;
end;

for i=2:pings-1
   if n_per_beam(1,i) == n_per_beam(1,i+1)
      for k=1:n_per_beam(1,i)
      Line_254(k+n_per_beam(2,i-1),22)=atan((Line_254(k+n_per_beam(2,i-1),3) - Line_254(n_per_beam(2,i)+k,3))/ sqrt(
(Line_254(k+n_per_beam(2,i-1),1)- Line_254(n_per_beam(2,i)+k,1))^2 + (Line_254(k+n_per_beam(2,i-1),2)-
Line_254(n_per_beam(2,i)+k,2))^2));
      end;
  elseif n_per_beam(1,i) < n_per_beam(1,i+1)
      for k=1:n_per_beam(1,i)
         Line_254(k+n_per_beam(2,i-1),22)=atan((Line_254(k+n_per_beam(2,i-1),3) - Line_254(n_per_beam(2,i)+k,3))/ sqrt(
(Line_254(k+n_per_beam(2,i-1),1)- Line_254(n_per_beam(2,i)+k,1))^2 + (Line_254(k+n_per_beam(2,i-1),2)-
Line_254(n_per_beam(2,i)+k,2))^2));
      end;
  elseif n_per_beam(1,i) > n_per_beam(1,i+1)
      for k=1:n_per_beam(1,i)
         if k<=n_per_beam(1,i+1)
```

```
        Line_254(k+n_per_beam(2,i-1),22)=atan((Line_254(k+n_per_beam(2,i-1),3) - Line_254(n_per_beam(2,i)+k,3))/ sqrt( (Line_254(k+n_per_beam(2,i-1),1)- Line_254(n_per_beam(2,i)+k,1))^2 + (Line_254(k+n_per_beam(2,i-1),2)- Line_254(n_per_beam(2,i)+k,2))^2));
        else
        Line_254(k+n_per_beam(2,i-1),22) = 0;
        end;
      end;
   end;
end;

   if n_per_beam(1,1) ==n_per_beam(1,2)
      for i = 1:n_per_beam(2,1)
      Line_254(i,22)=atan((Line_254(i,3) - Line_254(n_per_beam(2,1)+i,3))/ sqrt( (Line_254(i,1)- Line_254(n_per_beam(2,1)+i,1))^2 + (Line_254(i,2)- Line_254(n_per_beam(2,1)+i,2))^2));
      end;

   elseif n_per_beam(1,1)<n_per_beam(1,2)
      for i = 1:n_per_beam(2,1)
      Line_254(i,22)=atan((Line_254(i,3) - Line_254(n_per_beam(2,1)+i,3))/ sqrt( (Line_254(i,1)- Line_254(n_per_beam(2,1)+i,1))^2 + (Line_254(i,2)- Line_254(n_per_beam(2,1)+i,2))^2));
      end;

    elseif n_per_beam(1,1)>n_per_beam(1,2)
       for i=1:n_per_beam(2,1)
          if i<=n_per_beam(1,2)
          Line_254(i,22)=atan((Line_254(i,3) - Line_254(n_per_beam(2,1)+i,3))/ sqrt( (Line_254(i,1)- Line_254(n_per_beam(2,1)+i,1))^2 + (Line_254(i,2)- Line_254(n_per_beam(2,1)+i,2))^2));
          else
          Line_254(i,22) = 0;
          end;
       end;
   end;


for i=1:n_per_beam(1,pings) %Alongtrack slope for the last ping is assumed to be equal to zero, because there is not a next ping to use as parameter to compute it.
   Line_254(n_per_beam(2,pings-1)+i,22) = 0;
end;

% Now the last ping must be removed from the files, because it is not possible compute alongtrack slope for them. Other files %must be reduced too, as follows:

Line_254 = Line_254(1:n_per_beam(2,pings-1),:);
averages = averages(1:pings-1,:);
size_Line_254 = size_Line_254 - n_per_beam(1,pings);
ping = ping-1;
pings = pings-1;
clearvars amount aux aux_file i ping sum_n_per_beam k amount aux i k time_of_interest_sec

%% 6 - Ensonified area, per beam.
% First step is to define what is the angle that limits the near nadir and the oblique zones. Equations 10 and 11 are used as criteria %to determine which is the crossover angle. Depending on the average depth for each ping, c is chosen from the sound speed %profile as show below:

c = zeros(1,pings);
for i = 1:pings
   diff_d = 1000;
   aux = 1;
      while diff_d > 0.5
         c (i) = Absorption_201406231438(aux+1,2);
         diff_d = abs (averages(i,3) - Absorption_201406231438(aux,1));
         aux = aux+1;
```

```
      end;
  end;
averages(:,4) = c';

% Now that all parameters to compute the crossover angle are known, the limit angle will be computed for each ping, as shown
%below:

theta_limit = zeros(3,pings);
for i = 1:pings
   theta_limit(1,i) = acos (( 1 + (averages(i,4) * pulselength)/(2*averages(i,3)))^-1);
end;

for i = 1:pings
   theta_limit(2,i) = asin (-(beamwidth_across * averages(i,3))/(averages(i,4) * pulselength) + sqrt(1 + ((beamwidth_across *
averages(i,3))/(averages(i,4) * pulselength))^2));
end;

% Comparing Lines 1 and 2 from theta_limit. For each column, the largest values are going to be assumed as the crossover angle.

for i = 1:pings
   if theta_limit(1,i)> theta_limit(2,i)
   theta_limit(3,i) = theta_limit(1,i);
   else
   theta_limit(3,i) = theta_limit(2,i);
   end;
end;
averages(:,5) = theta_limit(3,:)';

% Adding two new columns to Line_254 file:
aux = 1;
for i = 1:pings
   while Line_254(aux,6) == i && aux <size_Line_254
      Line_254 (aux,18) = averages(i,4); %c in water, immediately above the seafloor
      Line_254 (aux,19) = averages(i,5); %theta_limit (near-nadir/oblique incidence angle)
      aux = aux+1;
   end;
end;

%Last row:
Line_254 (aux,18) = Line_254 (aux-1,18); %c in water, immediately above the seafloor
Line_254 (aux,19) = Line_254 (aux-1,19); %theta_limit (near-nadir/oblique incidence angle)

% Now, the ensonified area for each beam is going to be  computed, taking into consideration the theta limit calculated above.
%Just to remember, BTS = BS + 10*Log10(area).

%10*log10(Area for a non-flat seafloor):
for i = 1:size_Line_254
   if abs(Line_254(i,10)*pi/180) <= abs(Line_254(i,19)) %Near-Nadir condition
      Line_254(i,20) = 10*log10(abs((beamwidth_across * beamwidth_along * (Line_254(i,35))^2)/ (cos(Line_254(i,16)) *
cos(Line_254(i,22)))));
   else %Oblique zone.
   Line_254(i,20) = 10*log10(    abs(    (    Line_254(i,18) * pulselength * beamwidth_along * Line_254(i,35)    )    / (    2 *
cos(Line_254(i,10))*pi/180 - Line_254(i,16) * cos(Line_254(i,22))   )   )   );
   end;
end;

% 10log10(Area for a flat seafloor):
for i = 1:size_Line_254
   if abs(Line_254(i,10)*pi/180) <= abs(10*pi/180) %Near-Nadir condition. The default crossover angle for KM EM2040 is 10º.
      Line_254(i,21) = 10*log10((abs(beamwidth_across * beamwidth_along * (Line_254(i,35)^2))));
  else
```

```
   Line_254(i,21) = 10*log10(abs(0.5 * Line_254(i,18) * pulselength * beamwidth_along * (Line_254(i,35))/sin(Line_254(i,10)*pi/180)));
    end;
end;
clearvars i aux max_SSP_depth diff_d aux c Diff_SSP_MBES_Depths theta_limit Time

%% 7 - Backscattering and Seabed Reflectivity - EM Technical Note - Hammerstad, E. (2000)
% First is to compute BS model using KM criteria, whose Crossover angle is equal to 10º

for i=1:size_Line_254
   if Line_254(i,35)<Line_254(i,27)
   Line_254(i,39) = Line_254(i,28);
   elseif Line_254(i,35) >= Line_254(i,27)*1/(cos(10*pi/180))
   Line_254(i,39) = Line_254(i,29) - 5*log10((Line_254(i,35)/Line_254(i,27))^2) * ((Line_254(i,35)/Line_254(i,27))^2-1
   elseif Line_254(i,35) > Line_254(i,27) && Line_254(i,35) < Line_254(i,27)*1/(cos(10*pi/180))
   Line_254(i,39) = ((Line_254(i,28) + 3.162 * sqrt((Line_254(i,35)/Line_254(i,27))-1) * (Line_254(i,29)- Line_254(i,28)) - 5*log10((Line_254(i,35)/Line_254(i,27))^2) * ((Line_254(i,35)/Line_254(i,27))^2-1)));
   end;
end;

% Second is to compute BS model using the crossover angle computed in
% section #6
for i=1:size_Line_254
   if Line_254(i,35)<Line_254(i,27)
   Line_254(i,40) = Line_254(i,28);
   elseif Line_254(i,35) >= Line_254(i,27)*1/(cos(Line_254(i,19)))
   Line_254(i,40) = (Line_254(i,29) - 5*log10((Line_254(i,35)/Line_254(i,27))^2) * ((Line_254(i,35)/Line_254(i,27))^2-1));
   elseif Line_254(i,35) > Line_254(i,27) && Line_254(i,35) < Line_254(i,27)*1/(cos(Line_254(i,19)))
   Line_254(i,40) = (Line_254(i,28) + 3.162 * sqrt((Line_254(i,35)/Line_254(i,27))-1) * (Line_254(i,29)- Line_254(i,28)) - 5*log10((Line_254(i,35)/Line_254(i,27))^2) * ((Line_254(i,35)/Line_254(i,27))^2-1));
   end;
end;

% Now that the KM Model are known it is necessary to remove the effect of the KM model (column 39)  and the flat seafloor %area from the BS values in column 33 and then reapply the KM model with the new criteria (column 40) and remove the sloped %seafloor area effect to get back the BS information.

