

Multi-path Penalty Metric in Underwater Acoustic Communication for Autonomy and Human Decision-making

by

Lieutenant Bradli Anne Howard
Submarine Officer, UNITED STATES NAVY

B.S. 2015, Nuclear Engineering, NORTH CAROLINA STATE UNIVERSITY

Submitted to the Department of Mechanical Engineering and the Joint Program in Applied Ocean Science & Engineering in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

and the

WOODS HOLE OCEANOGRAPHIC INSTITUTION

September 2021

©2021 B. Howard. All rights reserved.

The author hereby grants to MIT and WHOI permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Author

Department of Mechanical Engineering, MIT
Applied Ocean Science & Engineering, WHOI
August 6, 2021

Certified by

Henrik Schmidt
Professor of Mechanical and Ocean Engineering, MIT
Thesis Supervisor

Accepted by

Nicolas Hadjiconstantinou
Professor of Mechanical Engineering, MIT
Chair, Department Committee on Graduate Students

Accepted by

Gordon Zhang
Associate Scientist with Tenure, Applied Ocean Physics & Engineering, WHOI
Chair, Joint Committee for Applied Ocean Science & Engineering

Multi-path Penalty Metric in Underwater Acoustic Communication for Autonomy and Human Decision-making

by

Lieutenant Bradli Anne Howard

Submitted to the Department of Mechanical Engineering and the Joint Program in Applied Ocean Science & Engineering on August 6, 2021 in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering.

ABSTRACT

A novel performance metric to improve underwater digital acoustic communication, called Multi-path Penalty (MPP), is proposed as an alternative to traditional signal-to-noise ratio (SNR) methods in the context of the Arctic Beaufort Sea. MPP and SNR are compared alongside a third performance metric, Minimum Achievable Error (MAE), which replicates the operation of a channel estimate-based decision feedback equalizer in an acoustic modem. The three metrics are then tested in a hardware-in-the-loop Virtual Ocean simulator for an autonomous undersea vehicle (AUV) communicating with a collaborator. Using field data of modem statistics obtained during ICEX20 and expanded data supplied by the simulator, calibration of the three metrics to modem packet success is evaluated, resulting in a proposed recalibration for MAE. The AUV's ability to communicate when adaptively choosing its depth is analyzed above and below the Beaufort Lens, and settings for MPP's engineering variables are obtained. The results show MPP generally improves reception and demodulation of acoustic transmissions over SNR by approximately 5% within an operational range of 8 km, while achieving similar results to the more robust metric MAE. MPP is an improved utility for underwater digital acoustic communication in both marine autonomy and as a tactical decision aid.

Thesis Supervisor: Henrik Schmidt
Title: Professor of Mechanical and Ocean Engineering, MIT

ACKNOWLEDGMENTS

This work would not be possible without the extraordinary support of the United States Navy, which provided funding for this research, my degree, and my livelihood as an active duty submarine officer.

The Ice-Tethered Profiler data were collected and made available by the Ice-Tethered Profiler Program (Toole et al., 2011; Krishfield et al., 2008) based at the Woods Hole Oceanographic Institution (<https://www2.whoi.edu/site/itp/>).

Thank you to the excellent faculty and teaching staff at MIT and WHOI, who enabled me to expand my professional knowledge to the academic realm. Particular thanks to Dr. Erin Fischell, Dr. Julien Bonnell, Dr. Mike Benjamin, Dr. John Leonard, Dr. Sam Laney, Dr. Anna Michel, Caileigh Fitzgerald, Leanora Fraser, Kris Kipp, and Dr. Misha Novitsky.

During ICEX20, our lab found immeasurable support from Arctic Submarine Lab, the Underwater Development Center (UWDC), UUVRON, DEVRON-5, and all the staff, sailors and marines who make living in an Arctic ice camp possible year on year. Many, many thanks to John Joseph of the Naval Postgraduate School who allowed me to use the coolest CTD probe ever – so cool that King Neptune himself stole it out of jealousy. Thank you Naval Undersea Warfare Center (NUWC) for allowing us to expend several XCTD-1 probes.

Thank you to our entire ICEX20 MIT-WHOI-Bluefin Team including Professor Henrik Schmidt, Dr. Toby Schneider, Oscar Víquez Rojas, Dr. Rui Chen, LCDR Dan Goodwin, Dennis Giaya, Dan MacDonald, and Josiah DeLange. Thank you also to the extended LAMSS family members Dr. Supun Randeni, Blake Cole, Kristen Railey, Craig Evans, and Geoff Fox.

Thank you to (now Dr.) EeShan Bhatt and (soon-to-be Dr.) Oscar Víquez Rojas for being the best peer mentors a newbie grad student could have — and amazing friends.

Thank you to Dr. Toby Schneider and Dr. Jim Preisig for patiently answering my questions and supporting my extensive use of Netsim for this research.

Most importantly, thank you to my research advisor Professor Henrik Schmidt for sending me to the Arctic, encouraging me to learn, and teaching me how to investigate problems the Navy doesn't know are problems yet.

Finally, thank you to my husband, Duncan, and our cat Mochi — the bright side of quarantine during a global pandemic.

CONTENTS

1	INTRODUCTION	12
1.1	Environmental Variability in the Arctic Ocean	13
1.2	The Arctic Beaufort Sea as a Communication-denied Challenge	14
2	THEORY	15
2.1	Sound in Water	15
2.1.1	Ray Tracing	18
2.2	Signal-to-Noise Ratio (SNR)	21
2.3	Minimum Achievable Error (MAE)	22
2.4	Multi-path Penalty (MPP)	25
3	METHODS & SETUP	26
3.1	LAMSS Autonomy & Utilities	28
3.1.1	Adaptive Depth Behavior	29
3.1.2	Netsim: Hardware-in-the-Loop	31
3.2	Mission Setup to Test Metrics	32
4	RESULTS	37
4.1	Basic Metric Plots & Comparison	37
4.2	Single Depth Source & Receiver Analysis	49
4.2.1	Metric Plot Data at Single Depths	49
4.2.2	Single Depth Netsim Packet Success	52
4.2.3	MAE Recalibration	55
4.3	Determining p	57
4.3.1	Effect of Varied MPP vs. MAE	57
4.3.2	Depth Adaptive Autonomy with varied Penalty, p	61
4.4	Determining b	64
4.5	Validating Choice of τ	66
5	CONCLUSION & FUTURE WORK	68

A APPENDIX	70
A.1 Function to Calculate MPP in Matlab: mpp	70
A.2 Function to Calculate MAE in Matlab: minae	75
BIBLIOGRAPHY	85

LIST OF FIGURES

2.1	Example Beaufort Sea CTD Profile	17
2.2	Illustration of Snell's Law	18
2.3	Example Ray-tracing Plot (Munk Profile)	19
2.4	Example Transmission Loss Plot (Munk Profile)	20
3.1	LAMSS ICEX20 Beaufort Sea Experiment Setup	27
3.2	LAMSS Metric Calculation Flowchart	31
3.3	Netsim Hardware Setup	32
3.4	Example Lawnmower Run Pattern	33
4.1	SNR, Source Depth 30 m & 90 m (ICEX20)	39
4.2	Utility SNR, Source Depth 30 m & 90 m (ICEX20)	40
4.3	MPP, Source Depth 30 m & 90 m (ICEX20), Various p	41
4.4	Utility MPP, Source Depth 30 m & 90 m (ICEX20), Various p	42
4.5	MPP, Source Depth 30 m & 90 m (ICEX20), Various b	43
4.6	Utility MPP, Source Depth 30 m & 90 m (ICEX20), Various b	44
4.7	MPP, Source Depth 30 m & 90 m (ICEX20), Various τ	45
4.8	Utility MPP, Source Depth 30 m & 90 m (ICEX20), Various τ	46
4.9	MAE, Source Depth 30 m & 90 m (ICEX20)	47
4.10	Utility MAE, Source Depth 30 m & 90 m (ICEX20)	48
4.11	SNR, MPP (Various p), and MAE for Single Source & Receiver Depths	50
4.12	SNR & MPP Converted to Utility for Single Source & Receiver Depths	51
4.13	NETSIM Packet Success	53
4.14	NETSIM Packet Success vs. MPP (Various p) Utility	54
4.15	NETSIM Packet Success vs. MAE Utility	55
4.16	Recalibrated MAE Utility, Source Depth 30 m & 90 m (ICEX20)	56
4.17	SNR vs. MAE Histogram & Range Plots, Source Depth 30 m	58
4.18	SNR vs. MAE Histogram & Range Plots, Source Depth 90 m	58
4.19	MPP vs. MAE Histogram & Range Plots, Source Depth 30 m, Various p	59

4.20	MPP vs. MAE Histogram & Range Plots, Source Depth 90 m, Various p	60
4.21	Adaptive Depth Simulation Results for Varied p , Shallow Receiver (30 m)	63
4.22	Adaptive Depth Simulation Results for Varied p , Deep Receiver (90 m)	63
4.23	Adaptive Depth Simulation Results for Varied b , Shallow Receiver (30 m)	65
4.24	Adaptive Depth Simulation Results for Varied b , Deep Receiver (90 m)	65
4.25	Adaptive Depth Simulation Results for Varied τ , Shallow Receiver (30 m)	67
4.26	Adaptive Depth Simulation Results for Varied τ , Deep Receiver (90 m)	67

LIST OF TABLES

4.1	MAE Assumptions & Modem Parameters	38
4.2	Adaptive Depth Simulation Results for Varied p	62
4.3	Adaptive Depth Simulation Results for Varied b	64
4.4	Adaptive Depth Simulation Results for Varied τ	66

1 INTRODUCTION

“The sea is everything.”

Jules Verne, 1861

On March 9, 2020, two Russian Federation Tu-142 maritime reconnaissance aircraft flew over the ice camp where I was conducting field work for this thesis.¹ Though the planes were almost certainly there for the US submarines surfaced through the ice nearby, this event was simply the latest evidence in the observation that the Arctic Ocean now, more than ever, is relevant in terms of national strategic interest. Though the Cold War is now several decades in the past, it remains a geopolitically important region for nearby countries including the United States, Canada, and the Russian Federation. Even the People’s Republic of China has ambitions in the Arctic as part of its larger global strategy, despite not being traditionally considered an Arctic player.²

As sea ice extent is reduced year on year, the region becomes more accessible to surface vessels for a longer time frame during the year, with greater access in general on the Arctic ice basin periphery.³ Better access also leads to increased interest in the mineral and biological wealth of the Arctic for fishing, mining, and other industry apart from simple trade. With this increased international focus it is of paramount importance, therefore, to continue scientific characterization of the Arctic environment and to subsequently develop tools that maximize the success of operations – both surfaced and submerged.

¹Woody, *Russian reconnaissance planes were intercepted by US and Canadian fighters as they lurked over a submarine exercise in the Arctic*.

²Doshi, Dale-Huang, and Zhang, *Northern Expedition: China’s Arctic Activities and Ambitions*.

³Timmermans, Toole, and Krishfield, “Warming of the interior Arctic Ocean linked to sea ice losses at the basin margins”.

1.1 ENVIRONMENTAL VARIABILITY IN THE ARCTIC OCEAN

In addition to the more routine oceanographic engineering challenges – pressure, corrosion, hydrodynamics, hydrophobic electronics, etc. – the Arctic presents further complications. Extreme meteorological conditions, seasonal accessibility, polar sun cycle, magnetic polar drift, relatively sparse satellite coverage, and the presence of sea ice all require careful consideration for any kind of operational undertaking.

Sea ice presents a particular challenge not only for ice-breaking surface vessels, but for submersibles as well. Many sea ice covered regions can be reached by navigating beneath the ice keel. However, many crewed and autonomous submersibles normally rely on satellites while surfaced to periodically update their navigation systems (using GPS) and externally communicate. These vessels must forgo these updates unless the vessel has the ability to sense the thickness of, and successfully break through, the overhead sea ice layer, as US submarines do. For smaller submersibles, successful navigation and communication must be accomplished by other means.

Usually this leads to underwater acoustics, with yet another set of environmental challenges related to physics and environmental variability particular to the Arctic Ocean. Namely, the presence of shadow zones, multi-path propagation, and multi-directional ambient noise. Traditionally, the generalized sound speed profile (SSP) of the Arctic Ocean has been modelled by a cool, isovelocity layer of water near an ice-covered surface, followed by a linearly increasing velocity as a function of depth.⁴ However, a stagnant, warm intrusion layer has been observed in the Arctic Beaufort Sea in recent years.⁵ This layer causes large shadow zones in range and depth where refracting sound does not reach, rather than the uniform, upwardly-refracting environment of the last several decades. These shadow zones can negatively impact acoustic communication for deployed sensors if not properly mitigated.⁶

A combination of the refractive and reflective properties of sound in this environment implies that multiple echoes, or arrivals, of the same source can be received at a single range and depth. This ‘multi-path’ sound structure further complicates underwater acoustics, and for digital acoustic modems in particular. Receiving multiple arrivals of the same message within the same span of time can cause the modem to incorrectly demodulate the received transmission, potentially degrading the efficacy of an acoustic communication network. Though there are methods, such as error-correcting codes, which can recover lost information bits, the multi-path propagation mode must be well understood and modelled. Lastly, the varied composition of the

⁴Jensen, Kuperman, Porter, and Schmidt, *Computational Ocean Acoustics*, p. 306.

⁵Toole, Timmermans, Perovich, Krishfield, Proshutinsky, and Richter-Menge, “[Influences of the ocean surface mixed layer and thermohaline stratification on Arctic Sea ice in the central Canada Basin](#)”.

⁶Chen and Schmidt, “[Temporal and spatial characteristics of the Beaufort Sea ambient noise environment](#)”.

overhead sea ice layer also impacts ambient noise in the environment. Ice cracking and movement can contribute to overall ambient noise in broad and narrow band frequency spectra, and can be highly directional.

1.2 THE ARCTIC BEAUFORT SEA AS A COMMUNICATION-DENIED CHALLENGE

In an effort to overcome one of the many challenges posed by Arctic operations, this thesis aims to describe an alternative metric for underwater acoustic communication. Referred to as the Multi-path Penalty (MPP), this metric takes advantage of the ocean environment to improve the ability of an asset to successfully communicate with one or more collaborators while submerged. The goal of this metric is to provide an autonomous vehicle or a human decision-maker with a tool to evaluate the relative communication utility of a dense grid of possible range and depth combinations. This tool should have improved performance over traditional methods while achieving comparable performance to more complicated, computationally intensive methods.

Though this work focuses primarily on high frequency active encoded digital transmission and passive decoded reception, similar benefits may be extrapolated to a passive contact tracking scenario. After presenting the calculation, three engineering variables unique to MPP are analyzed in the context of the aforementioned field work performed in the Beaufort Sea. In this analysis, real conductivity-temperature-depth (CTD) data is used by a robust ocean impulse response simulator to generate real signals for use by a hardware network of acoustic modems. With a common acoustic environment, the various metrics are compared in simulation and evaluated. Lastly, MPP settings for the Beaufort Sea are recommended based on this evaluation, and a framework to establish these settings in other oceans is presented.

2 THEORY

This section discusses, in brief, a summary of the physical principles directly applicable to calculating three underwater acoustics performance metrics: Signal-to-Noise Ratio (SNR), Minimum Achievable Error (MAE), and MPP. Beginning with how sound travels in the ocean environment, this overview will summarize the oceanographic considerations of ray tracing and acoustic modem operations to present the already established metrics of SNR and MAE. Lastly, it will propose a calculation for a novel metric, MPP, which is the main subject of this work.

2.1 SOUND IN WATER

Sound is the movement of a medium's particles in time and space, described as a wave oscillating outwardly from a source. The fundamental wave equation, below, describes the relationship between time, space, and the medium of interest for a wave u .¹

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u \quad (2.1)$$

The constant, c , referred to as the sound speed or velocity, is a property of the medium's composition in units of distance per time and is the principal component which governs a wave's behavior. While waves can be thought intuitively to be oscillating in time, this relationship implies that they simultaneously oscillate in space. An explicit and familiar expression of this duality is found in the following:

$$c = \lambda f \quad (2.2)$$

Where λ is the wavelength in units of distance, and f is the frequency in units of 'per time' or cycles per time (i.e Hz). Wavelength, therefore, is the distance travelled by one oscillation of the wave and frequency describes how long it takes. The final result is the sound speed, c , which, as a property of a medium, describes exactly how a wave will travel through it.

Although highly dependent on the medium's specific composition and state, a common single value approximation for sound speed in air is 340 m/s (\approx 1115 ft/s or \approx 760 mph). In water,

¹Jensen, Kuperman, Porter, and Schmidt, *Computational Ocean Acoustics*, p. 69.

sound travels roughly five times faster, or approximately 1500 m/s (\approx 4900 ft/s or \approx 3400 mph). However, the speed of sound is not homogeneous within a medium, and while approximations with a single value can be made to easily solve the wave equation, the true behavior of sound is much more complex.

In many oceanographic applications, a depth-dependent sound speed profile (SSP) is calculated from temperature, conductivity, and pressure data for a given vertical column of seawater. Typically, an instrument referred to as a CTD (Conductivity-Temperature-Depth) probe is lowered from the surface of the water to the bottom collecting this data, and then uses an empirical formula to determine the sound speed profile as a function of water depth, z . While there are several different formulations, a simple equation published by Medwin in 1975, which finds sound speed within 0.1 m/s for depths up to 1 km, is given below.²³

$$c = 1449.2 + 4.6T - 0.055T^2 + 0.00029T^3 + (1.34 - 0.01T)(S - 35) + 0.016z \quad (2.3)$$

Where sound speed (c) is in m/s, temperature (T) in degrees Celsius, salinity (S) in parts per thousand, and depth (z) in meters.⁴ More accurate calculations of sound speed typically include more terms and have more specific initial conditions, such as Mackenzie's nine term version, which may be used for depths up to 8 km.⁵

By using only a vertical profile of sound speed with depth, we make the assumption for the analysis in this work that the sound speed is otherwise range independent. This means we do not consider changes to sound speed with varying latitude and longitude, which is a reasonable assumption for calculations taking place over short ranges. It is important to note, however, that at very long ranges (e.g. trans-oceanic) additional complexity in the range dependence of the sound speed must be considered, particularly for depths below 1 km.⁶⁷ Oceanographic currents, fronts, and eddies also influence the horizontal sound speed field, and are similarly not considered for the purposes of this work.

²Medwin, *Speed of sound in water: A simple equation for realistic parameters*.

³Medwin and Clay, *Fundamentals of Acoustical Oceanography*.

⁴Jensen, Kuperman, Porter, and Schmidt, *Computational Ocean Acoustics*, p. 3.

⁵Mackenzie, "Nine-term equation for sound speed in the oceans".

⁶Spiesberger and Metzger, "A new algorithm for sound speed in seawater".

⁷Spiesberger and Metzger, "New estimates of sound speed in water".

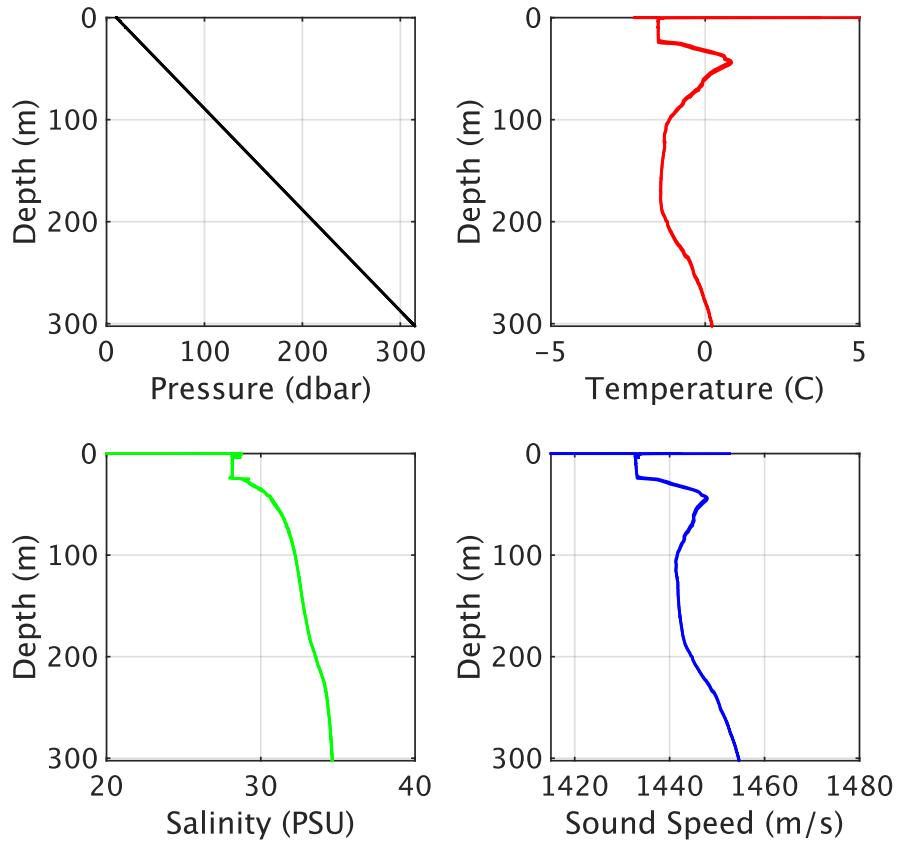


Figure 2.1: Raw data of pressure, temperature, salinity, and sound speed taken in the Beaufort Sea during March of 2020. Salinity is derived from conductivity, absolute pressure is converted to depth, and the Mackenzie equation is used to obtain the resulting sound speed in m/s.

An example Beaufort Sea profile of temperature, salinity, pressure, and resulting sound speed is shown in Figure 2.1. As expected from the large temperature terms present in Equation 2.3, the shape of the resulting profile is strongly influenced by the warm layer from 30 to 90 m. With sound speed thus defined as a function of depth, the column of water can then be modelled as an acoustic waveguide, bounded by the sea surface at the top and the sea floor at the bottom. In Arctic applications, the sea surface may also include an ice layer with a defined thickness and composition.

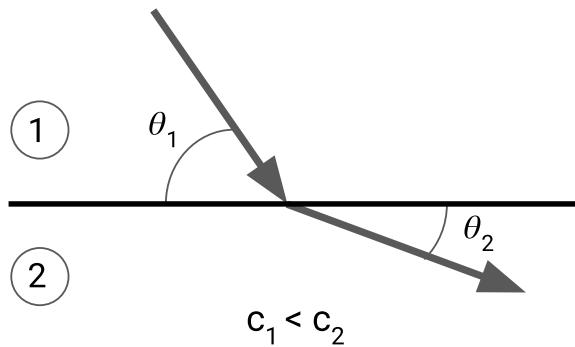


Figure 2.2: Illustration of Snell's Law

2.1.1 RAY TRACING

Waves of sound refract within the acoustic waveguide according to Snell's Law (Equation 2.4), which describes the conservation of energy and momentum for a wave over an interface of two distinct media as shown in Figure 2.2.

$$c_1 \sin \theta_1 = c_2 \sin \theta_2 \quad (2.4)$$

Since sound speed is not constant within the vertical water column, each incremental step taken by a wave results in a new θ_2 which changes the ray's direction. If c_2 is higher than c_1 , the ray will have a shallower θ_2 , refracting upwards. Conversely, if c_2 is lower than c_1 , the ray will have a steeper θ_2 , refracting downwards. Generally, rays travel outwardly from a source and refract towards depths with lower sound speed. A common phrase is that sound is 'lazy' in this way, meaning that cooler layers of water collect slow moving sound.

Ray tracing is a modelling method for rays of sound emanating from a source in a given waveguide. A set of ray-tracing equations derived from Equations 2.1 and 2.4 can be used to determine an individual ray's path and the pressure amplitude of the ray along it.⁸ Using a point-source approximation, a fan of rays with varied initial transmission angles can be solved to show the behavior of sound in a certain environment. An example Munk SSP alongside a plot of seven rays is shown in Figure 2.3.⁹ All the rays tend to bend toward the lowest region of sound speed, and oscillate around the minimum near 1200 m in depth.

With a large number of rays, an acoustic pressure field can be calculated to show the effect of the acoustic source at many ranges and depths. For each possible receiver location, the complex amplitudes of the rays that arrive at that specific location, referred to as eigenrays or 'arrivals', can be combined to represent the total acoustic pressure field. In this work we consider incoherent

⁸Jensen, Kuperman, Porter, and Schmidt, *Computational Ocean Acoustics*, Ch. 3.

⁹Jensen, Kuperman, Porter, and Schmidt, *Computational Ocean Acoustics*, p. 212.

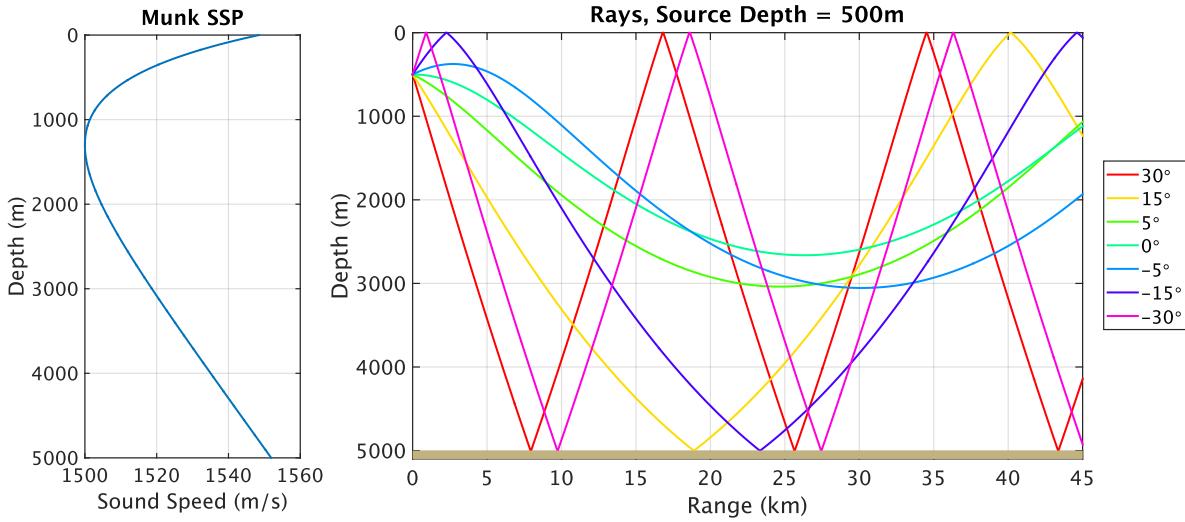


Figure 2.3: Example Ray-tracing Plot for a Non-linear SSP (Munk Profile). Ray-tracing Matlab program written in collaboration with Ariel Chouchana & Kurran Singh based on a template made by Dr. Rui Chen during the Environmental Ocean Acoustics course (2.681).

transmission loss, which is the determination of the pressure field by the sum of the square absolute complex amplitudes of arrivals (N) referenced in dB to an underwater source producing $1 \mu\text{P}$ of pressure at 1 m given by Equation 2.5.

$$\text{TL}(r, z) = -20 \log_{10} \left(\frac{\sum_{i=1}^N |A(r, z)_i|^2}{A_{\text{ref}}} \right) \quad (2.5)$$

For the Munk SSP discussed before, the transmission loss plot using this relationship is shown in Figure 2.4 for a 10 kHz source using 2000 rays. The plot shows that as rays propagate outwards from a source, the overall field loses amplitude, but less rapidly in some areas relative to others. In this way, transmission loss plots are used to characterize the propagation of sound generally in an ocean environment.

