Development and Testing of the AXBT Realtime Editing System (ARES)

by

Lieutenant Junior Grade Casey R. Densmore, United States Navy

B.S., United States Naval Academy (2018)

Submitted to the Department of Earth, Atmospheric
and Planetary Sciences
in partial fulfillment of the requirements for the degree of
Master of Science in Physical Oceanography
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
and the
WOODS HOLE OCEANOGRAPHIC INSTITUTION

September 2020

Author . . . .
Department of Earth, Atmospheric and Planetary Sciences
August 13, 2020

Certified by. . . . .
Steven R. Jayne
Senior Scientist
Woods Hole Oceanographic Institution
Thesis Supervisor

Accepted by . . . .
Glenn R. Flierl
Chairman, Joint Committee for Physical Oceanography
Massachusetts Institute of Technology

Development and Testing of the AXBT Realtime Editing System (ARES)

by

Lieutenant Junior Grade Casey R. Densmore,

United States Navy

Abstract

Airborne eXpendable BathyThermographs (AXBTs) are air-launched, single use temperature-depth probes that telemeter temperature observations as a VHF-modulated frequency. This study describes the AXBT Realtime Editing System (ARES), which was developed to receive and quality control temperature-depth profiles with no external hardware other than a VHF radio receiver. The ARES Data Acquisition System performs fast Fourier transforms on windowed segments of demodulated signal transmitted from the AXBT and uses the resulting spectra to identify valid temperature-depth observations. When evaluated using 389 profiles, the ARES data acquisition system produced temperature-depth profiles nearly identical to those generated using a Sippican MK-21 processor, while reducing the amount of noise from VHF interference included in those profiles. The ARES Profile Editor applies a series of automated checks to identify and correct common profile discrepancies, before displaying the profile on an editing interface that provides simple user controls to make additional corrections. When evaluated against 1,177 tropical Atlantic and Pacific AXBT profiles, the ARES automated quality control system successfully corrected 87% of the profiles without any manual intervention necessary. The ARES Data Acquisition and Profile Editing Systems performed exceptionally well when operationally tested with 44 AXBTs during Hurricane Dorian (2019), enabling high resolution observations across key oceanic features including Dorian's cold wake and the Gulf Stream. Necessary future work includes improvements on the automated quality control algorithm and evaluation against a more diverse dataset of temperature-depth profiles.

Thesis Supervisor: Steven R. Jayne
Title: Senior Scientist
Woods Hole Oceanographic Institution

Acknowledgments

I would not have reached this point without the influence of many people, and while I cannot list them all here within a reasonable word count, I am truly appreciative of their guidance and support.

First, I am thankful for my mom and grandparents, who encouraged my fascination with severe weather from a young age and supported me unconditionally along the way. You may have gotten more than you bargained for with the tornado chasing and hurricane hunting, but your encouragement enabled my interest to become a passion.

Second, I am grateful to the USNA Oceanography Department faculty who provided me with the tools necessary to succeed in the Joint Program. I would like to particularly thank my undergraduate research advisers CAPT Elizabeth Sanabia and Dr. Brad Barrett, who guided me through my first experiences researching and publishing peer-reviewed science.

CAPT Sanabia also brought me into TROPIC, which introduced me to one of the most amazing experiences in my life: flying with the Hurricane Hunters. I am grateful to CAPT Sanabia for her continued professional and academic mentorship over the past few years. I also want to thank the members of the 53rd Weather Reconnaissance Squadron for their support and patience over those years as we launched hundreds of AXBTs on training and storm missions.

Over TROPIC I met and worked with a lot of amazing people. I want to particularly thank Jordan, Grace, Shannon and Jake for the countless hours they spent testing various versions of ARES. Additionally, this work would not have been possible without technical guidance from Matt Kuhn, Jeff Kerling, and Kyle Rushing from the Naval Oceanographic Office.

Also during TROPIC I met Steve Jayne, who became my advisor for the MIT-WHOI Joint Program. Steve's guidance enabled me to select classes and research topics to take full advantage of the opportunities available to me in the Joint Program. Steve is one of the smartest people I have ever met, and I aspire to one day be a scientist of his caliber. I am also grateful to the professors and administrators in the Joint Program, as well as my classmates whom I met along the way. I would particularly like to thank Stephan Gallagher and Jeff Grabon, two exemplary Naval officers and even better friends who were willing to mentor a clueless Ensign starting his Naval career in an unconventional way.

Finally, I am eternally grateful to my wife for her constant love and support throughout my time in Massachusetts. Hannah, I love you more than I can put into words and I am so lucky to get to spend the rest of my life with you.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

12

14

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

1.1 Motivation

Obtaining in-situ temperature-depth profiles remains a necessity for observational oceanography. While significant advances in satellite capabilities have enabled global sea surface temperature observation, resolving subsurface features (e.g. mixed layer depth and ocean heat content) and observing conditions under extremely cloudy skies require in situ measurements [3]. The eXpendable BathyThermograph (XBT) is a commonly-used sensor capable of collecting a single temperature-depth profile, typically to several hundred meters depth. The aircraft-launched variant, the Airborne eXpendable BathyThermograph (AXBT), enables aircraft to measure upper-ocean temperatures in remote or inhospitable regions that are difficult to access by ship.

Applications for AXBTs range from scientific (e.g. observations of multi-scale processes, particularly in remote or inhospitable conditions) to military (e.g. underwater acoustics). One recent example of scientific use of AXBTs, which served as the motivation for this thesis, is the Training and Research in Oceanic and atmospheric Processes In Cyclones (TROPIC) field campaign [4], which has run from 2011-2020. TROPIC is conducted to make sustained observations of upper-ocean temperatures beneath tropical cyclones. ARES was developed in coordination with TROPIC as a data acquisition and quality control solution to replace the existing systems.

1.2 TROPIC Program Background

From 2011 through 2019, members of the TROPIC program deployed 1,524 AXBTs on nearly 200 flights, including missions through 30 named tropical cyclones in the Atlantic and eastern/central Pacific basins. These AXBTs were launched from WC-130J aircraft on a not-to-interfere basis during training and reconnaissance missions by the US Air Force 53rd Weather Reconnaissance Squadron (Hurricane Hunters). Successful AXBT profiles were quality controlled and transmitted to the Global Telecommunications System (GTS) for integration into numerical weather prediction models.
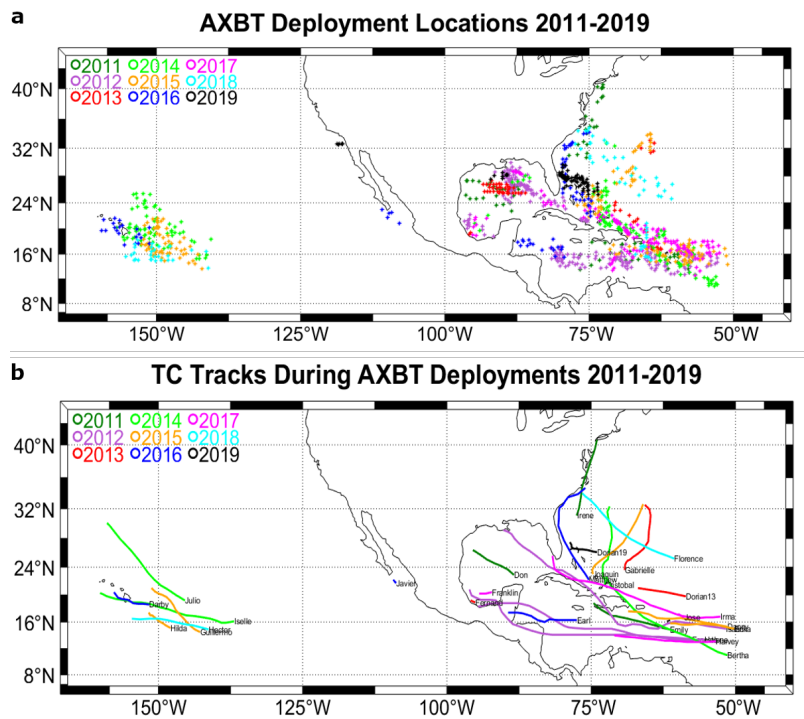
Figure 1-1: (a) AXBT deployment locations and (b) tracks of tropical cyclones in which AXBTs were deployed for TROPIC between 2011 and 2019. Figures from [1].

Through 2019, TROPIC participants collected AXBT data onboard the aircraft with a Mobile Ocean Observing System (MOOS; [4]) designed and built by the Naval Research Laboratory (NRL), Monterey, CA. This system includes a power converter, MK-10A Receiver, two Marantz audio recorders, MK-21 Data Acquisition System, and a processing computer, all encased within two approximately 2'x2'x3' boxes together weighing over 200 lbs. Signals received via the aircraft's VHF antenna are de-

modulated by the MK-10A Receiver before being simultaneously passed to a Marantz audio recorder and the MK-21 Data Acquisition System. The MK-21 calculates temperature and depth observations from the incoming audio stream and sends those observations to the processing computer, where they are simultaneously logged and plotted on the MK-21 software's user interface.

After an AXBT profile has been fully recorded, it is quality controlled using the System for At-Sea Environmental Analysis (SASEA), a software package operated in a MATLAB environment. During this quality control process, raw temperature-depth profiles are despiked and smoothed before inflection points are automatically detected for recording. Users then manually select additional points to ensure a sufficient vertical resolution to pass quality control systems for numerical models (e.g. every 10 m in the upper 100 m). These quality controlled profiles are then saved as both 1 m resolution text files and JJVV (BATHY) files, the latter of which are transmitted to the GTS. Additional information about the data acquisition and quality control processes for TROPIC can be found in [4].

While effective, this system has several disadvantages. First, the MOOS system is both large and heavy, making it difficult to carry on and off of the aircraft. Additionally, the system includes a large number of interconnected components with cables that must be attached and removed every time the system is setup and stored, introducing the possibility for user error or loose and/or degraded connectors that prevent AXBT signals from being properly received. Moreover, the quality control process requires both manual input of profile metadata (e.g. drop time and position) and manual profile editing, introducing the potential for operator error. This quality control system is also somewhat time-intensive (requiring approximately 10 minutes per profile for a well-trained operator), increasing the margin for error during rapid or extensive AXBT deployments. Finally, a major objective of TROPIC is to transition AXBT observation capabilities to the Hurricane Hunters. This places an additional emphasis on reducing requisite space (for integration into onboard systems) and time (due to the fast-paced work tempo for aircrew in the storm environment) constraints of the system, as well as the degree of training required for operators.

1.3 Contribution and Organization

The AXBT Realtime Editing System (ARES) was developed to accomplish several objectives:

1. Incorporate most hardware-defined features from MOOS as software-defined functions to reduce the amount of necessary hardware

2. Enable simultaneous processing of multiple AXBTs on different VHF channels

3. Integrate system date/time inputs and a connected GPS receiver to auto-populate fields on AXBT launch and minimize errors due to incorrectly entered drop information

4. Combine data acquisition and profile editing into a program with a single graphical user interface so users can seamlessly transition from data receipt to quality control

5. Minimize requisite oceanographic background knowledge on the part of users receiving and quality controlling AXBT-measured profiles

ARES is composed of two integrated subsystems: the Data Acquisition System, which demodulates and Fourier transforms received VHF signals and calculates the corresponding temperature depth profile, and the Profile Editing System, which applies automatic quality control checks to correct for common issues and presents the resulting profile to the user to apply any additional corrections before saving. This thesis and the accompanying research provide the following contributions:

- Developing the software-hardware suite for the AXBT Realtime Editing System (ARES)

- Describing the functionality of ARES

- Providing instructions to install and operate ARES, as well as to troubleshoot possible issues

22

- Evaluating the performance of the ARES Data Acquisition and Profile Editing Systems against the existing TROPIC dataset

- Examining the performance of ARES in an operational setting during aerial weather reconnaissance missions through Hurricane Dorian (2019)

- Presenting a case study of useful results from AXBT profiles collected and quality-controlled with ARES in Hurricane Dorian

The remainder of this thesis is organized as follows. Chapter 2 provides more detailed background about Airborne eXpendable BathyThermographs (AXBTs) and requirements for AXBT data acquisition and post-processing systems. Chapter 3 describes and evaluates the performance of the ARES Data Acquisition System. Chapter 4 describes and evaluates the performance of the ARES Profile Editing System. Chapter 5 examines the performance of ARES during Hurricane Hunter operations in Hurricane Dorian (2019) and presents some of the results enabled by collection of AXBT data at high spatial resolution with ARES. Chapter 6 includes a summary of ARES functionality and avenues for future work. Finally, the appendices include (A) installation and (B) troubleshooting guides and (C) additional technical details about ARES source code and compilation.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Background

2.1 AXBT System Overview

The Airborne eXpendable BathyThermograph (AXBT) is an air-launched, single use temperature depth probe capable of observing ocean temperatures in the upper 400 to 900 m (depending on the AXBT variant used). Components of an AXBT are shown in Fig. 2-1, left panel.