% Column 41: KM model (crossover 10º) + 10log10(flat seafloor area) + BS - 2TL (KM range method): Column 39 + Column %21 + Column 33 - Column 37 (KM Range method)

% Column 42: Column 41 - 10log10(sloped seafloor area) + 2TL (ray tracing method): Columns 41 - Column 40 - Column 20 + %Column 14

for i=1:size_Line_254
    Line_254(i,41) = Line_254(i,39) + Line_254(i,21) - Line_254(i,37) + Line_254(i,33);
    Line_254(i,42) = Line_254(i,41) - Line_254(i,20) + Line_254(i,14);
    Line_254(i,41) = Line_254(i,39) + Line_254(i,33); %Here is the BS considering a flat seafloor and using the 2TL model used by Kongsberg
    Line_254(i,44) = Line_254(i,39) + Line_254(i,33) + Line_254(i,21) - Line_254(i,20); %KM TL, slope Area
    Line_254(i,45) = Line_254(i,39) + Line_254(i,33) + Line_254(i,14) - Line_254(i,37);% Ray tracing TL and flat seafloor area.
    Line_254(i,43) = Line_254(i,29) + 20*log10(sin(abs(Line_254(i,15))));
end;

%% 8 - Data interpolation
% An Pre_Interp matrix need to be created in order to interpolate the data This matrix is a (400 x pings*4) matrix, where, for %each ping, the four columns are:

Pre_Interp = zeros(400,pings*6);
 column=1;
 row=1;
 for i=1:size_Line_254
          if Line_254(i,6)== column
```

```
            Pre_Interp(row,column*6-5) = Line_254(i,15)*180/pi; %%
            Pre_Interp(row,column*6-4) = 10^(Line_254(i,33)/10); %dB converted into real numbers
            Pre_Interp(row,column*6-3) = 10^(Line_254(i,41)/10); %dB converted into real numbers
            Pre_Interp(row,column*6-2) = 10^(Line_254(i,42)/10); %dB converted into real numbers
            Pre_Interp(row,column*6-1) = 10^(Line_254(i,44)/10); %dB converted into real numbers
            Pre_Interp(row,column*6-0) = 10^(Line_254(i,45)/10); %dB converted into real numbers
            row=row+1;
        else
        row=1;
        column=column+1;
        Pre_Interp(row,column*6-5) = Line_254(i,15)*180/pi; %%
        Pre_Interp(row,column*6-4) = 10^(Line_254(i,33)/10); %dB converted into real numbers
        Pre_Interp(row,column*6-3) = 10^(Line_254(i,41)/10); %dB converted into real numbers
        Pre_Interp(row,column*6-2) = 10^(Line_254(i,42)/10); %dB converted into real numbers
        Pre_Interp(row,column*6-1) = 10^(Line_254(i,44)/10); %dB converted into real numbers
        Pre_Interp(row,column*6-0) = 10^(Line_254(i,45)/10); %dB converted into real numbers
        row=row+1;
        end;
end;

 % Eliminating NaN values:
Pre_Interp(isnan(Pre_Interp))=0;
% The purpose here is to use only 200 pings and to do so, all columns greater than 800 need to be deleted from Pre_Interp

[~,num_columns_Pre_Interp]=size(Pre_Interp);

if  num_columns_Pre_Interp>1200
   for i=1:(num_columns_Pre_Interp-1200)
      Pre_Interp(:,(num_columns_Pre_Interp-1200)+1200-i+1)=[];
   end;
end;

[~,num_columns_Pre_Interp]=size(Pre_Interp);
% Before sorting the grazing angles, it is important to replace empty values, showed as "0", by previous values:

for i=1:num_columns_Pre_Interp/6
   for k=1:400 %num of beams
        if Pre_Interp(k,i*6-5)==0 && Pre_Interp(k,i*6-4)==0  %%
        Pre_Interp(k,i*6-5) = Pre_Interp(k-1,i*6-5)-0.001;
        Pre_Interp(k,i*6-4) = Pre_Interp(k-1,i*6-4);
        Pre_Interp(k,i*6-3) = Pre_Interp(k-1,i*6-3);
        Pre_Interp(k,i*6-2) = Pre_Interp(k-1,i*6-2);
        Pre_Interp(k,i*6-1) = Pre_Interp(k-1,i*6-1);
        Pre_Interp(k,i*6-0) = Pre_Interp(k-1,i*6-0);
        end;
   end;
end;

% Sorting Rows, per ping
for i=1:num_columns_Pre_Interp/6
   Temp=Pre_Interp(:,i*6-5:i*6);
   Temp_Sort = sortrows(Temp);
   Pre_Interp(:,i*6-5:i*6) = Temp_Sort;
end;
clearvars Temp Temp_Sort

% Here, all consecutive angles that are equals will be changed by and increment of 0.01º. To eliminate values of X that are
%consecutive and not distinct is mandatory to proceed with the interpolation:

aux=0.01;
for i=1:num_columns_Pre_Interp/6
   for k=1:399
```

```
      if Pre_Interp(k+1,i*6-5)==Pre_Interp(k,i*6-5) && Pre_Interp(k,i*6-5)>0 %%
        Pre_Interp(k+1,i*6-5) = Pre_Interp(k,i*6-5)-aux; %%
        aux=aux+0.01; %%
      elseif Pre_Interp(k+1,i*6-5)==Pre_Interp(k,i*6-5) && Pre_Interp(k,i*6-5)<0 %%
        Pre_Interp(k+1,i*6-5) = Pre_Interp(k,i*6-5)+aux*-1; %%
        aux=aux+0.01;
      end;
   end;
end;

% To determine the interval of interpolation, the first step is to check out the maximum and the minimum angle that must be
%used, in order to use the same interpolation criteria and angles for the entire dataset.

% separating positive and negative angles!
Pre_Interp_starboard = Pre_Interp;
Pre_Interp_port = Pre_Interp;

for i=1:num_columns_Pre_Interp/6
   for k=1:400
      if Pre_Interp_starboard(k,i*6-5)<0
        Pre_Interp_starboard(k,i*6-5)=0;
        Pre_Interp_starboard(k,i*6-4)=0;
        Pre_Interp_starboard(k,i*6-3)=0;
        Pre_Interp_starboard(k,i*6-2)=0;
        Pre_Interp_starboard(k,i*6-1)=0;
        Pre_Interp_starboard(k,i*6-0)=0;
     end;
      if Pre_Interp_port(k,i*6-5)>0
        Pre_Interp_port(k,i*6-5)=0;
        Pre_Interp_port(k,i*6-4)=0;
        Pre_Interp_port(k,i*6-3)=0;
        Pre_Interp_port(k,i*6-2)=0;
        Pre_Interp_port(k,i*6-1)=0;
        Pre_Interp_port(k,i*6-0)=0;
      end;
   end;
end;

Pre_Interp_starboard(1:190,:)=[];
Pre_Interp_port(211:400,:)=[];

% Eliminating angles = 0
% Portside
for i=1:num_columns_Pre_Interp/6
   for k=1:210
       if k==1 && Pre_Interp_port(1,i*6-5)==0 % this conditional is necessary
%        in cases where only starboard BS information was record by the system.
         Pre_Interp_port(:,i*6-5)= Pre_Interp_port(:,i*6-5-6);
         Pre_Interp_port(:,i*6-4)= Pre_Interp_port(:,i*6-4-6);
         Pre_Interp_port(:,i*6-3)= Pre_Interp_port(:,i*6-3-6);
         Pre_Interp_port(:,i*6-2)= Pre_Interp_port(:,i*6-2-6);
         Pre_Interp_port(:,i*6-1)= Pre_Interp_port(:,i*6-1-6);
         Pre_Interp_port(:,i*6-0)= Pre_Interp_port(:,i*6-0-6);
       elseif Pre_Interp_port(k,i*6-5)==0
         Pre_Interp_port(k,i*6-5)= Pre_Interp_port(k-1,i*6-5)+0.001;
         Pre_Interp_port(k,i*6-4)= Pre_Interp_port(k-1,i*6-4);
         Pre_Interp_port(k,i*6-3)= Pre_Interp_port(k-1,i*6-3);
         Pre_Interp_port(k,i*6-2)= Pre_Interp_port(k-1,i*6-2);
         Pre_Interp_port(k,i*6-1)= Pre_Interp_port(k-1,i*6-1);
         Pre_Interp_port(k,i*6-0)= Pre_Interp_port(k-1,i*6-0);
       end;
   end;
```

```
end;

% Starboard side
for i=1:num_columns_Pre_Interp/6
   for k=2:210
      if Pre_Interp_starboard(210-k+1,i*6-5)==0
         Pre_Interp_starboard(210-k+1,i*6-5)= Pre_Interp_starboard(210-k+1+1,i*6-5)-0.001;
         Pre_Interp_starboard(210-k+1,i*6-4)= Pre_Interp_starboard(210-k+1+1,i*6-4);
         Pre_Interp_starboard(210-k+1,i*6-3)= Pre_Interp_starboard(210-k+1+1,i*6-3);
         Pre_Interp_starboard(210-k+1,i*6-2)= Pre_Interp_starboard(210-k+1+1,i*6-2);
         Pre_Interp_starboard(210-k+1,i*6-1)= Pre_Interp_starboard(210-k+1+1,i*6-1);
         Pre_Interp_starboard(210-k+1,i*6-0)= Pre_Interp_starboard(210-k+1+1,i*6-0);
      end;
   end;
end;

% Determining the max and min angles for port and starboard:
angles_starboard=zeros(210,num_columns_Pre_Interp/6);
angles_port=zeros(210,num_columns_Pre_Interp/6);
for i =1:num_columns_Pre_Interp/6
   angles_starboard(:,i) = Pre_Interp_starboard(:,i*6-5); %%
   angles_port(:,i) = Pre_Interp_port(:,i*6-5); %%
end;
min_angle_starboard = min(min(angles_starboard));
min_angle_port = min(min(angles_port));
max_angle_starboard = max(max(angles_starboard));
max_angle_port = max(max(angles_port));

% Now that the start and end angle, 'min_angle' and 'max_angle', respectively
% are known, 'xi' represents the new angles to be used in interpolation.
% Note that the interval of interpolation is 1º

% xi=fix(min_angle)+1:1:fix(max_angle)-1;
xi_starboard=fix(min_angle_starboard):1:fix(max_angle_starboard);
length_xi_starboard = length(xi_starboard);

xi_port=fix(min_angle_port):1:fix(max_angle_port);
length_xi_port = length(xi_port);

% 'Interpolation' Matrix is going to storage angles and interpolated values,
% for each ping. 'BS_interp' is going to be used to storage only the
% interpolated values for BS. The average of each line of this matrix,
% conjugated with 'xi'angles is the final result.