Although ray methods are computationally quick and physically intuitive, there are notable caveats and assumptions. The source frequency is assumed to be high (i.e. wavelength is short) and surface compositions are often approximated. When calculating the pressure field, regions where no rays propagate will have an artificially asymptotic result. This caustic effect can generally be reduced by including a sufficiently large amount of rays in the pressure field calculation, but the effect may still be observed in shadow zone regions.

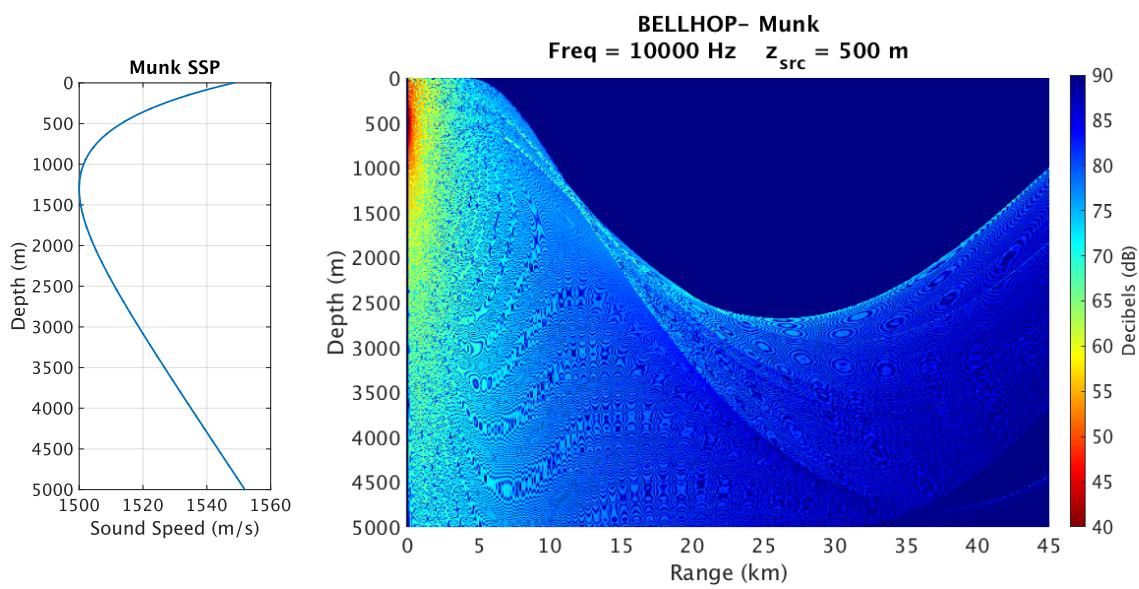


Figure 2.4: Example Transmission Loss Plot for a Non-linear SSP (Munk Profile) for a 10 kHz source using 2000 rays. Generated with BELLHOP.

2.2 SIGNAL-TO-NOISE RATIO (SNR)

The traditional performance metric in underwater acoustics for communication between a set of transducers is the Signal-to-Noise Ratio (SNR). It represents the combined effect of a source, the ambient noise present in the environment, and the transmission loss of the source from a known SSP. The relationship, also sometimes referred to as the sonar equation, is given below.

$$\text{SNR} = \text{SL} - \text{NL} - \text{TL} \quad (2.6)$$

Where SL (Source Level) is the acoustic source level, NL (Noise Level) is the level of background noise, and TL is as described in Equation 2.5.¹⁰ All terms are conveyed in decibels (dB) referenced to an acoustic level 1 m from a source producing 1 μP of pressure.

SNR can be modelled entirely on the known acoustic environment, and in this format does not include losses inherent to the hardware of the receiver. It assumes only that a receiver would sense the resulting signal level at a given depth and range. While the transmission loss calculation takes into account the complex phase component of an arrival's amplitude, the total SNR field represents a static moment in time from a continuous source transmission. It assumes that all possible arrival paths are being received at the same moment in time from a continuously transmitting source, when in reality this may not be the case from a pulsed or windowed transmitter.

Sometimes, the simple reception of sound *in general* from a known source location may be sufficient for a platform's operation. In a positioning method known as ultra-short baseline (USBL), a transmitter mounted on a platform with a known location (such as a ship with GPS) sends an acoustic pulse to an underwater collaborator (such as an AUV). The time it takes for the pulse to arrive at the collaborator can be converted into a range and, depending on the number of receivers present, direction to the transmitter. The collaborator then transmits a similar pulse, and the first platform can perform the same calculation in reverse. In this example, the timing of the acoustic pulse – and therefore receiving the pulse at all – is the main component that matters.

Other applications use sound to instead send a digital acoustic message. In this context, information is modulated into an acoustic pulse by a modem (MODulation-DEModulation), transmitted, received, and then demodulated for use by the receiving platform.

¹⁰Jensen, Kuperman, Porter, and Schmidt, *Computational Ocean Acoustics*, p. 711.

2.3 MINIMUM ACHIEVABLE ERROR (MAE)

While good for estimating reception likelihood based on the environment alone, SNR may not fully account for modem-specific considerations. Multi-path propagation complicates the timing of discrete-time signals processing equipment, or operations where the timing and duration of a signal is a predominant factor. Therefore it would be desirable to have a metric that can also inform the *quality* of message reception and subsequently the likelihood of successful decoding.

In general, underwater digital acoustic communication can be characterized as lossy, while having high latency and low bandwidth. Unlike electromagnetic forms of digital communication, the speed of sound in water limits the volume and speed of information transfer, requiring creative means to encode as much data as reliably as possible. Often, information in known formats (such as marine industry standardized NMEA 0183 sentences) can be manipulated with algorithms to compress data to use as small a number of zeros and ones as possible.¹¹ Binary data is then translated in a modulation routine into the transmission itself.

One such modulation routine is referred to as Phase-Shift Keying (PSK). Given that sound is a wave, it follows a sinusoidal pattern with a defined phase. By shifting the phase of a transmission to represent a 0 or a 1, a modem encodes digital information into an analog signal. In Quadrature Phase-Shift Keying, the sum of a modulated sine (quadrature component) and cosine (in-phase component) are used to represent four combinations of binary numbers rather than a single 0 or 1 individually (i.e. 00, 01, 10, & 11). Since the ambient environment can have a strong dilating and distorting effect on an original signal, a high degree of fidelity in message decoding is important to minimize symbol loss.

This inherently imposes additional requirements beyond the simple ability to receive a transmission. When the transmission arrives, it needs to be sufficiently intact such that the modem can interpret the message correctly. Timing becomes extremely important to modem operations; when to begin demodulating, accounting for Doppler shift between the platforms, discriminating between two arrivals of the same message, etc. Receivers in multi-path propagation environments must deal with ancillary arrivals of the same message as ‘noise’ and cope with ice-bounce effects as well as all other noise sources.

In this case, underwater acoustic modems may employ an adaptive decision feedback equalizer to reduce inter-symbol interference in multi-path environments.¹² As part of the overall modem hardware, the equalizer’s function is to take an input from the ambient environment, estimate the effect it will have, and attempt to reverse that effect on the incoming signal in order to improve

¹¹Shannon, “A mathematical theory of communication”.

¹²Gallimore, Partan, Vaughn, Singh, Shusta, and Freitag, “The WHOI micromodem-2: A scalable system for acoustic communications and networking”.

the quality of the decoded data. That discrete-time estimation, or decision feedback element, takes both the time-varying acoustic environment and the hardware properties of the modem into account.

A detailed analysis of the calculation of a performance estimate called MAE for three different types of equalizers can be found in J. Preisig (2005).¹³ The same adaptive estimation that an equalizer is making may be replicated mathematically and used independently as a predictive performance metric. MAE can then be used to evaluate the same grid of receiver ranges and depths as SNR, but includes the modem's own adaptive decision feedback as part of the model. A brief summary of this method follows, which is based on the operation of a channel estimate-based decision feedback equalizer (CE-DFE). Example code used to calculate MAE from a BELLHOP arrivals file is included in Appendix A.2.

From a known set of arrivals and the receiver's heading and speed, the Doppler shift for each eigenray, denoted by i , can be calculated as follows.

$$f_{di} = 1 + \frac{v_{rS} \cos(\theta_{Si}) - v_{rR} \cos(\theta_{Ri})}{c} \quad (2.7)$$

And the delay by,

$$d_{di} = \cos|d_i|(1 + v_{rR} \cos(\theta_{Ri})) \quad (2.8)$$

Where f_{di} is the Doppler shift of the eigenray, θ_{Si} is the source declination angle, θ_{Ri} is the receiver declination angle, v_{rS} is the relative velocity of the source to the receiver, and v_{rR} is the relative velocity of the receiver to the source.

The Doppler scaled waveform impulse response is then generated using the Doppler shift and delay of each eigenray, the modem bandwidth (B), carrier frequency (F_c), sampling frequency (F_s), and desired number of discrete fast Fourier transform samples (nfft). Then, given ambient noise level (NL), channel baud, and equalizer feed forward taps, a matrix of cross spectral density and down-sampled noise covariance can be estimated. The sync-point, or point of maximum impulse response, is determined in order to set the scale of a discrete Green's function, which is then combined with the noise correlation to calculate MAE as below.

$$\text{MAE} = -10 \log_{10} \frac{1}{1 + \hat{g}_0^h \hat{Q}^{-1} \hat{g}_0} \quad (2.9)$$

Where \hat{Q} is the estimated effective noise correlation matrix, and \hat{g}_0 is a column vector of the estimated impulse response at discrete time $n = 0$.

¹³Preisig, "Performance analysis of adaptive equalization for coherent acoustic communications in the time-varying ocean environment".

In summary, this calculation takes both the environment and modem into account to evaluate performance. However, MAE can be time consuming and computationally expensive since it requires many instances of impulse response to be generated for a single possible receiver location. More practically, modem parameters may not always be available to a decision-maker even if computational resources are not limited. While SNR is a faster, lighter performance metric, MAE is much more robust – a metric that would provide an ‘in-between’ level of evaluation might be desired instead.

2.4 MULTI-PATH PENALTY (MPP)

A third performance metric, MPP, is an informed modification of the traditional SNR calculation.

$$MPP = SL - NL - TL - MPL \quad (2.10)$$

Where MPL is then given by:

$$MPL = -10 \log_{10} \left(1 - \frac{P_{mpp}}{P_{\max}} \right)^p - b N \quad (2.11)$$

P_{mpp} is the power of the second strongest ray arriving less than τ seconds before or after the next ray, P_{\max} is the power of the strongest arriving ray, p is the multi-path penalty factor, N is the total number of top and bottom bounces on the eigenrays, and b is the bounce penalty factor.

MPP simply penalizes receiver ranges and depths that experience multiple strong arrivals, and paths that experience many bounces. Ideally, this new metric would combine the advantages of both SNR and MAE. By introducing the variables p , b , and τ , the goal is to recapture some of the advantages associated with a modem decision feedback equalizer. Since MAE takes arrival timing, strength, and bounce effects into account, these new variables would modify the more traditional SNR calculation to achieve a lightweight, better performance metric.

While the ray methods described previously provide the arrival inputs to this calculation, the three engineering variables (p , b , and τ) are arbitrary and must be empirically determined. Since the Arctic environment of the Beaufort Lens is a challenging multi-path environment, the main goal of this thesis will be to determine the settings of the three engineering variables for the Beaufort Sea to improve digital acoustic communication. The working hypotheses of this metric are:

- Proper settings of p , b , and τ for a given environment will maximize MPP's performance.
- MPP will produce performance improvement over SNR because it takes the power, timing, and physical path of each arrival into account.
- MPP and MAE will produce similar performance because both take the multi-path ambient environment and arrival timing into account.

3 METHODS & SETUP

In March of 2020, the Laboratory for Autonomous Marine Sensing Systems (LAMSS)¹ participated in the US Navy's ICEX20 on the sea ice of the Beaufort Sea, North of Prudhoe Bay, Alaska. The on-ice science team consisted of Prof. Henrik Schmidt, Dr. Toby Schneider (GobySoft, LLC), graduate students Oscar Víquez & Dr. Rui Chen, LCDR Daniel Goodwin and myself.

In the experiment a Bluefin-21 AUV, *Macrura*, was deployed under the ice in an acoustic network consisting of four ice-moored buoys arranged in a square with an area of roughly 4 km². Each buoy was configured with two hydrophones, one placed at 30 m and the other at 90 m depth. One set of diagonal buoys were configured with the ability to transmit at 30 m, and the other set at 90 m. Figure 3.1 shows the acoustic range and equipment setup.² This underwater acoustic network employed WHOI Micromodem-2 modules³ with a carrier frequency of 10 kHz for digital acoustic messages on a time-synchronized, alternating 15 second transmission schedule between the buoys and *Macrura*.

While the primary research objective during ICEX20 was to demonstrate a novel under-ice navigation framework, it also provided the opportunity to evaluate modem performance in the real environment. MPP was integrated into the autonomy suite prior to the experiment, and was calculated alongside SNR for several deployments by the modem network software. As a rough estimate, p of 3, b of 3, and τ of 20 ms were used in the configuration.

The amount of data gained in ICEX20 regarding modem performance was somewhat sparse, but statistics of modem performance between the four buoys and with *Macrura* were collected. This initial result, showing average packet success with range for the buoy combinations of 30 m and 90 m alongside predicted MPP and SNR was published by Dr. Toby Schneider shortly after the experiment.⁴ Average packet success in this case is the percentage of transmitted messages that were successfully received and decoded for a given range bin. For both transmission layers,

¹[The Laboratory for Autonomous Marine Sensing Systems](#).

²Randeni, Schneider, and Schmidt, “[Construction of a high-resolution under-ice AUV navigation framework using a multidisciplinary virtual environment](#)”.

³Gallimore, Partan, Vaughn, Singh, Shusta, and Freitag, “[The WHOI micromodem-2: A scalable system for acoustic communications and networking](#)”.

⁴Schneider, Schmidt, and Randeni, “[Self-Adapting Under-Ice Integrated Communications and Navigation Network](#)”.

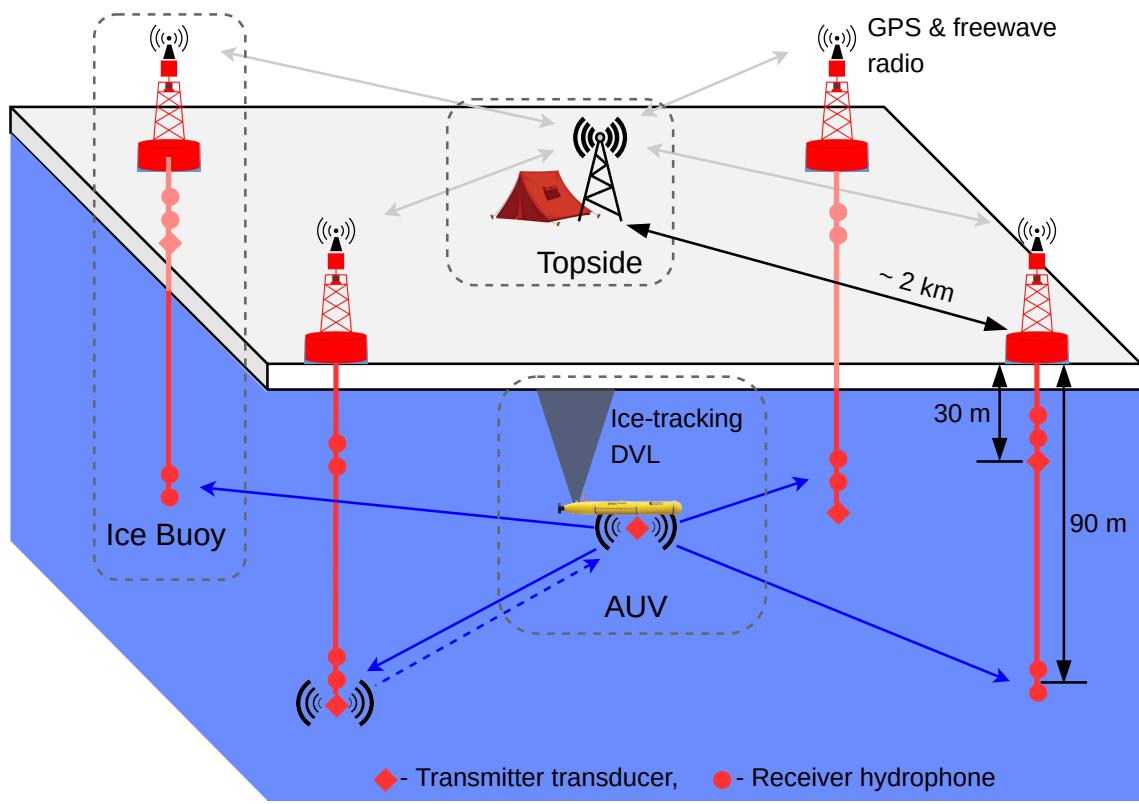


Figure 3.1: LAMSS ICEX20 Beaufort Sea Experiment Setup. Figure Credit: Dr. Supun Randeni & Dr. EeShan Bhatt

packet success more closely followed MPP than SNR over all ranges. In particular, SNR overestimates packet success in the shallow transmit, shallow receive scenario at all ranges.

Throughout the five day set of AUV operations, ten CTD casts were opportunistically collected for use by the navigation framework, and to examine acoustic environment variability over the course of the experiment. The first five in chronological order were obtained with a retrievable RBR Concerto-3 CTD profiler, and the latter five were obtained with expendable Lockheed Martin XCTD-1 modules. The environmental variation and impact on the acoustic operations of the vehicle navigation system is the subject of ongoing research.

In the following analysis, the real data from a CTD (RBR) profile obtained on March 9, 2020 at 1202 AKST (71.184, -142.407) is used in the ray-tracing and arrival generating software as well as within the LAMSS autonomy suite. Below 300 m, the profile is supplemented with a smoothed curve of modelled and real data from two external resources, based on methods presented by E. Bhatt (2021).⁵ The modelled data is from the Hybrid Coordinate Ocean Model (HYCOM)⁶ and the real data is from the WHOI Ice-tethered Profiler (ITP) program.⁷ Bathymetry in the operational region of ICEX20 was not flat, but is assumed to be constant at 2680 m for this analysis.

3.1 LAMSS AUTONOMY & UTILITIES

The autonomy software used in this research is a collection of applications implemented with MOOS-IvP (Mission Oriented Operating Suite-Interval Programming), Ocean Acoustics Library (OALIB)⁸ ocean modeling tools, and various third-party interfaces — collectively referred to as the LAMSS autonomy codebase. MOOS-IvP⁹ extends the MOOS¹⁰ robot middleware with an autonomous helm, a suite of helm behaviors, run-time utilities and post-mission analysis tools. Behaviors in the IvP Helm are coordinated using multi-objective optimization.

A MOOS-IvP enabled vehicle contains a database process called the MOOSDB, which is a publish-subscribe application that coordinates other applications running in parallel. Applications may subscribe to a number of variables, for example ‘NAV_X’ and ‘NAV_Y’, which may be primarily updated by an app publishing data from a GPS receiver. At a controlled timing interval, these apps interact with the database and perform their functions. A single MOOSDB and con-

⁵Bhatt, “A Virtual Ocean framework for environmentally adaptive, embedded acoustic navigation on autonomous underwater vehicles”.

⁶Chassignet, Hurlburt, Smedstad, Halliwell, Hogan, Wallcraft, Baraille, and Bleck, “The HYCOM (HYbrid Coordinate Ocean Model) data assimilative system”.

⁷Toole, Krishfield, Timmermans, and Proshutinsky, *The Ice-Tethered profiler: Argo of the Arctic*.

⁸*Acoustics Toolbox (BELLHOP)*.

⁹*MOOS-IvP*.

¹⁰*MOOS*.

nected apps are referred to as as ‘MOOS community’, and it is possible to run multiple communities on a single vehicle, or single communities on multiple vehicles. Apps to control hardware are common, but more often MOOS-IvP is employed to control autonomous decision-making as a ‘backseat’ computer while a separate, interfaced ‘frontseat’ computer translates backseat commands into direct vehicle control signals. This makes it easy to separate the autonomy software from the actual vehicle for testing and simulation. In the LAMSS codebase, multiple communities can be launched on a single Ubuntu machine to simulate a vehicle backseat, frontseat, and a command community (generally referred to as ‘topside’ in AUV mission sets).

The LAMSS autonomy codebase includes apps to handle acoustic communications, under-ice navigation, environmentally adaptive vehicle behaviors, mission & contact management, array processing, and real time ocean environment simulation & prediction, among others. In particular, LAMSS includes the Virtual Ocean simulator, which is able to produce real signals propagated through the ocean environment and send them with proper timing to real hardware, if configured.¹¹ Given environmental parameters for sound speed, noise frequency band & directionality, ice properties, bathymetry, floor properties, and a number of other environmental inputs, the Virtual Ocean can reproduce the impulse response of the ocean onto an original, real transmission. The simulator accurately reproduces attenuation, refraction, reflection, and ambient noise, resulting in a high-fidelity ‘received’ signal being passed to receiving modem hardware in real-time with proper time delay. In this work, the ability to simulate and recreate the ICEX20 experiment in a high fidelity autonomy simulation with real modem hardware is employed to investigate the various metrics and settings.

3.1.1 ADAPTIVE DEPTH BEHAVIOR

The vehicle can be configured to adaptively seek the best depth to communicate with a collaborator based on SNR, MPP (with desired p , b , & τ), or MAE. The program controlling this input to the depth behavior, called pDepthAdapt, interfaces with other applications to calculate SNR, MPP, or MAE directly and convert it to an equivalent utility for depth choice comparison. The autonomy suite then decides which depth is most desired and directs the frontseat computer to achieve and maintain the depth, which is updated based on the vehicle’s current range to the receiver target.

The ray method software employed to generate arrivals for the range and depth possibilities is BELLHOP, an acoustic ray-tracing software maintained by OAL.¹² Given an environmental

¹¹Schneider and Schmidt, “NETSIM: A realtime virtual ocean hardware-in-the-loop acoustic modem network simulator”.

¹²Porter, *The BELLHOP Manual and User’s Guide*.

parameter definition file ‘env’ it calculates and stores the eigenrays for the specified ranges and depths in an arrivals file ‘arr’. This file is then read by pDepthAdapt and the configured metric calculation is performed.

Using the real data collected in ICEX20, a metric utility conversion to normalize data was devised:

$$\overline{M} = \max\left(\min\left(\frac{80(M - 30)}{50} + 20, 100\right), 0\right) \quad (3.1)$$

Where M is a metric value for a given receiver location and the resulting \overline{M} is the normalized value expressed in percent utility from 0 to 100. That is, for any metric level greater than or equal to 80 dB, the equivalent utility is 100%. For metric levels lower than or equal to 17.5 dB the equivalent utility is 0%. This method roughly conforms a given matrix of receiver depth and range metrics to a bounded value of utility for use by the MOOS-IvP decision helm. Converting to percent utility is also a more intuitive value for a human decision-maker than one reported in ‘performance metric decibels’.

The main drawback of this metric conversion, apart from its basis on the sparse ICEX20 dataset, is how it treats the magnitude of negative values. The SNR calculation does not produce negative numbers, but the MPL calculation of Equation 2.11 allows for increasingly negative values as p and b become larger. The MAE calculation results in the magnitude of a complex number and thus does not produce negative numbers. Therefore, it can be expected that as MPP becomes more negative, the effect of an increasing penalty will become fractionally smaller as the negative values at a certain location ‘floor’ to 0% utility.

It is also important to note that while pDepthAdapt is running, competing inputs are also impacting the control decisions made by the MOOS-IvP helm for depth choice. For example, in real and simulated experiments, *Macrura* is limited by separate behaviors to prevent exceeding 200 m for hardware limit concerns and from impacting the overhead ice keel. The helm then chooses depth that best solves the input constraints, and continues to re-solve the problem at the predetermined interval with the most recently published inputs.

To summarize, a known environment is loaded into the LAMSS autonomy suite, then pDepthAdapt interfaces with BELLHOP to compute arrivals, perform the metric calculation, and convert the metric grid to utility before sending the information to the vehicle’s IvP-Helm decision-maker. The engineering variables present in the calculation of MPP are p , τ , and b as discussed in the previous chapter, while MAE depends on many inputs from the vehicle itself as well as the onboard modem. A visual of this information flow path is shown in Figure 3.2.