Figure 2-1: (Left) components of an AXBT (from left to right: storage canister, AXBT casing, surface float, and thermistor probe) and (right) a typical AXBT deployment diagram. Figure from `www.hurricanescience.org/science/observation/` `aircraftrecon/expendableairborneinstruments/` (images originally provided by the USCG International Ice Patrol).

The leftmost item in Fig. 2-1 is a plastic canister commonly used to store AXBTs. Launch methods vary depending on launching platform, but in many cases (including for deployments associated with this thesis) the AXBT (gray cylinder second from the left in Fig. 2-1, left panel) is removed from the canister and placed in an A-sized (approximately 13 cm in diameter) launch tube (Fig. 2-2). This tube is connected to the airframe and has a mechanism (e.g. a lever) to briefly open a seal in the base of the tube, allowing the pressure difference between the aircraft interior and surrounding environment to push the AXBT out of the tube. The AXBT then descends towards the surface beneath a small parachute; descent times can range between 120 and 150 seconds for a launch from 10,000 feet, depending on atmospheric conditions.
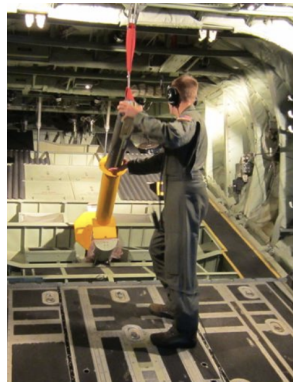
Figure 2-2: A Hurricane Hunter loads an AXBT into its launch tube.

Upon reaching the surface, the entire AXBT submerges (Fig. 2-1, right panel) and deploys a surface float (small red object second from the right in Fig. 2-1, left panel) that separates from the parachute and plastic casing (which self-scuttles). The surface float inflates a small bag to maintain positive buoyancy, and a saltwater-activated battery triggers the float to begin transmitting an unmodulated VHF carrier frequency from its antenna. This carrier frequency is one of 99 standard sonobuoy VHF frequencies between 136 and 173.5 MHz, and is either selected by the operator prior to launch or built into the AXBTs hardware (depending on the AXBT model). Simultaneously, water-soluble adhesive retaining a thermistor probe (rightmost object in Fig. 2-1, left panel) in the base of the surface float dissolves, releasing the thermistor to descend through the water column. The thermistor remains connected to the

surface float via a very thin copper wire through which it transmits temperature observations. The maximum observation depth of an AXBT is limited by the length of this wire; after the thermistor exceeds this rated depth, the wire breaks, the AXBT stops transmitting, and the surface float self-scuttles [5, 6].

As the probe descends, ambient ocean temperature alters the resistivity (and therefore resistance) of an integrated thermistor element. The thermistor's resistance in turn modifies the voltage of the integrated circuit, which is the input to a voltage-controlled oscillator (VCO) in the probe. The VCO outputs an alternating current whose frequency is proportional to the input voltage (and thus thermistor resistivity and ocean temperature). This alternating current travels up the wire to the surface float, and it is the audio-range frequency of this alternating current that is frequency modulated (FM) into the VHF carrier frequency and transmitted to the observing aircraft for data acquisition and processing. Recovering a temperature-depth profile from this signal requires empirical frequency-to-temperature and fall rate equations, discussed in the following section.

2.2 AXBT Processing

2.2.1 Data Acquisition

Processing signal from an AXBT and returning a temperature-depth profile from signal received through a VHF antenna involves repeating the following steps several times per second for the duration of AXBT profile transmission:

1. Demodulating a segment of FM VHF signal to recover the original AXBT signal (including a temperature-encoded audio-range frequency)

2. Applying a signal processing technique (e.g. Fourier transform) to recover the peak frequency from that AXBT signal segment

3. Applying empirical frequency-to-temperature and fall rate equations to determine the corresponding temperature-depth point for that signal segment

27

Demodulating incoming VHF FM signal and outputting a PCM audio stream requires a radio receiver, which may fall into one of two categories: hardware- or software-defined. Hardware-defined receivers demodulate signals from one or more fixed frequencies, and generally have physical control inputs (e.g. knobs, switches) to control settings such as demodulated signal volume. For example, the Mobile Ocean Observing System (MOOS), used during TROPIC from 2011-2019, integrates a hardware-defined MK-10A Receiver capable of demodulating FM signals from VHF channels 12, 14, and 16 (170.5, 172.0, and 173.5 MHz, respectively).

Software-defined receivers function similarly to their hardware-defined counterparts, but are controlled by a connected computer which also often handles much of the signal processing tasks required for demodulation. These receivers require an application programming interface (API) through which the computer communicates with the receiver. Though requiring an operating computer may be considered a disadvantage, software-defined receivers provide multiple advantages over hardware-defined receivers. First, they enable seamless integration of the receiver with other relevant computer software (in this context, controlling whether a selected receiver is actively demodulating, as well as its target VHF channel/frequency, from within an AXBT data acquisition program). Additionally, software-defined receivers typically require less analog hardware, potentially mitigating spatial and financial limitations. Given these advantages, ARES was developed with integrated support for the WiN-RADIO W39GSBE software-defined receiver (discussed further in Chapter 3).

Regardless of demodulation method, windowed segments of the output audio stream from the receiver must be processed to identify peak frequencies at discrete times and subsequently determine the corresponding observed temperatures and depths. The current industry standard for this task is the Sippican MK-21 Data Acquisition System. This processor and its accompanying software are designed to process received (and for airborne variants, demodulated) signals from a range of ship- and air-launched expendable ocean probes. Processed profiles are transferred to the operating computer in realtime and saved for further quality control and use. In many data acquisition systems, audio streams may also be directed to an audio recorder so

profile recordings can be reprocessed to correct for errors. It should be noted that other AXBT data acquisition systems exist, such as the Ocean Data Acquisition and Analysis Recorder (ODAAR; [7]).

Although these tasks typically require external hardware in existing AXBT data acquisition systems, both the signal processing and audio recording can be accomplished entirely on the processing computer. This enables the development of a system with significantly less required hardware (advantageous due to weight, space, and cost constraints) but which also increases the burden on the processing computer. ARES was developed so these tasks are software-defined, and their data flow and computational expense are both examined in Chapter 3.

2.2.2   Empirical Conversion Equations

Temperature and depth conversion equations are the source of a range of literature aiming to identify uncertainty and correct for a host of error sources such as internal noise and thermal lag (e.g. [2, 5, 6, 8, 9, 10, 11]). The standard Navy fall rate equation to determine observation depth from elapsed time since the first signal was identified, based on presumed fall rate of the probe (1.52 m/s) is simply $z = 1.52\Delta t$. The corresponding conversion relating temperature (^{o}C) and frequency (Hz) originally specified by Naval Air Systems Command (NAVAIR) to AXBT manufacturers is a simple linear equation (Eq. 2.1; [6]; personal communication, J. Kerling, 2020).

$$f = 1440 + 36T$$
$$T = -40 + (2.778 \times 10^{-2})f$$
(2.1)

Several studies have sought to identify more accurate, higher order frequency-to-temperature conversion equations. For example, [12] tested temperature calibrations for 48 Sippican AXBTs and identified a cubic conversion equation:

$$T = -66.8857 + (7.0273 \times 10^{-2})f - (2.1807 \times 10^{-5})f^2 + (3.6311 \times 10^{-9})f^3 \quad (2.2)$$

Additionally, Sippican released a fifth-order conversion [5]:

$$T = -126.662 + (0.219954)f - (1.70509 \times 10^{-4})f^2 + (7.70543 \times 10^{-8})f^3$$
$$- (1.7958 \times 10^{-11})f^4 + (1.73823 \times 10^{-15})f^5 \qquad (2.3)$$

However, [6] determined that Eq. (2.2) underestimated temperatures by approximately 0.1^oC and Eq. (2.3) overestimated them by approximately 0.2 - 0.5^oC, and proposed and tested a linear equation demonstrated to be more accurate:

$$T = -38.6045 + (2.71075 \times 10^{-2})f \qquad (2.4)$$

A more recent, similar but not identical fifth-order Sippican conversion (Eq. 2.5) also exists (personal communication, J. Kerling, 2020). The use of this (or an extremely similar) updated Sippican conversion equation in the MK-21 Data Acquisition System was verified by examining MK-21 output for known frequency inputs using a BANDIT Box. The BANDIT Box is useful for testing sonobuoy data acquisition systems by outputting a VHF modulated test signal (for AXBTs, this signal is a single frequency following the standard Navy conversion, Eq. 2.1). Because the VHF-modulated frequencies generated and output by the BANDIT box are known, observed residuals (Fig. 2-3, black lines) between temperatures input to the MK-21 Data Acquisition System and those output from the MK-21 following the proprietary equation were used to identify a best fit estimate (Fig. 2-3, red line), which is compared to the updated Sippican conversion (Eq. 2.5; Fig. 2-3, green line). The root-mean-square difference between the residual best fit and modified Sippican equation are less than 0.02^oC, well within the threshold for AXBT instrumental error.

$$T = -126.662 + (0.219954)f - (1.70596 \times 10^{-4})f^2 + (7.705434 \times 10^{-8})f^3$$
$$- (1.79581 \times 10^{-11})f^4 + (1.73823 \times 10^{-15})f^5 \qquad (2.5)$$
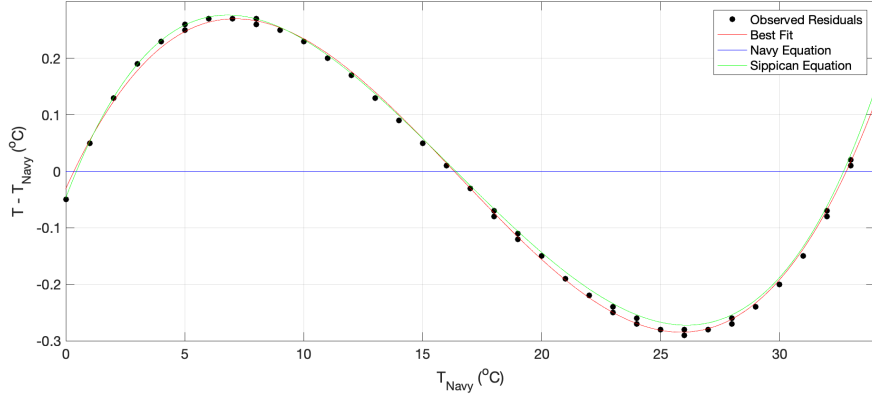
Figure 2-3: Residual temperature differences between the standard Navy and MK-21 conversion equations (black lines), a best fit equation for those residuals (red line), and the documented Sippican conversion equation (Eq. 2.5; green line).

Similar analyses exist for AXBT probe fall rate equations. As with frequency conversions, Sippican released a higher order depth equation (Eq. 2.6; [5]).

$$z = 1.5926\Delta t - 0.00018\Delta t^2 \tag{2.6}$$

However, [6] identified a depth equation that the author found to be more accurate for both shallow and deep AXBTs:

$$z = 1.6325\Delta t + (1.5553 \times 10^{-5})\Delta t^2 \tag{2.7}$$

It is worth noting that the conversion equations listed here are far from comprehensive, but are simply provided as examples of the range of equations tested and employed over the past several decades. For a more thorough discussion, see [11].