Interpolation_33_port = zeros(length_xi_port,num_columns_Pre_Interp/3);
Interpolation_41_port = zeros(length_xi_port,num_columns_Pre_Interp/3);
Interpolation_42_port = zeros(length_xi_port,num_columns_Pre_Interp/3);
Interpolation_44_port = zeros(length_xi_port,num_columns_Pre_Interp/3);
Interpolation_45_port = zeros(length_xi_port,num_columns_Pre_Interp/3);

BS_33_port = zeros(length_xi_port,num_columns_Pre_Interp/6);
BS_41_port = zeros(length_xi_port,num_columns_Pre_Interp/6);
BS_42_port = zeros(length_xi_port,num_columns_Pre_Interp/6);
BS_44_port = zeros(length_xi_port,num_columns_Pre_Interp/6);
BS_45_port = zeros(length_xi_port,num_columns_Pre_Interp/6);

Interpolation_33_starboard = zeros(length_xi_starboard,num_columns_Pre_Interp/3);
Interpolation_41_starboard = zeros(length_xi_starboard,num_columns_Pre_Interp/3);
Interpolation_42_starboard = zeros(length_xi_starboard,num_columns_Pre_Interp/3);
Interpolation_44_starboard = zeros(length_xi_starboard,num_columns_Pre_Interp/3);
Interpolation_45_starboard = zeros(length_xi_starboard,num_columns_Pre_Interp/3);
```

```
BS_33_starboard = zeros(length_xi_starboard,num_columns_Pre_Interp/6);
BS_41_starboard = zeros(length_xi_starboard,num_columns_Pre_Interp/6);
BS_42_starboard = zeros(length_xi_starboard,num_columns_Pre_Interp/6);
BS_44_starboard = zeros(length_xi_starboard,num_columns_Pre_Interp/6);
BS_45_starboard = zeros(length_xi_starboard,num_columns_Pre_Interp/6);

for i=1:num_columns_Pre_Interp/6
   Interpolation_33_port(:,i*2-1)=xi_port';
   yi_33_port=interp1(Pre_Interp_port(:,i*6-5),Pre_Interp_port(:,i*6-4),xi_port); %% %%%
   Interpolation_33_port(:,i*2) = yi_33_port';
   BS_33_port(:,i) = yi_33_port';

   Interpolation_41_port(:,i*2-1)=xi_port';
   yi_41_port=interp1(Pre_Interp_port(:,i*6-5),Pre_Interp_port(:,i*6-3),xi_port); %% %%%
   Interpolation_41_port(:,i*2) = yi_41_port';
   BS_41_port(:,i) = yi_41_port';

   Interpolation_42_port(:,i*2-1)=xi_port';
   yi_42_port=interp1(Pre_Interp_port(:,i*6-5),Pre_Interp_port(:,i*6-2),xi_port); %% %%%
   Interpolation_42_port(:,i*2) = yi_42_port';
   BS_42_port(:,i) = yi_42_port';

   Interpolation_44_port(:,i*2-1)=xi_port';
   yi_44_port=interp1(Pre_Interp_port(:,i*6-5),Pre_Interp_port(:,i*6-1),xi_port); %% %%%
   Interpolation_44_port(:,i*2) = yi_44_port';
   BS_44_port(:,i) = yi_44_port';

   Interpolation_45_port(:,i*2-1)=xi_port';
   yi_45_port=interp1(Pre_Interp_port(:,i*6-5),Pre_Interp_port(:,i*6-0),xi_port); %% %%%
   Interpolation_45_port(:,i*2) = yi_45_port';
   BS_45_port(:,i) = yi_45_port';

   Interpolation_33_starboard(:,i*2-1)=xi_starboard';
   yi_33_starboard=interp1(Pre_Interp_starboard(:,i*6-5),Pre_Interp_starboard(:,i*6-4),xi_starboard); %% %%%
   Interpolation_33_starboard(:,i*2) = yi_33_starboard';
   BS_33_starboard(:,i) = yi_33_starboard';

   Interpolation_41_starboard(:,i*2-1)=xi_starboard';
   yi_41_starboard=interp1(Pre_Interp_starboard(:,i*6-5),Pre_Interp_starboard(:,i*6-3),xi_starboard); %% %%%
   Interpolation_41_starboard(:,i*2) = yi_41_starboard';
   BS_41_starboard(:,i) = yi_41_starboard';

   Interpolation_42_starboard(:,i*2-1)=xi_starboard';
   yi_42_starboard=interp1(Pre_Interp_starboard(:,i*6-5),Pre_Interp_starboard(:,i*6-2),xi_starboard); %% %%%
   Interpolation_42_starboard(:,i*2) = yi_42_starboard';
   BS_42_starboard(:,i) = yi_42_starboard';

   Interpolation_44_starboard(:,i*2-1)=xi_starboard';
   yi_44_starboard=interp1(Pre_Interp_starboard(:,i*6-5),Pre_Interp_starboard(:,i*6-1),xi_starboard); %% %%%
   Interpolation_44_starboard(:,i*2) = yi_44_starboard';
   BS_44_starboard(:,i) = yi_44_starboard';

   Interpolation_45_starboard(:,i*2-1)=xi_starboard';
   yi_45_starboard=interp1(Pre_Interp_starboard(:,i*6-5),Pre_Interp_starboard(:,i*6-0),xi_starboard); %% %%%
   Interpolation_45_starboard(:,i*2) = yi_45_starboard';
   BS_45_starboard(:,i) = yi_45_starboard';
end;

% Now that the angles and interportaled values for BS, for each ping, are known, it is time to average the data and convert it into
%dB again

BS_33_port(isnan(BS_33_port))=0;
```

```
BS_41_port(isnan(BS_41_port))=0;
BS_42_port(isnan(BS_42_port))=0;
BS_44_port(isnan(BS_44_port))=0;
BS_45_port(isnan(BS_45_port))=0;

BS_33_port(isinf(BS_33_port))=0;
BS_41_port(isinf(BS_41_port))=0;
BS_42_port(isinf(BS_42_port))=0;
BS_44_port(isinf(BS_44_port))=0;
BS_45_port(isinf(BS_45_port))=0;

Final_Interp_33_port(:,1) = Interpolation_33_port(:,1);
Final_Interp_41_port(:,1) = Interpolation_41_port(:,1);
Final_Interp_42_port(:,1) = Interpolation_42_port(:,1);
Final_Interp_44_port(:,1) = Interpolation_44_port(:,1);
Final_Interp_45_port(:,1) = Interpolation_45_port(:,1);

BS_33_starboard(isnan(BS_33_starboard))=0;
BS_41_starboard(isnan(BS_41_starboard))=0;
BS_42_starboard(isnan(BS_42_starboard))=0;
BS_44_starboard(isnan(BS_44_starboard))=0;
BS_45_starboard(isnan(BS_45_starboard))=0;

BS_33_starboard(isinf(BS_33_starboard))=0;
BS_41_starboard(isinf(BS_41_starboard))=0;
BS_42_starboard(isinf(BS_42_starboard))=0;
BS_44_starboard(isinf(BS_44_starboard))=0;
BS_45_starboard(isinf(BS_45_starboard))=0;

Final_Interp_33_starboard(:,1) = Interpolation_33_starboard(:,1);
Final_Interp_41_starboard(:,1) = Interpolation_41_starboard(:,1);
Final_Interp_42_starboard(:,1) = Interpolation_42_starboard(:,1);
Final_Interp_44_starboard(:,1) = Interpolation_44_starboard(:,1);
Final_Interp_45_starboard(:,1) = Interpolation_45_starboard(:,1);

for i=1:length_xi_port;
Final_Interp_33_port(i,2) = 10*log10(mean(BS_33_port(i,:)));
Final_Interp_41_port(i,2) = 10*log10(mean(BS_41_port(i,:)));
Final_Interp_42_port(i,2) = 10*log10(mean(BS_42_port(i,:)));
Final_Interp_44_port(i,2) = 10*log10(mean(BS_44_port(i,:)));
Final_Interp_45_port(i,2) = 10*log10(mean(BS_45_port(i,:)));
end;
for i=1:length_xi_starboard;
Final_Interp_33_starboard(i,2) = 10*log10(mean(BS_33_starboard(i,:)));
Final_Interp_41_starboard(i,2) = 10*log10(mean(BS_41_starboard(i,:)));
Final_Interp_42_starboard(i,2) = 10*log10(mean(BS_42_starboard(i,:)));
Final_Interp_44_starboard(i,2) = 10*log10(mean(BS_44_starboard(i,:)));
Final_Interp_45_starboard(i,2) = 10*log10(mean(BS_45_starboard(i,:)));
end;

%% 9 - Filtering

windowSize = 5;
b = (1/windowSize)*ones(1,windowSize);
a=1;
y5_33_port(:,2) = filter(b,a,Final_Interp_33_port(:,2));
y5_33_starboard(:,2) = filter(b,a,Final_Interp_33_starboard(:,2));
y5_41_port(:,2) = filter(b,a,Final_Interp_41_port(:,2));
y5_41_starboard(:,2) = filter(b,a,Final_Interp_41_starboard(:,2));
y5_42_port(:,2) = filter(b,a,Final_Interp_42_port(:,2));
y5_42_starboard(:,2) = filter(b,a,Final_Interp_42_starboard(:,2));
y5_44_port(:,2) = filter(b,a,Final_Interp_44_port(:,2));
```

```
y5_44_starboard(:,2) = filter(b,a,Final_Interp_44_starboard(:,2));
y5_45_port(:,2) = filter(b,a,Final_Interp_45_port(:,2));
y5_45_starboard(:,2) = filter(b,a,Final_Interp_45_starboard(:,2));