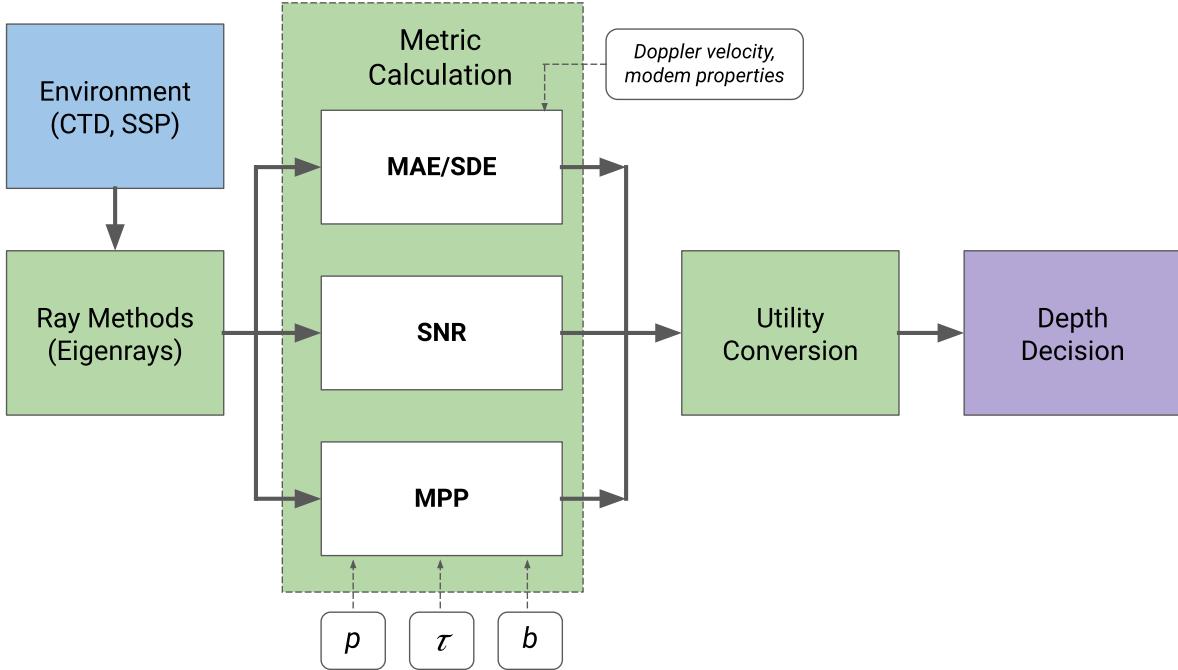


Figure 3.2: LAMSS Metric Calculation Flowchart. Blue represents data obtained outside of the LAMSS autonomy framework, green is the computational flow through the application pDepthAdapt, and purple is made either by an autonomous vehicle’s IvP-Helm or a human decision-maker.

3.1.2 NETSIM: HARDWARE-IN-THE-LOOP

In advance of the ICEX20 experiment, engineering tests of the autonomy suite on actual hardware were performed to simulate as closely as possible real-time operations. Apart from testing *Macrura* itself, a micromodem setup mimicking the topside hardware arrangement for the experiment was also tested.

The hardware setup used, called Netsim, consists of an audio simulation server, GPS module for Precise Positioning Service (PPS) timing, four independent WHOI Micromodem-2 modules, an off the shelf audio mixer, and an internet router for remote testing.¹³ The connections and basic configuration are shown in Figure 3.3. The modems encode, generate, transmit, receive and decode actual acoustic messages exactly as they would if deployed in the field. Logged modem data include actual SNR in and out, Mean Square Error (MSE), packet success, and a variety of other modem statistics.¹⁴ The server, running the LAMSS Virtual Ocean discussed previously, uses the audio mixer to appropriately synthesize the input signal at the receiving modem with the environmental impulse response modelled for the receiver location. A computer running a

¹³Schneider and Schmidt, “NETSIM: A realtime virtual ocean hardware-in-the-loop acoustic modem network simulator”.

¹⁴*Micromodem-2 User’s Guide (Version number: 1.2)*.

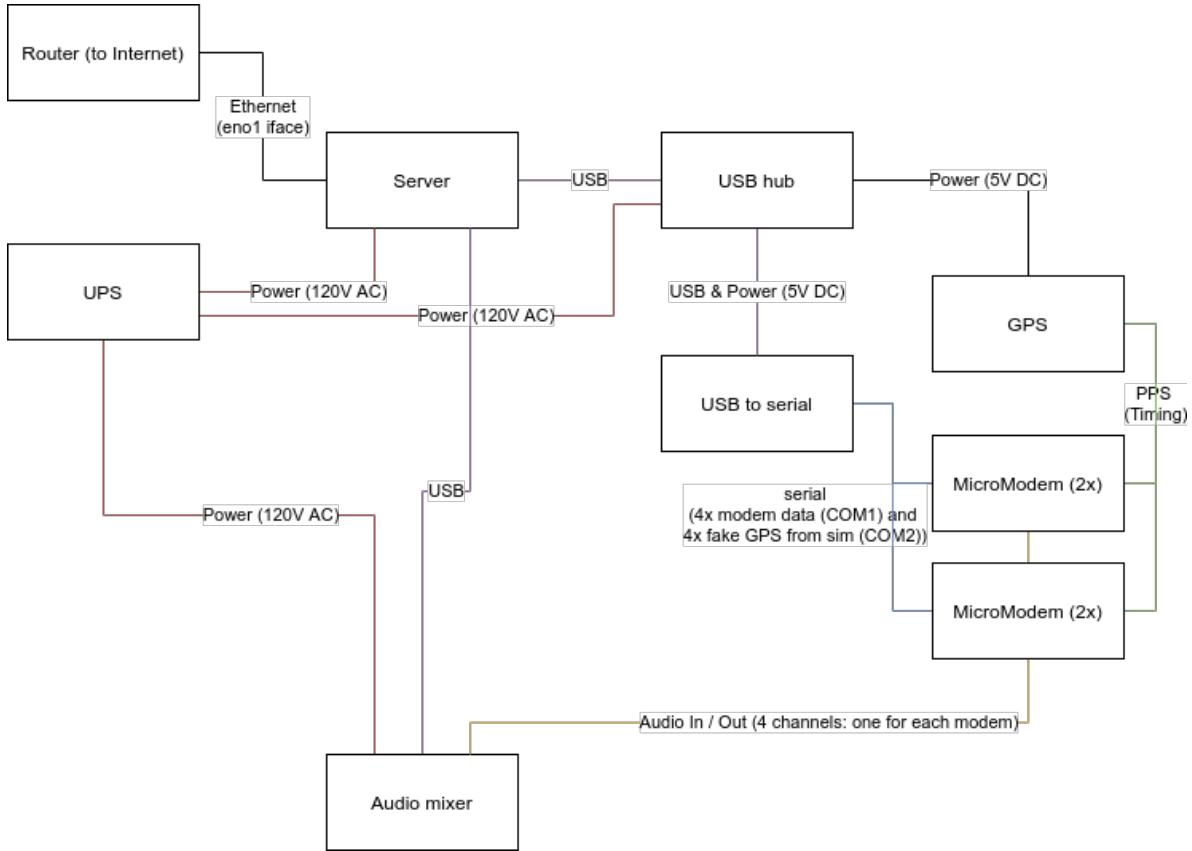


Figure 3.3: Netsim Hardware-in-the-Loop Server, Remote Interface, Audio Mixer, & Micromodem Setup.
 Figure Credit: Dr. Toby Schneider

LAMSS autonomy simulation can remotely connect to Netsim and make use of the modems as if an actual operation were taking place.

The same framework is now used to test the performance of the various communication metric configurations. In this way, Netsim may be used to generate a larger dataset than was acquired during ICEX20 to evaluate the response of SNR, MPP, and MAE.

3.2 MISSION SETUP TO TEST METRICS

The simplest setup is to send a simulated *Macrura* to perform a ‘lawnmower’ pattern, while allowing it to choose its own depth to maximize communication packets with a single buoy (H1) at a single receiving depth. Since the ICEX20 acoustic environment has a strong layer and shadow zone effect, simulation missions are set up to obtain sufficient data for a collaborator receiver above the Beaufort Lens (Shallow, 30 m), and then repeated with the receiver below the lens (Deep, 90 m). Simulation missions are also configured to significantly challenge the range be-

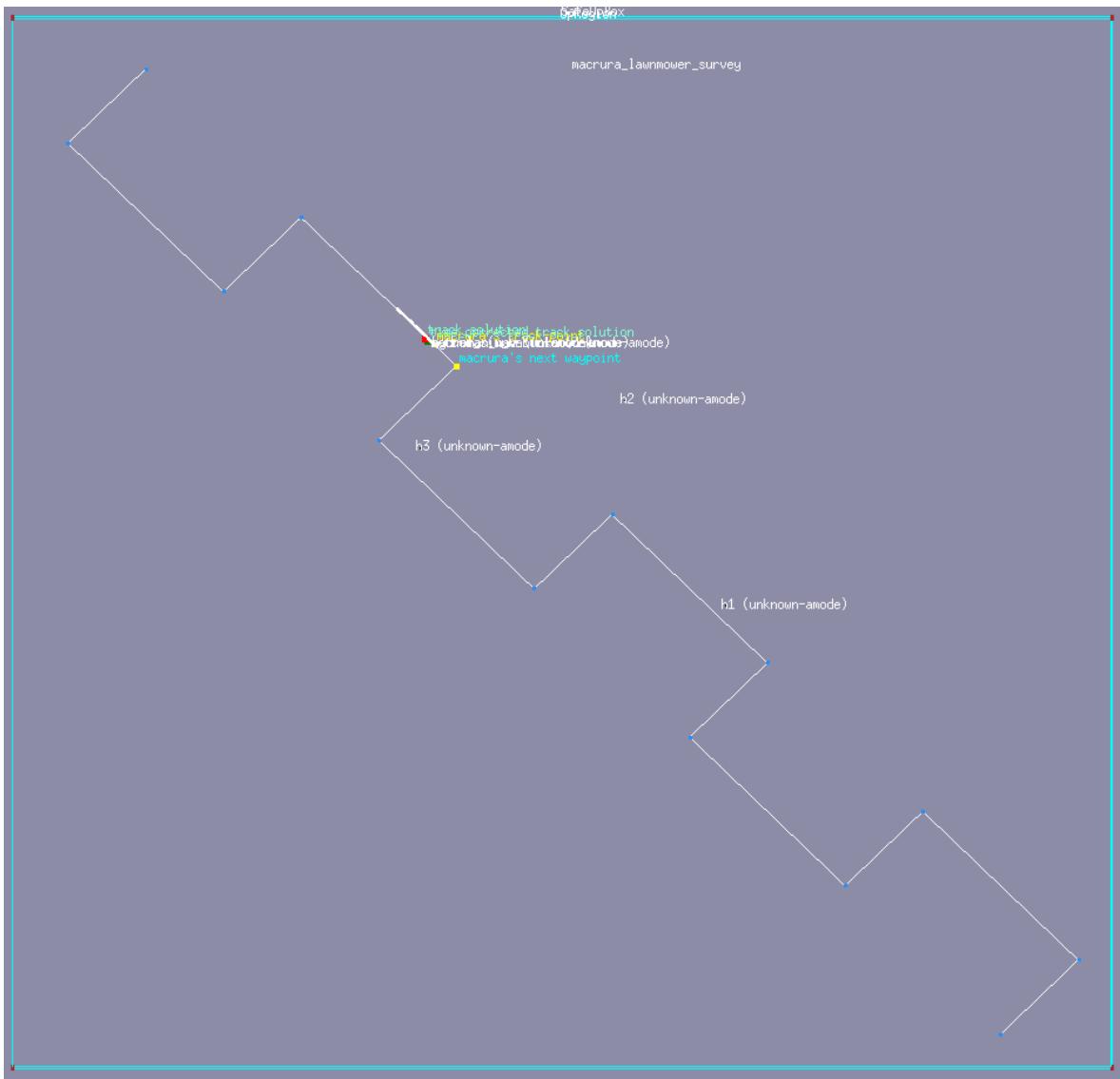


Figure 3.4: An example lawnmower run pattern, as shown by the MOOS app pMarineViewer. The outer blue cyan line is *Macrura*'s operational box. The white path shows the pattern, with blue dots at the corners representing waypoint objectives. The three buoy locations are labelled 'h1', 'h2', and 'h3'. The overlapping points and text represent actual and corrected track solutions for *Macrura*'s location as produced by the navigation simulation and ICNN.

tween *Macrura* and H1, out to approximately 8 km. An example of a lawnmower pattern is shown in Figure 3.4. Of note, Netsim has a total of four Micromodems, which allows for three buoys plus one for *Macrura* rather than four individual buoys as were used in ICEX20.

Ideally, a single ‘fire and forget’ configuration of MPP settings for the general environment would be most desirable during a vehicle deployment. While the analysis is performed specifically for the observed environment of the Beaufort Sea, the same method could be employed to determine the best settings for a different SSP. The simulation test plan procedure is outlined as follows.

1. Load the desired environment and bathymetry into Netsim and the local LAMSS configuration.
2. Validate packet success statistics at known depths for the loaded environment. With *Macrura* limited to a single depth, collect modem statistics to all three receiver buoys as below:
 - *Macrura* shallow, buoys shallow.
 - *Macrura* shallow, buoys deep.
 - *Macrura* deep, buoys shallow.
 - *Macrura* deep, buoys deep.

Calculate actual packet success for appropriate range bins. Determine if an adjustment is needed to the metric-utility conversion (Equation 3.1) by comparing packet success to predicted SNR, MAE, and varied settings of MPP over the same range bins.

3. Determine SNR performance. With *Macrura* employing an adaptive depth behavior, collect modem statistics to a single collaborator buoy (H1) as below:
 - *Macrura* adaptive (SNR), H1 shallow.
 - *Macrura* adaptive (SNR), H1 deep.

Calculate percent of messages received and percent decoded.

4. Determine MAE performance. With *Macrura* employing an adaptive depth behavior, collect modem statistics to a single collaborator buoy (H1) as below:
 - *Macrura* adaptive (MAE), H1 shallow.
 - *Macrura* adaptive (MAE), H1 deep.

Calculate percent of messages received and percent decoded.

5. Determine best settings for p in the given environment. Set an initial b to 0 and τ to 20 ms.

With *Macrura* employing an adaptive depth behavior, collect modem statistics to a single collaborator buoy (H1) as below:

- *Macrura* adaptive (MPP, $p = 1$), H1 shallow.
- *Macrura* adaptive (MPP, $p = 1$), H1 deep.
- *Macrura* adaptive (MPP, $p = 2$), H1 shallow.
- *Macrura* adaptive (MPP, $p = 2$), H1 deep.
- *Macrura* adaptive (MPP, $p = 3$), H1 shallow.
- *Macrura* adaptive (MPP, $p = 3$), H1 deep.
- *Macrura* adaptive (MPP, $p = 4$), H1 shallow.
- *Macrura* adaptive (MPP, $p = 4$), H1 deep.
- *Macrura* adaptive (MPP, $p = 5$), H1 shallow.
- *Macrura* adaptive (MPP, $p = 5$), H1 deep.

Calculate percent of messages received and percent decoded, choose the best result to set p for the desired operational range.

6. Determine best settings for b in the given environment. Set p to the best performing value as determined previously, and hold τ to 20 ms. With *Macrura* employing an adaptive depth behavior, collect modem statistics to a single collaborator buoy (H1) as below:

- *Macrura* adaptive (MPP, $b = 1$), H1 shallow.
- *Macrura* adaptive (MPP, $b = 1$), H1 deep.
- *Macrura* adaptive (MPP, $b = 2$), H1 shallow.
- *Macrura* adaptive (MPP, $b = 2$), H1 deep.
- *Macrura* adaptive (MPP, $b = 3$), H1 shallow.
- *Macrura* adaptive (MPP, $b = 3$), H1 deep.
- *Macrura* adaptive (MPP, $b = 4$), H1 shallow.
- *Macrura* adaptive (MPP, $b = 4$), H1 deep.
- *Macrura* adaptive (MPP, $b = 5$), H1 shallow.
- *Macrura* adaptive (MPP, $b = 5$), H1 deep.

Calculate percent of messages received and percent decoded, choose the best result to set b for the desired operational range.

7. Validate τ . Set p and b to the best performing values as determined previously. With *Macrura* employing an adaptive depth behavior, collect modem statistics to a single collaborator buoy (H1) as below:

- *Macrura* adaptive (MPP, $\tau = 15$ ms), H1 shallow.
- *Macrura* adaptive (MPP, $\tau = 15$ ms), H1 deep.
- *Macrura* adaptive (MPP, $\tau = 25$ ms), H1 shallow.
- *Macrura* adaptive (MPP, $\tau = 25$ ms), H1 deep.

Calculate percent of messages received and percent decoded, choose the best result to set τ for the desired operational range. Repeat the entire procedure as necessary to obtain sufficient message samples to constrain the setting variance.

4 RESULTS

4.1 BASIC METRIC PLOTS & COMPARISON

The following series of figures show the results of the SNR, MPP, and MAE calculation for the single assumed Beaufort Sea SSP used in all Netsim ICEX20 simulations. In all these figures, the plots are repeated for a shallow (30 m) and deep (90 m) transmitter to illustrate the effect above and below the Beaufort Lens. MPP is repeated several times at different configurations to show the effect of varying p , b , and τ .

Figure 4.1 shows the SNR result in dB, calculated using Equation 2.6 implemented in Matlab (Appendix A.1), with a source level of 183 dB re $1 \mu\text{P}$, 1 m and noise level of 40 dB $1 \mu\text{P}$, 1 m. These assumed levels are the same used in the Netsim simulator and during ICEX20. SNR converted to utility with Equation 3.1 is shown in Figure 4.2. The resulting plot is a smooth representation largely driven by the incoherent transmission loss treatment of the arrival data.

Using the same assumed source and noise levels, plots of MPP for varied p of 1, 2, 3 and 10 are shown in Figure 4.3. Only p is varied, with b set to 0 and τ set to 20 ms. Similarly, the same calculation converted to utility is shown in Figure 4.4. As expected, the plot is similar to the SNR representation, with specific areas penalized based on arrival characterization.

A penalty of 10 is shown only for illustrative purposes, but was not configured in the test plan. The extreme penalty shows large areas in both the shallow and deep source where multi-path areas are nearly removed. Further analysis would be required to determine how much of the pattern is due to cell-edge caustics introduced by the ray methods over the specific range and depth grid. However, the resulting plot and the impact the large penalty has on the grid of possibilities is exactly what *Macrura*, another AUV, or a human would employ to make a depth decision, caustics included.

Figure 4.5 shows MPP for varied b of 1, 2, 3 and 10, with p held at 3 and τ to 20 ms. Again, a bounce penalty of 10 was not used in the calculation, but is shown for illustrative purposes. As bounce penalty increases, the bounce paths predictably fade – most notably in the shallow case near the ice sheet. The shallow case also shows a progressively penalized area in the upwardly refracting region near 5 km. In the deep case, the bottom bounce penalty appears to have a less

Modem Parameter (Assumed)	Value
Source Level	183 dB
Noise Level	40 dB
Relative velocities	0 m/s
Bandwidth	4 kHz
Sampling Frequency (Fs)	40 kHz
Symbol rate	18 baud
Feed Forward Taps	21
Feed Back Taps	20
Discrete Fourier transform bins	2048

Table 4.1: MAE Assumptions & Modem Parameters

significant effect, given the predominantly direct path arrival structure. Figure 4.6 shows the utility conversion.

When τ is varied with p held to 3 and b to 0, it does not appear to significantly change within the same order of magnitude. Figures 4.7 and 4.8 show τ set to 15 ms, 20 ms, and 25 ms, with little change. Thus, the change between adaptive depth performance for varied τ will likely be small.

For the MAE plots and the Netsim missions, the assumptions for modem parameters are shown in Table 4.1. The parameters were chosen to match the SNR & MPP scenarios and are an estimate of parameters for a WHOI Micromodem-2. Figure 4.9 shows the result of the calculation implemented in Matlab (Appendix A.2). While the general shape of the contour plot is similar at both source depths to the SNR and MPP plots, the result shows a higher bias toward locations closer to the source location. Additionally, though the MAE results are also in decibels, the result does not fall below 0 as it does in MPP. Another interesting feature is a near-vertical line near approximately 2.8 km in both the shallow and deep plots. This could be a result from a bottom bounce path of the assumed bathymetry.

Figure 4.10 shows the conversion to utility. The first observation of this result is that the metric conversion may not be appropriate to scale MAE in the same way it applies to SNR and MPP. While the metric conversion was tied to the ICEX packet success data, the scale dampens the MAE result, which has a lower range in dB. This implies that the AUV may have a harder time distinguishing utility between bins for low MAE dB. Recalibration of MAE appears to be required, and will be discussed in Section 4.2.3.

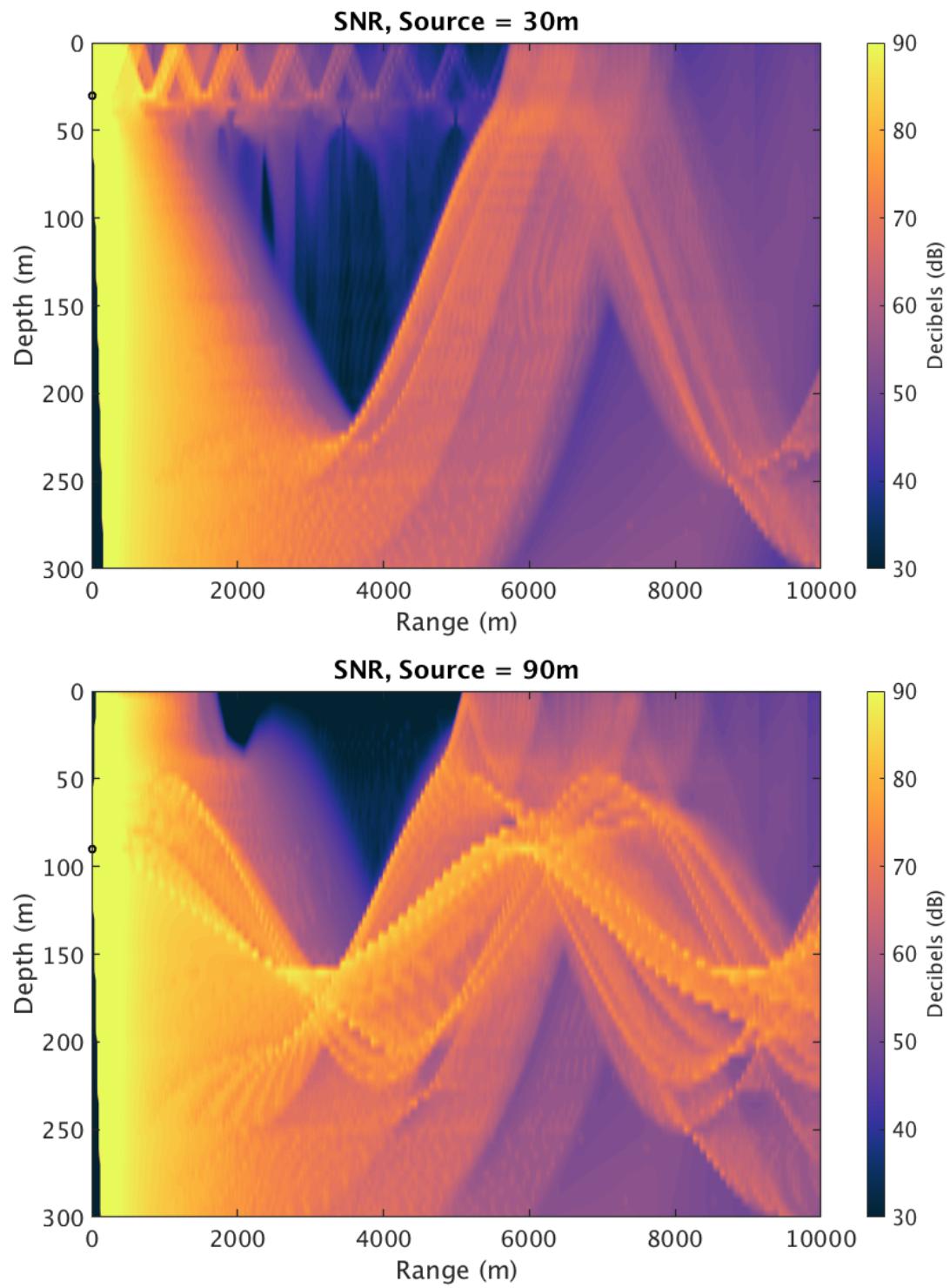


Figure 4.1: Signal-to-Noise Ratio (in dB) for various depths and ranges using arrivals modelled in BELL-HOP for a 10 kHz source based on a CTD cast collected during ICEX20 (9 MAR 2012 02 AKST, CTD-2).