2.2.3 AXBT Profile Quality Control

After the AXBT has finished transmitting, the received profile must be quality controlled to correct for noise, interference, and AXBT measurement errors. In TROPIC, this was previously accomplished with the System for At-Sea Environmental Analysis (SASEA; [13]). SASEA provides an interface for analysis and quality control from a range of ship- and air-launched probes. Options for XBT and AXBT

profile quality control include profile smoothing, surface correction, and individual point addition and removal. Although fully automated processing systems are known to exist, these systems are not documented in published literature (S. Paul, NOAA AOC, 2020, personal communication).

Various organizations have developed guidelines for XBT/AXBT quality control routines. For example, NOAA's Atlantic Oceanographic and Meteorological Laboratory (AOML) references a quality control cookbook for XBTs developed by Australia's Commonwealth Scientific and Industrial Research Organisation (CSIRO) XBT program [14]. This document includes guidelines for common ocean thermal structures and seasonal variability, quality control procedures to identify common profile discrepancies, and quality control flags to summarize those discrepancies. A similar (though much older) manual published by the Naval Oceanographic Office provided guidance for XBT quality control for the US Navy [15].

These quality control guidelines and routines all detail basic requisite background oceanographic knowledge, including an understanding of typical thermal structures for the region and season in which the XBT/AXBT was launched. Additionally, these routines detail common discrepancies and failure modes, as well as a series of quality control flags used to categorize the quality of any XBT/AXBT profile. Together, these highlight the potential for an advanced XBT/AXBT quality control interface. Such an interface could automatically apply checks and corrections for common profile failure modes, provide an interface for users to verify profile accuracy and apply additional edits if necessary, integrate global climatology and bathymetry data to guide both automated and user-driven profile quality control, and include inputs for users to select from a range of quality control flags to categorize profile accuracy in a standardized manner.

Chapter 3

ARES Data Acquisition System

The AXBT Realtime Editing System (ARES) Data Acquisition System is designed to receive digital pulse code modulated (PCM) audio data containing an AXBT signal and produce a viable temperature-depth profile from that data. ARES is compatible with WiNRADIO software-defined radio receivers, which demodulate a VHF signal transmitted from an AXBT and exporting the resulting PCM data to ARES for processing. Additionally, previously recorded WAV files can be imported into ARES to generate a temperature-depth profile. ARES integrates the signal processing capabilities of the MK-21 or similar hardware and audio recorders as software-defined functions, reducing the equipment necessary to launch and process data from AXBTs. The remainder of this chapter is organized into three sections: Section 3.1 describes the ARES Data Acquisition System interface, Section 3.2 discusses necessary hardware setup, data flow, signal processing details, and additional features of the Data Acquisition System, and Section 3.3 evaluates the capabilities of the Data Acquisition System using raw (PCM) audio data from profiles launched during the TROPIC field program from 2011-2019.

3.1   Data Acquisition System Interface Description

3.1.1   Layout

An ARES Data Acquisition System graphical user interface (GUI) tab is opened by default when starting ARES. Additional Data Acquisition System tabs can be opened by either selection *Options > New Data Acquisition System Tab* or using the keybinding ctrl+N. The Data Acquisition System window is divided into three sections (Fig. 3-1):

1. Raw, realtime temperature-depth profile plot (left)

2. Tabular temperature-depth profile and signal information (lower right)

3. Datasource, VHF, and profile metadata configurations (upper right)
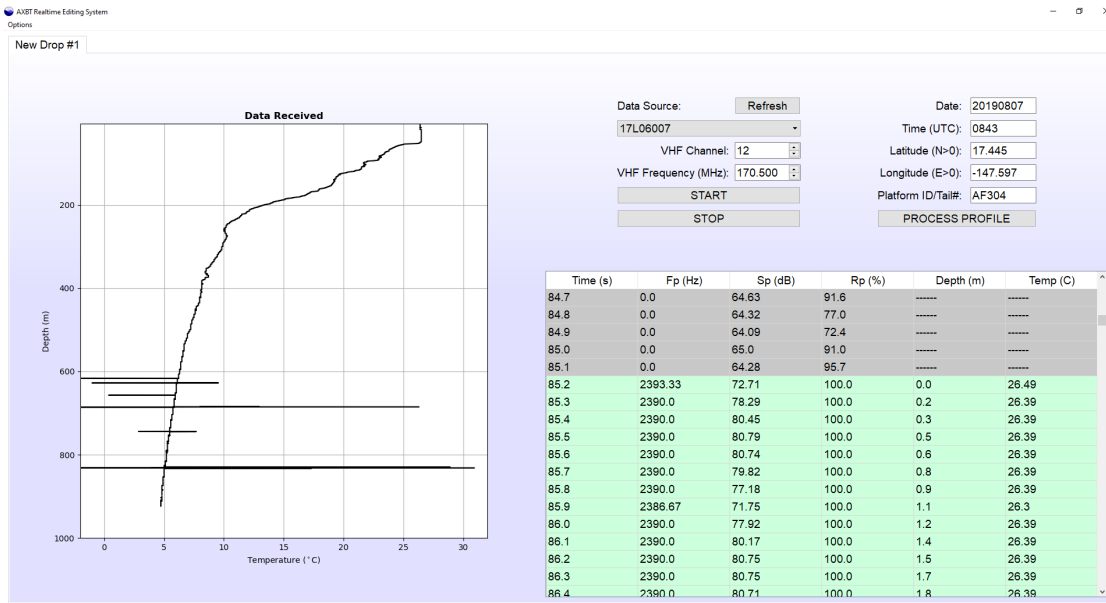
Figure 3-1: The ARES Data Acquisition System graphical user interface.

The upper-right corner of the GUI provides options for the user to control the Data Acquisition System, and includes two columns with user inputs. At the top of the left column, a dropdown menu allows the user to select the data source from which AXBT data is being processed. The datasource selection dropdown menu

automatically has two default options: "Test", which processes a test signal (from a raw audio file distributed with ARES), or "Audio", which prompts the user to select a raw WAV file to be reprocessed. Additionally, on Windows systems with the WiNRADIO drivers installed, any connected WiNRADIO receivers will be listed by serial number in order to process data in realtime. Because each receiver can only demodulate one channel at a time, ARES prevents the user (with an accompanying warning message) from selecting the serial number for a WiNRADIO that is actively receiving in a different tab.

If a connected WiNRADIO receiver is selected, the two spinboxes immediately below the datasource selection menu (labelled "VHF Channel" and "VHF Frequency") allow the user to select the VHF frequency that the receiver demodulates, either directly by frequencies or by selecting the corresponding sonobuoy VHF channel (selecting a new frequency will update the selected channel, and vice versa). The VHF channel and frequency being demodulated can be changed while a receiver is actively processing data.

Buttons for the user to start and stop data acquisition are located at the bottom of the left column. Data acquisition should be started when the profile is launched, to ensure that the recorded date, time, and (as applicable) position are representative of the moment the AXBT was launched. Data acquisition can be restarted after being stopped (except when reprocessing from audio files), and the time elapsed will be preserved from the initial start selection. However, for realtime processing, any data received while the Data Acquisition System was stopped is lost. After processing has been initiated (by selecting start), the user cannot switch between Test/Audio/Receiver data sources. The selected receiver can be changed if multiple receivers are connected, but to do this the user must stop processing, select a different receiver, and then resume processing.

While actively processing, the table in the lower-left section of the screen is updated in realtime with incoming data from an AXBT. The table's columns are (from right to length): time (seconds), frequency (Hz), signal level (dB), signal ratio (%), depth (meters) and temperature (^{o}C). The time and frequency fields list the time

elapsed since "Start" was selected and the peak frequency of the signal currently being received. If a signal's peak frequency is in the expected range corresponding to realistic temperature values (defined by user settings, default 1300-2800 Hz) and the observed signal level and signal-to-noise ratio (SNR; these values are displayed in the S_P and R_P columns in the table, respectively and described in more detail in Section 3.2.2) meet the required minimum thresholds (which are defined in the user settings as described in the next section), then the temperature and depth are calculated and appended to the profile plot, and the corresponding row in the table will appear green. If the minimum signal level and SNR thresholds are not met, temperature and depth are not calculated or appended to the profile plot, and the corresponding row in the table will appear gray.

The right column of text provides inputs to record AXBT drop information: date, time, latitude, longitude, and platform identifier (e.g. aircraft tail number). For realtime drops (when the selected datasource is a WiNRADIO serial number), the date, time will be autopopulated from the system date and time, and the position will be populated from a connected GPS receiver (if configured in ARES preferences, as discussed in the following section). This autopopulation option is configurable in the ARES settings. After processing AXBT data and entering the corresponding drop information, selecting "Process Profile" will first prompt the user to save the raw profile data (this is configurable in ARES settings) before loading the raw profile into a Profile Editor tab (discussed in Chapter 4).

3.1.2   Settings

Data Acquisition System Settings

The Data Acquisition System settings are divided into four sections: autopopulation options, file save formats, Data Acquisition System thresholds and windows, and additional settings. The autopopulation options (top left) control whether ARES automatically updates the recorded drop information (in the right column in the Data Acquisition System tab) whenever "Start" is selected for the first time in a tab. Au-

topopulation is not used when data is being reprocessed from a raw audio file (when "Audio" is the selected data source). Four file save formats (discussed in Section 3.2.3) are also selectable. Below these options are three additional user settings that control user prompts that appear when attempting to save the raw data and transition to a Profile Editor mode.
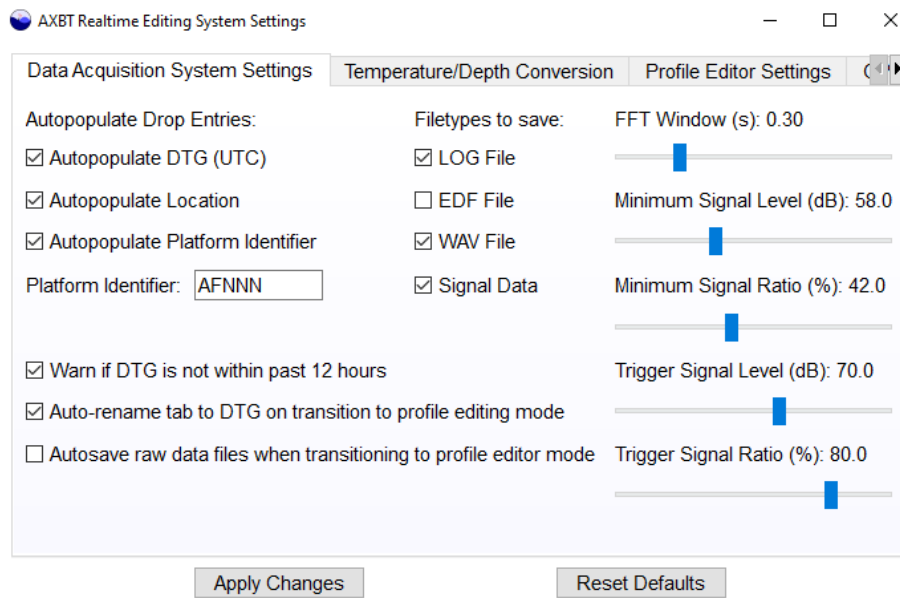
Figure 3-2: The ARES Data Acquisition System settings window.

The rightmost column of settings control the windows and thresholds used to process AXBT data. The "FFT Window" is the length of each window of data (in seconds) used to determine peak frequency and corresponding temperature. Increasing this window increases the frequency and temperature resolution (see Δf, Eq. 3.2 in Section 3.2.2) and reduces the effects of transient interference and noise. However, increasing the window also increases computational expense and slows the program, reducing the rate at which data can be processed (this effect is compounded when processing multiple profiles simultaneously) and decreasing the raw profile's vertical resolution. The minimum signal level and signal ratio are thresholds used to distinguish valid AXBT data from noise: increasing the thresholds by moving the sliders to the right makes the Data Acquisition System more selective, reducing the amount of data considered valid. These minimum thresholds are applied to the values listed in

37

the S_P and R_P (respectively) columns of the table (discussed in more detail in Section 3.2.2). Finally, the trigger signal level and ratio are additional constraints applied to to the minimum signal level and ratio to identify the first datapoint transmitted from an AXBT and trigger the start of the temperature-depth profile. Because AXBT signals typically begin stronger and weaken with depth as the aircraft generally moves away from the buoy (see Section 3.3), using a higher constraint here can help to prevent weak VHF interference from prematurely triggering data collection.