%Cleaning blunders that can appear after filtering in the first angles, due to moving window size issue:
% y5_33_port(:,3) = 10.^(y5_33_port(:,2)/10);
% y5_33_starboard(:,3) = 10.^(y5_33_starboard(:,2)/10);
% y5_41_port(:,3) = 10.^(y5_41_port(:,2)/10);
% y5_41_starboard(:,3) = 10.^(y5_41_starboard(:,2)/10);
% y5_42_port(:,3) = 10.^(y5_42_port(:,2)/10);
% y5_42_starboard(:,3) = 10.^(y5_42_starboard(:,2)/10);
% avg_y5_33_port = mean(y5_33_port(1:25,3));
% avg_y5_41_port = mean(y5_41_port(1:25,3));
% avg_y5_42_port = mean(y5_42_port(1:25,3));
% avg_y5_33_starboard = mean(y5_33_starboard(1:25,3));
% avg_y5_41_starboard = mean(y5_41_starboard(1:25,3));
% avg_y5_42_starboard = mean(y5_42_starboard(1:25,3));

y5_33_port(:,1) = Final_Interp_33_port(:,1);
y5_41_port(:,1) = Final_Interp_41_port(:,1);
y5_42_port(:,1) = Final_Interp_42_port(:,1);
y5_44_port(:,1) = Final_Interp_44_port(:,1);
y5_45_port(:,1) = Final_Interp_45_port(:,1);
y5_33_starboard(:,1) = Final_Interp_33_starboard(:,1);
y5_41_starboard(:,1) = Final_Interp_41_starboard(:,1);
y5_42_starboard(:,1) = Final_Interp_42_starboard(:,1);
y5_44_starboard(:,1) = Final_Interp_44_starboard(:,1);
y5_45_starboard(:,1) = Final_Interp_45_starboard(:,1);


% Selecting only grazing angles from 25º to 75º, which represents incident angles from 15º to 65º:
y5_33_port_raw = y5_33_port;
length_y5_33_port = length(y5_33_port);
for i=1:length_y5_33_port;
   if y5_33_port(length_y5_33_port+1-i,1)*-1<25;
      y5_33_port(length_y5_33_port+1-i,:)=[];
   elseif y5_33_port(length_y5_33_port+1-i,1)*-1>75;
      y5_33_port(length_y5_33_port+1-i,:)=[];
   end;
end;
clearvars length_y5_33_port

y5_41_port_raw = y5_41_port;
length_y5_41_port = length(y5_41_port);
for i=1:length_y5_41_port;
   if y5_41_port(length_y5_41_port+1-i,1)*-1<25;
      y5_41_port(length_y5_41_port+1-i,:)=[];
   elseif y5_41_port(length_y5_41_port+1-i,1)*-1>75;
      y5_41_port(length_y5_41_port+1-i,:)=[];
   end;
end;
clearvars length_y5_41_port

y5_42_port_raw = y5_42_port;
length_y5_42_port = length(y5_42_port);
for i=1:length_y5_42_port;
   if y5_42_port(length_y5_42_port+1-i,1)*-1<25;
      y5_42_port(length_y5_42_port+1-i,:)=[];
   elseif y5_42_port(length_y5_42_port+1-i,1)*-1>75;
      y5_42_port(length_y5_42_port+1-i,:)=[];
   end;
end;
```

```
clearvars length_y5_42_port

y5_44_port_raw = y5_44_port;
length_y5_44_port = length(y5_44_port);
for i=1:length_y5_44_port;
   if y5_44_port(length_y5_44_port+1-i,1)*-1<25;
      y5_44_port(length_y5_44_port+1-i,:)=[];
   elseif y5_44_port(length_y5_44_port+1-i,1)*-1>75;
      y5_44_port(length_y5_44_port+1-i,:)=[];
   end;
end;
clearvars length_y5_44_port

y5_45_port_raw = y5_45_port;
length_y5_45_port = length(y5_45_port);
for i=1:length_y5_45_port;
   if y5_45_port(length_y5_45_port+1-i,1)*-1<25;
      y5_45_port(length_y5_45_port+1-i,:)=[];
   elseif y5_45_port(length_y5_45_port+1-i,1)*-1>75;
      y5_45_port(length_y5_45_port+1-i,:)=[];
   end;
end;
clearvars length_y5_45_port

y5_33_starboard_raw = y5_33_starboard;
length_y5_33_starboard = length(y5_33_starboard);
for i=1:length_y5_33_starboard;
   if y5_33_starboard(length_y5_33_starboard+1-i,1)<25;
      y5_33_starboard(length_y5_33_starboard+1-i,:)=[];
   elseif y5_33_starboard(length_y5_33_starboard+1-i,1)>75;
      y5_33_starboard(length_y5_33_starboard+1-i,:)=[];
   end;
end;
clearvars length_y5_33_starboard

y5_41_starboard_raw = y5_41_starboard;
length_y5_41_starboard = length(y5_41_starboard);
for i=1:length_y5_41_starboard;
   if y5_41_starboard(length_y5_41_starboard+1-i,1)<25;
      y5_41_starboard(length_y5_41_starboard+1-i,:)=[];
   elseif y5_41_starboard(length_y5_41_starboard+1-i,1)>75;
      y5_41_starboard(length_y5_41_starboard+1-i,:)=[];
   end;
end;
clearvars length_y5_41_starboard

y5_42_starboard_raw = y5_42_starboard;
length_y5_42_starboard = length(y5_42_starboard);
for i=1:length_y5_42_starboard;
   if y5_42_starboard(length_y5_42_starboard+1-i,1)<25;
      y5_42_starboard(length_y5_42_starboard+1-i,:)=[];
   elseif y5_42_starboard(length_y5_42_starboard+1-i,1)>75;
      y5_42_starboard(length_y5_42_starboard+1-i,:)=[];
   end;
end;
clearvars length_y5_42_starboard

y5_44_starboard_raw = y5_44_starboard;
length_y5_44_starboard = length(y5_44_starboard);
for i=1:length_y5_44_starboard;
   if y5_44_starboard(length_y5_44_starboard+1-i,1)<25;
      y5_44_starboard(length_y5_44_starboard+1-i,:)=[];
```

```
   elseif y5_44_starboard(length_y5_44_starboard+1-i,1)>75;
      y5_44_starboard(length_y5_44_starboard+1-i,:)=[];
   end;
end;
clearvars length_y5_44_starboard

y5_45_starboard_raw = y5_45_starboard;
length_y5_45_starboard = length(y5_45_starboard);
for i=1:length_y5_45_starboard;
   if y5_45_starboard(length_y5_45_starboard+1-i,1)<25;
      y5_45_starboard(length_y5_45_starboard+1-i,:)=[];
   elseif y5_45_starboard(length_y5_45_starboard+1-i,1)>75;
      y5_45_starboard(length_y5_45_starboard+1-i,:)=[];
   end;
end;
clearvars length_y5_45_starboard

%% 10 - Plotting results
disp(' ')
disp('Do you want to plot grazing angles or incident angles? Type "1" for grazing angles,')
grazingorincident=input('"2" for incident angles or "3" for both grazing and incident angles => ');
disp(' ')
if grazingorincident ==1
% Grazing Angle, from 25º to 75º
figure
plot(y5_33_port(:,1)*-1,y5_33_port(:,2)*2, 'r-.', 'LineWidth', 3.0);
hold on
plot(y5_42_port(:,1)*-1,y5_42_port(:,2), 'r', 'LineWidth', 3.0);
hold on
plot(y5_33_starboard(:,1),y5_33_starboard(:,2)*2, 'g-.', 'LineWidth', 3.0);
hold on
plot(y5_42_starboard(:,1),y5_42_starboard(:,2), 'g', 'LineWidth', 3.0);
hold on
title({'\fontsize{20} Multifrequency Backscatter '; ...
['\fontsize{20} Line 254 - Sample # ' num2str(sample_number) ' - Frequency: ' num2str(100*frequency) ' kHz - Pulse Length: '
num2str(pulselength*10^6) ' \mus']})
xlabel('Grazing Angle (º)', 'Fontsize', 30);
ylabel('Backscatter Strength (dB re 1m2)', 'Fontsize', 30);
legend('BS datagram Port Filtered', 'New BS Port Filtered','BS datagram Starboard Filtered', 'New BS Starboard
Filtered','Location', 'southeast');
set(gca,'fontsize',30)
grid;
axis([0 90 -60 0]);

% Comparing TL and areas (only starboard side)
figure
plot(y5_42_starboard(:,1),y5_42_starboard(:,2), 'b', 'LineWidth', 3.0);
hold on
plot(y5_41_starboard(:,1),y5_41_starboard(:,2), 'r', 'LineWidth', 3.0);
hold on
plot(y5_44_starboard(:,1),y5_44_starboard(:,2), 'g', 'LineWidth', 3.0);
hold on
plot(y5_45_starboard(:,1),y5_45_starboard(:,2), 'k', 'LineWidth', 3.0);
hold on
title({'\fontsize{20} Comparing different TL and Areas combination '; ...
['\fontsize{20} Line 254 - Sample # ' num2str(sample_number) ' - Frequency: ' num2str(100*frequency) ' kHz - Pulse Length: '
num2str(pulselength*10^6) ' \mus']})
xlabel('Grazing Angle (º)', 'Fontsize', 30);
ylabel('Backscatter Strength (dB re 1m2)', 'Fontsize', 30);
legend('Ray tracing Range & Sloped Area', 'KM Range & KM Area','KM Range & Sloped Area', 'Ray tracing Range and KM
area','Location', 'southeast');
set(gca,'fontsize',30)
```

```
grid;
axis([0 90 -60 0]);
end;