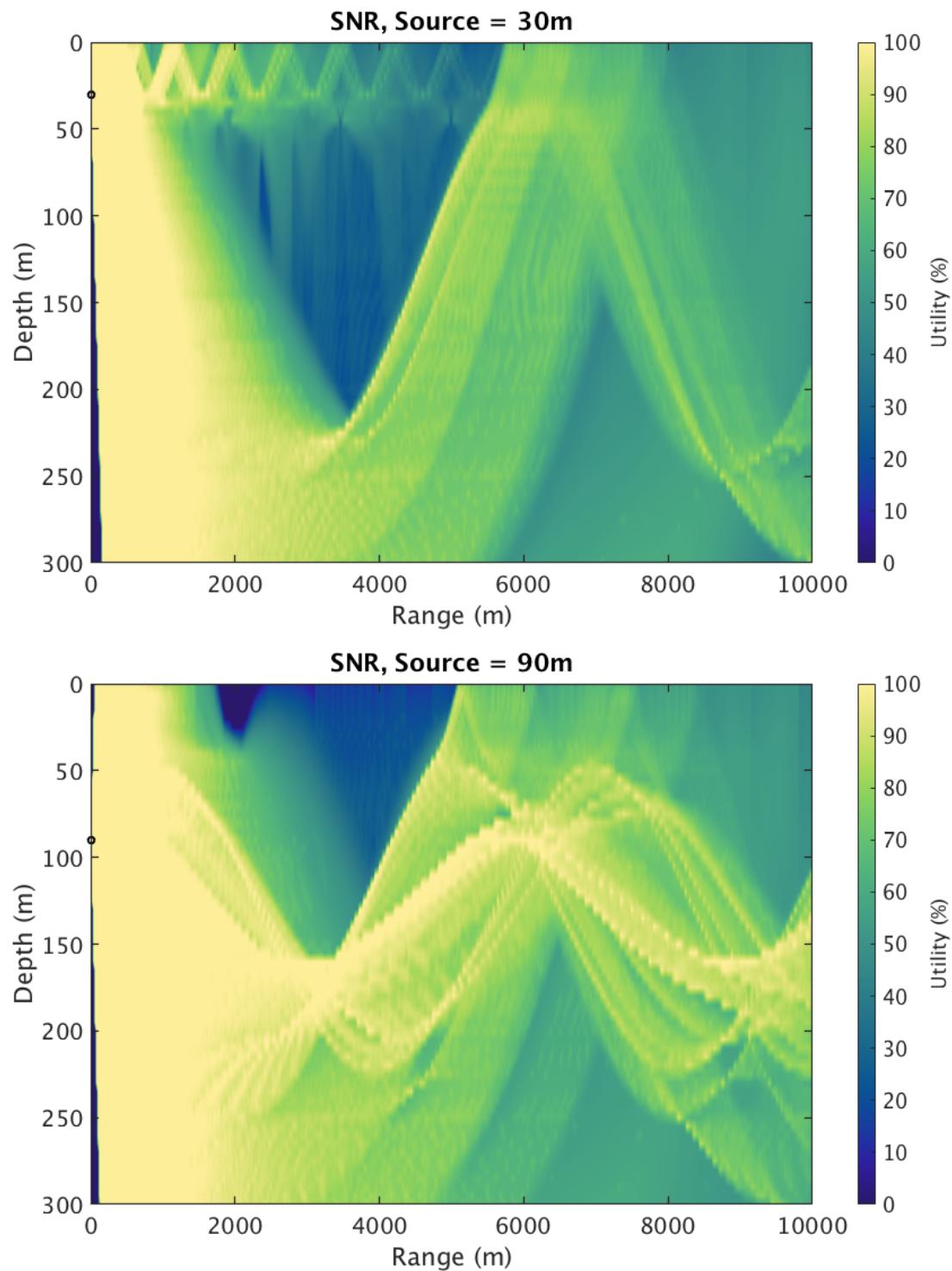


Figure 4.2: Utility converted Signal-to-Noise Ratio (in % utility) for various depths and ranges using arrivals modelled in BELLHOP for a 10 kHz source based on a CTD cast collected during ICEX20 (9 MAR 20 1202 AKST, CTD-2).

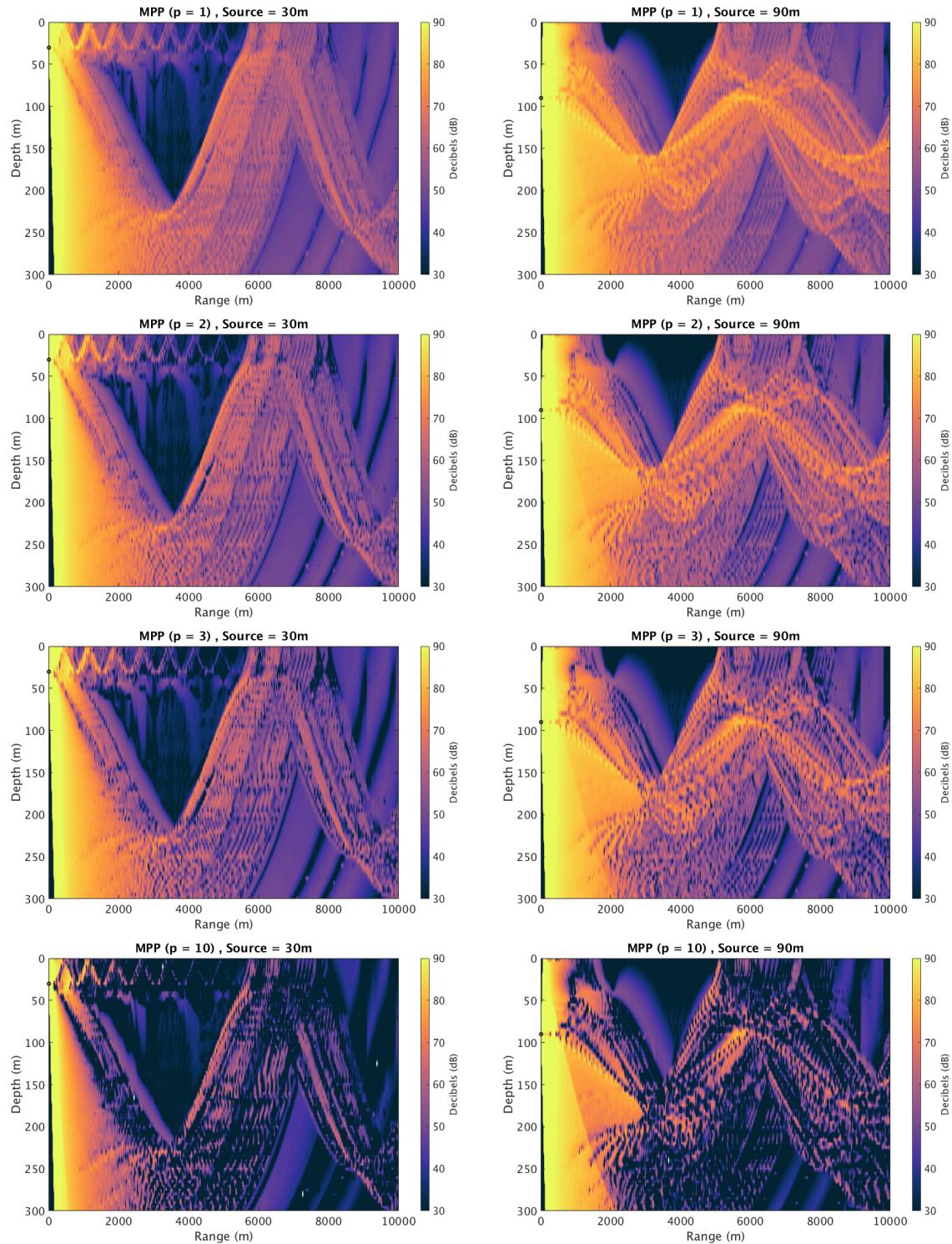


Figure 4.3: Multi-path Penalty (various p) (in dB) for various depths and ranges using arrivals modelled in BELLHOP for a 10 kHz source based on a CTD cast collected during ICEX20 (9 MAR 2012 02 AKST, CTD-2).

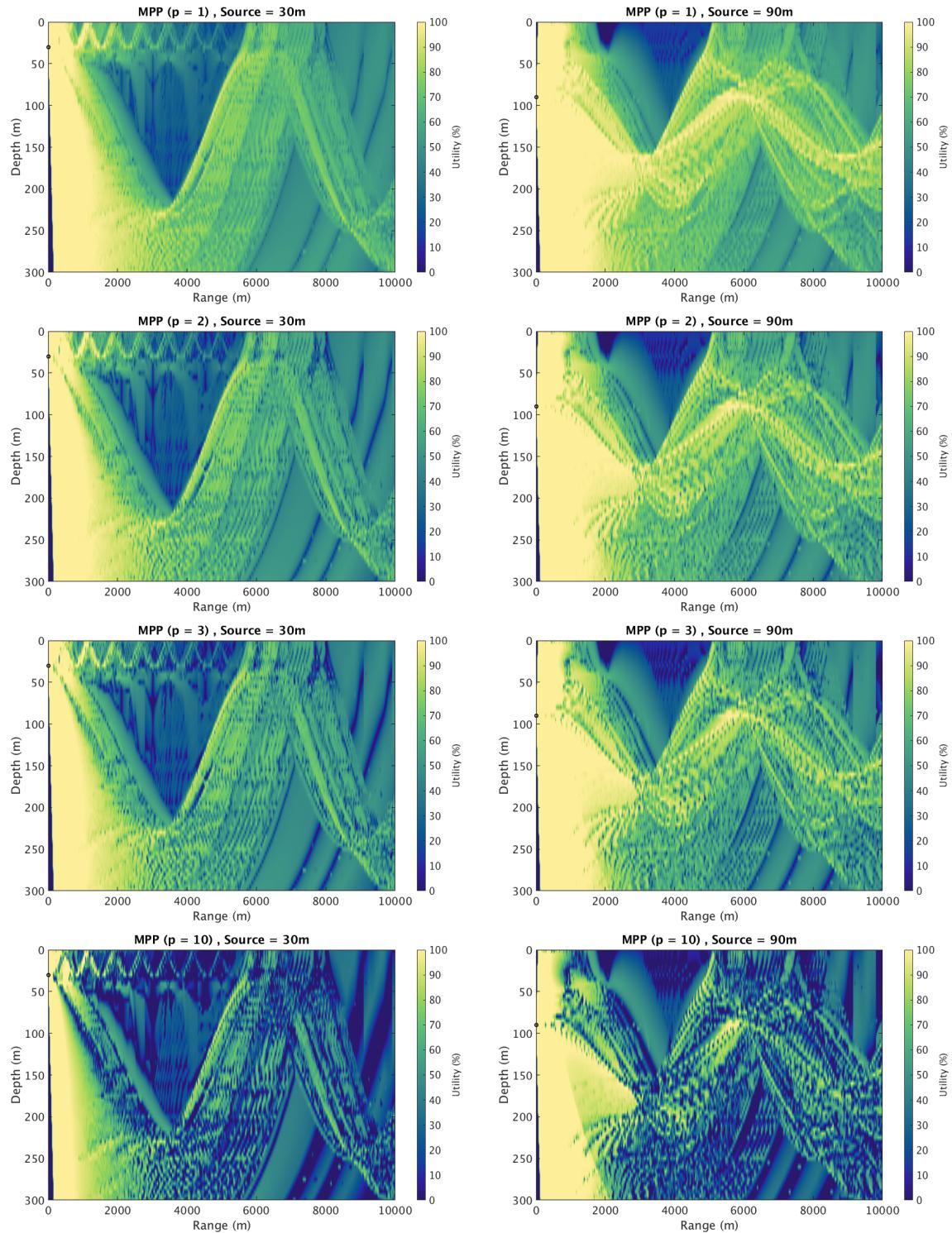


Figure 4.4: Utility converted Multi-path Penalty (various p) (in % utility) for various depths and ranges using arrivals modelled in BELLHOP for a 10 kHz source based on a CTD cast collected during ICEX20 (9 MAR 2012 02 AKST, CTD-2).

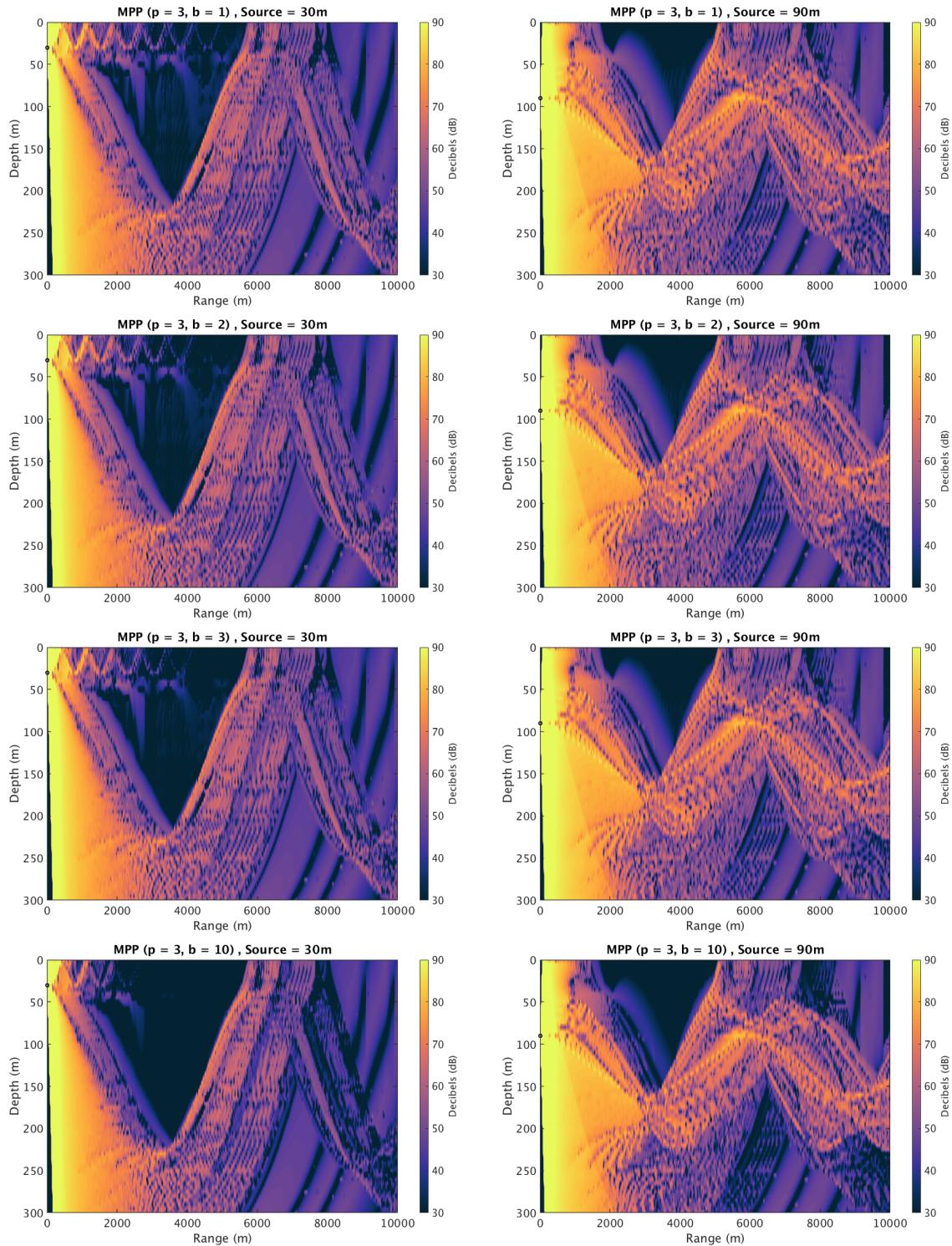


Figure 4.5: Multi-path Penalty (various b) (in dB) for various depths and ranges using arrivals modelled in BELLHOP for a 10 kHz source based on a CTD cast collected during ICEX20 (9 MAR 2012 02 AKST, CTD-2).

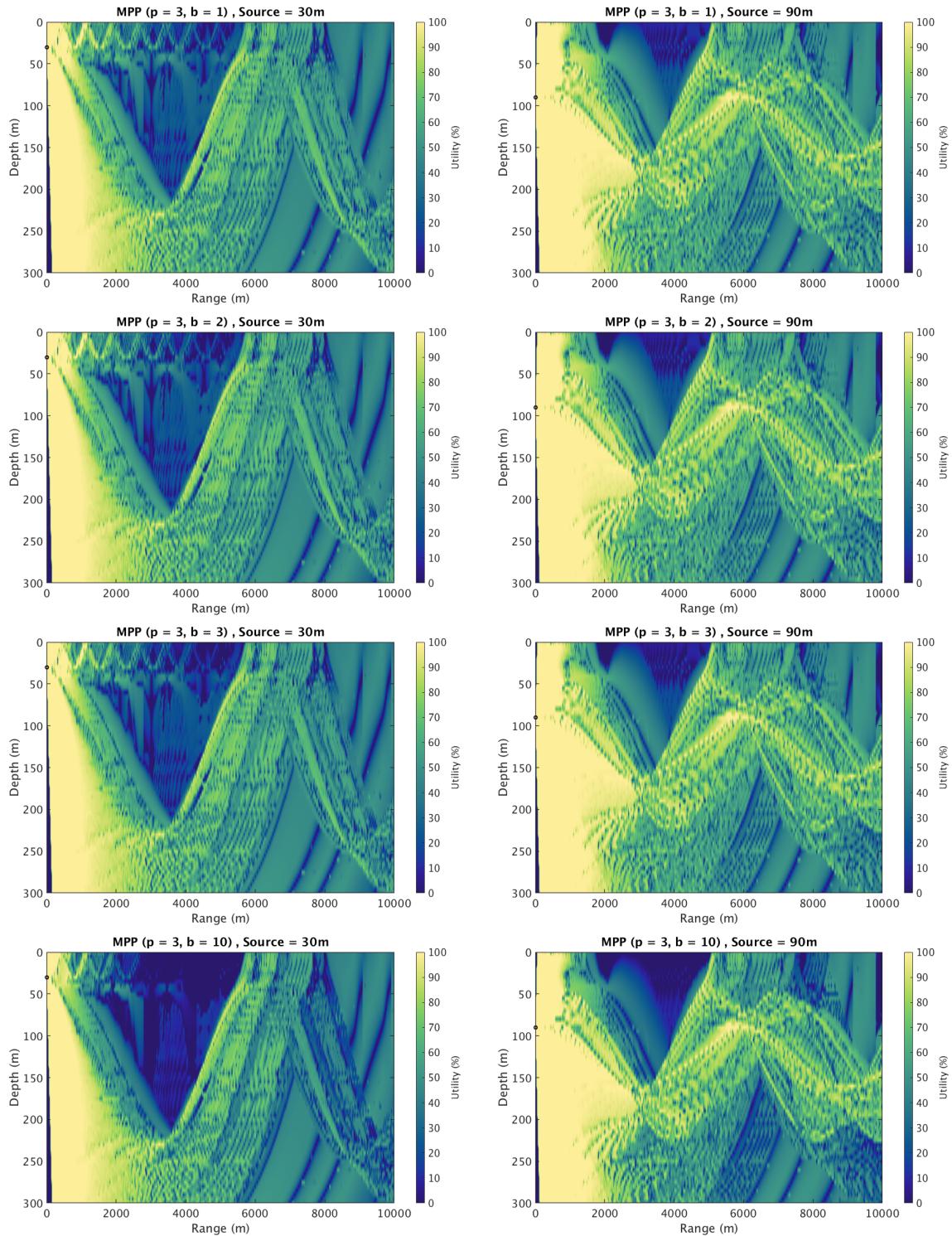


Figure 4.6: Utility converted Multi-path Penalty (various b) (in % utility) for various depths and ranges using arrivals modelled in BELLHOP for a 10 kHz source based on a CTD cast collected during ICEX20 (9 MAR 2012 02 AKST, CTD-2).

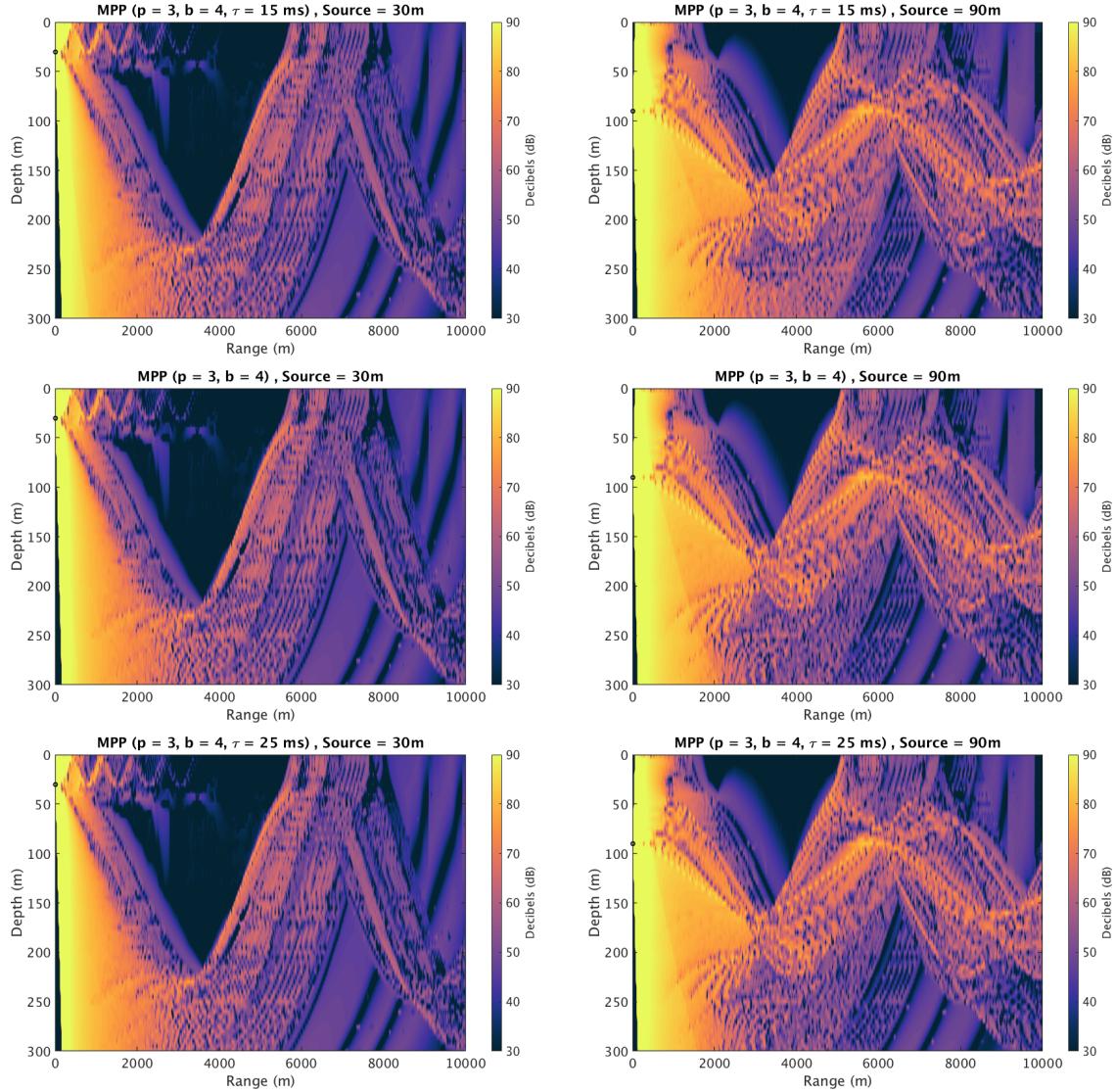


Figure 4.7: Multi-path Penalty (various τ) (in dB) for various depths and ranges using arrivals modelled in BELLHOP for a 10 kHz source based on a CTD cast collected during ICEX20 (9 MAR 2012 02 AKST, CTD-2). The middle row of plots represents $\tau = 20$ ms, as do all others in the analysis unless specifically denoted otherwise.

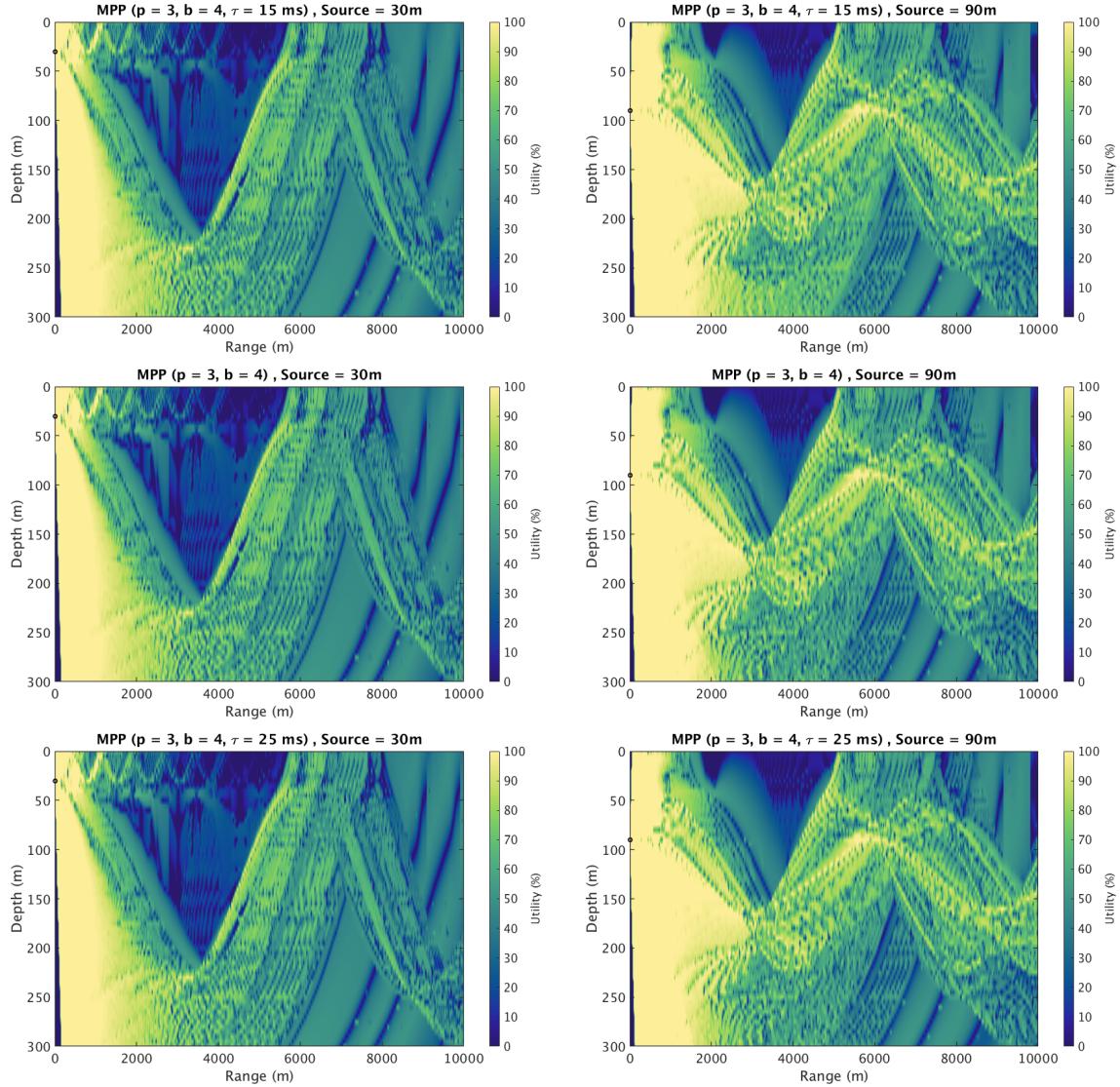


Figure 4.8: Utility converted Multi-path Penalty (various τ) (in % utility) for various depths and ranges using arrivals modelled in BELLHOP for a 10 kHz source based on a CTD cast collected during ICEX20 (9 MAR 2012 02 AKST, CTD-2). The middle row of plots represents $\tau = 20$ ms, as do all others in the analysis unless specifically denoted otherwise.