Temperature/Depth Conversion Equation Settings

ARES includes the ability to use customized linear, quadratic, or cubic conversion equations for AXBTs (Fig. 3-3). Two columns are provided: one for frequency-to-temperature and the other for time-to-depth conversions. To adjust the conversion equations being used, enter the coefficient for each term in its respective box. The equation at the top of the respective column should update to reflect the coefficients in the boxes. If non-numeric characters are entered, the equation will not update. The equations used by ARES will not be updated until "Apply Changes" is selected. Additionally, if the contents of the input boxes do not match their respective equations (e.g. due to a non-numeric character as described above), the coefficients used by ARES are based on the equations at the top of each column, not the contents of the input boxes.

Finally, at the bottom of the tab there are two additional inputs to select the valid frequency range. The default range of frequencies ARES accepts as valid AXBT signal is 1300 Hz to 2800 Hz, corresponding to approximately -3.9^{o}C to 37°C. However, because ARES may be used in locations where only a subset of these temperatures are expected (e.g. an AXBT launched in the Southern Ocean during austral winter would most likely not observe 20°C ocean temperatures), this input can be used to apply more precise constraints on the peak frequencies (and therefore temperatures) that ARES accepts as valid observations.
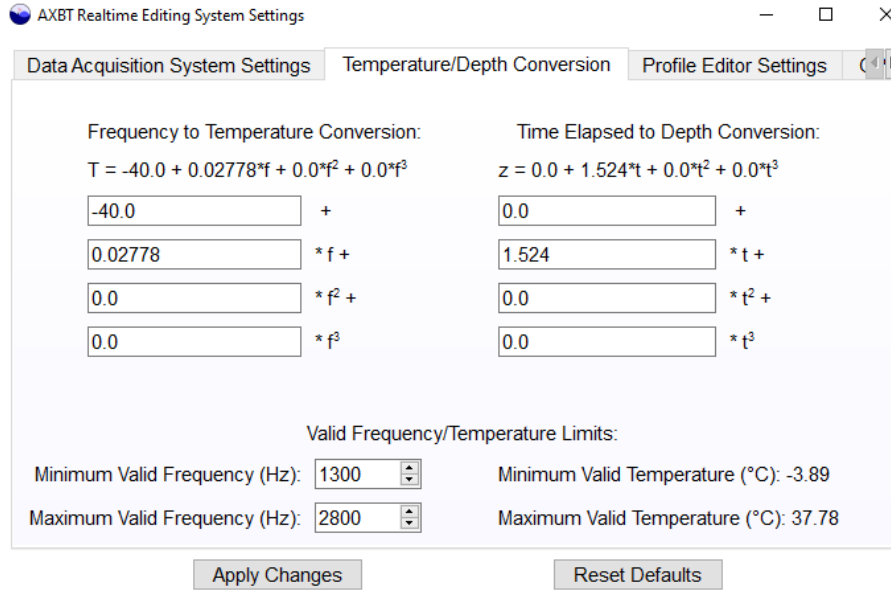
Figure 3-3: The ARES temperature/depth conversion equation settings window.

GPS Configuration Settings

The GPS settings tab allows the user to select a serial port for ARES to open and collect GPS data in order to autopopulate drop latitude and longitude when processing is started (more information on the GPS NMEA stream and serial configuration is provided in Section 3.2.3). The dropdown menu lists all available serial interfaces (it is worth reiterating that all available interfaces are listed, not just those that are transmitting valid GPS NMEA data streams). If multiple ports are available and the correct port is unknown, the user can try connecting to each port one at a time. If an active GPS receiver is connected to the selected port, this will populate the most recent GPS fix from that receiver. Otherwise, a warning message will appear stating that a GPS data stream could not be identified.

3.2   Procedural Overview

In order to be processed by ARES, VHF data received by an appropriate antenna are demodulated by a WiNRADIO receiver into pulse code modulated (PCM) data (collected at 64 kHz) before being transferred to the computer and appended to a
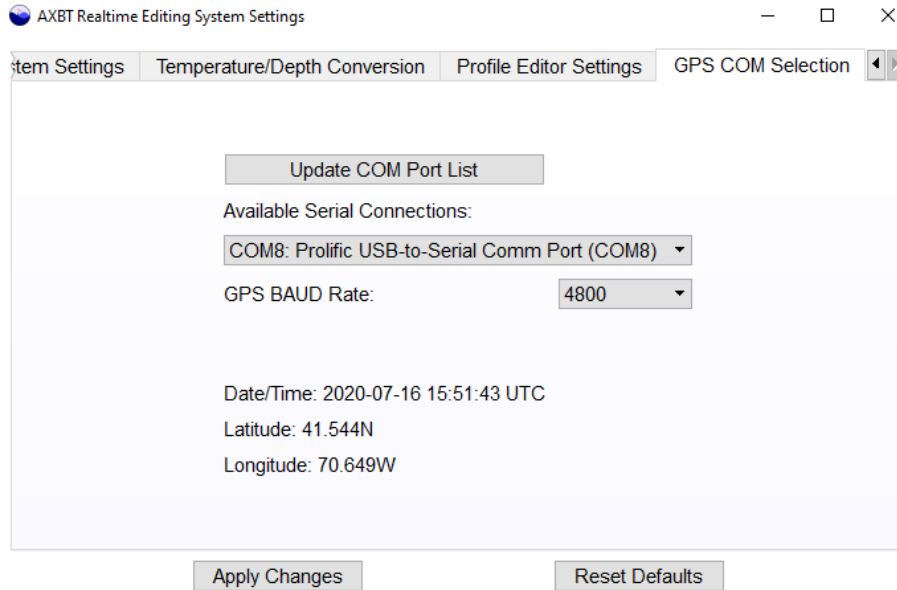
Figure 3-4: The ARES GPS configuration settings window.

length-conserving buffer containing one second of data (as new data are appended to the tail of the buffer, an equal number of points are removed from the head). All signal processing and subsequent operations are carried out on PCM data that has been appended to this signal processing thread. All operations prior to PCM data being appended to the Data Acquisition System thread are described in detail in Section 3.2.1, and all following computations are discussed in Section 3.2.2.

3.2.1 Radio Receiver Integration

External Hardware Configuration

The software-defined radio receiver currently integrated with ARES is a WiNRA-DIO WR-G39WSBe Sonobuoy Receiver (Fig. 3-5; `www.winradio.com/home/g39wsbe.htm`). It demodulates transmitted signals from standard sonobuoy VHF carrier frequencies between 136 and 173.5 MHz, and outputs analog (SMA) and digital (serial) demodulated data. A single receiver weighs 0.43 kg, measures 16.6 cm x 9.7 cm x 4.1 cm, and requires 12V direct current at 0.5A. The receiver has four connections: power (standard 2.5 mm jack), VHF in (SMA male), analog out (SMA male; unused in this application), and digital out (serial). A serial-to-USB converter for the digital

Figure 3-5: A WiNRADIO G39WSBE software-defined receiver (image from `www.winradio.com/home/g39wsbe.htm`). Ports on the front-left face of the receiver are: (top left) analog audio out (SMA), (top right) VHF in (SMA), (bottom left) power (2.5 mm), and (bottom right) digital audio out (serial).

output enables easy integration with most modern computers.

A single receiver can only demodulate one VHF frequency at a time, and thus only process good data from one AXBT at a time (although there is potential for an advanced system to track multiple peaks in spectral density simultaneously, this is extremely prone to errors and thus limiting operations to one AXBT per channel at a time is optimal). The time from launch it takes a single AXBT to fully transmit (approximately 11.5 minutes from 10,000 feet, including 120 seconds fall time and an approximately 1.5 ms^{-1} descent rate to 850 m), constrains the spatial resolution at which AXBTs can be launched as a function of aircraft ground speed. Thus, ARES was configured for the operation of up to six receivers in parallel (though this may be further limited by computer clock rate and random access memory).

An example of an array of WiNRADIO receivers is shown in Fig. 3-6. This array included three radio receivers that connect to a processing computer via a single USB connection with an unpowered USB hub (with a fourth port available for a GPS receiver). The included power adapter converts standard AC current to 2A 12V DC,
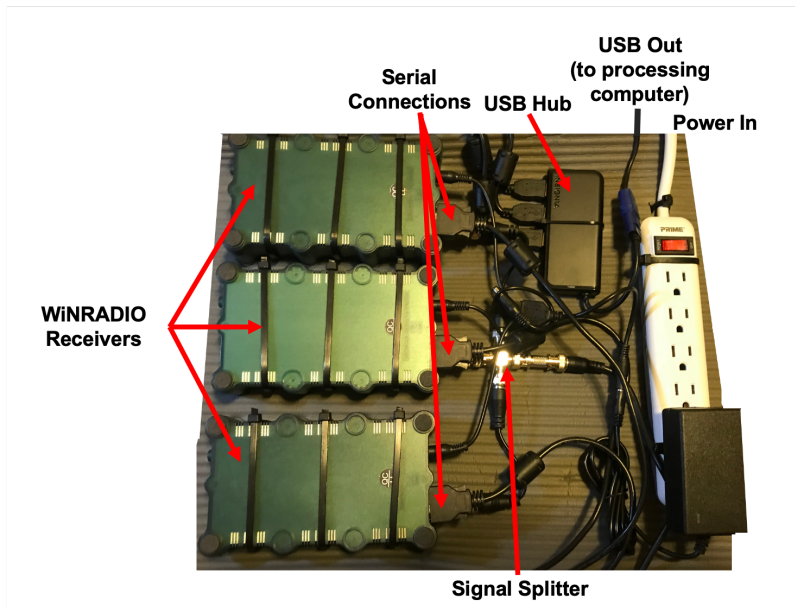
Figure 3-6: The WiNRADIO receiver array used in this study.

and thus a single adapter was used with a three-way splitter to provide power to all three WiNRADIOs. Additionally, a BNC splitter was used to connect a single VHF antenna output to all three receivers. SMA-to-BNC converters were used for the VHF input to connect the WiNRADIO's SMA input to this splitter and thus the aircraft's VHF antenna. In testing, no significant drop in signal level or corresponding processor performance was observed when using the BNC splitter. This configuration of three receivers enabled deployment of deep-water AXBTs (rated to 850m) every 3-4 minutes, or shallow-water AXBTs (rated to 400m) every 2 minutes.

Hardware systems accompanying ARES (following the design of the example array) require three connections: a standard 60Hz/110V input alternating current wall power supply, a BNC coaxial connection to a VHF antenna for RF signal input, and USB output connected to the processing computer running an instance of ARES. VHF signal received by the connected antenna are demodulated by the WiNRADIO receiver, which outputs stream of PCM data collected at 64 kHz from a serial port. This data is transferred to the processing computer using a serial to USB converter and appended to a buffer on the computer using the receiver's application-programming interface. This buffer is periodically accessed by the ARES Data Ac-
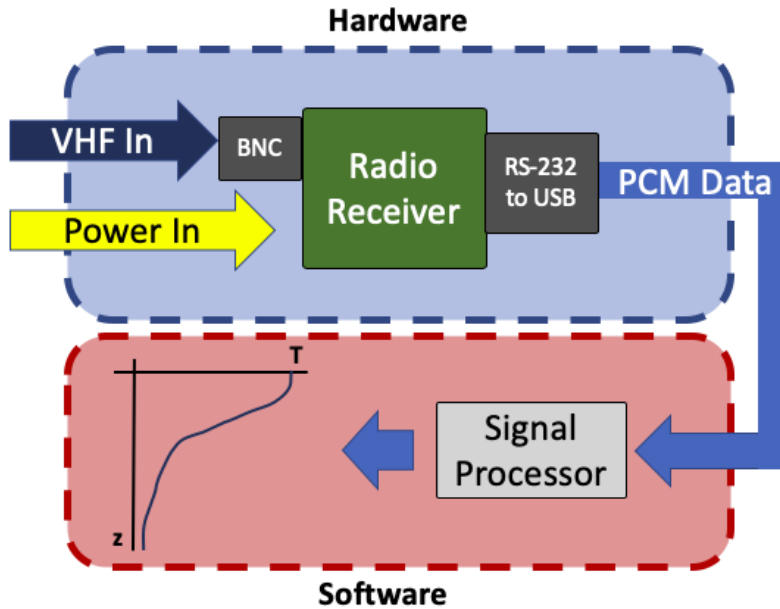
Figure 3-7: Process to demodulate a received AXBT VHF signal and generate the encoded temperature-depth profile.

quisition System interface to generate a temperature-depth profile from the received data stream (Fig. 3-7).