% Computing incident angles
y5_33_port_i = y5_33_port;
y5_41_port_i = y5_41_port;
y5_42_port_i = y5_42_port;
y5_44_port_i = y5_44_port;
y5_45_port_i = y5_45_port;

y5_33_starboard_i = y5_33_starboard;
y5_41_starboard_i = y5_41_starboard;
y5_42_starboard_i = y5_42_starboard;
y5_44_starboard_i = y5_44_starboard;
y5_45_starboard_i = y5_45_starboard;

for i=1:length(y5_33_port_i)
   if y5_33_port(i,1) >= 0
      y5_33_port_i(i,1) = 90-y5_33_port(i,1);
   elseif y5_33_port(i,1) < 0
      y5_33_port_i(i,1) = -90-y5_33_port(i,1);
   end;
end;

for i=1:length(y5_41_port_i)
   if y5_41_port(i,1) >= 0
      y5_41_port_i(i,1) = 90-y5_41_port(i,1);
   elseif y5_41_port(i,1) < 0
      y5_41_port_i(i,1) = -90-y5_41_port(i,1);
   end;
end;

for i=1:length(y5_42_port_i)
   if y5_42_port(i,1) >= 0
      y5_42_port_i(i,1) = 90-y5_42_port(i,1);
   elseif y5_42_port(i,1) < 0
      y5_42_port_i(i,1) = -90-y5_42_port(i,1);
   end;
end;

for i=1:length(y5_44_port_i)
   if y5_44_port(i,1) >= 0
      y5_44_port_i(i,1) = 90-y5_44_port(i,1);
   elseif y5_44_port(i,1) < 0
      y5_44_port_i(i,1) = -90-y5_44_port(i,1);
   end;
end;

for i=1:length(y5_45_port_i)
   if y5_45_port(i,1) >= 0
      y5_45_port_i(i,1) = 90-y5_45_port(i,1);
   elseif y5_45_port(i,1) < 0
      y5_45_port_i(i,1) = -90-y5_45_port(i,1);
   end;
end;

   y5_33_port_i = sortrows(y5_33_port_i);
   y5_41_port_i = sortrows(y5_41_port_i);
   y5_42_port_i = sortrows(y5_42_port_i);
   y5_44_port_i = sortrows(y5_44_port_i);
   y5_45_port_i = sortrows(y5_45_port_i);
```

```
for i=1:length(y5_33_starboard_i)
   if y5_33_starboard(i,1) >= 0
      y5_33_starboard_i(i,1) = 90-y5_33_starboard(i,1);
   elseif y5_33_starboard(i,1) < 0
      y5_33_starboard_i(i,1) = -90-y5_33_starboard(i,1);
   end;
end;

for i=1:length(y5_41_starboard_i)
   if y5_41_starboard(i,1) >= 0
      y5_41_starboard_i(i,1) = 90-y5_41_starboard(i,1);
   elseif y5_41_starboard(i,1) < 0
      y5_41_starboard_i(i,1) = -90-y5_41_starboard(i,1);
   end;
end;

for i=1:length(y5_42_starboard_i)
   if y5_42_starboard(i,1) >= 0
      y5_42_starboard_i(i,1) = 90-y5_42_starboard(i,1);
   elseif y5_42_starboard(i,1) < 0
      y5_42_starboard_i(i,1) = -90-y5_42_starboard(i,1);
   end;
end;

for i=1:length(y5_44_starboard_i)
   if y5_44_starboard(i,1) >= 0
      y5_44_starboard_i(i,1) = 90-y5_44_starboard(i,1);
   elseif y5_44_starboard(i,1) < 0
      y5_44_starboard_i(i,1) = -90-y5_44_starboard(i,1);
   end;
end;

for i=1:length(y5_45_starboard_i)
   if y5_45_starboard(i,1) >= 0
      y5_45_starboard_i(i,1) = 90-y5_45_starboard(i,1);
   elseif y5_45_starboard(i,1) < 0
      y5_45_starboard_i(i,1) = -90-y5_45_starboard(i,1);
   end;
end;

   y5_33_starboard_i = sortrows(y5_33_starboard_i);
   y5_41_starboard_i = sortrows(y5_41_starboard_i);
   y5_42_starboard_i = sortrows(y5_42_starboard_i);
   y5_44_starboard_i = sortrows(y5_44_starboard_i);
   y5_45_starboard_i = sortrows(y5_45_starboard_i);

if grazingorincident ==2
% Incident Angles, from 15º to 65º
figure
plot(y5_33_port_i(:,1)*-1,y5_33_port_i(:,2)*2, 'r-.', 'LineWidth', 3.0);
hold on
plot(y5_42_port_i(:,1)*-1,y5_42_port_i(:,2), 'r', 'LineWidth', 3.0);
hold on
plot(y5_33_starboard_i(:,1),y5_33_starboard_i(:,2)*2, 'g-.', 'LineWidth', 3.0);
hold on
plot(y5_42_starboard_i(:,1),y5_42_starboard_i(:,2), 'g', 'LineWidth', 3.0);
hold on
title({'\fontsize{20} Multifrequency Backscatter '; ...
['\fontsize{20} Line 254 - Sample # ' num2str(sample_number) ' - Frequency: ' num2str(100*frequency) ' kHz - Pulse Length: '
num2str(pulselength*10^6) ' \mus']})
xlabel('Incident Angle (º)', 'Fontsize', 30);
```

```
ylabel('Backscatter Strength (dB re 1m2)', 'Fontsize', 30);
legend('BS datagram Port Filtered', 'New BS Port Filtered','BS datagram Starboard Filtered', 'New BS Starboard Filtered','Location', 'southwest');
set(gca,'fontsize',30)
grid;
axis([0 90 -60 0]);

% Comparing TL and areas (only starboard side)
figure
plot(y5_42_starboard_i(:,1),y5_42_starboard_i(:,2), 'b', 'LineWidth', 3.0);
hold on
plot(y5_41_starboard_i(:,1),y5_41_starboard_i(:,2), 'r', 'LineWidth', 3.0);
hold on
plot(y5_44_starboard_i(:,1),y5_44_starboard_i(:,2), 'g', 'LineWidth', 3.0);
hold on
plot(y5_45_starboard_i(:,1),y5_45_starboard_i(:,2), 'k', 'LineWidth', 3.0);
hold on
title({'\fontsize{20} Comparing different 2TL and Areas combination '; ...
['\fontsize{20} Line 254 - Sample # ' num2str(sample_number) ' - Frequency: ' num2str(100*frequency) ' kHz - Pulse Length: ' num2str(pulselength*10^6) ' \mus']})
xlabel('Incident Angle (º)', 'Fontsize', 30);
ylabel('Backscatter Strength (dB re 1m2)', 'Fontsize', 30);
legend('Ray tracing Range & Sloped Area', 'KM Range & KM Area','KM Range & Sloped Area', 'Ray tracing Range and KM area','Location', 'southwest');
set(gca,'fontsize',30)
grid;
axis([0 90 -60 0]);
end;

if grazingorincident ==3

% Grazing Angle, from 25º to 75º
figure
plot(y5_33_port(:,1)*-1,y5_33_port(:,2)*2, 'r-.', 'LineWidth', 3.0);
hold on
plot(y5_42_port(:,1)*-1,y5_42_port(:,2), 'r', 'LineWidth', 3.0);
hold on
plot(y5_33_starboard(:,1),y5_33_starboard(:,2)*2, 'g-.', 'LineWidth', 3.0);
hold on
plot(y5_42_starboard(:,1),y5_42_starboard(:,2), 'g', 'LineWidth', 3.0);
hold on
title({'\fontsize{20} Multifrequency Backscatter '; ...
['\fontsize{20} Line 254 - Sample # ' num2str(sample_number) ' - Frequency: ' num2str(100*frequency) ' kHz - Pulse Length: ' num2str(pulselength*10^6) ' \mus']})
xlabel('Grazing Angle (º)', 'Fontsize', 30);
ylabel('Backscatter Strength (dB re 1m2)', 'Fontsize', 30);
legend('BS datagram Port Filtered', 'New BS Port Filtered','BS datagram Starboard Filtered', 'New BS Starboard Filtered','Location', 'southeast');
set(gca,'fontsize',30)
grid;
axis([0 90 -60 0]);

% Comparing TL and areas (only starboard side)
figure
plot(y5_42_starboard(:,1),y5_42_starboard(:,2), 'b', 'LineWidth', 3.0);
hold on
plot(y5_41_starboard(:,1),y5_41_starboard(:,2), 'r', 'LineWidth', 3.0);
hold on
plot(y5_44_starboard(:,1),y5_44_starboard(:,2), 'g', 'LineWidth', 3.0);
hold on
plot(y5_45_starboard(:,1),y5_45_starboard(:,2), 'k', 'LineWidth', 3.0);
hold on
```

```
title({'\fontsize{20} Comparing different TL and Areas combination '; ...
['\fontsize{20} Line 254 - Sample # ' num2str(sample_number) ' - Frequency: ' num2str(100*frequency) ' kHz - Pulse Length: ' num2str(pulselength*10^6) ' \mus']})
xlabel('Grazing Angle (º)', 'Fontsize', 30);
ylabel('Backscatter Strength (dB re 1m2)', 'Fontsize', 30);
legend('Ray tracing Range & Sloped Area', 'KM Range & KM Area','KM Range & Sloped Area', 'Ray tracing Range and KM area','Location', 'southeast');
set(gca,'fontsize',30)
grid;
axis([0 90 -60 0]);