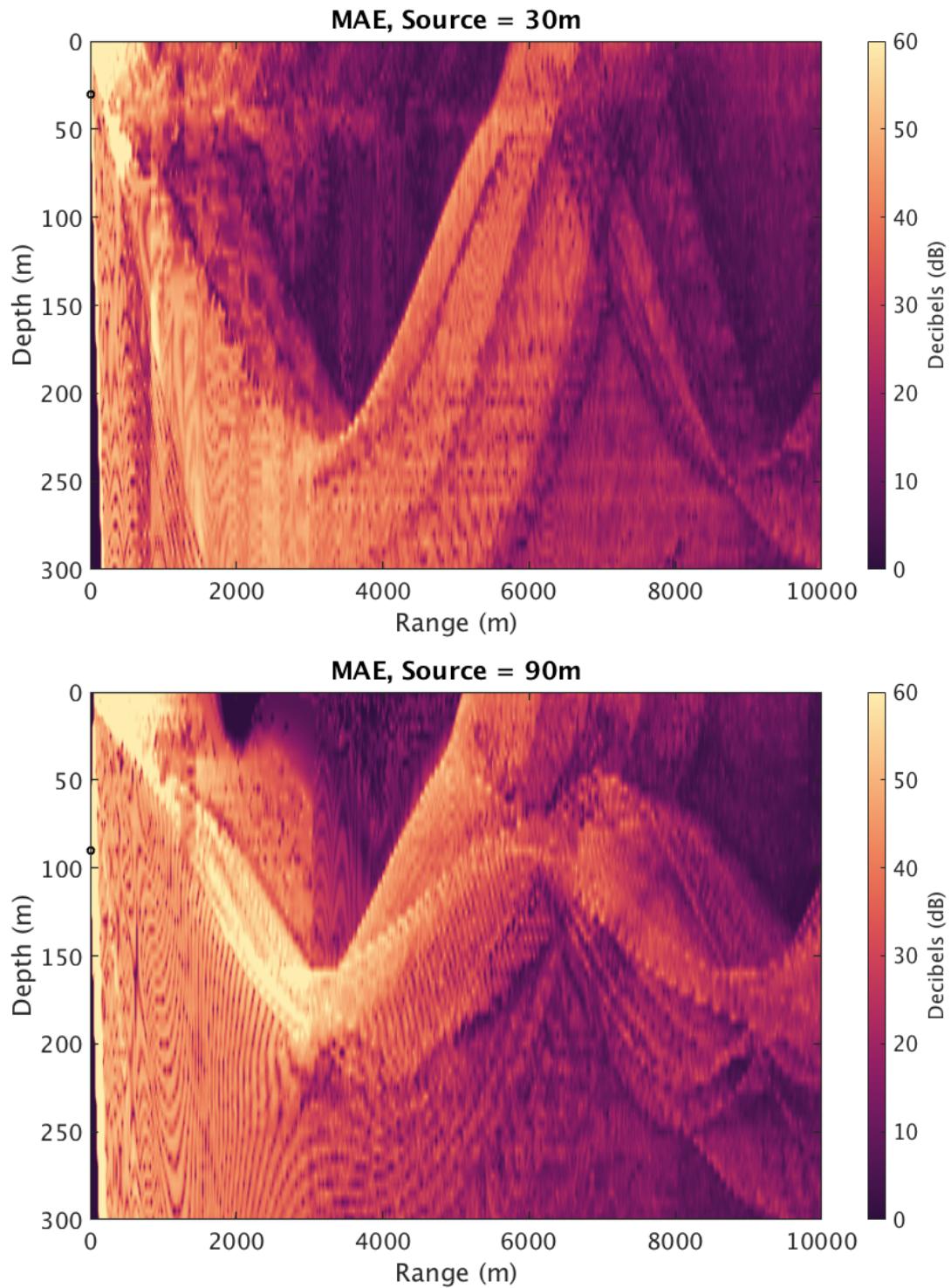


Figure 4.9: Minimum Achievable Error (in dB) for various depths and ranges using arrivals modelled in BELLHOP for a 10 kHz source based on a CTD cast collected during ICEX20 (9 MAR 2012 02 AKST, CTD-2). MAE was calculated at each receiver point assuming a stationary source and receiver (i.e. zero Doppler case)

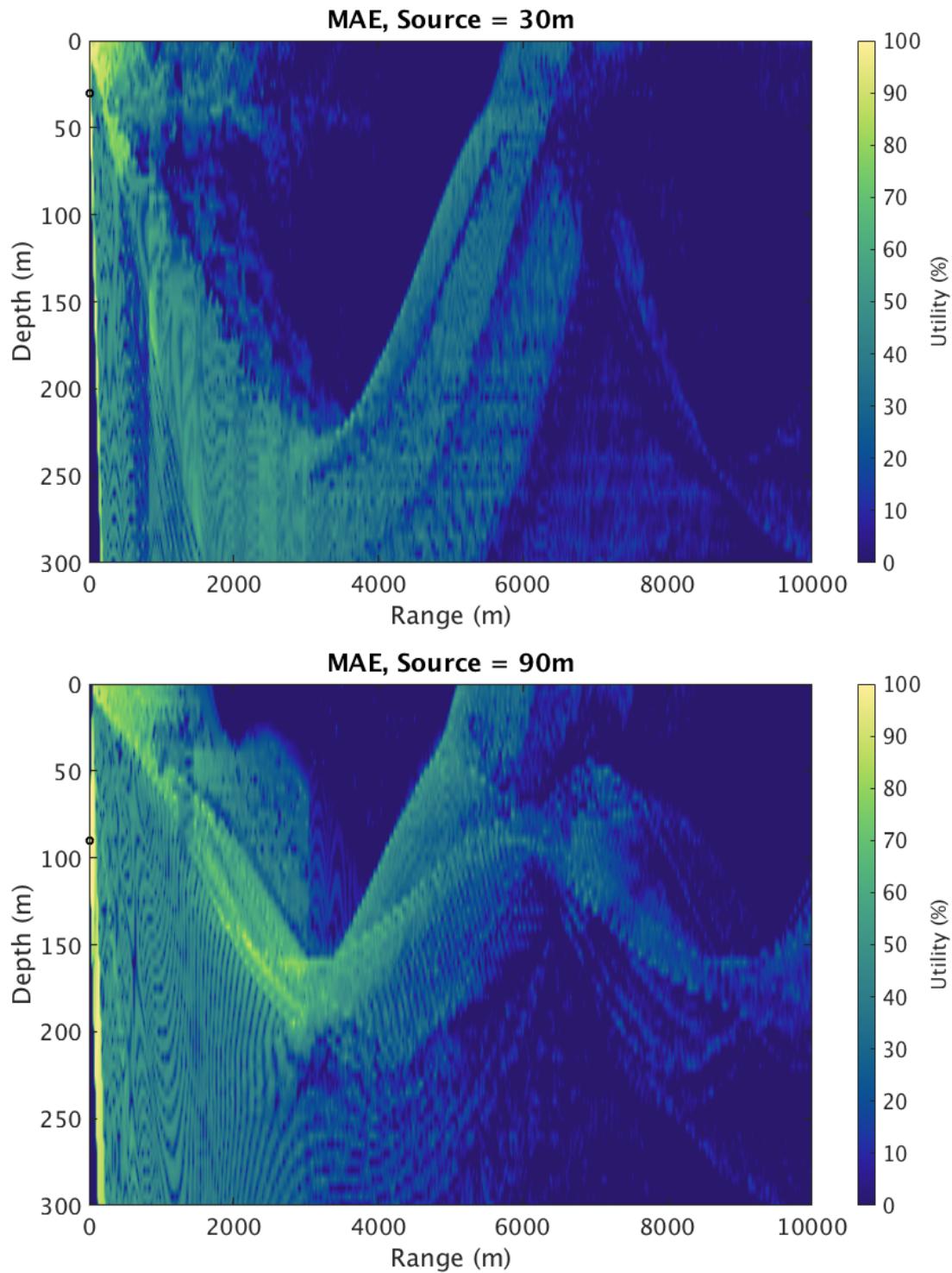


Figure 4.10: Utility converted Minimum Achievable Error (in % utility) for various depths and ranges using arrivals modelled in BELLHOP for a 10 kHz source based on a CTD cast collected during ICEX20 (9 MAR 2012 02 AKST, CTD-2). MAE was calculated at each receiver point assuming a stationary source and receiver (i.e. zero Doppler case)

4.2 SINGLE DEPTH SOURCE & RECEIVER ANALYSIS

4.2.1 METRIC PLOT DATA AT SINGLE DEPTHS

Using the data from the previous plots, a horizontal slice at either the shallow or deep receiver range is taken and plotted relative to the other metrics. Figure 4.11 shows SNR and MPP with varied p on the left axis, and MAE on the right for shallow and deep combinations of the source (transmitter - tx) and receiver (rx). SNR is plotted as the solid black line at the very top of all the MPP lines, which overlap on SNR at certain ranges where no penalty is applicable. The curve for SNR on all the plots is substantially more flat than any of the MPP curves. Even though MAE is plotted on a smaller axis, SNR is still smoother in general, which is supported by Figures 4.1 and 4.2 shown previously.

In all the plots, the additive behavior of penalty for MPP on certain ranges where multi-path is present is easier to see than in the full-field receiver grid of the previous Figures. In the first plot (shallow to shallow) the influence of the many top bounce paths present in the upper layer near the ice are apparent, and the return of rays from below the layer near 6 km is also visible. The second plot (shallow to deep) shows even the near-range effect of MPP, which becomes apparent within 1 km, and the earlier arrival (in-range) of the returning below-layer paths from the source at around 5 km. The shadow zone for this Beaufort Sea environment is well observed in the third and fourth plots from 2 - 3 km and then 3 - 4 km, respectively.

Figure 4.11 also supports the idea that MPP more closely matches MAE than SNR. On all the plots, even a modest penalty value appears to modify the line to better trend with MAE, particularly at multi-path rich regions. For example, on the third plot (deep to shallow) the region between 3 and 5 km is modelled as a flat, uniform line in SNR, but MAE and MPP capture the more complex environmental problem.

As a visual for the ‘flooring’ effect of the metric utility conversion, Figure 4.12 shows the same data as Figure 4.11. Since the previous section noted that the metric utility conversion would not be an appropriate choice for MAE, the conversion is shown only for MPP. All four plots suggest that none of the depicted p values will return a completely flat result when employed in the autonomy suite, giving room to evaluate p in the test procedure to find an ideal value.

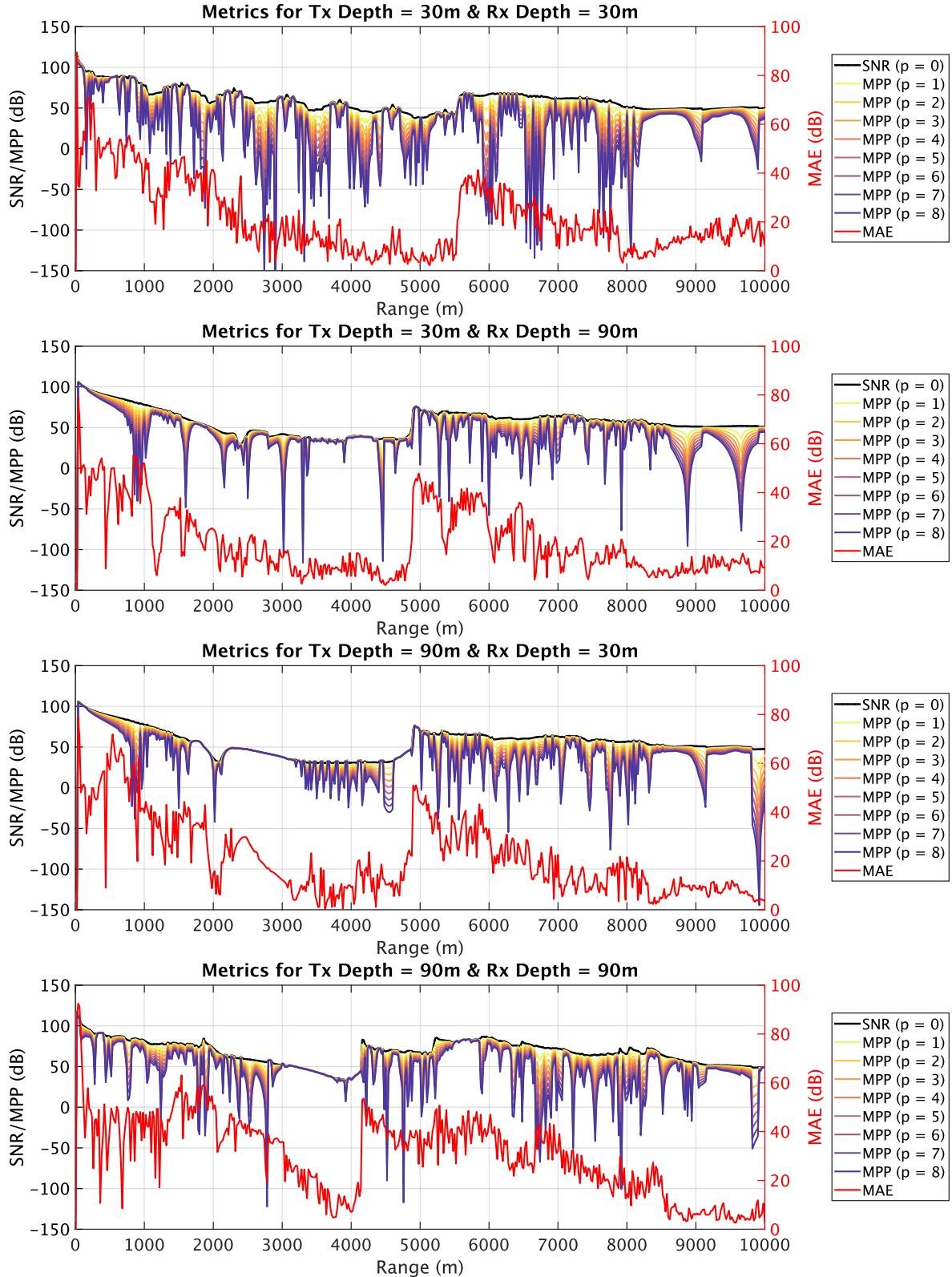


Figure 4.11: SNR, MPP (Various p), and MAE for Single Source & Receiver Depths

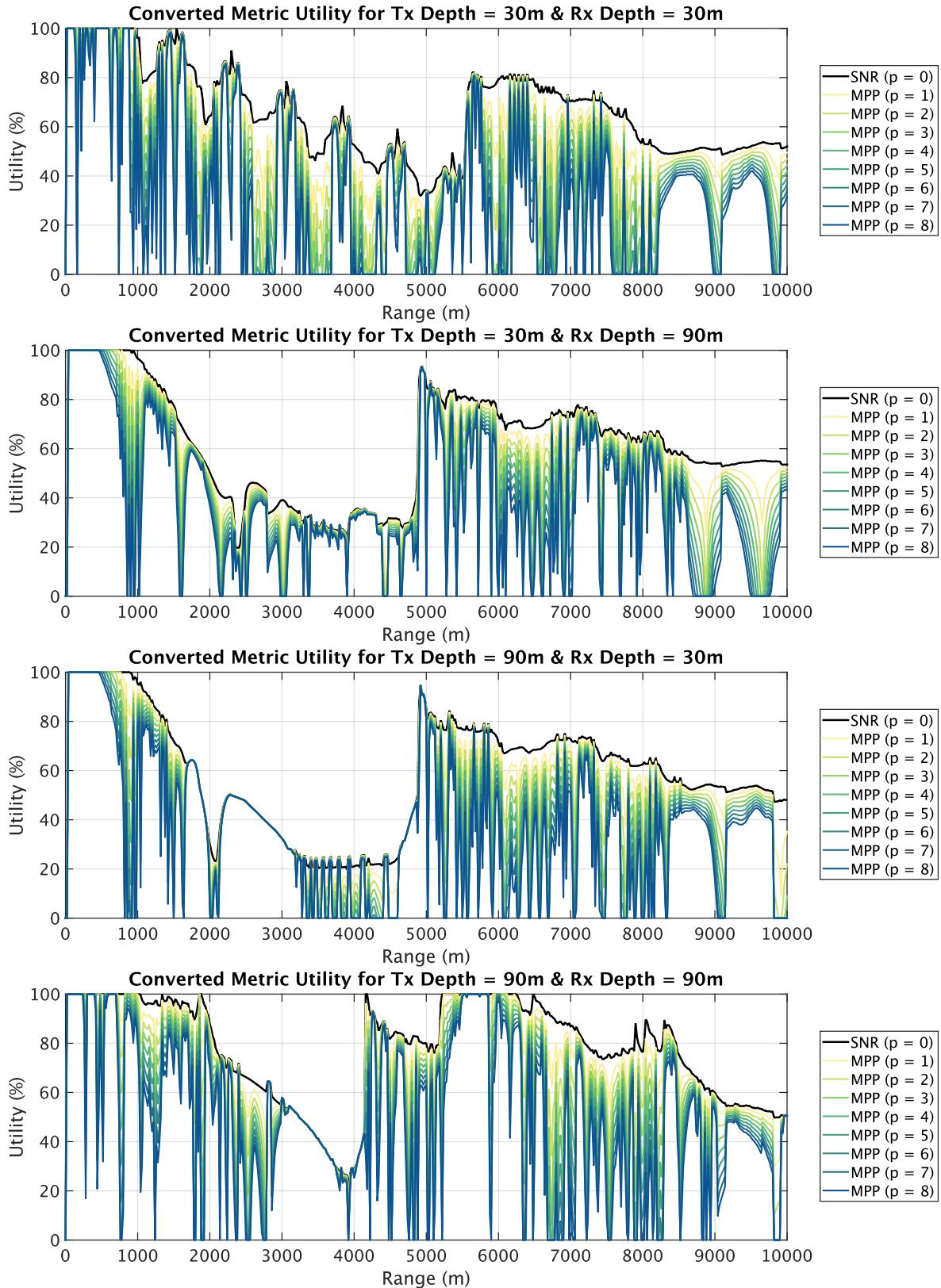


Figure 4.12: SNR & MPP Converted to Utility for Single Source & Receiver Depths. This conversion to a percent utility is the normalization used by *Macrura's* adaptive depth behavior.

4.2.2 SINGLE DEPTH NETSIM PACKET SUCCESS

Netsim is now employed with *Macrura* at a single depth to obtain simulated packet success statistics with range. Figure 4.13 shows the packet success at the receiver for either the shallow or deep location with range. The left column of plots represents transmissions from *Macrura* held at 30 m, and the right column at 90 m. The first row closely mirrors the result of T. Schneider (2020)¹ in format, with packet success in red and predicted SNR in dashed blue for the two receiver depths. The second row has the same packet success data, but has MPP with varied p ($b = 0, \tau = 20$ ms) instead of SNR. The third row shows the MAE comparison, and the fourth shows the modem-reported SNR-in statistics, which are closely correlated to packet success as expected.

The most interesting result of this plot is that for the shallow-shallow case (and to a lesser extent, the shallow-deep case), all metrics under-predict packet success. Both the Virtual Ocean software used by Netsim and the metric calculations utilize the same BELLHOP output file and settings to model the overhead ice layer, which suggests this is unexpected. However, referring to Figure 4.12, the utility conversion appears to adequately cover this discrepancy with a much flatter curve in general for the 2 - 6 km range in that case. Re-plotting packet success with MPP converted to utility, Figure 4.14 indeed shows better approximation at the 5 km reference point. This supports the implementation of the conversion metric as derived from the real ICEX20 data for MPP, since packet success in this case is not obtained from a depth decision.

¹Schneider, Schmidt, and Randeni, “Self-Adapting Under-Ice Integrated Communications and Navigation Network”.

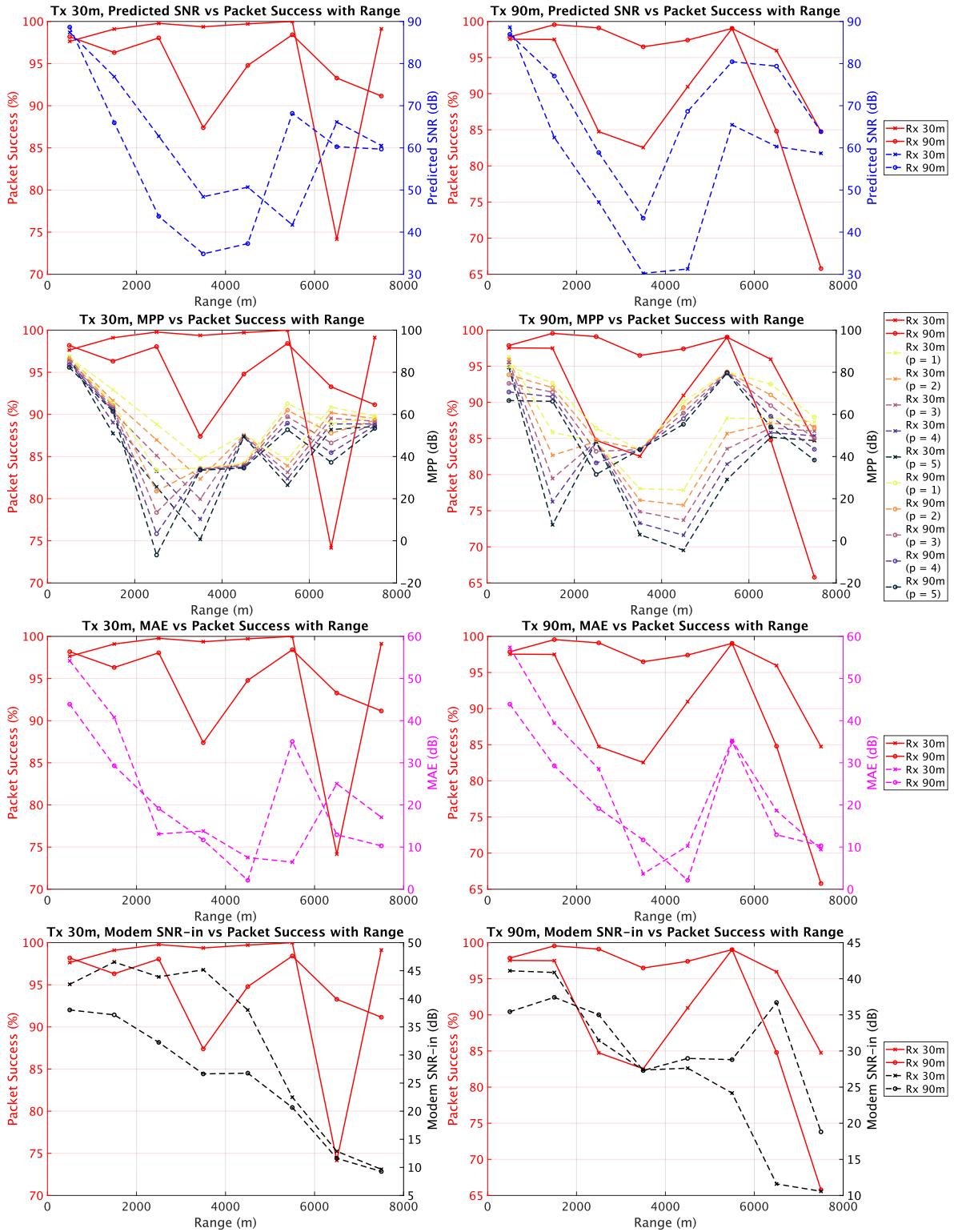


Figure 4.13: NETSIM Packet Success compared for various metrics. Data generated from single depth lawnmower runs between *Macrura* and buoy H1.

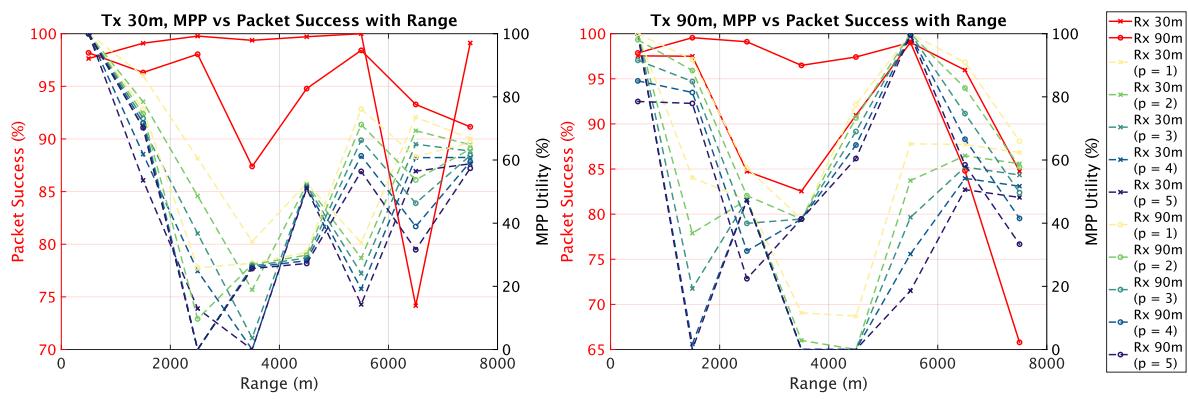


Figure 4.14: NETSIM Packet Success vs. MPP (Various p) Utility, calculated using the utility conversion based on data from ICEX20.