WiNRADIO API Interaction

In addition to the required proprietary system drivers (which currently exist for Windows only), a dynamic-link library (DLL) provides the application-programming interface (API) necessary for the processing computer to communicate with and control the WiNRADIO receivers. This was used in conjunction with the Python ctypes module [16] to handle all receiver communication and control exclusively in Python. This API includes a large range of functions for receiver communication and control (an exhaustive list is available at `www.winradio.com/home/g39wsb_sdk.htm`). While the exact API functions used by the data acquisition system would vary when integrated with a different brand of software-defined receiver, the format would be similar and all operations after demodulated PCM data is accessed by the processing computer would be identical.

The WiNRADIO API is accessed by Python using the ctypes module following the syntax in the example below:

```
1  wrdll = ctypes.windll.LoadLibrary('path-to-DLL-file')
```

Here, wrdll is a class with a library of nested API functions imported from the DLL file. This library includes a large range of functions for receiver communication and control (an exhaustive list is available at `http://winradio.com/home/g39wsb_sdk.htm`), but the primary functions used in ARES are listed in Table 3.1.

Table 3.1: WiNRADIO application-programming interface functions integrated in the ARES Data Acquisition System.

Function	Inputs	Output	Purpose
Open	char *SerialNum	int hRadio	Generate handle for a specific receiver using UTF-8 encoded serial
SetPower	int hRadio bool status	bool success	Set receiver power: true=on, false=off
InitializeDemodulator	int hRadio	bool success	Begin demodulating input VHF signal
SetFrequency	int hRadio dword frequency	bool success	Set VHF frequency to demodulate
SetVolume	int hRadio int volume	bool success	Set volume of output PCM data (0-31)
GetSignalStrengthdBm	int hRadio	int sigstrength	Approximate signal strength for current VHF frequency
SetupStreams	int hRadio void *callback	bool success	Direct the stream of demodulated PCM data to a user-written callback function
GetRadioList	Info *RadioList int BufferSize int *InfoSize	int total	Populate a list of structures with info about all connected radio recievers
CloseRadioDevice	int hRadio	bool success	Power off the receiver

Before passing input arguments to functions nested in the library, Python variables must be converted to C-type variables (e.g. 16-bit signed integer, unsigned long integer, character array pointer) using the ctypes module. After these conversions, executing commands from the DLL file is straightforward. An example is provided in Appendix C.3.1 demonstrating use of the API-defined function "Open" to access a connected receiver. This requires a pointer to a null-terminated, utf-8 encoded char-

acter array containing the serial number for the receiver to be activated, and returns an integer handle used as an input argument for all subsequent commands interacting with that receiver.

In order to control multiple receivers, ARES uses the API-defined function "GetRadioList", which provides information about all connected receivers. This function requires a preallocated array of c-type structures (described ind detail on the WiNRADIO website), with empty fields to be populated by the API. Implementation of this function in a Python environment is demonstrated in Appendix C.3.1.

To enable simultaneous processing of multiple AXBTs from multiple receivers in parallel, the ARES Data Acquisition System handles the data stream from each radio (and all associated computations) as individual threads, and uses PyQt's multithreading support to handle processing. Integration of multithreading is discussed further in Section 3.2.3, but it is worth noting here that each thread is a class with nested functions, as shown in Fig. 3-8. The `__init__()` function saves all input variables (e.g. Data Acquisition System settings) to the class, and either opens and initializes the selected radio receiver if processing data live, or opens and reads in PCM data from a selected WAV file if reprocessing from raw audio. The `run()` function starts selecting segments of PCM data either using the most recent data received from the WiNRADIO or sequential segments from a source audio file, and processing them as described in Section 3.2.2. The `changecurrentfrequency()` function (only applicable when processing in realtime from a receiver) changes the VHF channel that the receiver is demodulating while it is actively demodulating data. Finally, `changethresholds()` adjusts the Data Acquisition System settings if the user changes them while the current thread is active and `abort()` allows the user to terminate the current thread and finish processing.

A WiNRADIO API function, 'SetupStreams', is used to control PCM data transfer from the receiver to the processing computer. This assigns a callback function that is executed every time a buffer is filled (16-bit PCM data is collected at 64 kHz in a 2 KB buffer, so this occurs at approximately 30 Hz). In ARES, this callback function appends the data to both a WAV file and a buffer on the computer from which PCM
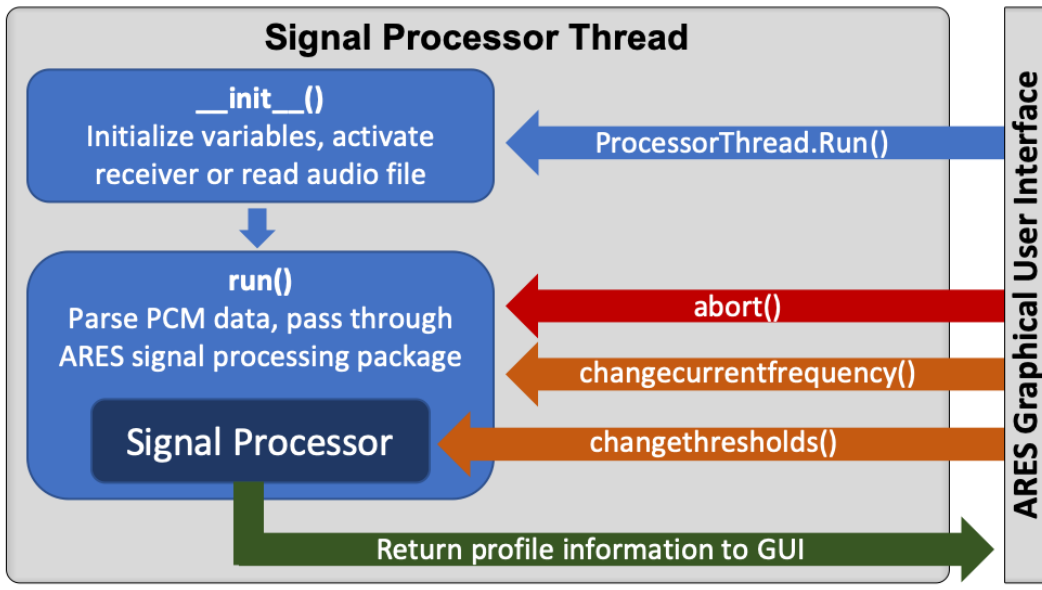
Figure 3-8: Diagram of nested functions in the ARES Data Acquisition System thread class, with functions listed on arrows using PyQt signals (arrows out of the Data Acquisition System thread block) and slots (arrows into the thread block) to pass information to and from the ARES user interface.

segments are pulled by the data acquisition system and analyzed, following Fig. 3-9. An example of the callback function declaration with SetupStreams is also provided in Appendix C.3.1.

3.2.2 Data Flow

Whether PCM data are read from a WAV file or accessed realtime from a connected VHF receiver, the subsequent signal processing sequence is identical. First peak frequency, signal level, and signal-to-noise ratio are determined using a discrete Fourier transform of each segment of tapered PCM data, and then these values are used to infer the temperature-depth profile observed by the AXBT (Fig. 3-9).

Signal Processing

Peak frequency, which is empirically related to the AXBT-observed ocean temperature, is determined by applying fast Fourier transforms to small subsets of data. Unless otherwise specified, a window length of 0.3 seconds is used (the effects of

46
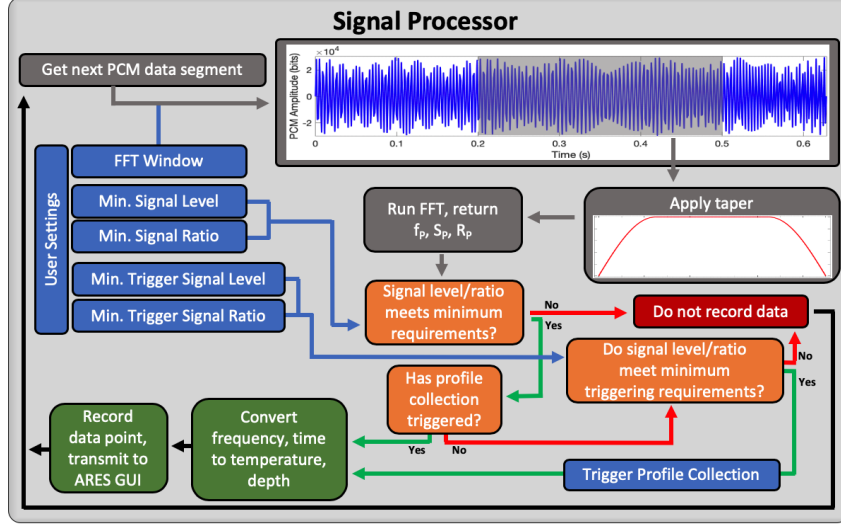
Figure 3-9: Sequence to pull and process segments of PCM data to identify valid temperature-depth measurements transmitted from an AXBT.

this window length are considered in the ARES Data Acquisition System evaluation section). Before a data subset is transformed into the frequency domain, a cosine (Tukey) taper is applied to the time series in order to minimize spectral leakage (Eq. 3.1). The taper window (T) is determined by the length of the PCM data subset (L) and a single parameter (α). The alpha parameter is the ratio of the tapered component of the time series to the total length of the time series, and is constrained by $0 \leq \alpha \leq 1$.

$$
T[i] = \begin{cases}
\frac{1}{2}\left[1 + cos\left(\frac{2\pi}{\alpha}\frac{i-1}{L-1} - \pi\right)\right], & i < \frac{\alpha}{2}(L-1) + 1 \\
\frac{1}{2}\left[1 + cos\left(\frac{2\pi}{\alpha} - \frac{2\pi}{\alpha}\frac{i-1}{L-1} - \pi\right)\right], & i > L - \frac{\alpha}{2}(L-1) \\
1, & \text{otherwise}
\end{cases} \tag{3.1}
$$

The frequency spectra for a given segment of PCM data are calculated using a fast Fourier transform (requiring only the PCM data subset and corresponding sampling frequency), using the `numpy.fft.fft()` function included in Python's Numpy module [17]. The transform of a discrete time series (s, of length L and sampling frequency f_s) into the frequency domain (S) is described in Eq. (3.2), where m and n are discrete indices corresponding to elements in the time and frequency domains, respectively.