% Comparing TL and areas (only starboard side)
figure
plot(y5_42_starboard_i(:,1),y5_42_starboard_i(:,2), 'b', 'LineWidth', 3.0);
hold on
plot(y5_41_starboard_i(:,1),y5_41_starboard_i(:,2), 'r', 'LineWidth', 3.0);
hold on
plot(y5_44_starboard_i(:,1),y5_44_starboard_i(:,2), 'g', 'LineWidth', 3.0);
hold on
plot(y5_45_starboard_i(:,1),y5_45_starboard_i(:,2), 'k', 'LineWidth', 3.0);
hold on
title({'\fontsize{20} Comparing different 2TL and Areas combination '; ...
['\fontsize{20} Line 254 - Sample # ' num2str(sample_number) ' - Frequency: ' num2str(100*frequency) ' kHz - Pulse Length: ' num2str(pulselength*10^6) ' \mus']})
xlabel('Incident Angle (º)', 'Fontsize', 30);
ylabel('Backscatter Strength (dB re 1m2)', 'Fontsize', 30);
legend('Ray tracing Range & Sloped Area', 'KM Range & KM Area','KM Range & Sloped Area', 'Ray tracing Range and KM area','Location', 'southwest');
set(gca,'fontsize',30)
grid;
axis([0 90 -60 0]);
end;

clearvars a b Final_Interp_33_port Final_Interp_41_port Final_Interp_42_port Final_Interp_33_starboard Final_Interp_41_starboard Final_Interp_42_starboard
clearvars ii_44_port yi_44_starboard yi_45_port yi_45_starboard
clearvars length_xi_port length_xi_starboard  size_Absorption_201406231438
clearvars Interpolation_33 Interpolation_41 Interpolation_42 Pre_Interp angles column i num_columns_Pre_Interp xi yi_33 yi_41 yi_42 max_angle min_angle BS_33 BS_41 BS_42
clearvars k columns row averages avg_distance num_beams size_Absorption_201446171703 size_Line_254 transducer_depth angles_port angles_starboard
clearvars aux avg_acrosstrackslope BS_33_port BS_41_port BS_42_port i Interpolation_33_port Interpolation_41_port Interpolation_42_port k BS_33_starboard BS_41_starboard BS_42_starboard Interpolation_33_starboard Interpolation_41_starboard Interpolation_42_starboard
clearvars xi_port yi_33_port yi_41_port yi_42_port xi_starboard yi_33_starboard yi_41_starboard yi_42_starboard average_sampling_rate
clearvars max_angle_port min_angle_port Pre_Interp_port std_acrosstrackslope max_angle_starboard min_angle_starboard Pre_Interp_starboard
clearvars beamwidth_across beamwidth_along Absorption_201406231438 Freq_MeanAbs_Sector Interpolation_45_port Interpolation_45_starboard pings whenend whenstart windowSize
clearvars BS_44_port BS_45_port BS_44_starboard BS_45_starboard Final_Interp_44_port Final_Interp_45_port Interpolation_44_port nterpolation_45_port Final_Interp_44_starboard Final_Interp_51_starboard Interpolation_44_starboard nterpolation_45_starboard
%% 11 - Saving files

% Column 1: Grazing Angle
% Column 2: BS
% Column 3: Incident Angle
% Column 4: BS
% Exporting Interpolated data port side:
HM_port(:,1) = y5_42_port(:,1);
HM_port(:,2) = y5_42_port(:,2);
```

```
HM_port(:,3) = y5_42_port_i(:,1);
HM_port(:,4) = y5_42_port_i(:,2);
save(['Port_Sample_' num2str(sample_number) '_Frequency_' num2str(frequency*100) '_PulseLength_' num2str(pulselength*10^6) '.txt'], 'HM_port', '-ascii');
disp(['File Port_Sample_' num2str(sample_number) '_Frequency_' num2str(frequency*100) '_PulseLength_' num2str(pulselength*10^6) '.txt was successfully saved']);

% Exporting Interpolated data starboard side:
HM_starboard(:,1) = y5_42_starboard(:,1);
HM_starboard(:,2) = y5_42_starboard(:,2);
HM_starboard(:,3) = y5_42_starboard_i(:,1);
HM_starboard(:,4) = y5_42_starboard_i(:,2);
disp(' ')
save(['Starboard_Sample_' num2str(sample_number) '_Frequency_' num2str(frequency*100) '_PulseLength_' num2str(pulselength*10^6) '.txt'], 'HM_starboard', '-ascii');
disp(['File Starboard_Sample_' num2str(sample_number) '_Frequency_' num2str(frequency*100) '_PulseLength_' num2str(pulselength*10^6) '.txt was successfully saved']);

% Exporting information for Statistics (only starboard side and incident angles):
HM_stats(:,1) = y5_42_starboard_i(:,1); % Incident angles
HM_stats(:,2) = y5_42_starboard_i(:,2); % Final BS with Ray tracing TL and sloped area
HM_stats(:,3) = y5_41_starboard_i(:,2); % Final BS with KM TL and flat area
HM_stats(:,4) = y5_44_starboard_i(:,2); % Final BS with KM TL and sloped area
HM_stats(:,5) = y5_45_starboard_i(:,2); % Final BS with Ray tracing TL and flat area
disp(' ')
save(['Statistics_Sample_' num2str(sample_number) '_Frequency_' num2str(frequency*100) '_PulseLength_' num2str(pulselength*10^6) '.txt'], 'HM_stats', '-ascii');
disp(['File Statistics_Sample_' num2str(sample_number) '_Frequency_' num2str(frequency*100) '_PulseLength_' num2str(pulselength*10^6) '.txt was successfully saved']);

% Saving the main matrix: Line_254:
disp(' ')
save(['Processed_File_Sample_' num2str(sample_number) '_Frequency_' num2str(frequency*100) '_PulseLength_' num2str(pulselength*10^6) '.txt']', 'Line_254', '-ascii');
disp(['Processed_File_Sample_' num2str(sample_number) '_Frequency_' num2str(frequency*100) '_PulseLength_' num2str(pulselength*10^6) '.txt was successfully saved']);
disp(' ')
toc
```

APPENDIX D - EXTRACTING INFORMATION FROM DATAGRAM FILES

The following is the entirety of the codes that were created for extracting needed information from Kongsberg datagram files.

I) Extracting information from PU Status output datagram:

```
% CCOM / JHC
% Anderson Pecanha and Commander Adriano Vieira de Souza (Brazilian Navy Officer)
% Created: Mar/20/2016
% Last Update: Apr/03/2016

function output = PUStatusOutput(filename)

clc
identifier = endianness('TYPE INPUT FILENAME HERE.all') ;
fp = fopen('TYPE INPUT FILENAME HERE .all','r') ;

      fid = fopen('TYPE OUTPUT FILENAME HERE.txt','w');

if identifier ~= -1

   while feof(fp) ~= 1

      if fread(fp,1) == 2 % 02h = 2 at base-ten is equivalent to STX - beginer's identifier

         if fread(fp,1) == 49   % 31h = 49 at base-ten

            %EM model number
            EM_model = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

            if EM_model == 2040

            %disp('-----------------------------Beginning of the Header-----------------------------')

            % ------------------ Date = year*10000 + month*100 + day
            Date = formated_date(swapped_unsigned_value(identifier,32,fread(fp,4)')) ;

            % ------------------ Time since midnight in milliseconds
            Time = hdecimal2hms(swapped_unsigned_value(identifier,32,fread(fp,4)')/1000/3600) ;

            fread(fp,59) ;

            % Depth to normal incidence in m 1U
            Depth = swapped_unsigned_value(identifier,8,fread(fp,1)') ; ;

            % Range to normal incidence in m 2U
            Range = swapped_unsigned_value(identifier,16,fread(fp,2)') ; ;

            fread(fp,14) ;

             VETOR = strcat(sprintf('%s %s',Date,Time ),sprintf(' %.2f %.2f',Depth,Range) ) ;
             disp(VETOR)
             fprintf(fid,'%s\n',VETOR);
         end ;
         end ;
```

```
        else
            %fread(fp,1) ; % skip one line
        end ;
        %fread(fp,1) ; % skip one line
    end ;

    fclose(fp) ;
end ;
output = 0 ;
fclose('all') ;
```

II) Extracting information from Raw range and angle 78 datagram:

```
% CCOM / JHC
% Anderson Pecanha and Commander Adriano Vieira de Souza (Brazilian Navy Officer)
% Created: Mar/20/2016
% Last Update: Apr/03/2016

function output = RawRangeAngle_78(filename)

clc
identifier = endianness('TYPE INPUT FILE NAME HERE.all') ;
fp = fopen('TYPE INPUT FILE NAME HERE.all','r') ;

      fid = fopen('TYPE OUTPUT FILE NAME HERE.txt','w');

if identifier ~= -1

   while feof(fp) ~= 1

      if fread(fp,1) == 2

         if fread(fp,1) == 78

            %EM model number
            EM_model = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

            if EM_model == 2040

            %disp('------------------------------Beginning of the Header------------------------------')

            % ------------------ Date = year*10000 + month*100 + day
            Date = formated_date(swapped_unsigned_value(identifier,32,fread(fp,4)')) ;

            % ------------------ Time since midnight in milliseconds
            Time = hdecimal2hms(swapped_unsigned_value(identifier,32,fread(fp,4)')/1000/3600) ;

            % ------------------ Ping counter (sequential counter)
            Ping_counter = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

            %System serial number
            serial_number = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

            % ------------------ Sound speed at transducer in m/s
            Sound_Speed_atTransducer = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

            %Number of transmit sectors = Ntx
            Ntx = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

            %Number of receiver beams in datagram = Nrx
            Nrx = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

            %Number of valid detections
            Number_Valid_Detections = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

            %Sampling frequency in Hz
            Sampling_Frequency = swapped_float_value(identifier,32,fread(fp,4)') ;

            %Dscale
            Dscale = swapped_unsigned_value(identifier,32,fread(fp,4)') ;

               aux1 = 1;
```

```
for i = 1:Ntx

%disp('------------------------------begin-----------------------------')

%Tilt angle re TX array in 0.01° - 2S
 Tilt_Angle_re_TX_array = swapped_signed_value(identifier,16,fread(fp,2)') ;

%Focus range in 0.1 m (0 = No focusing applied) - 2U
Focus_Range = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

%Signal length in s - 4F
Signal_Length = swapped_float_value(identifier,32,fread(fp,4)') ;

%Sector transmit delay re first TX pulse, in s - 4F
Sector_Transmit_Delay = swapped_float_value(identifier,32,fread(fp,4)') ;

        % ------------------ Centre frequency in Hz - 4F
Centre_Frequency_Mean_Absorption_Coeff_Transmit_Sector_Number(1,aux1) =
swapped_float_value(identifier,32,fread(fp,4)') ;
        aux1 = aux1 + 1;
        % ------------------ Mean absorption coeff. in 0.01 dB/km - 2U
Centre_Frequency_Mean_Absorption_Coeff_Transmit_Sector_Number(1,aux1) =
swapped_unsigned_value(identifier,16,fread(fp,2)') ;
        aux1 = aux1 + 1;

%Signal waveform identifier - 1U
Signal_Waveform_Identifier = swapped_unsigned_value(identifier,8,fread(fp,1)') ;

        % ------------------ Transmit sector number - 1U
Centre_Frequency_Mean_Absorption_Coeff_Transmit_Sector_Number(1,aux1) =
swapped_unsigned_value(identifier,8,fread(fp,1)') ;
        aux1 = aux1 + 1;

%Signal bandwidth in Hz - 4F
Signal_Bandwidth = swapped_float_value(identifier,32,fread(fp,4)') ;