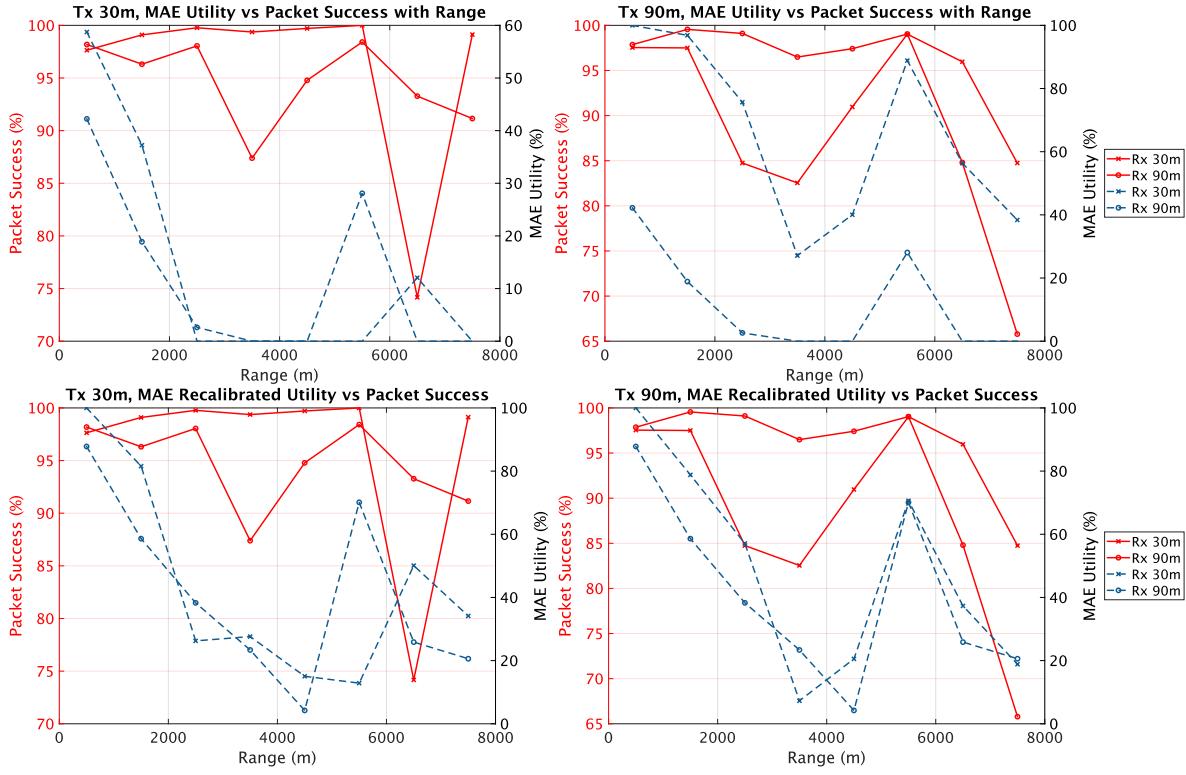


Figure 4.15: NETSIM Packet Success vs. MAE Utility. The first plot shows MAE calculated using the initial utility conversion, while the second plot shows a recalibrated utility conversion derived from NETSIM packet success data.

4.2.3 MAE RECALIBRATION

MAE, however, does not appropriately scale to observed packet success when converted to utility, made very apparent in the first plot of Figure 4.15. As expected, MAE is poorly correlated in the mid-range in particular, and inappropriately floors to 0 at several points. The conversion for MAE should, therefore, be modified to more closely match the packet success statistics obtained by Netsim.

$$\overline{\text{MAE}} = \min(2 |\text{MAE}|, 100) \quad (4.1)$$

Equation 4.1 is now implemented to properly scale MAE. Since MAE does not produce negative values and the expression takes the absolute value of MAE, the minimum condition is dropped. In order to preserve 0 dB MAE as 0% utility, an additive term is also omitted. Observing in Figure 4.13 that when MAE is 50 dB, packet success is near 100%, this value is used as the maximum utility. Figure 4.16 now shows the improved version of Figure 4.10.

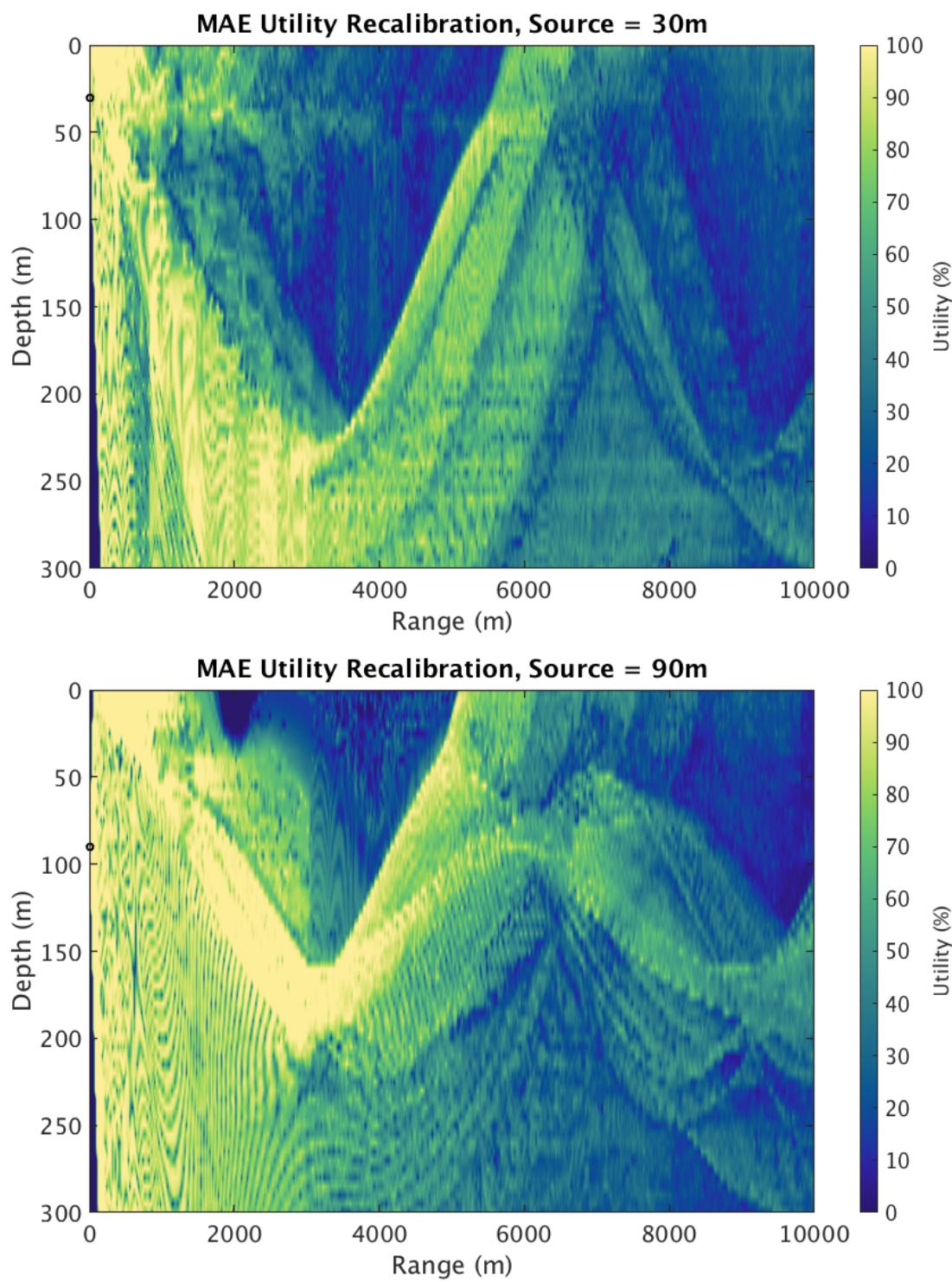


Figure 4.16: Utility converted Minimum Achievable Error (in % utility) using recalibrated conversion for various depths and ranges using arrivals modelled in BELLHOP for a 10 kHz source based on a CTD cast collected during ICEX20 (9 MAR 2012 02 AKST, CTD-2). MAE was calculated at each receiver point assuming a stationary source and receiver (i.e. zero Doppler case).

4.3 DETERMINING p

After analysis of the metric utility conversion, the settings for MPP to best reflect packet success and overall communication performance are desired. First, predicted MPP is plotted against MAE in an attempt to observe a correlation, if any. Then the results of Netsim missions with varied configurations of p are performed with an adaptive depth behavior.

4.3.1 EFFECT OF VARIED MPP vs. MAE

As previously discussed, MAE provides a high fidelity metric taking into account actual modem performance as well as the environmental impulse response. Ideally, a trend between MPP and MAE would help to make an informed choice of MPP settings in the given environment. Figure 4.17 and Figure 4.18 show SNR plotted versus MAE for all receiver points of the calculation performed at 30 m and 90 m, respectively. Similarly, Figure 4.19 and Figure 4.20 show the same, but for MPP with p of 1, 2, and 3. In all four figures, the left-most plot is a receiver point histogram, while the right-most is the same data but plotted sequentially from nearest receiver range to farthest.

When p is increased, it means that higher multi-path receiver points have a higher penalty applied; the first observation is that this tends to spread out the absolute range of values within the MPP set over MAE. Both the histogram and the by-range plots show this effect along the increasingly negative y-axis where MPP is plotted. Interestingly, three horizontal hot spots of histogram data (for MPP \approx 35, 50, and 60 dB) in the shallow source plots do not appear to significantly move with increased p . A similar hot spot in the deep source case near 50 dB MPP is also largely unmoved. Based on the by-range plots, these hot spots appear to be influenced by receiver points in the far range.

The by-range plots do suggest that MAE and MPP are loosely correlated in range, but no major shape change between the varied p suggests one would be better than another when considering all the points. Generally, the data shows a relationship with interesting edge effects, with a large spread of MAE and MPP values making up the main body of the cluster. At high MAE and MPP, there appears to be a more linear trend in the receiver points, which are almost entirely in the near range.

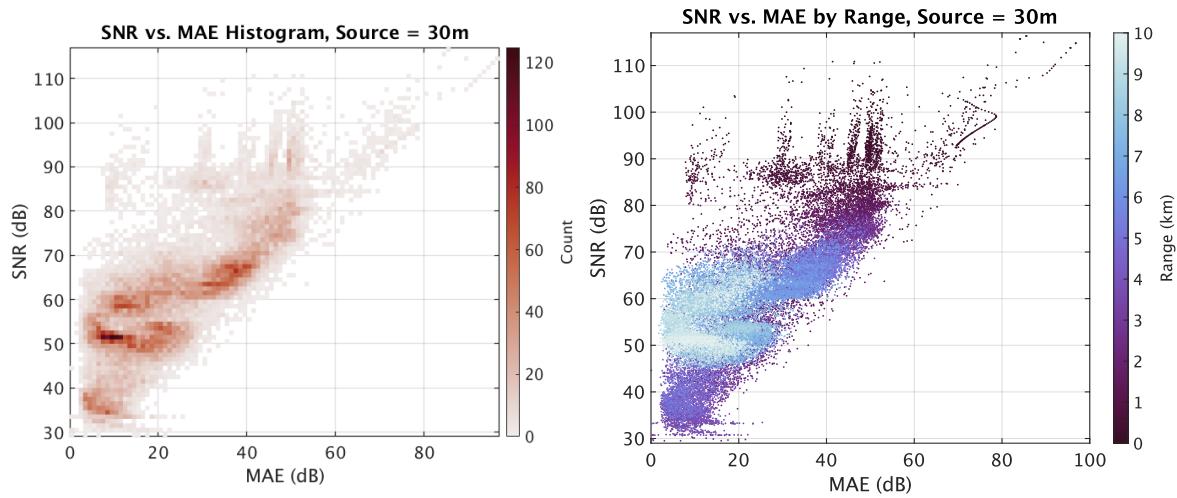


Figure 4.17: SNR vs. MAE Histogram & Range Plots for a 10 kHz source at 30 m depth. Each count or point represents a single receiver (depth, range) for the given environment. On the histograms, entries for which MAE = 0 for all ranges were excluded for display purposes, but were generally higher than the maximum range for counts. For the range plot, the points are colored by the receiver range (i.e. multiple points may have the same color).

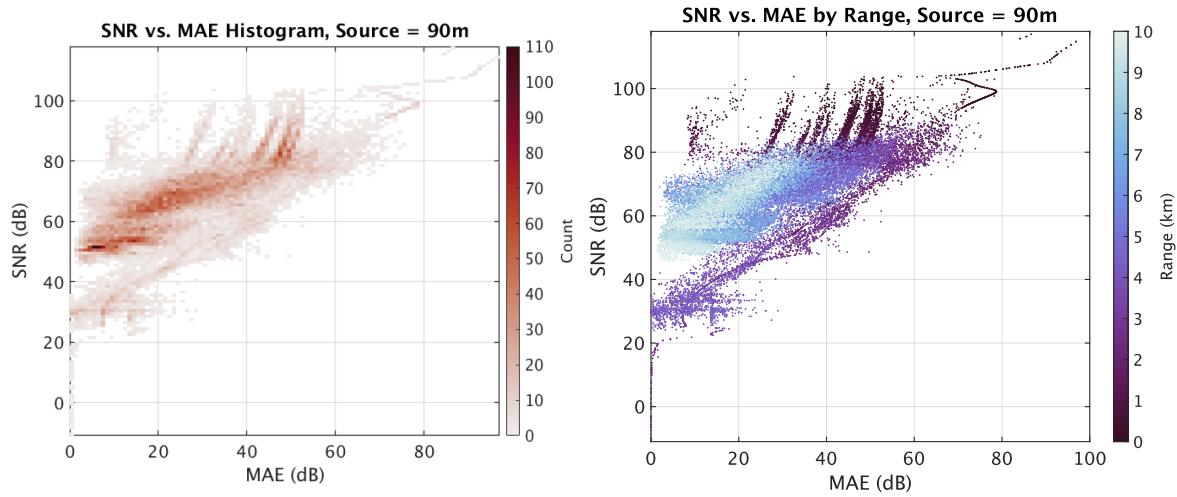


Figure 4.18: SNR vs. MAE Histogram & Range Plots for a 10 kHz source at 90 m depth. Each count or point represents a single receiver (depth, range) for the given environment. On the histograms, entries for which MAE = 0 for all ranges were excluded for display purposes, but were generally higher than the maximum range for counts. For the range plot, the points are colored by the receiver range (i.e. multiple points may have the same color).

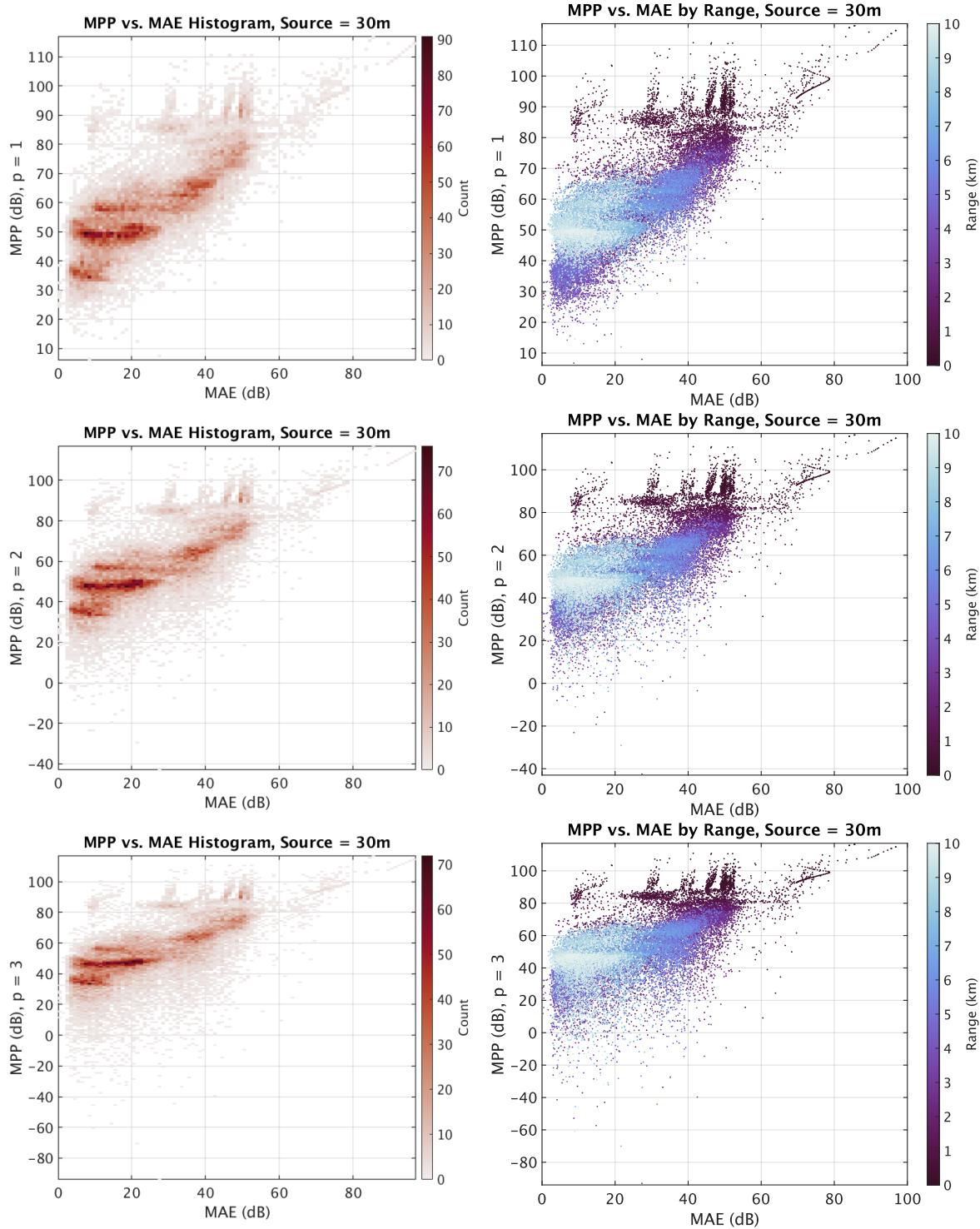


Figure 4.19: MPP vs. MAE Histogram & Range Plots for a 10 kHz source at 30 m depth and p set to 1, 2, & 3. Each count or point represents a single receiver (depth, range) for the given environment. On the histograms, entries for which MAE = 0 for all ranges were excluded for display purposes, but were generally higher than the maximum range for counts. For the range plot, the points are colored by the receiver range (i.e. multiple points may have the same color).

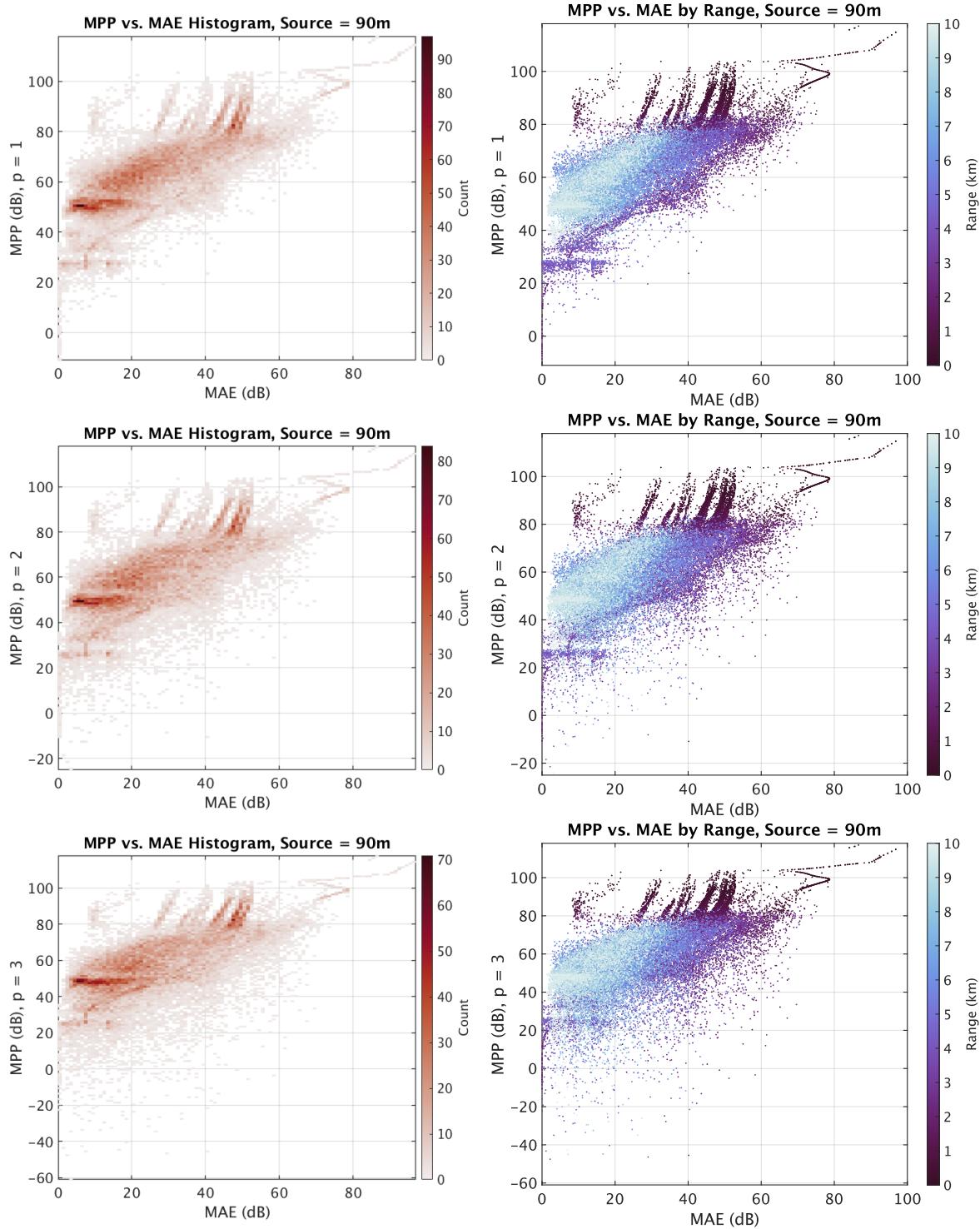


Figure 4.20: MPP vs. MAE Histogram & Range Plots for a 10 kHz source at 90 m depth and p set to 1, 2, & 3. Each count or point represents a single receiver (depth, range) for the given environment. On the histograms, entries for which $\text{MAE} = 0$ for all ranges were excluded for display purposes, but were generally higher than the maximum range for counts. For the range plot, the points are colored by the receiver range (i.e. multiple points may have the same color).

4.3.2 DEPTH ADAPTIVE AUTONOMY WITH VARIED PENALTY, p

The goal of this part of the experimental procedure was to vary the metric configuration used by the vehicle in its adaptive depth behavior, observe whether or not communication performance was improved, and subsequently determine which value of p was best. For each scenario, two full simulations of approximately 335 messages each were performed. As described in the test plan description, b was set to 0 and τ to 20 ms.

Table 4.2 constitutes the main results. The trend is plotted in Figure 4.21 for the shallow receiver case, and Figure 4.22 for the deep receiver case. A 95% Clopper-Pearson confidence interval is reported for each scenario, where messages vary in range from 0 - 8 km and are considered statistically independent. Sources of error in this data include differences in simulation launch time, internet connection variation (from localhost to Netsim hardware), hardware noise effects, and PPS timing drift (if any).

In the shallow receiver case, SNR performed the best out of any metric, in both percent received and percent decoded. MAE appeared to achieve the median of the other metrics for percent received, but performed more poorly in terms of percent decoded. However, MAE was executed with the *previous* metric utility conversion, and is therefore expected to produce suboptimal results. The best MPP results were for $p = 3$, though the overall trend in performance was downward from the SNR baseline.

However, in the deep receiver case, MPP did perform better than SNR at all values of p for percent decoded, with 3 producing the best result. MAE performed the best out of all the metrics (despite the earlier metric utility conversion method) in percent decoded, with SNR performing significantly worse. Percent received for all metrics was nearly the same, and uniformly high.

One reason that p of 3 was the best result may have to do with the original ICEX20 dataset that informed Equation 3.1. That empirical conversion was designed to calibrate SNR and MPP (with p at 3) to the modem packet statistics with range collected during the course of the ICEX20 experiment. Since p of 3 was a best engineering estimate at the time, that decision may implicitly affect the results in this case. However, the scaling effect in the utility conversion was shown in Section 4.2.1 to vary highly with changing p , so even if a bias was introduced in the calibration, the results, at the very least, validate MPP performance as compared to SNR and MAE.

Since p of 3 tended to perform the best in both scenarios, this value is set as the multi-path penalty for the subsequent sections. The caveat that it did not perform better than SNR in the shallow case remains.

<i>p</i> Variation						
H1-Shallow (30 m)						
Metric	<i>Macrura</i> Sent	H1 Received		H1 Decoded		
		No. (%)	CI* (%)	No. (%)	CI* (%)	
SNR	674	629 (93.3)	91.2 - 95.1	611 (90.6)	88.2 - 92.7	
MPP ($p = 1$)	673	599 (89.0)	86.4 - 91.3	576 (85.6)	82.7 - 88.2	
MPP ($p = 2$)	673	594 (88.3)	85.6 - 90.6	568 (84.4)	81.4 - 87.1	
MPP ($p = 3$)	674	612 (90.8)	88.4 - 92.9	587 (87.1)	84.3 - 89.5	
MPP ($p = 4$)	673	598 (88.9)	86.2 - 91.1	563 (83.7)	80.6 - 86.4	
MPP ($p = 5$)	675	580 (85.9)	83.1 - 88.5	545 (80.7)	77.6 - 83.7	
MAE	677	602 (88.9)	86.3 - 91.2	563 (83.2)	80.1 - 85.9	
H1-Deep (90 m)						
Metric	<i>Macrura</i> Sent	H1 Received		H1 Decoded		
		No. (%)	CI* (%)	No. (%)	CI* (%)	
SNR	676	671 (99.3)	98.3 - 99.8	578 (85.5)	82.6 - 88.1	
MPP ($p = 1$)	675	668 (99.0)	97.9 - 99.6	603 (89.3)	86.8 - 91.6	
MPP ($p = 2$)	664	662 (99.7)	98.9 - 99.9	598 (90.1)	87.5 - 92.2	
MPP ($p = 3$)	671	667 (99.4)	98.5 - 99.8	612 (91.2)	88.8 - 93.2	
MPP ($p = 4$)	674	667 (99.0)	97.8 - 99.5	610 (90.1)	88.0 - 92.6	
MPP ($p = 5$)	674	669 (99.3)	98.3 - 99.8	608 (90.2)	87.7 - 92.3	
MAE	677	670 (99.0)	97.9 - 99.6	629 (92.9)	90.7 - 94.7	

Table 4.2: NETSIM Adaptive Depth Simulation Results for a given mission set with p varied between runs.