47

$$S[n] = \sum_{m=0}^{L-1} s_m e^{-i2\pi mn/L},$$

$$n = 0, 1, 2, ..., \frac{L}{2},$$

$$\Delta f = \frac{f_s}{L}, \quad f = n\Delta f \tag{3.2}$$

Resulting frequency spectra (S) are collapsed into three characteristic parameters: the peak frequency (f_P) and the corresponding signal level (S_P) and signal-to-noise ratio $(R_P;$ defined mathematically in Eq. 3.3). Only signals from 1300-2800 Hz, corresponding to a realistic temperature range of -3.88^oC to 37.7^oC, are considered. The signal level depends on the maximum spectral value in the 1300-2800 Hz band (corresponding to the peak frequency that the AXBT is presumed to be transmitting), and the signal-to-noise ratio (SNR) is the ratio of that maximum spectral value in the 1300-2800 Hz band to the maximum value in the entire band. Thus, the SNR ranges between 0 and 1, where it is 0 if no nonzero values are measured in the 1300-2800 Hz band and 1 if the most powerful received signal is in the 1300-2800 Hz band. These parameters are used to distinguish AXBT signal from noise and to calculate the corresponding temperature if necessary. To be considered a valid AXBT signal, the signal level and SNR must both satisfy pre-set thresholds (appropriate thresholds are considered in the Data Acquisition System evaluation). Because possible maximum values of the signal spectra span several orders of magnitude, the peak signal level is expressed in decibels (dB re 1 bit, referenced in this study hereafter as dB).

$$K = \arg\max S_k, \quad f[k] \in [1300, 2800] \tag{3.3a}$$

$$f_P = f[K] \tag{3.3b}$$

$$S_P = 10\log_{10}\left(\frac{S[K]}{1 \text{ bit}}\right) \tag{3.3c}$$

$$R_P = \frac{S[K]}{\max S} \tag{3.3d}$$

Temperature-Depth Conversion

For signals that satisfy the minimum signal level, frequency range, and SNR requirements, temperature and depth are determined from the peak frequency and elapsed time between the current observation and first observation. ARES uses the general empirical relationship between measured temperature (^{o}C) and transmitted frequency (Hz) (Eq. 3.4; noting that the coefficients 1440 and 36 have units of Hz and $Hz\ ^{o}C^{-1}$, respectively) and standard fall rate, 1.52 ms^{-1} [6].

$$f_P = 1440 + 36T \tag{3.4}$$

Thus, given the peak frequency (f_P) and elapsed time (Δt) since the first observation (when it is presumed the probe is at the surface), the corresponding temperature (T) and depth (z, positive down) are calculated as:

$$T = -40 + (2.778x10^{-2})f \tag{3.5a}$$

$$z = 1.52\Delta t \tag{3.5b}$$

Determining time elapsed requires tracking when the first valid signal was recorded. Because AXBT signals are typically stronger when transmission begins and decay over time as distance between the float and the aircraft increases, in order to prevent VHF interference from prematurely triggering the profile, higher minimum signal level and signal-to-noise ratio thresholds are required for a signal to be accepted as the surface value. Thus, AXBT profiles are processed by repeating the six steps below at the data acquisition system's sampling frequency of approximately 10 Hz (this process is represented visually in Fig. 3-9):

1. A subset of PCM data is pulled from the input data stream and tapered.

2. The tapered PCM subset is transformed into the frequency domain.

3. Peak frequency, signal-to-noise ratio, and signal level are calculated.

4. If the signal level and SNR are above the minimum thresholds and profile collection has already been triggered, time elapsed is calculated and the corresponding temperature and depth values are determined and recorded.

5. If profile collection has not been triggered but the signal level and SNR satisfy the trigger thresholds, the observation time is recorded to determine future elapsed times, profile collection is triggered, and the surface temperature is calculated and recorded.

3.2.3 Additional Features

GPS Integration

In order to reduce potential for error by manual entry, ARES was designed to accept input from any GPS receiver that transfers information with a NMEA stream to automatically record position. This is accomplished using Python's pyserial and pynmea2 modules, which were developed to manage serial data connections and parse GPS NMEA-formatted data, respectively. Using pyserial, ARES scans for and provides a list of all available serial ports. Given a user-specified port, ARES can open the port, parse incoming NMEA data and record latitude and longitude (as long as the selected port is connected to a GPS with a valid NMEA stream), either when data recording is started or to test the connection to a GPS receiver. Examples of using pyserial to retrieve a list of available serial ports (potential GPS receiver connections) and parsing NMEA feed information including time and position from a connected serial port are provided in Appendix C.3.1.

File Types

Raw data from a processed profile can be saved in one of five formats: raw audio (WAV), ASCII signal data, LOG, EDF, and NVO files. WAV files are the most raw form of data available, and recording WAV files is advantageous as it enables reprocessing the profile using different signal processing schemes. However, while some filesystems record creation and "last modified" datetimes, WAV files cannot

store drop metadata such as time and position. In ARES, temporary WAV files are generated when a receiver is activated, and PCM data is constantly appended to the file as the WiNRADIO continues passing data to the processing computer.

The ASCII signal data file records time elapsed, peak frequency, signal level, and signal-to-noise ratios in a comma-delimited format (in that order). This enables profile testing and adjustments (e.g. changes to conversion equations or minimum signal thresholds) without repeating the computationally-expensive signal processing necessary to generate a time series of peak frequencies and accompanying signal levels from a source audio file. Like the WAV file, temporary signal data files are created when a receiver is activated and continuously updated as more data is streamed from the receiver. For both the raw audio and signal data files, when the user "saves" the file, they are simply renaming and moving those files from a temporary directory managed by the processing computer's operating system (e.g. Wondershare for Windows 10).

Finally, the LOG, EDF, and NVO files are all various ASCII file formats for storing the raw data. LOG files are columnar ASCII files including time elapsed, depth, peak frequency, and temperature, but do not include position information. EDF and NVO files include the observed temperature-depth profile as well as time and position information, and the option for additional metadata (e.g. drop number, AXBT type, and additional information). An NVO file format example is provided in Section 4.1.5. When these files are saved by the users, text files are actually generated from the raw profile information stored within the program's primary class.

Multithreading

The AXBT Realtime Editing System uses multithreading to facilitate processing data from multiple AXBTs simultaneously (either in realtime or from raw audio files). Multithreading and multiprocessing are two methods for handling parallel computations. Multiprocessing requires a central processing unit (CPU) that has multiple cores, each of which are able to run computations simultaneously and separately from one another. Multithreading only requires one core, but instead manages multiple threads (individual tasks) that operate simultaneously from the user's perspective

but alternate CPU use at high frequency so each thread receives a requisite amount of computation time from the CPU.

The PyQt5 library includes multithreading support through several classes: QThread-Pool, QRunnable, pyqtSignal and pyqtSlot. The QThreadPool class is initialized with the program to provide a parent to manage a limited number of individual threads (ARES currently supports up to six threads) running simultaneously in the background. The QRunnable class is the framework for the individual threads (e.g. in Fig. 3-8 the "Signal Processor thread" is a QRunnable class). This class contains an initialization function that is executed when the QRunnable class is created and a run function that is called separately by the parent thread after initializing the QRunnableClass (in ARES, these functions are executed consecutively when the user selects the option to start processing on the graphical user interface). Finally, pyqtSignal and pyqtSlot are the PyQt implementation of signals and slots, which enable transfer of information and commands between individual threads and the parent process. In ARES, this transfer is between the user interface and individual Data Acquisition System threads. For example, ARES supports the user changing the VHF frequency being demodulated and Data Acquisition System minimum thresholds while the Data Acquisition System thread is active, and this is accomplished using signals and slots (labelled changethresholds in Fig. 3-8). Additionally, every time a new segment of PCM data is processed and either converted into a temperature-depth datapoint or identified as a null value, that information is transmitted via a signal back to the parent thread (the user interface), where it is stored with the other profile information and used to update the profile plot and table.

3.3 Processor Evaluation

3.3.1 Methodology

To evaluate the accuracy of the ARES signal processing package, 531 temperature-depth profiles collected with the MOOS system from 2011-2019 during the TROPIC

field campaign were reprocessed from archived raw audio (.wav) files. These files contain the same format of PCM audio data as the demodulated AXBT signal that is output to the processing computer from a VHF radio receiver. Of these 531 recorded profiles, 459 profiles met minimum signal level requirements for reprocessing and 389 of those 459 profiles had corresponding ASCII files with temperature-depth profiles collected via MK-21, enabling a direct comparison between ARES Data Acquisition System and MK-21 output. Given the MK-21's wide use in scientific and military applications, it can be considered an industry standard for AXBT processing. Thus, in this study the profiles processed with the MK-21 are considered to be a baseline against which the accuracy of the ARES Data Acquisition System is evaluated.

This source audio files were reprocessed at 5 Hz using windowed segments of PCM data 0.3 seconds in length (corresponding to 13,230 datapoints given a 44.1 kHz sampling frequency for all files), resulting in a 33% overlap in PCM data with each adjacent observation. Segments were tapered with a cosine taper ($\alpha = 0.5$) before being Fourier transformed into the frequency domain. The resulting profiles were then examined to optimize the minimum signal level and signal-to-noise ratio thresholds applied, determine the accuracy of collected temperature-depth profiles, and consider the effects of taper and window size on the resulting profiles.

3.3.2 Minimum thresholds for S_P and R_P

Because profiles were originally obtained with a mix of shallow- and deep-water AXBTs from multiple manufacturers and processed with multiple receivers using varying settings, the peak signal levels of the associated raw audio files also vary greatly (corresponding to peak signals spanning several orders of magnitude; Fig. 3-10a). To standardize the raw data analyzed, profiles with a peak signal level of less than 75 dB were excluded, leaving 459 profiles for additional analysis.

To determine optimal signal level and signal-to-noise ratio thresholds for reprocessing of the remaining 459 profiles, distributions of observed signal levels (Fig. 3-10b) and signal-to-noise ratios (Fig. 3-10c) were examined. The signal level distribution can be divided into three distributions of signal levels: those greater than 70 dB, those
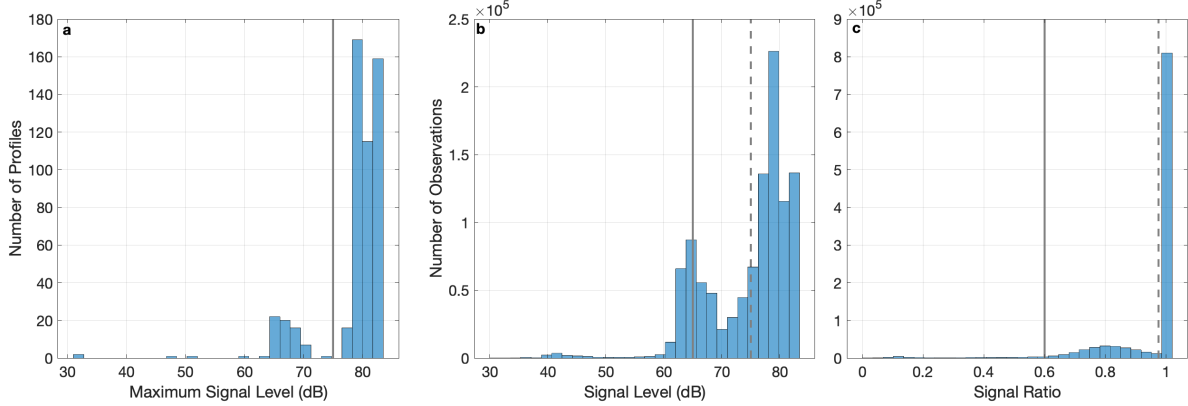
Figure 3-10: (a) Distribution of maximum signal levels (dB) for the 531 reprocessed profiles. The vertical gray line denotes the required minimum peak signal level for all profiles analyzed ($S_{P_{MAX}} = 75$ dB). (b,c) Distributions of (b) signal levels (dB) and (c) signal-to-noise ratios for all observations in the 459 profiles whose maximum signal level exceeds the threshold in (a). Vertical lines represent minimum signal level and ratio thresholds (solid and dashed lines correspond to minimum and trigger thresholds, respectively) applied when evaluating Data Acquisition System performance.

between 60 and 70 dB, and those less than 60 dB. The first two distributions are most likely dominated by valid AXBT signal whereas the third distribution consists primarily of noise. The bimodal nature of the two greater distributions highlights the remaining variability in signal levels, due to some combination of variability among AXBTs, residual differences in processing configurations, and ambient conditions (including aircraft altitude, distance from float, and atmospheric conditions).

Comparably, signal-to-noise ratios are separated into three distinct categories. The first (and by far the largest) category consists of SNR values equal to unity, which occurs when the dominant signal received by the aircraft is within the expected frequency for AXBTs (1300-2800 Hz). A second peak ranging from 0.6 to 0.9 consists of some combination of signals and noise. In this case, the greatest signal level is observed at a frequency outside of the expected range for AXBTs but a signal of approximately the same order of magnitude is observed within the AXBT frequency band. Finally, a third distribution below 0.5 consists of primarily electronic noise. In this study it is assumed that the first distributions of signal level and SNR observations consist almost exclusively of good data, the second distributions are composed of a mix of signal and noise, and the third distributions represent noise.