%disp('------------------------------end-----------------------------')

end ;
aux1 = 1 ;
%pause
aux2 = 1 ;
for j = 1:Nrx

%disp('------------------------------begin-----------------------------')

        % ------------------ Beam pointing anlge re RX array in 0.01° - 2S
Beam_Pointing_Anlge_Transmit_Sector_Number2(1,aux2) = swapped_signed_value(identifier,16,fread(fp,2)') ;
        aux2 = aux2 + 1;
        % ------------------ Transmit sector number - 1U
 Beam_Pointing_Anlge_Transmit_Sector_Number2(1,aux2) = swapped_unsigned_value(identifier,8,fread(fp,1)') ;
        aux2 = aux2 + 1;
        %disp(sprintf(' %.2f ',Beam_Pointing_Anlge_Transmit_Sector_Number2)) ;

%Detection info - 1U
Detection_Info = swapped_unsigned_value(identifier,8,fread(fp,1)') ;

%Detection window length in samples - 2U
Detection_Window = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

%Quality factor - 1U
Quality_Factor = swapped_unsigned_value(identifier,8,fread(fp,1)') ;
```

```
            %D corr - 1S
            D_corr = swapped_signed_value(identifier,8,fread(fp,1)') ;

            %Two way travel time in s - 4F
                        TWTT(1,aux2)= swapped_float_value(identifier,32,fread(fp,4)');

          %Reflectivity (BS) in 0.1 dB resolution - 2S
           Beam_Pointing_Anlge_Transmit_Sector_Number2(1,aux2) = swapped_signed_value(identifier,16,fread(fp,2)') ;
                        aux2=aux2+1;

            %Real time cleaning info - 1S
            Real_Time_Cleaning_Info = swapped_signed_value(identifier,8,fread(fp,1)') ;

            %Spare - 1U
            Spare = swapped_unsigned_value(identifier,8,fread(fp,1)') ;

            %disp('------------------------------end------------------------------')

            end ;
            aux2 = 1 ;

          %Spare
          Spare = swapped_unsigned_value(identifier,8,fread(fp,1)') ;

          %End identifier = ETX (Always 03h)
          fread(fp,1,'uint8')' ;

          %Check sum of data between STX and ETX 2U
          fread(fp,1,'uint16')' ;

          VETOR = strcat(sprintf('%s %s',Date,Time ),sprintf(' %07.f %.3f',Ping_counter,Sound_Speed_atTransducer),sprintf('
%.2f ',Centre_Frequency_Mean_Absorption_Coeff_Transmit_Sector_Number),sprintf(' %.0f %.0f %.0f
',Beam_Pointing_Anlge_Transmit_Sector_Number2), sprintf(' %.6f ',TWTT)) ;

          disp(VETOR)
          fprintf(fid,'%s\n',VETOR);
        end ;
        end ;

      else
      end ;
   end ;
end ;

output = 0 ;
fclose(fp) ;
fclose('all') ;
```

III) Extracting information from Seabed image data 89 datagram:

```
% CCOM / JHC
% Anderson Pecanha and Commander Adriano Vieira de Souza (Brazilian Navy Officer)
% Created: Mar/20/2016
% Last Update: Apr/03/2016

function output = SeabedImageData_89(filename)

clc
identifier = endianness('TYPE INPUT FILE NAME HERE.all') ;
fp = fopen('TYPE INPUT FILE NAME HERE.all','r') ;

      fid = fopen('TYPE OUTPUT FILE NAME HERE.txt','w');

if identifier ~= -1

   while feof(fp) ~= 1

      if fread(fp,1) == 2

         if fread(fp,1) == 89
clc
            EM_model = swapped_unsigned_value(identifier,16,fread(fp,2)') ;
            if EM_model == 2040 % 2040 EM model number

               % EM model

               %Date = year*10000 + month*100 + day (Example: Sep 26, 2005 = 20050926
               Date_SBI = formated_date(swapped_unsigned_value(identifier,32,fread(fp,4)')) ;

               %Time since midnight in milliseconds
               Time_SBI = hdecimal2hms(swapped_unsigned_value(identifier,32,fread(fp,4)')/1000/3600) ;

               %Ping counter (sequential counter)
               Ping_counter = swapped_unsigned_value(identifier,16,fread(fp,2)') ;
               disp(sprintf('%s %.0f','Ping counter (sequential counter) = ',Ping_counter));

               %System serial number
               serial_number = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

               %Sampling frequency in Hz
               frequency_SBI = swapped_float_value(identifier,32,fread(fp,4)') ;

               %Range to normal incidence used to correct sample
               %amplitudes in no. of samples
               Rangetonormalincidence_SBI = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

               %Normal incidence BS in 0.1 dB (BSN)
                Normal_incidence_SBI = swapped_signed_value(identifier,16,fread(fp,2)') ;

               %Oblique BS in 0.1 dB (BSO)
               ObliqueBS_SBI = swapped_signed_value(identifier,16,fread(fp,2)') ;

 fprintf(fid,'%s %s %07.2f %07.2f
%d\n',Date_SBI,Time_SBI,Normal_incidence_SBI,ObliqueBS_SBI,Rangetonormalincidence_SBI);

               %Tx beamwidth along in 0.1°
               Txbeamwidth_SBI = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

               %TVG law crossover angle in 0.1°
```

```
            TVGlaw_SBI = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

            %Number of valid beams (N)
            Numberofvalidbeams_SBI = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

               %cont = 0 ;
               %---------- pause

               for AUX = 1:Numberofvalidbeams_SBI
                  %Sorting direction
                  Sortingdirection_SBI = swapped_signed_value(identifier,8,fread(fp,1)') ;

                  %Detection info
                  Detectioninfo_SBI = swapped_unsigned_value(identifier,8,fread(fp,1)') ;

                  %Number of samples per beam = Ns
                  Ns_SBI = swapped_unsigned_value(identifier,16,fread(fp,2)') ;
                  %cont = cont + Ns_SBI ;

                  %Center sample number
                  Centersamplenumber_SBI = swapped_unsigned_value(identifier,16,fread(fp,2)') ;

               end ;
         end ;
      end ;
   end ;
end ;

fclose('all')
```

IV) Functions:

```
% CCOM / JHC
% Anderson Pecanha and Commander Adriano Vieira de Souza (Brazilian Navy Officer)
% Created: Mar/16/2016
% Last Update: Apr/03/2016
```

a) dump_Raw_Range_beamAngle.m

```
function output = dump_RawRange_BeamAngle(filename)
clc
identifier = endianness(filename) ;
fp = fopen(filename,'r') ;

if identifier ~= -1

   while feof(fp) ~= 1

      if fread(fp,1) == 2

         if fread(fp,1) == 70%%%(F)%102

            disp('------------------------------Beginning of the Header------------------------------')

            %EM model number
            EM_model = swapped_unsigned_value(identifier,16,fread(fp,2)') ;
            disp(sprintf('%s %.0f','EM model number = ',EM_model));

            %Date = year*10000 + month*100 + day
            Date_Depth = formated_date(swapped_unsigned_value(identifier,32,fread(fp,4)')) ;
            disp(sprintf('%s %s','Date = ',Date_Depth));

            %Time since midnight in milliseconds
            Time_Depth = hdecimal2hms(swapped_unsigned_value(identifier,32,fread(fp,4)')/1000/3600) ;
            disp(sprintf('%s %s','Time = ',Time_Depth));

            %Ping counter (sequential counter)
            Ping_counter = swapped_unsigned_value(identifier,16,fread(fp,2)') ;
            disp(sprintf('%s %.0f','Ping counter (sequential counter) = ',Ping_counter));

            %System serial number
            serial_number = swapped_unsigned_value(identifier,16,fread(fp,2)') ;
            disp(sprintf('%s %.0f','System serial number = ',serial_number));

            %Maximum number of beams possible
            Maximum_number_beams_possible = swapped_unsigned_value(identifier,8,fread(fp,1)') ;
            disp(sprintf('%s %.0f','Maximum number of beams possible = ',Maximum_number_beams_possible));

            %Number of valid receive beams = N
            N = swapped_unsigned_value(identifier,8,fread(fp,1)') ;
            disp(sprintf('%s %.0f','Number of valid receive beams = N = ',N));

            %Sound speed at transducer in dm/s
            Sound_speed_transducer = swapped_unsigned_value(identifier,16,fread(fp,2)')/10 ;
            disp(sprintf('%s %.2f','Sound speed at transducer in m/s = ',Sound_speed_transducer));

            disp('---------------------------------End of the Header---------------------------------')
             pause
```

```
            for i = 1:N

            disp('-----------------------------begin-----------------------------')
%             Beam pointing angle in 0.01°

                        if identifier == 0 % Big-Endian
                           Beam_pointing_angle = fread(fp,1,'int16','ieee-be')'/100 ;
                        else
                           Beam_pointing_angle = fread(fp,1,'int16','ieee-le')'/100 ;
                        end ;

            disp(sprintf('%s %.2f','Beam pointing angle in 0.01 = ',Beam_pointing_angle)) ;

%             Transmit tilt angle in 0.01°
                        if identifier == 0 % Big-Endian
                           Transmit_tilt_angle = fread(fp,1,'int16','ieee-be')'/100 ;
                        else
                           Transmit_tilt_angle = fread(fp,1,'int16','ieee-le')'/100 ;
                        end ;
                    disp(sprintf('%s %.2f','Transmit tilt angle in 0.01 = ',Transmit_tilt_angle) );

%             Range (two--way travel time)
            two_way_travel_time = swapped_unsigned_value(identifier,16,fread(fp,2)') ;
            disp(sprintf('%s %.2f','Range (two--way travel time) = ',two_way_travel_time));

%             Reflectivity (BS) in 0.5 dB resolution) (Example: --20 dB = 216

                        if identifier == 0 % Big-Endian
                           Reflectivity = fread(fp,1,'int8','ieee-be')' ;
                        else
                           Reflectivity = fread(fp,1,'int8','ieee-le')' ;
                        end ;

            %Reflectivity = swapped_signed_value(identifier,8,fread(fp,1)') ;
            disp(sprintf('%s %.1f','Reflectivity (BS) in 0.5 dB resolution) (Example: --20 dB = 216) = ',Reflectivity)) ;