* 95% Clopper-Pearson confidence interval.

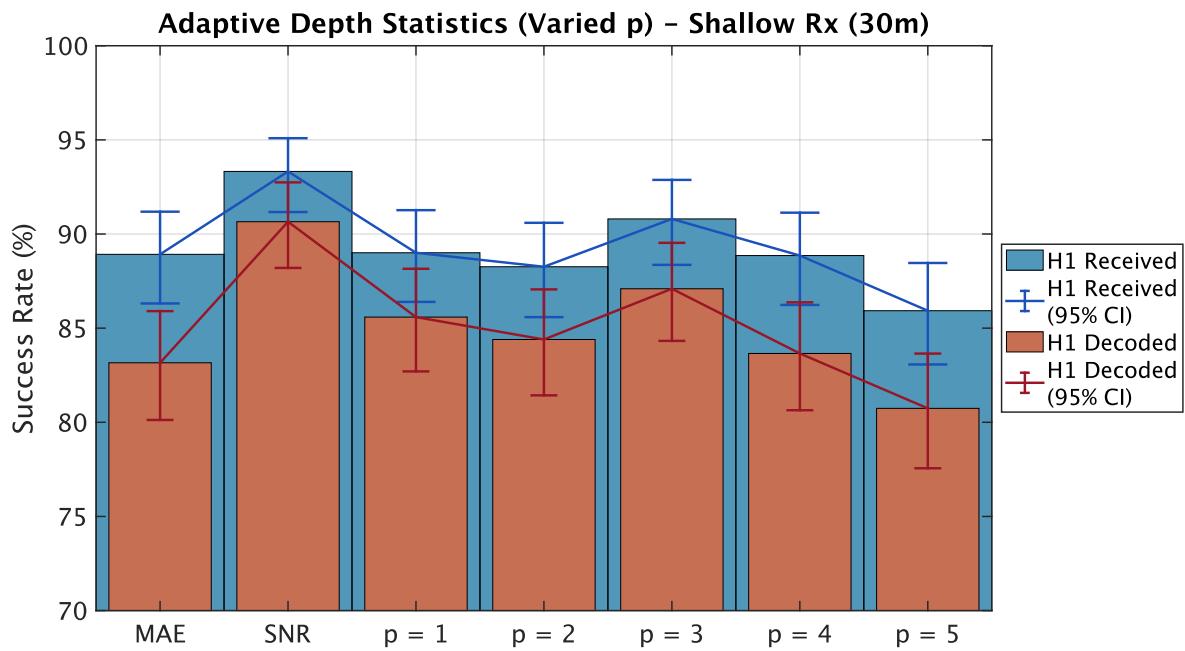


Figure 4.21: Adaptive Depth Simulation Results for Varied p , Shallow Receiver (30 m)

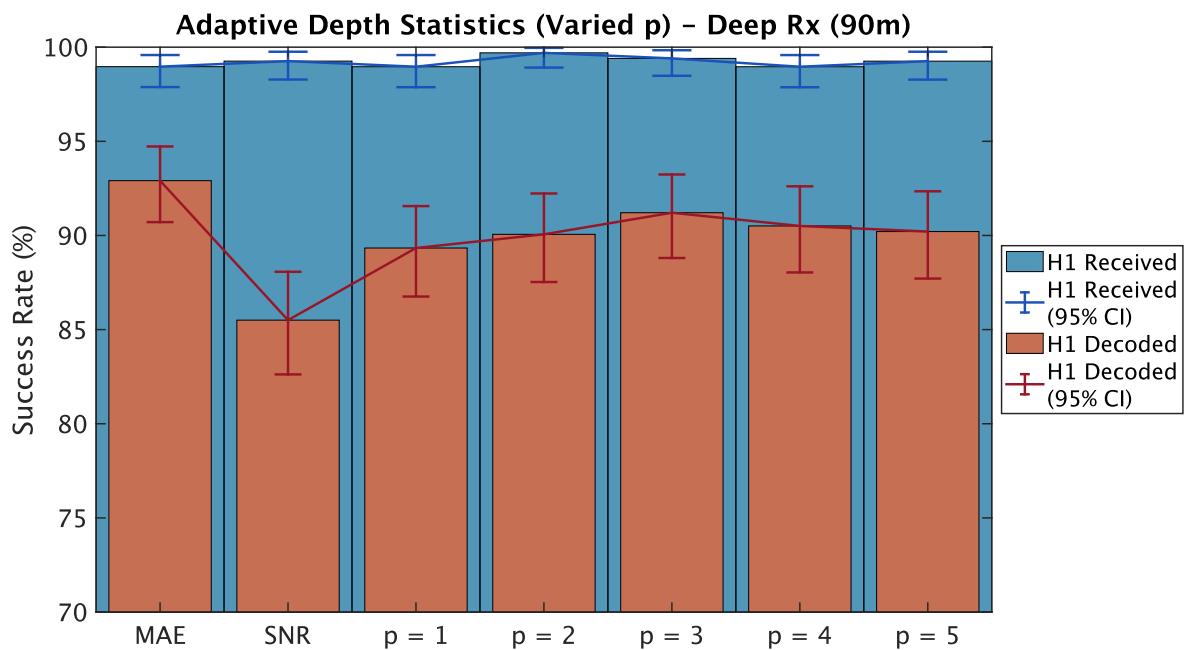


Figure 4.22: Adaptive Depth Simulation Results for Varied p , Deep Receiver (90 m)

<i>b</i> Variation						
H1-Shallow (30 m)						
Metric	<i>Macrura</i> Sent	H1 Received		H1 Decoded		
		No. (%)	CI* (%)	No. (%)	CI* (%)	
MPP (<i>b</i> = 1)	673	567 (84.3)	81.3 - 86.9	520 (77.3)	73.9 - 80.4	
MPP (<i>b</i> = 2)	673	596 (88.6)	85.9 - 90.9	570 (84.7)	81.8 - 87.3	
MPP (<i>b</i> = 3)	673	602 (89.5)	86.9 - 91.7	566 (84.1)	81.1 - 86.8	
MPP (<i>b</i> = 4)	675	619 (91.7)	89.4 - 93.7	597 (88.4)	85.8 - 90.8	
MPP (<i>b</i> = 5)	673	559 (83.1)	80.0 - 85.8	530 (78.8)	75.5 - 81.8	
H1-Deep (90 m)						
Metric	<i>Macrura</i> Sent	H1 Received		H1 Decoded		
		No. (%)	CI* (%)	No. (%)	CI* (%)	
MPP (<i>b</i> = 1)	675	675 (100)	99.5 - 100	615 (91.1)	88.7 - 93.2	
MPP (<i>b</i> = 2)	674	671 (99.6)	98.7 - 99.9	613 (91.0)	88.5 - 93.0	
MPP (<i>b</i> = 3)	674	669 (99.3)	98.3 - 99.8	613 (91.1)	88.5 - 93.0	
MPP (<i>b</i> = 4)	675	670 (99.3)	98.3 - 99.8	600 (89.0)	86.3 - 91.2	
MPP (<i>b</i> = 5)	673	666 (99.0)	97.9 - 99.6	612 (90.9)	88.5 - 93.0	

Table 4.3: NETSIM Adaptive Depth Simulation Results for a given mission set with b varied between runs. p is set to 3, and τ to 20 ms. * 95% Clopper-Pearson confidence interval.

4.4 DETERMINING b

Repeating the simulation missions, this time with p set to 3 and varying b , the main result is shown in Table 4.3, and plotted in Figures 4.23 and 4.24. Unfortunately, b does not significantly improve reception in the shallow case to make up for the difference between MPP and SNR. Though the shallow case shows that a b of 4 is essentially equivalent to b of 0, there is a small dip in performance for b of 4 in the deep case. In general, there is little to no difference between the results for variation in the deep case, which is expected since there are many more bounce path possibilities above the Beaufort Lens.

The packet success analysis showed that when transmitting and receiving above the Beaufort Lens, packet success is unexpectedly high when compared to any of the three metrics (Figure 4.13). Since the variation of b does not significantly improve performance for a set p , b may be set to 4 for the subsequent τ variation with little impact.

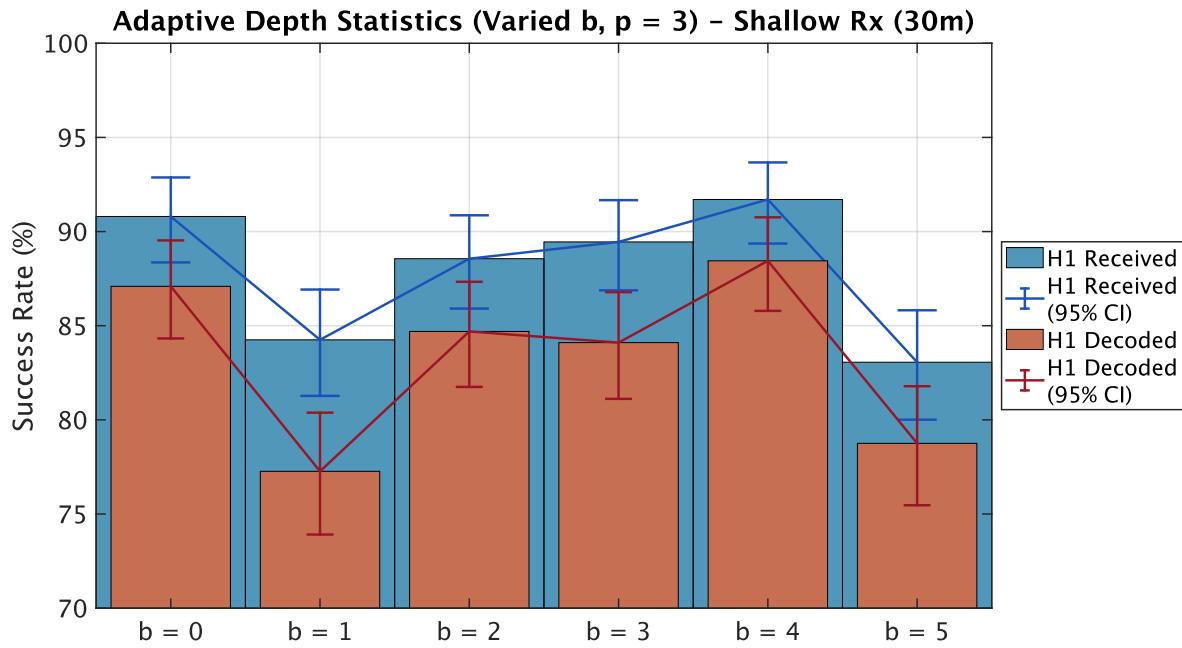


Figure 4.23: Adaptive Depth Simulation Results for Varied b , Shallow Receiver (30 m)

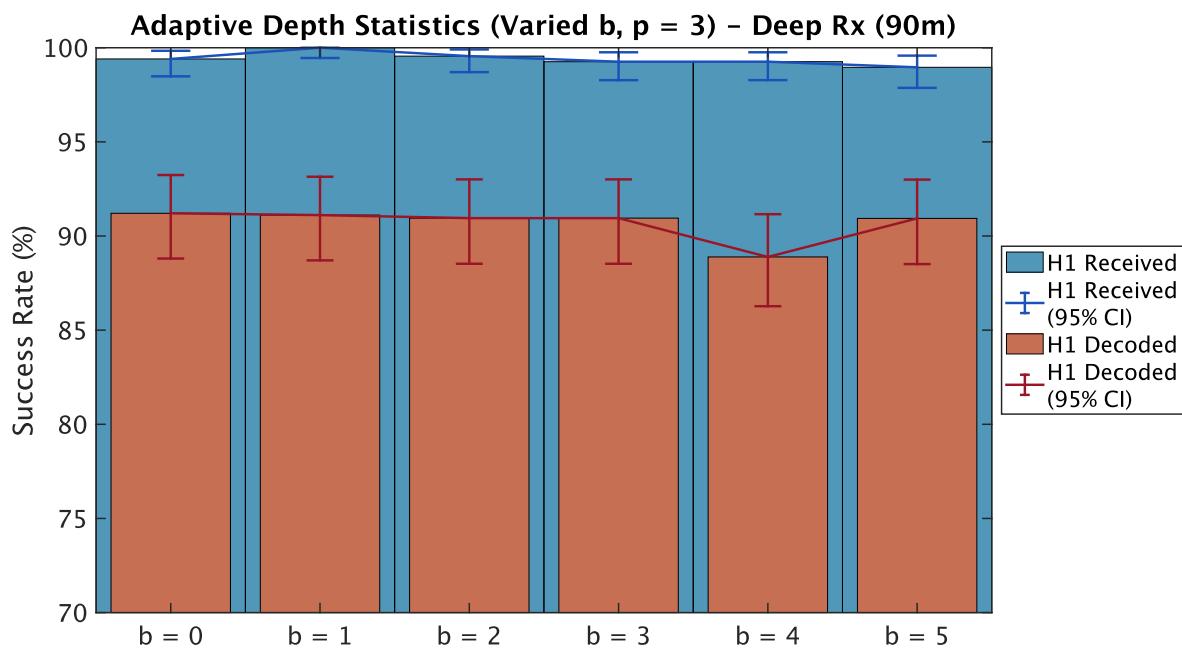


Figure 4.24: Adaptive Depth Simulation Results for Varied b , Deep Receiver (90 m)

τ Variation						
H1-Shallow (30 m)						
Metric	<i>Macrura</i> Sent	H1 Received		H1 Decoded		
		No. (%)	CI* (%)	No. (%)	CI* (%)	
MPP ($\tau = 15$ ms)	675	582 (86.2)	83.4 - 88.7	535 (79.3)	76.0 - 82.3	
MPP ($\tau = 20$ ms)	675	619 (91.7)	89.4 - 93.7	597 (88.4)	85.8 - 90.8	
MPP ($\tau = 25$ ms)	674	604 (89.6)	87.1 - 91.8	568 (84.3)	81.3 - 86.9	
H1-Deep (90 m)						
Metric	<i>Macrura</i> Sent	H1 Received		H1 Decoded		
		No. (%)	CI* (%)	No. (%)	CI* (%)	
MPP ($\tau = 15$ ms)	673	668 (99.3)	98.3 - 99.8	608 (90.3)	87.9 - 92.5	
MPP ($\tau = 20$ ms)	675	670 (99.3)	98.3 - 99.8	600 (88.9)	86.3 - 91.2	
MPP ($\tau = 25$ ms)	673	671 (99.7)	98.9 - 99.9	615 (91.4)	89.0 - 93.4	

Table 4.4: NETSIM Adaptive Depth Simulation Results for a given mission set with τ varied between runs. p is set to 3, and b to 4. * 95% Clopper-Pearson confidence interval.

4.5 VALIDATING CHOICE OF τ

As indicated by Figures 4.7 and 4.8, a small change to τ does not have a significant impact on the results shown in Table 4.4. In fact, the shallow case results (Figure 4.25) show that 20 ms is an appropriate choice, while the deep case (Figure 4.26) is statistically insignificant.

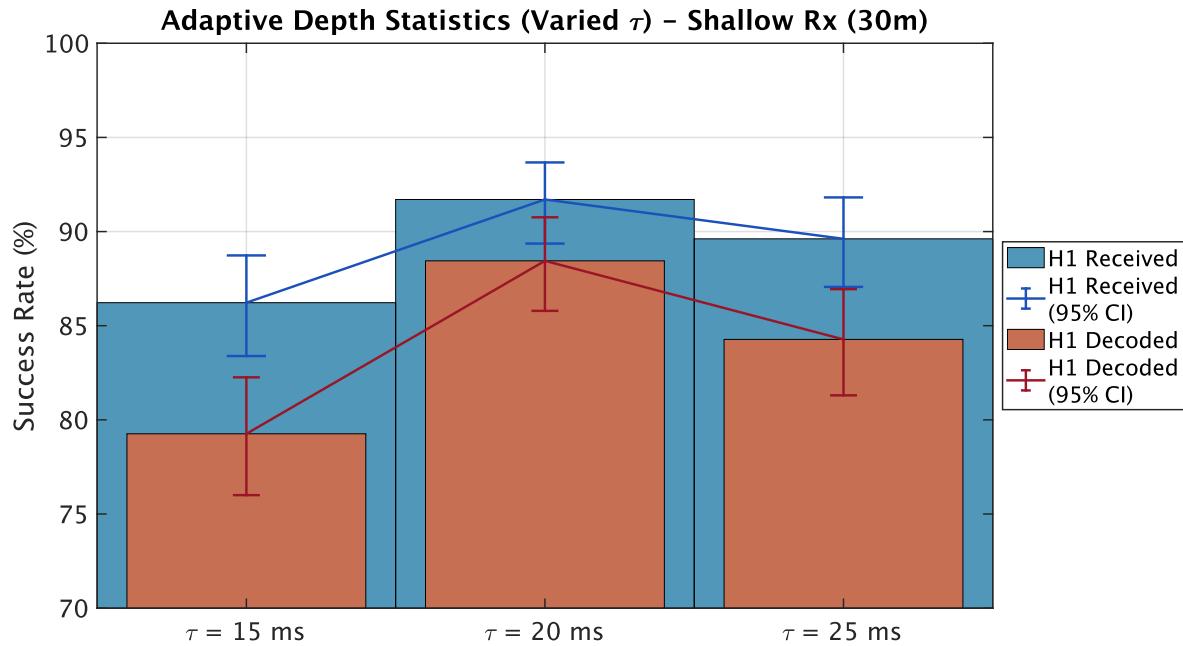


Figure 4.25: Adaptive Depth Simulation Results for Varied τ , Shallow Receiver (30 m)

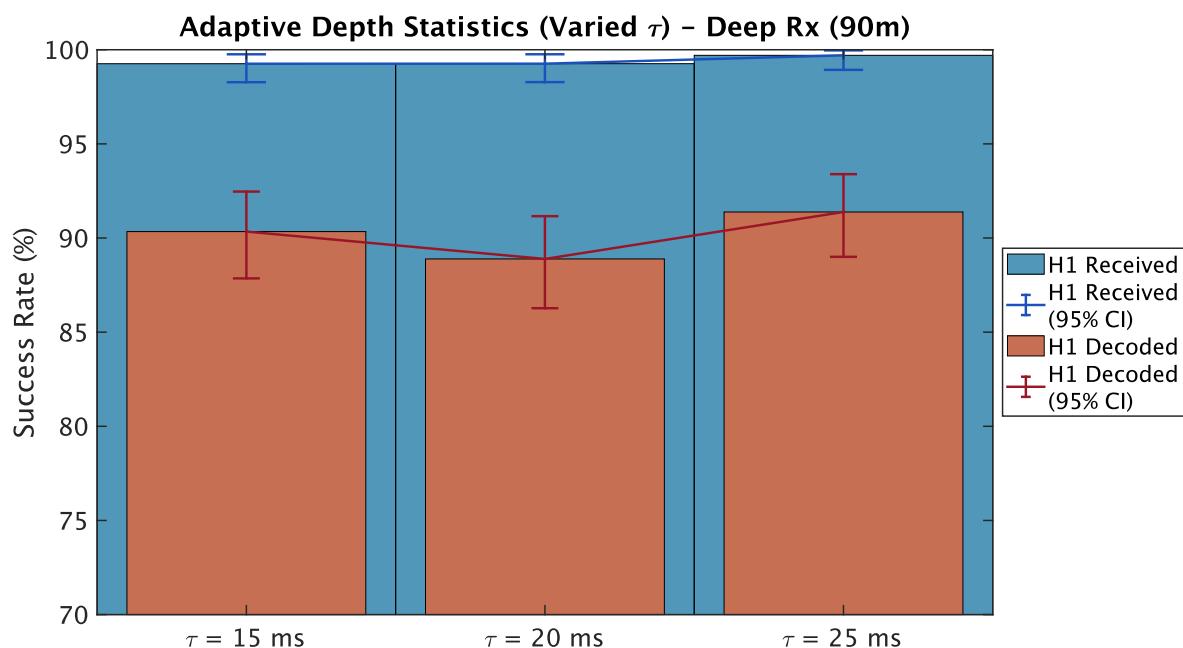


Figure 4.26: Adaptive Depth Simulation Results for Varied τ , Deep Receiver (90 m)

5

CONCLUSION & FUTURE WORK

In order to improve predictive metrics of underwater acoustic performance, this analysis has explored SNR, MAE, and a new metric called MPP. Given real environmental data obtained during ICEX20, the metrics were individually calculated over a receiver grid of depth and range appropriate for autonomous vehicle operation. These results are compared to each other in Section 4.1, and demonstrate the decision aid that an autonomous vehicle or a human decision-maker would use to make a depth selection. Various settings of MPP are also shown and compared to show the effect of the varying parameters onto the predictive result.

Next, a high fidelity Virtual Ocean was employed with hardware-in-the-loop (Netsim) to closely simulate the ICEX20 acoustic environment in the Beaufort Sea. Packet success statistics were obtained between a static, constant-depth receiving buoy and a range-varying vehicle transmitting at constant depth. These packet success statistics are shown to match those obtained during ICEX20, validating the simulator's ability to accurately replicate the environment. With this validation, Equation 4.1 was proposed to better scale MAE, rather than using Equation 3.1 which is more appropriate for SNR and MPP. The expansion of available modem statistics from ≈ 2 km during ICEX20 out to 8 km with Netsim constitutes the first contribution of this analysis, and the subsequent recommendation of Equation 4.1 to calibrate MAE is the second.

The analysis then focuses on demonstrating the performance of all three metrics in an autonomy mission for the purpose of comparing the overall results. With the autonomous vehicle now allowed to adapt its depth to collaborate with a shallow receiver and a deep receiver, the mission is repeated with the metric set to SNR, MAE, and MPP at various settings of p . It is important to note that this experiment forced the vehicle to operate at significant distance to the collaborator and constitutes a robust performance scenario over 8 km. The most interesting result of this test was to show that $p = 3$, $b = 4$, $\tau = 20$ ms tended to be the best settings among the MPP cases. This experiment also showed that, in the deep collaborator case, MAE is shown to produce the best performance in terms of percent of messages successfully decoded, and that both MAE and MPP tended to perform better than, or just as well as, SNR.

Based on this analysis, MAE should be used, if practicable, to predict modem performance in a given environment. It generally outperforms both SNR and MPP and incorporates the most

information to give an accurate picture of the modem’s response to the environment. Since MAE is a software replication of a channel estimate-based decision feedback equalizer’s operation, it closely matches how that type of modem will actually behave. Even if not employing this specific type of modem – or, alternately, no modem at all – the metric includes the time component of acoustics to a predictive measure, which SNR does not. In the absence of MAE, MPP should be used, particularly in ocean applications where multi-path has a strong effect.

This analysis also proposes a test procedure in Section 3.2 to take an ambient acoustic environment and explore it, in order to pick the best metric and settings. As a contribution, this procedure can be repeated for operational areas other than the Beaufort Sea – for instance, the Irminger Sea – and can be used in future work as a supplement to applications where acoustic communication is important.

Future work on MPP or MAE could include how the metrics fare at extreme ranges, or in environments where bathymetry is extremely deep. While *Macrura* and many other AUVs are limited in range due to other engineering concerns, improvements in deployment duration and inter-vehicle collaboration are extending their operational range over time. While 8 km is significant for 10 kHz modem transmissions, an interesting study could include lower frequency and longer distance to examine the utility of MPP as a performance metric.