54
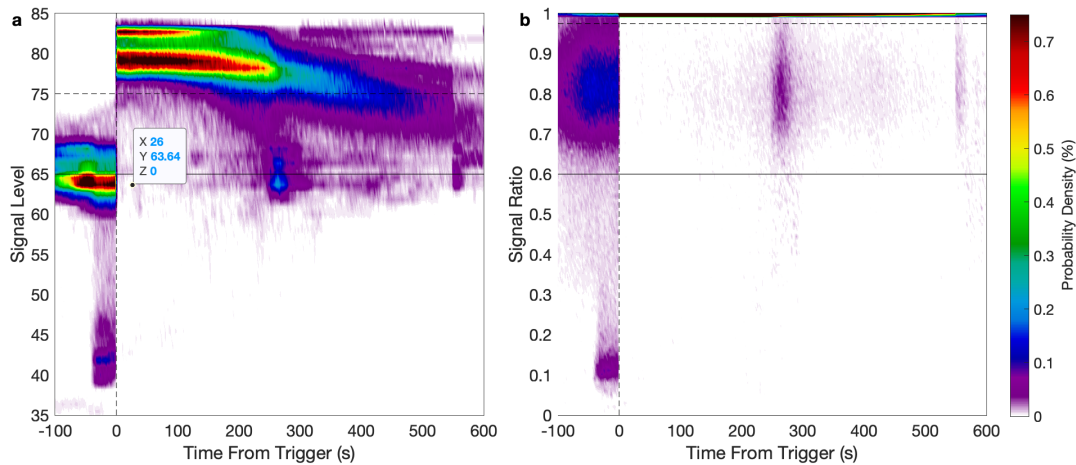
Figure 3-11: Distributions of observations for all observation times versus (a) signal levels and (b) signal-to-noise ratios. Profiles were standardized so zero seconds (vertical lines) denotes the first observation that satisfied the minimum triggering thresholds. Triggering and general thresholds are denoted by horizontal dashed and solid lines, respectively.

Additional context for optimal trigger and minimum thresholds is provided in Fig. 3-11. Almost all signal levels remain below 75 dB in the absence of a signal when profile collection is triggered, jumping above that threshold upon AXBT signal initiation. However, SNR values span almost the full zero to unity range prior to profile initiation, nearly unanimously reaching unity upon identification of valid AXBT signal. Based on this information, to optimize inclusion of good data and exclusion of noise, the minimum signal level and SNR for observations to be considered good data (Fig. 3-10, solid lines) were set as 65 dB and 0.6 respectively, and the minimum trigger signal level and SNR necessary to begin profile collection were set as 75 dB and 0.975 respectively.

3.3.3 Profile Accuracy

A comparison of MK-21- and ARES-processed temperature-depth profiles suggests that the ARES signal processing module performs comparably to the MK-21 (Fig. 3-12). The 2-m binned temperatures exhibit a nearly one-to-one relationship (note the agreement between solid and dashed lines, and $r^2 = 0.9977$), with two exceptions.
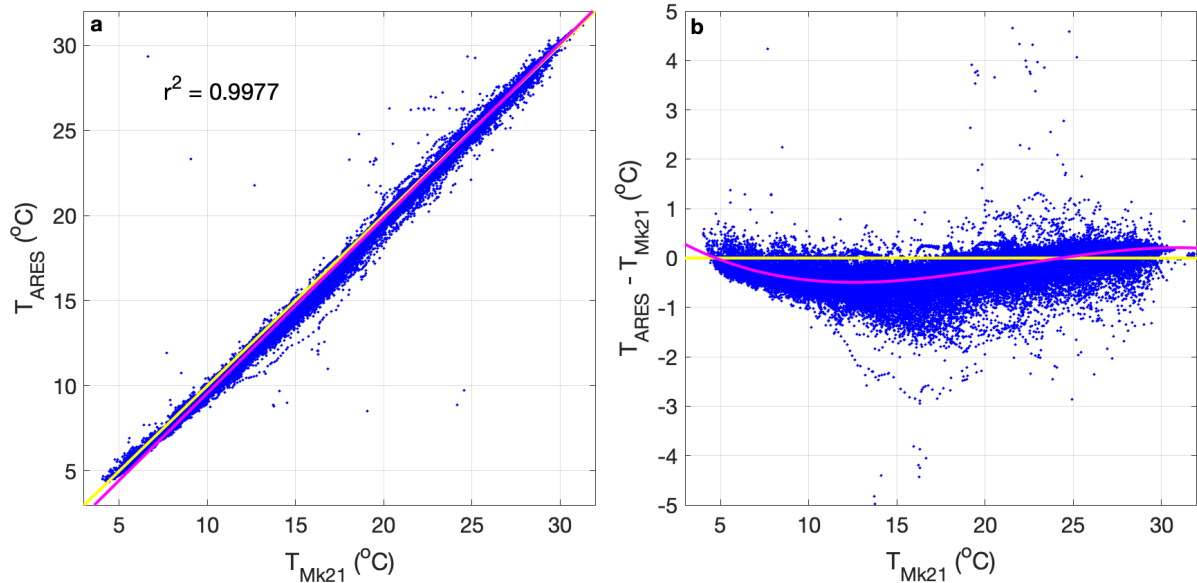
Figure 3-12: Comparison of 2m bin-averaged temperatures processed by the MK-21 and by ARES. Yellow lines denote a 1:1 relationship and magenta lines represent best fits for the data.

First, a small number of points more than 1°C from the 1:1 line along either axis are likely due to noise that was not filtered by either the ARES or MK-21 signal processing systems. Second, there is a systematic shift towards negative residuals (ARES observations are warmer than corresponding MK-21 observations) for temperatures between approximately 7°C and 23°C and towards positive residuals (MK-21 observations are warmer than corresponding ARES observations) for temperatures below 7°C and above 23°C (Fig. 3-12b). This shift is likely due to differences in the conversion equations (notably the probe fall rate equation) used in each scheme. ARES currently uses the standard Navy linear fall rate equation [6], whereas the MK-21 uses a higher-order, proprietary fall rate equation. It should be noted that the purpose of this analysis is to evaluate the signal processing scheme (e.g. tapering, transforming, and excluding noise) rather than to compare effectiveness of different AXBT temperature and depth conversion equations, which have been studied extensively in previous literature (e.g. [5, 6, 18]). Substituting different conversion equations in the source code is trivial and accomplishable as a user-defined setting.

To quantify which processor better removed interference from temperature-depth
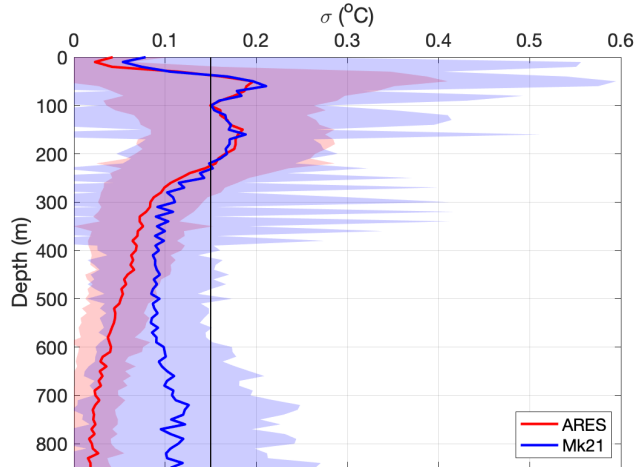
Figure 3-13: Average 10m standard deviations for temperature-depth profiles processed by the MK-21 and by ARES (shading denotes one standard deviation from the mean deviations). The solid black line denotes rated temperature accuracy for Sippican AXBTs [2].

profiles, standard deviation profiles were calculated by subdividing each profile into 10-m bins, determining the standard deviation for each bin, and comparing for MK-21- and ARES-processed profiles (Fig. 3-13). Because temperature changes due to noise $(O(1-10)^oC)$ are much greater than typical valid temperature changes on scales of 10 m or less, these binned standard deviations for a profile should be indicative of the "spikiness" due to interference projected onto the profile within each 10 m range. Lower standard deviations indicate less variability due to noise that was projected onto the raw temperature-depth profile by the Data Acquisition System.

ARES- and MK-21-processed profiles exhibited similar 10-m standard deviations from approximately 30 m to 200 m. Both profiles have local maxima near the thermocline, where ocean temperatures can change drastically on a scale of 10 m or less. However, ARES-processed standard deviations are lower both at the surface (in the upper 30 m) and at depths greater than approximately 200 m. An increase in interference at depth is likely due to a decrease in AXBT signal strength as the receiving aircraft translates away from the float, decreasing the signal-to-noise ratio. Hence, the application of a minimum signal ratio threshold in ARES likely accounts for its better performance below 200 m.

3.3.4 Tapering and FFT Window Optimization

Adjusting the window size (W) of the discrete Fourier transform results in a tradeoff between computational expense and temperature resolution (ΔT, Eq. 3.6). Specifically, as the window size is increased, the window length (L) increases and the frequency step size (Δf, in Hz) in the transformed spectra (Eq. 3.2) decreases, which also decreases the corresponding temperature step size (through Eq. 3.4). This is because the sampling frequency (f_s), which relates window length (L, number of PCM datapoints) to both window size (W, in seconds) and frequency step, cancels out (Eq. 3.6; where the coefficient 36 has units $Hz\ ^oC^{-1}$).

$$
\begin{aligned}
\Delta T = \frac{\Delta f}{36} &= \left(\frac{1}{36}\right)\left(\frac{f_s}{L}\right) = \left(\frac{1}{36}\right)\left(\frac{f_s}{f_s W}\right) \\
&= \frac{1}{36W}
\end{aligned}
\tag{3.6}
$$

This corresponds to a temperature resolution of 0.093^oC for the standard window size of 0.3 seconds (Table 3.2). To further evaluate the resolution versus computational efficiency tradeoff, ten raw audio files were reprocessed using four different window sizes (0.1, 0.2, 0.3, and 0.5 seconds) both with and without a cosine taper. Performance effects of window size are shown in Table 3.2, with all performance times and standard deviations expressed as ratios to the comparable performance time using a 0.3-second window and no taper. Using window sizes of 0.1, 0.2 and 0.5 seconds, ARES processed files in 40%, 70%, and 285% the amount of time it took to process an identical file with a 0.3-second window (respectively). However, the corresponding temperature resolution increased from 0.28^oC for a 0.1 second window to 0.006^oC for a 0.5-second window. Based on these results, 0.3 seconds remains a suitable window size as it provides a temperature resolution much less than AXBT instrument error [6] while enabling ARES to process multiple profiles simultaneously without consuming all of the processing computer's resources.

The negligible effect of adding a cosine taper on processing time is also apparent in Table 3.2. The mean residual between untapered and tapered temperatures was

Table 3.2: Audio file processing time means and standard deviations, and corresponding temperature resolution, for several combinations of FFT windows and taper use. Processing time means and standard deviations (comma-separated) are expressed as the ratio for a given processing time to the corresponding time for the same file using no taper and a window of 0.3 seconds.

FFT Window (s)	Resolution (^oC)	No Taper	With Taper
0.1	0.28	0.40, 0.06	0.39, 0.07
0.2	0.14	0.70, 0.10	0.71, 0.11
0.3	0.09	1, 0	1.0, 0.03
0.5	0.06	2.85, 0.19	2.91, 0.11

0.02^oC, with the distribution ranging from $0.003\text{-}0.1^oC$ (not shown). Additionally, mean 10-m temperature standard deviations (the same metric presented in Fig. 3-13) are 0.114^oC and 0.116^oC for tapered and non-tapered profiles. Despite the minimal effect of tapering on mean statistics, the potential for reduced interference on a case-by-case basis combined with the minimal associated increase in computational expense are sufficient to justify retaining Tukey taper application as a component of the signal processing data flow.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

ARES Profile Editing System

The AXBT Realtime Editing System (ARES) Profile Editing System enables users to quality control AXBT temperature-depth profiles, guided by an automated quality control algorithm and further aided by climatology and bathymetry data for the profile location to reduce the background oceanographic knowledge necessary on the part of the user.

The remainder of this chapter is organized into three sections: Section 4.1 describes the ARES Profile Editor interface, Section 4.2 describes the capabilities of the automated quality control "autoQC" algorithm, and Section 4.3 evaluates the performance of the Profile Editing System using AXBT profiles collected and originally quality-controlled during the TROPIC field program from 2011-2019.