%             Beam number
            Beam_number = swapped_unsigned_value(identifier,8,fread(fp,1)') ;
            disp(sprintf('%s %.0f','Beam number = ',Beam_number));
            disp('-----------------------------end-----------------------------')
            end ;

          %Spare
          Spare = swapped_unsigned_value(identifier,8,fread(fp,1)') ;
          disp(sprintf('%s %.0f','Spare = ',Spare));

          %End identifier = ETX (Always 03h)
          fread(fp,1,'uint8')' ;

          pause %optional
          clc   %optional
        end ;

      else
         fread(fp,1) ; % skip one line
      end ;
      fread(fp,1) ; % skip one line
    end ;

    fclose(fp) ;
end ;
```

```
output = 0 ;
fclose(fp) ;
fclose('all') ;
```

b) endianness.m

```
function identifier = endianness(filename)

    fp = fopen('TYPE INPUT FILE NAME HERE.all' ,'r') ;

    if fp == -1
       disp('****** Cannot open the file. Please, check out the input file name and try again! ******') ;
       identifier = -1 ;
    else
       % based on the beginning of file the pointer will offset its
       % position indicator 8 bytes toward
       % here the pointer will read 1-byte size corresponding the beginning of the date and time fields
       fseek(fp, 8, 'bof');
       A = fread(fp,1) ;
          if A < 10
             identifier = 0 ; %'Big endian' ;
          else
             identifier = 1 ; %'Little endian' ;
          end ;
       fclose(fp) ;
    end ;
```

c) formated_date.m

```
function output_date = formated_date(input_date) ;

YYYYMMDD = input_date ;
DD = YYYYMMDD - 100*fix(YYYYMMDD/100);

%Check if the variable DD is < 10. If the answer is true DD_str will receive a zero on the left side
if length(num2str(DD)) == 1
   DD_str = cat(2,'0',num2str(DD)) ;
else
   DD_str = num2str(DD) ;
end ;

YYYYMM = (YYYYMMDD - DD)/100 ;
MM = (YYYYMM - 100*fix(YYYYMM/100)) ;

%Check if the variable MM is < 10. If the answer is true MM_str will receive a zero on the left side
if length(num2str(MM)) == 1
   MM_str = cat(2,'0',num2str(MM)) ;
else
   MM_str = num2str(MM) ;
end ;

YYYY = (YYYYMM - MM)/100 ;

output_date = cat(2,num2str(YYYY),'/',MM_str,'/',DD_str) ;
```

d) hdecimal2hms.m

```
function HMS = hdecimal2hms(time) ;
hour = fix(time) ;

%Check if the variable hour is < 10. If the answer is true hour_str will receive a zero on the left side
if length(num2str(hour)) == 1
   hour_str = cat(2,'0',num2str(hour)) ;
else
   hour_str = num2str(hour) ;
end ;

minutes = fix(( time - fix(time) )*60) ;

%Check if the variable minutes is < 10. If the answer is true minutes_str will receive a zero on the left side
if length(num2str(minutes)) == 1
   minutes_str = cat(2,'0',num2str(minutes)) ;
else
   minutes_str = num2str(minutes) ;
end ;

seconds = ( ( time - fix(time) )*60 - minutes )*60 ;

%Check if the variable seconds is < 10. If the answer is true seconds_str will receive a zero on the left side
if length(sprintf('%0.6f',seconds)) == 8
   seconds_str = cat(2,'0',sprintf('%0.6f',seconds)) ;
else
   seconds_str = sprintf('%0.6f',seconds) ;
end ;

   HMS = cat(2,hour_str,':',minutes_str,':',seconds_str) ;
```

e) swapped_float_value.m

```
function output = swapped_float_value(identifier,nr_bits,value)

if identifier == 0 % Big-Endian
   if nr_bits == 8
      VV = strcat(dec2bin(value(1),8)) ;
   elseif nr_bits == 16
      VV = strcat(dec2bin(value(1),8),dec2bin(value(2),8)) ;
   elseif nr_bits == 32
      VV = strcat(dec2bin(value(1),8),dec2bin(value(2),8),dec2bin(value(3),8),dec2bin(value(4),8)) ;
   elseif nr_bits == 64
      VV                                                                                   =
strcat(dec2bin(value(1),8),dec2bin(value(2),8),dec2bin(value(3),8),dec2bin(value(4),8),dec2bin(value(5),8),dec2bin(value(6),8),d
ec2bin(value(7),8),dec2bin(value(8),8)) ;
   end ;

elseif identifier == 1 % Little-Endian
   if nr_bits == 8
      value = fliplr(value) ;
      VV = strcat(dec2bin(value(1),8)) ;
   elseif nr_bits == 16
      value = fliplr(value) ;
      VV = strcat(dec2bin(value(1),8),dec2bin(value(2),8)) ;
   elseif nr_bits == 32
      value = fliplr(value) ;
      VV = strcat(dec2bin(value(1),8),dec2bin(value(2),8),dec2bin(value(3),8),dec2bin(value(4),8)) ;
   elseif nr_bits == 64
      value = fliplr(value) ;
      VV                                                                                   =
strcat(dec2bin(value(1),8),dec2bin(value(2),8),dec2bin(value(3),8),dec2bin(value(4),8),dec2bin(value(5),8),dec2bin(value(6),8),d
ec2bin(value(7),8),dec2bin(value(8),8)) ;
   end ;

end ;

   if nr_bits == 8
      output = typecast(uint8(bin2dec(VV)),'single')     ;
   elseif nr_bits == 16
      output = typecast(uint16(bin2dec(VV)),'single');
   elseif nr_bits == 32
      output = typecast(uint32(bin2dec(VV)),'single');
   elseif nr_bits == 64
      output = typecast(uint64(bin2dec(VV)),'single') ;
   end ;
```

f) swapped_signed_value.m

```
function output = swapped_signed_value(identifier,nr_bits,value)

if identifier == 0 % Big-Endian
   if nr_bits == 8
      VV = strcat(dec2bin(value(1),8)) ;
   elseif nr_bits == 16
      VV = strcat(dec2bin(value(1),8),dec2bin(value(2),8)) ;
   elseif nr_bits == 32
      VV = strcat(dec2bin(value(1),8),dec2bin(value(2),8),dec2bin(value(3),8),dec2bin(value(4),8)) ;
   elseif nr_bits == 64
      VV                                                                                                                                                =
strcat(dec2bin(value(1),8),dec2bin(value(2),8),dec2bin(value(3),8),dec2bin(value(4),8),dec2bin(value(5),8),dec2bin(value(6),8),d
ec2bin(value(7),8),dec2bin(value(8),8)) ;
   end ;

elseif identifier == 1 % Little-Endian
   if nr_bits == 8
      value = fliplr(value) ;
      VV = strcat(dec2bin(value(1),8)) ;
   elseif nr_bits == 16
      value = fliplr(value) ;
      VV = strcat(dec2bin(value(1),8),dec2bin(value(2),8)) ;
   elseif nr_bits == 32
      value = fliplr(value) ;
      VV = strcat(dec2bin(value(1),8),dec2bin(value(2),8),dec2bin(value(3),8),dec2bin(value(4),8)) ;
   elseif nr_bits == 64
      value = fliplr(value) ;
      VV                                                                                                                                                =
strcat(dec2bin(value(1),8),dec2bin(value(2),8),dec2bin(value(3),8),dec2bin(value(4),8),dec2bin(value(5),8),dec2bin(value(6),8),d
ec2bin(value(7),8),dec2bin(value(8),8)) ;
   end ;

end ;

   if nr_bits == 8
      output = typecast(uint8(bin2dec(VV)),'int8')     ;
   elseif nr_bits == 16
      output = typecast(uint16(bin2dec(VV)),'int16');
   elseif nr_bits == 32
      output = typecast(uint32(bin2dec(VV)),'int32');
   elseif nr_bits == 64
      output = typecast(uint64(bin2dec(VV)),'int64') ;
   end ;
```

g) swapped_unsigned_value.m

```
function output = swapped_unsigned_value(identifier,nr_bits,value)

if identifier == 0 % Big-Endian
   if nr_bits == 8
      VV = strcat(sprintf('%0*X',2,value(1))) ;
   elseif nr_bits == 16
      VV = strcat(sprintf('%0*X',2,value(1)),sprintf('%0*X',2,value(2))) ;
   elseif nr_bits == 32
      VV = strcat(sprintf('%0*X',2,value(1)),sprintf('%0*X',2,value(2)),sprintf('%0*X',2,value(3)),sprintf('%0*X',2,value(4))) ;
   elseif nr_bits == 64
      VV =
strcat(sprintf('%0*X',2,value(1)),sprintf('%0*X',2,value(2)),sprintf('%0*X',2,value(3)),sprintf('%0*X',2,value(4)),sprintf('%0*X',
2,value(5)) ...
         ,sprintf('%0*X',2,value(6)),sprintf('%0*X',2,value(7)),sprintf('%0*X',2,value(8))) ;
   end ;

elseif identifier == 1 % Little-Endian
   if nr_bits == 8
      value = uint8(fliplr(value)) ;
      VV = strcat(sprintf('%0*X',2,value(1))) ;
   elseif nr_bits == 16
      value = uint16(fliplr(value)) ;
      VV = strcat(sprintf('%0*X',2,value(1)),sprintf('%0*X',2,value(2))) ;
   elseif nr_bits == 32
      value = uint32(fliplr(value)) ;
      VV = strcat(sprintf('%0*X',2,value(1)),sprintf('%0*X',2,value(2)),sprintf('%0*X',2,value(3)),sprintf('%0*X',2,value(4))) ;
   elseif nr_bits == 64
      value = uint64(fliplr(value)) ;
      VV =
strcat(sprintf('%0*X',2,value(1)),sprintf('%0*X',2,value(2)),sprintf('%0*X',2,value(3)),sprintf('%0*X',2,value(4)),sprintf('%0*X',
2,value(5)) ...
         ,sprintf('%0*X',2,value(6)),sprintf('%0*X',2,value(7)),sprintf('%0*X',2,value(8))) ;
   end ;
end ;
output = hex2dec(VV) ;
```