A APPENDIX

A.1 FUNCTION TO CALCULATE MPP IN MATLAB: MPP

```
function [MPP,SNR,Pos,MPL_MASK] = mpp(filename,penalty,bounce_penalty, ...
                                         Source_Level_dB,Noise_Level_dB)
% mpp - Returns Multi-path Penalty (MPP) calculation for a given .arr file
% generated with the Ocean Acoustics Toolbox
% (http://oalib.hlsresearch.com/AcousticsToolbox/).
%
%     MPP = SL - NL - TL - MPL
%     SNR = SL - NL - TL
%
%     TL = -10*log10( Total Power )
%     MPL = -10*log10([1 - Pmpp/Pmax]^p) - bN, where Pmpp is the second most
%     powerful ray arriving tau seconds later, Pmax is the most powerful ray,
%     p is the path penalty, b is the bounce penalty, N is the # of bounces
%
%     The calculation for MPP is derived from the
%     source code of the Laboratory for Autonomous Marine Sensing Systems
%     (LAMSS) pDepthAdapt function (Henrik Schmidt, Oscar Viquez -
%     http://lamss.mit.edu).
%
% Assembled by: Bradli Howard (bradli@mit.edu).
%
% [MPP,SNR,Pos,MPL_MASK] = mpp(filename,penalty,bounce_penalty,
%                               Source_Level_dB,Noise_Level_dB)
%
% filename      - file path to desired .arr file. No other extensions
%                  accepted.
```

```

%   penalty      - (optional) desired multi-path penalty for calculation.
%                      Default is 3.
%
%   bounce_penalty - (optional) desired bounce penalty. Default is 3.
%
%   Source_Level_dB - (optional) source level, default is 183dB (ICEX20).
%
%   Noise_Level_dB - (optional) noise level, default is 40dB (ICEX20).

if ~exist('penalty','var')
    penalty = 3;
end
if ~exist('bounce_penalty','var')
    bounce_penalty = 3;
end
if ~exist('Source_Level_dB','var')
    Source_Level_dB = 183; % icex20 default
end
if ~exist('Noise_Level_dB','var')
    Noise_Level_dB = 40; % icex20 default
end

%%%%% read_arrivals_asc.m (From Acoustics Toolbox - citation at start) %%%%
%%% The following is modified only to remove 3D arrivals file section %%%
fid = fopen(filename,'r'); % open the file
if (fid == -1)
    error([mfilename,: Arrivals file cannot be opened'])
end
% read the 2D/3D flag to determine the file format
flag = fscanf(fid,'%s',1);
% check for the case of erroneously reading a BINARY format arrivals file
if ~strcmp(flag,''2D'') && ~strcmp(flag,''3D'')
    error([mfilename,: not an ASCII format Arrivals file?'])
end
% proceed accordingly for the Bellhop 2D vs 3D format
if strcmp(flag,''2D'')

```

```

Pos.freq = fscanf(fid,'%f',1);    % acoustic frequency
Nsz      = fscanf(fid,'%i',1);    % number of source depths
Pos.s.z = fscanf(fid,'%f',Nsz);   % source depths
Nrz      = fscanf(fid,'%i',1);    % number of receiver depths
Pos.r.z = fscanf(fid,'%f',Nrz);   % receiver depths
Nrr      = fscanf(fid,'%i',1);    % number of receiver ranges
Pos.r.r = fscanf(fid,'%f',Nrr);   % receiver ranges
% pre-allocate memory for the Arr arrivals structure array
Arr = repmat(struct('Narr',{int16(0)}), ...
             'A', {single(1.0i)}, ...
             'delay', {single(1.0i)}, ...
             'SrcDeclAngle', {single(0.0)}, ...
             'RcvrDeclAngle',{single(0.0)}, ...
             'NumTopBnc', {int16(0)}, ...
             'NumBotBnc', {int16(0)}),Nrr,Nrz,Nsz);
% loop over sources, rcv depths and rcv ranges to read all arrival info
for isd = 1:Nsz
    % read the maximum number of arrivals for this source
    Narrmx2 = fscanf(fid,'%i',1);
    for irz = 1:Nrz
        for irr = 1:Nrr
            % read the number of arrivals at this receiver
            Narr = fscanf(fid,'%i',1);
            Arr(irr,irz,isd).Narr = int16(Narr);
            % read and store all the arrivals, if there are any
            if Narr > 0
                da = single(fscanf(fid,'%f',[8,Narr]));
                Arr(irr,irz,isd).A = single( ...
                    da(1,1:Narr).*exp(1.0i*da(2,1:Narr)*pi/180.0));
                Arr(irr,irz,isd).delay = single( ...
                    da(3,1:Narr) + 1.0i*da(4,1:Narr));
                Arr(irr,irz,isd).SrcDeclAngle = da(5,1:Narr);
                Arr(irr,irz,isd).RcvrDeclAngle = da(6,1:Narr);
                Arr(irr,irz,isd).NumTopBnc     = int16(da(7,1:Narr));
                Arr(irr,irz,isd).NumBotBnc    = int16(da(8,1:Narr));
            end
        end
    end
end

```

```

        end % if any arrivals
    end % next receiver range
end % next receiver depth
end % next source
else % end of read 2D file format data
    error([mfilename,: 3D arrivals not supported.'])
end % end of read 3D file format data
fclose(fid);
eigenray_table = Arr;
meta = Pos;

%%%%% End read_arrivals_asc.m / Begin calculation from pDepthAdapt %%%%%%
num_ranges = length(meta.r.r);
num_depths = length(meta.r.z);
SNR = zeros(num_depths,num_ranges);
MPP = SNR;
MPL_MASK = SNR;
for j = 1:num_ranges
    for i = 1:num_depths
        % Determine total power and maximum power eigenray
        if eigenray_table(j,i).Narr < 1
            SNR(i,j) = -150; MPP(i,j) = -150;
        else
            N = eigenray_table(j,i).Narr;
            pow_max = 1e-30;
            power = 1e-30;
            for iray = 1:N
                pow_ray = (abs(eigenray_table(j,i).A(iray)))^2;
                power = power + pow_ray;
                if pow_ray > pow_max
                    iray_max = iray;
                    pow_max = pow_ray;
                    max_tau = abs(eigenray_table(j,i).delay(iray));
                    max_power_bounces=eigenray_table(j,i).NumTopBnc(iray) ...
                        + eigenray_table(j,i).NumBotBnc(iray);
                end
            end
        end
    end
end

```

```

    end
end
max_rat = 0;
for iray = 1:eigenray_table(j,i).Narr
    pow_ray = (abs(eigenray_table(j,i).A(iray)))^2;
    rat = pow_ray/pow_max;
    if (iray ~= iray_max && rat > max_rat && ...
        abs(max_tau-abs(eigenray_table(j,i).delay(iray))) ...
        < 0.02)
        max_rat = rat;
    end
end
SNR(i,j) = Source_Level_dB - Noise_Level_dB - -10*log10(power);
MPL_MASK(i,j) = (1-max_rat)^penalty;
MPP(i,j) = Source_Level_dB - Noise_Level_dB - ...
            - 10*log10(power*(1-max_rat)^penalty) ...
            - double(bounce_penalty*max_power_bounces);
end
end
end % function end

```

A.2 FUNCTION TO CALCULATE MAE IN MATLAB: MINAE

```
function [MAE,SDE,Pos] = minae(filename, vr_R, vr_S, Bandwidth, Fs, ...
    Source_Level_dB, Noise_Level_dB, Nbaud, num_ff_taps, num_fb_taps, nfft)
% minae - Minimum Achievable Error (MAE) & Soft Decision Error (SDE)
%   calculation for a given .arr file from the Acoustics Toolbox
%   (http://oalib.hlsresearch.com/AcousticsToolbox/). MAE/SDE calculation
%   derived from J. Preisig (jpreisig@jpanalytics.com) & H. Schmidt
%   (henrik@mit.edu).
% Assembled by Bradli Howard (bradli@mit.edu).
%
% [MAE,SDE,Pos] = minae(filename, vr_R, vr_S, Bandwidth, Fs, Source_Level_dB,
%                      Noise_Level_dB, Nbaud, num_ff_taps, num_fb_taps, nfft)
%
% filename      - 2-D single source BELLHOP .arr file
% vr_R          - Relative velocity of receiver to source
% vr_S          - Relative velocity of source to receiver
% Bandwidth     - Modem, double sided
% Fs            - Modem sampling frequency
% Source_Level_dB - Source level in decibels re 1uPa,1m
% Noise_Level_dB - Ambient noise level in decibels re 1uPa,1m
% Nbaud         - baud for modem
% num_ff_taps   - Number feed-forward taps, modem parameter
% num_fb_taps   - Number feed-back taps, modem parameter
% nfft          - number discrete Fourier Transform bins

%% Set defaults
if ~exist('vr_R', 'var')
    vr_R = 0;
end
if ~exist('vr_S', 'var')
    vr_S = 0;
end
if ~exist('Bandwidth', 'var')
    Bandwidth = 4000;
```

```

end
if ~exist('Fs','var')
    Fs = 40000;
end
if ~exist('Source_Level_dB','var')
    Source_Level_dB = 183;
end
if ~exist('Noise_Level_dB','var')
    Noise_Level_dB = 40;
end
if ~exist('Nbaud','var')
    Nbaud = 18;
end
if ~exist('num_ff_taps','var')
    num_ff_taps = 21;
end
if ~exist('num_fb_taps','var')
    num_fb_taps = 20;
end
if ~exist('nfft','var')
    nfft = 2048;
end

%%%% read_arrivals_asc.m (From Acoustics Toolbox - citation at start) %%%
disp('Opening arr file...');

fid = fopen(filename,'r'); % open the file
if (fid == -1)
    error([mfilename,: Arrivals file cannot be opened'])
end
% read the 2D/3D flag to determine the file format
flag = fscanf(fid,'%s',1);
% check for the case of erroneously reading a BINARY format arrivals file
if ~strcmp(flag,'''2D''') && ~strcmp(flag,'''3D''')
    error([mfilename,: not an ASCII format Arrivals file?'])
end

```

```

% proceed accordingly for the Bellhop 2D vs 3D format
if strcmp(flag,''2D''')
    Pos.freq = fscanf(fid,'%f',1); % acoustic frequency
    Nsz      = fscanf(fid,'%i',1); % number of source depths
    Pos.s.z = fscanf(fid,'%f',Nsz); % source depths
    Nrz     = fscanf(fid,'%i',1); % number of receiver depths
    Pos.r.z = fscanf(fid,'%f',Nrz); % receiver depths
    Nrr     = fscanf(fid,'%i',1); % number of receiver ranges
    Pos.r.r = fscanf(fid,'%f',Nrr); % receiver ranges
    % pre-allocate memory for the Arr arrivals structure array
    Arr = repmat( struct( 'Narr', {0}, ...
        'A',           {0}, ...
        'delay',       {0}, ...
        'doppler',     {0}, ... % added for MAE
        'impulse',     {0}, ... % added for MAE
        'SrcDeclAngle', {0}, ...
        'RcvrDeclAngle', {0}, ...
        'NumTopBnc',   {0}, ...
        'NumBotBnc',   {0}),Nrr,Nrz,Nsz);

    % loop over sources, rcv depths and rcv ranges to read all arrival info
    for isd = 1:Nsz
        % read the maximum number of arrivals for this source
        Narrmx2 = fscanf(fid,'%i',1);
        for irz = 1:Nrz
            for irr = 1:Nrr
                % read the number of arrivals at this receiver
                Narr = fscanf(fid,'%i',1);
                Arr(irr,irz,isd).Narr = Narr;
                % read and store all the arrivals, if there are any
                if Narr > 0
                    da = fscanf(fid,'%f',[8,Narr]);
                    Arr(irr,irz,isd).A = ...
                        da(1,1:Narr).*exp(1.0i*da(2,1:Narr)*pi/180.0);
                    Arr(irr,irz,isd).delay = ...
                        da(3,1:Narr) + 1.0i * da(4,1:Narr);
                end
            end
        end
    end
end

```

```

        Arr(irr,irz,isd).SrcDeclAngle = da(5,1:Narr);
        Arr(irr,irz,isd).RcvrDeclAngle = da(6,1:Narr);
        Arr(irr,irz,isd).NumTopBnc      = da(7,1:Narr);
        Arr(irr,irz,isd).NumBotBnc     = da(8,1:Narr);

        end % if any arrivals
    end % next receiver range
end % next receiver depth
end % next source
else % end of read 2D file format data
    error([mfilename,: 3D arrivals not supported.'])
end % end of read 3D file format data
fclose(fid);
eigenray_table = Arr;
num_depths = length(Pos.r.z); num_ranges = length(Pos.r.r);
Fc = Pos.freq;

%%% imp_response & modem_ray_arrivals - written for single receiver pt
if num_ranges > 50
    disp('Warning: Large number of receivers. This may take a long time!');
    prompt = 'Continue? (y/n): ';
    x = input(prompt,'s');
    if x == 'y'
        disp('Calculating impulse response... ');
    else
        error('Aborting... ');
    end
end
oneossp = 1/1450;
delta_t = 1/Fs;
for j = 1:num_ranges
    disp(['Range ' num2str(j) ' of ' num2str(num_ranges) ' ...'])
    for i = 1:num_depths
        num_rays = eigenray_table(j,i).Narr;
        eigenray_table(j,i).doppler = zeros(1,num_rays);
    end
end

```

```

for iray = 1:num_rays
    % calculate eigenray delay & doppler
    src_ray_cos = cosd(eigenray_table(j,i).SrcDeclAngle(iray));
    rcv_ray_cos = cosd(eigenray_table(j,i).RcvrDeclAngle(iray));
    temp = abs(eigenray_table(j,i).delay(iray));
    eigenray_table(j,i).delay(iray) = temp ...
        *(1 + vr_R*rcv_ray_cos*oneossp);
    eigenray_table(j,i).doppler(iray) = 1 + (src_ray_cos*vr_S ...
        - rcv_ray_cos*vr_R)*oneossp;
end

max_delay = max(eigenray_table(j,i).delay);
min_delay = min(eigenray_table(j,i).delay);
num_samples = round((max_delay-min_delay)*Fs + nfft);
eigenray_table(j,i).impulse = zeros(1,num_samples);

for iray = 1:num_rays
    %calculate doppler scaled waveform
    center_freq = Fc/eigenray_table(j,i).doppler(iray);
    band = Bandwidth/eigenray_table(j,i).doppler(iray);
    a = 0.5*band;
    imp_res = zeros(1,nfft);
    for k = 0:nfft-1
        t = k*delta_t;
        if (k == 0)
            w = 1;
        else
            w = sin(2*pi*a*t)/(2*a*pi*t);
        end
        imp_res(k+1) = w*cos(2*pi*center_freq*t);
    end
    t1 = eigenray_table(j,i).delay(iray) - min_delay;
    idx1 = round(t1*Fs);
    idx2 = idx1 + length(imp_res) - 1;
    if (idx1 < num_samples && idx2 >=0)

```

```

        idx_off = -idx1;
        if idx1 < 0
            idx1 = 0;
        end
        if idx2 < num_samples-1
            idx2 = num_samples-1;
        end
        ind = idx1;
        while (ind <= idx2 && ind + idx_off < length(imp_res))
            temp = eigenray_table(j,i).impulse(ind+1);
            eigenray_table(j,i).impulse(ind+1) = temp ...
                + imp_res(ind+idx_off+1) ...
                * abs(eigenray_table(j,i).A(iray));
            ind = ind + 1;
        end
    end
end
end
end
end
disp('Doppler Impulse Response done, calculating MAE...')

%%%% Noise & Covariance & Cross-spectral Density
minimum_frequency = Fc - 0.5*Bandwidth;
maximum_frequency = Fc + 0.5*Bandwidth;
delta_f = Fs/nfft;
i_first = round(minimum_frequency/delta_f);
i_last = round(maximum_frequency/delta_f);
noise_level = 10^(Noise_Level_dB/10);
FreqVec = (i_first:i_last)*delta_f;
h_lpf = ones(1,Nbaud)/Nbaud;
if (num_ff_taps/2~=round(num_ff_taps/2)) % checking odd taps/taps
    ffsampleoffsets = -(num_ff_taps-1)/2:(num_ff_taps-1)/2 ...
        + min(2,floor(num_ff_taps/2));
else
    ffsampleoffsets = -num_ff_taps/2+1:num_ff_taps/2 ...

```

```

+ min(1,(num_ff_taps-2));
end
fsr = 2;
dsfactor = Nbaud/fsr;
NoiseCovariance = ones(1,length(FreqVec))*noise_level*Fs*nfft/2;

% equalizer config
numcausalffftaps = -ffsampleoffsets(1);
numcausalffftaps = ffsampleoffsets(end)+1;

% integration kernel
integration_kernel = zeros(2*Nbaud-1,1);
for offset = 0:Nbaud-1
    integration_kernel(Nbaud+offset) = (delta_t^2) ...
        *sum(h_lpf(1:Nbaud-offset).*conj(h_lpf(1+offset:Nbaud))) ...
        .*exp(-1i*2*pi*Fc*offset*delta_t);
    if (offset~=0)
        integration_kernel(Nbaud-offset) = ...
            conj(integration_kernel(Nbaud+offset));
    end
end

max_delay_samps_at_Fs = (num_ff_taps-1)*dsfactor + Nbaud;
invFT = delta_f*exp((1i*2*pi*FreqVec.')*(0:max_delay_samps_at_Fs-1));
Rv_spatial_temporal = NoiseCovariance*invFT;

% correlation of baseband and downsampled noise covariance
Rv_bbds = zeros(num_ff_taps,num_ff_taps);
for kk1 = 1:num_ff_taps
    array_ind1 = mod(kk1-1,1)+1;
    delay1 = (kk1-array_ind1);
    for kk2 = kk1:num_ff_taps
        array_ind2 = mod(kk2-1,1)+1;
        delay2 = (kk2-array_ind2);
        delayoffset_at_Fs = (delay2 - delay1)*dsfactor;

```

```

% samples at the passband sample rate
ii = delayoffset_at_Fs;
temp = 0;
while (ii < delayoffset_at_Fs+2*Nbaud-1) && ...
(ii < length(Rv_spatial_temporal))
temp = temp + Rv_spatial_temporal(ii+1) ...
.*integration_kernel(ii+1-delayoffset_at_Fs);
ii = ii + 1;
end
Rv_bbds(kk1,kk2) = temp*exp(-1i*2*pi*Fc*delayoffset_at_Fs*delta_t);
Rv_bbds(kk2,kk1) = conj(temp);
end
end

%%%% Calculate MAE
MAE = zeros(num_depths,num_ranges); SDE = zeros(num_depths,num_ranges);
for j = 1:num_ranges
for i = 1:num_depths
pbCIR = (eigenray_table(j,i).impulse);
cirlen = length(pbCIR);
expvec = exp(-2*1i*pi*(Fc*(0:cirlen-1)/Fs));
startind = floor((Nbaud-1)/2);
bbdslength = floor((cirlen-1)/dsfactor + 1);
temp2 = conv(h_lpf,pbCIR.*expvec);
% downsample
bbds_cir = temp2(startind+(0:bbdslength-1)*dsfactor);
% apply source level / find sync point and align lengths
power_delay_profile = abs(bbds_cir).^2;
[~,syncpoint] = max(power_delay_profile);

% adds right and/or left zero padding if needed
if (syncpoint-1 < numacausalfftaps)
numpad = numacausalfftaps - syncpoint + 1;
bbds_cir = [zeros(1,numpad) bbds_cir];
syncpoint = syncpoint + numpad;

```

```

        bbdslength = bbdslength + numpad;
    end
    if (bbdlength - syncpoint) < numcausalfftaps - 1
        numpad = numcausalfftaps - 1 - bbdslength + 1;
        bbds_cir = [bbd_cir zeros(1,numpad)];
        bbdslength = bbdslength + numpad;
    end
    bbds_cir = (10^(Source_Level_dB/20))*bbds_cir;

    % Populate baseband, downsample Green's function
    numsamprateGcolumns = bbdslength + num_ff_taps - 1;
    samprateGsynccolumn = bbdslength + 1 - syncpoint ...
        + numcausalfftaps;
    temp = floor((samprateGsynccolumn-1)/fsr);
    samprateGcolumnsforsymrateG = (samprateGsynccolumn-fsr ...
        *temp:fsr:numsamprateGcolumns);
    Gsynccolumn = temp + 1;
    numGcolumns = length(samprateGcolumnsforsymrateG);
    numGrows = num_ff_taps;
    G = zeros(numGrows,numGcolumns);
    for curfftap = 1:num_ff_taps
        samprateGrow = [zeros(1,curfftap-1),fliplr(bbds_cir), ...
            zeros(1,numsamprateGcolumns-bbdslength-curfftap+1)];
        G(curfftap,:) = samprateGrow(samprateGcolumnsforsymrateG);
    end
    go = G(:,Gsynccolumn);
    if (Gsynccolumn - 1 < num_fb_taps)
        error('num_fb_taps is too large')
    end
    Go = [G(:,(1:Gsynccolumn-num_fb_taps-1)) G(:,Gsynccolumn+1:end)];
    Qnoise = Go*Go' + Rv_bbds;
    predictedMAE = 1/(1+real(go'/Qnoise*go));
    % account for finite averaging window length
    totalnumtaps = num_ff_taps + num_fb_taps;
    avewinmultiplier = 5;

```

```
rls_lambda = 1 - 1/(avewinmultiplier * totalnumtaps);
effwindowlen = 1/(1-rls_lambda);
finite_window_scaling_factor = (effwindowlen -1) ...
    /(effwindowlen - 1 - totalnumtaps);
predictedSDE = finite_window_scaling_factor*predictedMAE;
MAE(i,j) = -10*log10(predictedMAE);
SDE(i,j) = -10*log10(predictedSDE);

end
end
disp('Done!');
end %function end
```

BIBLIOGRAPHY

1. *Acoustics Toolbox (BELLHOP)*. Ocean Acoustics Library. URL: https://oalib-acoustics.org/AcousticsToolbox/index_at.html.
2. E. C. Bhatt. "A Virtual Ocean framework for environmentally adaptive, embedded acoustic navigation on autonomous underwater vehicles". MIT-WHOI, 2021.
3. E. P. Chassignet, H. E. Hurlbert, O. M. Smedstad, G. R. Halliwell, P. J. Hogan, A. J. Wallcraft, R. Baraille, and R. Bleck. "The HYCOM (HYbrid Coordinate Ocean Model) data assimilative system". *Journal of Marine Systems* 65, 1-4 SPEC. ISS. 2007, pp. 60–83. ISSN: 09247963. doi: [10.1016/j.jmarsys.2005.09.016](https://doi.org/10.1016/j.jmarsys.2005.09.016).
4. R. Chen and H. Schmidt. "Temporal and spatial characteristics of the Beaufort Sea ambient noise environment". *The Journal of the Acoustical Society of America* 148, 6 2020, pp. 3928–3941. ISSN: 0001-4966. doi: [10.1121/10.0002955](https://doi.org/10.1121/10.0002955).
5. R. Doshi, A. Dale-Huang, and G. Zhang. *Northern Expedition: China's Arctic Activities and Ambitions*. Brookings, 2021. URL: <https://www.brookings.edu/research/northern-expedition-chinas-arctic-activities-and-ambitions/>.
6. E. Gallimore, J. Partan, I. Vaughn, S. Singh, J. Shusta, and L. Freitag. "The WHOI micromodem-2: A scalable system for acoustic communications and networking". In: 2010. ISBN: 9781424443321. doi: [10.1109/OCEANS.2010.5664354](https://doi.org/10.1109/OCEANS.2010.5664354).
7. F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt. *Computational Ocean Acoustics*. Ed. by W. M. Hartmann. Second. Springer, 2011. ISBN: 978-1-4419-8677-1. doi: [10.1007/978-1-4419-8678-8](https://doi.org/10.1007/978-1-4419-8678-8).
8. K. V. Mackenzie. "Nine-term equation for sound speed in the oceans". *Journal of the Acoustical Society of America* 70, 3 1981, pp. 807–812. ISSN: NA. doi: [10.1121/1.386920](https://doi.org/10.1121/1.386920). URL: <http://asa.scitation.org/doi/10.1121/1.386920>.
9. H. Medwin. *Speed of sound in water: A simple equation for realistic parameters*. 1975. doi: [10.1121/1.380790](https://doi.org/10.1121/1.380790). URL: <http://asa.scitation.org/doi/10.1121/1.380790>.
10. H. Medwin and C. S. Clay. *Fundamentals of Acoustical Oceanography*. Academic Press, 1998.

11. *Micromodem-2 User's Guide (Version number: 1.2)*. Woods Hole Oceanographic Institution Acoustic Communications Group, 2017.
12. MOOS. Oxford Robotics Group. URL: <https://www.robots.ox.ac.uk/~mobile/MOOS/wiki/pmwiki.php/Main/HomePage>.
13. MOOS-IvP. MIT Center for Ocean Engineering and Department of Mechanical Engineering. URL: <https://oceanai.mit.edu/moos-ivp/pmwiki/pmwiki.php>.
14. M. B. Porter. *The BELLHOP Manual and User's Guide*. 2011.
15. J. C. Preisig. "Performance analysis of adaptive equalization for coherent acoustic communications in the time-varying ocean environment". *The Journal of the Acoustical Society of America* 118, 1 2005, pp. 263–278. ISSN: 0001-4966. doi: [10.1121/1.1907106](https://doi.org/10.1121/1.1907106).
16. S. Randeni, T. Schneider, and H. Schmidt. "Construction of a high-resolution under-ice AUV navigation framework using a multidisciplinary virtual environment". *IEEE/OES Autonomous Underwater Vehicles Symposium*, 2020. doi: [10.1109/AUV50043.2020.9267950](https://doi.org/10.1109/AUV50043.2020.9267950).
17. T. Schneider and H. Schmidt. "NETSIM: A realtime virtual ocean hardware-in-the-loop acoustic modem network simulator". *4th Underwater Communications and Networking Conference, UComms*, 2018. doi: [10.1109/UComms.2018.8493188](https://doi.org/10.1109/UComms.2018.8493188).
18. T. Schneider, H. Schmidt, and S. Randeni. "Self-Adapting Under-Ice Integrated Communications and Navigation Network". *5th Underwater Communications and Networking Conference (UComms)*, 2020.
19. C. E. Shannon. "A mathematical theory of communication". *The Bell System Technical Journal* 27:3, 1948, pp. 379–423. doi: [10.1002/j.1538-7305.1948.tb01338.x](https://doi.org/10.1002/j.1538-7305.1948.tb01338.x).
20. J. L. Spiesberger and K. Metzger. "New estimates of sound speed in water". *Journal of the Acoustical Society of America* 89, 4 1991, pp. 1697–1700. ISSN: NA. doi: [10.1121/1.401002](https://doi.org/10.1121/1.401002). URL: <http://asa.scitation.org/doi/10.1121/1.401002>.
21. J. L. Spiesberger and K. Metzger. "A new algorithm for sound speed in seawater". *The Journal of the Acoustical Society of America* 89, 6 1991, pp. 2677–2688. ISSN: 0001-4966. doi: [10.1121/1.400707](https://doi.org/10.1121/1.400707). URL: <http://asa.scitation.org/doi/10.1121/1.400707>.
22. *The Laboratory for Autonomous Marine Sensing Systems*. LAMSS Autonomy Suite. URL: <https://oceanai.mit.edu/lamss/pmwiki/pmwiki.php?n>Main.HomePage>.
23. M. L. Timmermans, J. Toole, and R. Krishfield. "Warming of the interior Arctic Ocean linked to sea ice losses at the basin margins". *Science Advances* 4, 8 2018, eaat6773. ISSN: 23752548. doi: [10.1126/sciadv.aat6773](https://doi.org/10.1126/sciadv.aat6773). URL: <http://advances.sciencemag.org/>.

24. J. M. Toole, M. L. Timmermans, D. K. Perovich, R. A. Krishfield, A. Proshutinsky, and J. A. Richter-Menge. "Influences of the ocean surface mixed layer and thermohaline stratification on Arctic Sea ice in the central Canada Basin". *Journal of Geophysical Research: Oceans* 115, C10 2010, 2009JC005660. ISSN: 0148-0227. doi: [10.1029/2009JC005660](https://doi.org/10.1029/2009JC005660). URL: <https://onlinelibrary.wiley.com/doi/10.1029/2009JC005660>.
25. J. M. Toole, R. A. Krishfield, M. L. Timmermans, and A. Proshutinsky. *The Ice-Tethered profiler: Argo of the Arctic*. 2011. doi: [10.5670/oceanog.2011.64](https://doi.org/10.5670/oceanog.2011.64).
26. C. Woody. *Russian reconnaissance planes were intercepted by US and Canadian fighters as they lurked over a submarine exercise in the Arctic*. 2020. URL: <https://www.businessinsider.com/russian-planes-lurked-over-navy-submarine-exercise-in-the-arctic-2020-3>.