4.1 Profile Editor Interface Description

4.1.1 Layout

The profile editor interface is divided into three sections (Fig. 4-1):

1. Interactive profile temperature-depth plot (left)

2. Static profile location and bathymetry plot (bottom right)

3. Profile editing tools (top right)
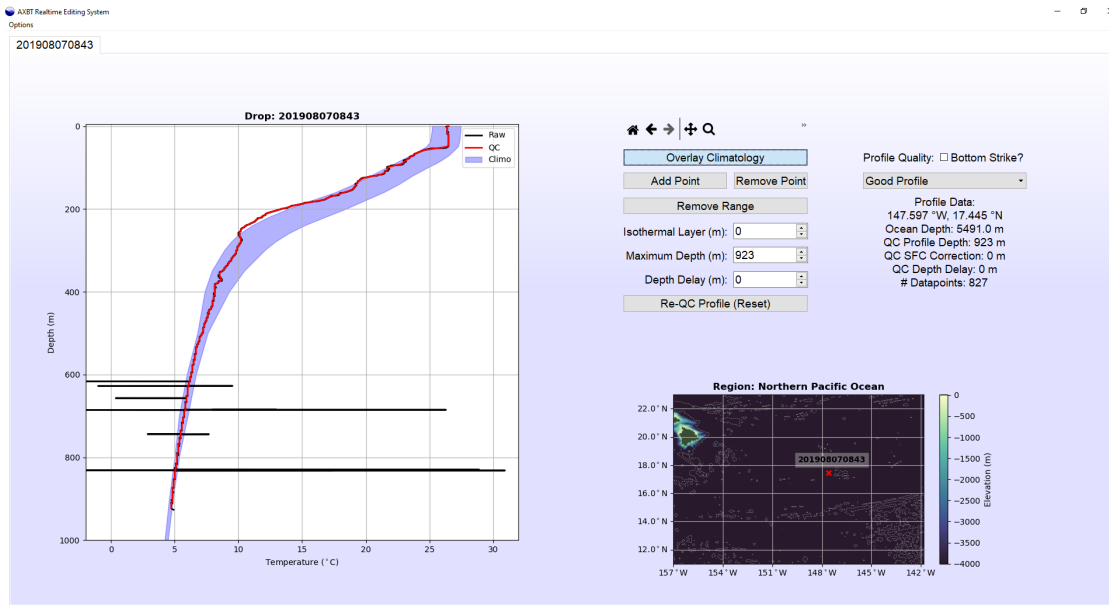
Figure 4-1: The ARES profile editing graphical user interface.

The profile temperature-depth plot overlays three profiles: raw data (black), quality-controlled data (red), and a climatological profile (using the Generalized Digital Ocean Model climatology; [19]) based on the AXBT's position and month launched, shaded for uncertainty (blue). All user edits are made to the quality-controlled (red) temperature-depth profile, which can be exported to several output file formats by ARES once all necessary corrections have been applied. When the profile editor interface is first loaded, the quality-controlled profile is generated using an automated quality control ("autoQC") algorithm that corrects for common modes of VHF interference and compares the data to climatology and bathymetry (discussed further in Section 4.2). The location plot overlays the AXBT's recorded position with contoured bathymetric data (1 arcminute resolution data from the NOAA ETOPO1 global relief dataset; [20]), to provide the user with additional information about the region in which the AXBT was launched and the surrounding bathymetry. Color contours are applied over the range 0 m to 4000 m using the cmocean "topo" colormap, and bathymetric features deeper than 4000 m are represented with dashed white contours at a 500m interval.

The top right section of the profile editor interface includes the tools necessary

to manipulate and edit the temperature-depth profile. Along the top of the plot is a toolbar that includes controls to pan and zoom around the temperature-depth plot in order to inspect and edit the profile in greater detail. It is worth noting that the controls in this toolbar do not edit the profile, and only change the user's view (for example, the back and forward arrows do not undo and redo edits to the profile, rather they allow the user to move back and forward to the previous and next views of the profile, respectively). The remaining interactive widgets are organized into two columns. The "Overlay Climatology" button in the left column allows the user to toggle the visibility of the blue climatology shading. All other buttons and spinboxes in the left column allow the user to edit the quality-controlled (red) profile (each of these corrections are discussed in detail in Section 4.1.3). Finally, the right column includes profile information. The text box includes data such as exact position, ocean depth at that position, and the number of datapoints in the quality-controlled profile. The dropdown menu provides a list of common profile discrepancies that can be used to flag a profile with uncorrectable features (such as an isothermal profile due to a probe that failed to deploy properly). These quality control flags are further discussed in Section 4.1.4. The "Bottom Strike" checkbox is used to note if it appears that the probe reached the ocean floor (e.g. due to erroneous warning or a profile whose maximum depth exceeds the indicated ocean depth at its position), as this information is incorporated in the output JJVV file (Section 4.1.5).

4.1.2   Opening a Profile Editor Tab

Raw temperature-depth profile can be loaded into a profile editor interface by one of two means:

1. Loaded directly from the ARES signal processor interface by selecting "Process Profile" (see Section 3.1) after acquiring data in realtime or reprocessing from an audio file

2. Read from an ASCII file format (see supported input ASCII formats in Section 4.1.5) using the ARES ASCII file selection interface (Fig. 4-2)
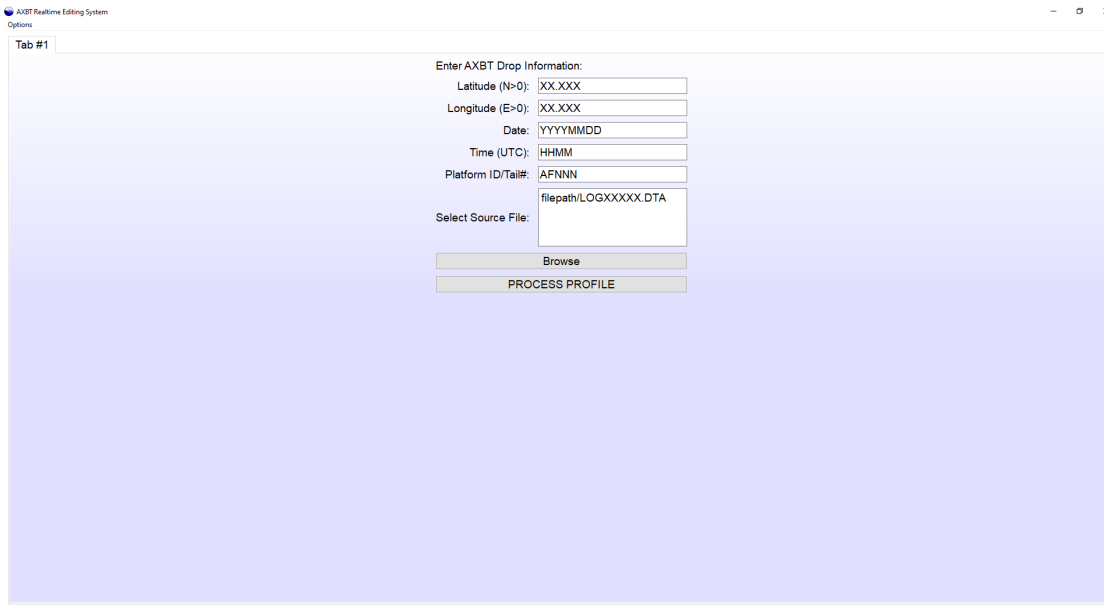
Figure 4-2: The ARES Profile Editor ASCII file selection interface.

Users can open an ASCII file selection interface tab to load a supported raw text file by either selecting "Options > New Profile Editing Tab" or using the keybinding "ctrl+P". This interface includes user inputs for drop date, time, and position, as some supported file formats (e.g. .DTA files) do not include that information. If a selected raw data file does include position and time, information is not required in those fields, but any valid positions or times entered there will supersede those in the raw data file.

4.1.3   Editing Components

Users can make four categories of edits to the quality-controlled profile (listed in order of appearance on the GUI, from top to bottom):

1. Add or remove individual points or a range of points

2. Specify a surface isothermal region to remove erroneous surface spikes

3. Truncate the base of the profile to correct for bottom strikes or excessive interference

4. Vertically shift the profile to correct for VHF interference

Individually removing points or spikes may be necessary when the autoQC algorithm's spike removal feature fails to identify all erroneous datapoints. This is especially common when there is extreme variability due to VHF interference that saturates the standard deviation filter and increases the deviation necessary to identify a datapoint as erroneous. Additionally, if too many points are truncated and important profile characteristics are missed, adding individual points may be necessary. Individual points can be added or removed and ranges of points can be removed with a point selection tool integrated into the ARES Profile Editor interface.

Occasionally, AXBT surface observations include erroneous spikes due to probe acclimation or interference. The ARES Profile Editor provides an option to manually create an isothermal layer at the surface by setting the temperature of all datapoints above a specified depth equal to the observed temperature at that depth using the "Isothermal Layer" feature. For example, if the surface temperature erroneously starts at 26^oC at the surface, but warms to a realistic value of 28^oC at 5 m, setting the isothermal layer value to 5 m will change the temperatures of all datapoints above 5 m to 28^oC, removing the surface spike. However, care must be taken by the operator to distinguish diurnal surface heating and cooling effects from erroneous features.

In some cases, it is necessary to truncate base of the profile, either to remove data from a bottom strike or to remove excessively variable (strong-interference) data. These profiles can be truncated with the "Maximum Depth" feature.

Finally, excessive VHF interference (either noise or an unexpected signal) that falsely triggers the start of profile collection can be overlooked by the autoQC algorithm if the gap between interference and valid data is less than 5 m or the interference extends without gaps beyond 50 m. If the correct profile surface is easily identifiable, the entire profile can be shifted upward manually using the "Depth Delay" feature. User-specified surface isothermal layer generation and profile truncation are not applied until after the vertical shift, so applying a vertical shift of 200 m and profile truncation at 400 m would first shift the profile upwards by 200 m, and then truncate all data that were below 600 m on the raw profile).

4.1.4 Quality Control Flags

In addition to profile edits, users can apply quality control flags to note any profile discrepancies. First, a checkbox allows users to note whether the profile was a bottom strike. This information is encoded in the JJVV output file format (described further in Section 4.1.5). Additionally, a dropdown menu allows users to select one of ten quality control (QC) codes depending on the state of the quality-controlled profile (Table 4.1). QC code 1 (Good Data) should be used for a profile that has been successfully quality-controlled (whether edits were necessary or not) and is of sufficient quality to assimilate in models or use for other means. QC code 0 (No QC) should be used when, due to time or other operational constraints, no quality control was conducted on the profile. QC code 8 (Action Required) should be used when the profile was reviewed but additional steps (e.g. point removal) are necessary before the profile is accurate. The remaining QC codes describe common issues with AXBTs. Only profiles with QC codes 1, 3, and 9 (case-dependent for the latter two codes) should be used for other applications.

4.1.5 Output File Formats

In addition to loading data processed either in realtime or from a WAV file using the Data Acquisition System, ARES supports loading data from LOG, EDF, and NVO ASCII files. It is also possible to load profiles from JJVV and FIN files (described in the next paragraph), but not recommended as these file formats generally contain lower resolution data that has already undergone post-processing.

ARES supports exporting profile data to two ASCII file formats (FIN and JJVV), one binary format (WMO BUFR), and both profile and location/bathymetry plots. FIN files contain drop metadata and a quality-controlled temperature depth profile subsampled and interpolated to a 1m resolution. These files are formatted identically to NVO files (see Section 3.2.3), the only exception being that NVO files include all raw profile information whereas FIN files only include quality-controlled, 1m resolution data. An example of NVO/FIN file format is provided in Fig. 4-3a.

Table 4.1: ARES quality control codes for post-processed AXBT temperature-depth profiles.

Code	Name	Description
0	No QC	No quality control was conducted
1	Good Data	Any profile discrepancies were corrected
2	No Signal	No signal was received from the AXBT
3	Spotty/ Interference	Signal was received but was either missing sections of data or was degraded by excessive VHF interference
4	Bad Data	Data was received but was bad quality for a reason not covered by a different code
5	Isothermal	Temperature profile was unexpectedly isothermal
6	Late Start/ Hung Probe	Either AXBT did not start transmitting on time, or the probe failed to detach from the surface float for some period after the transmission started
7	Slow Falling	Probe sank slowly through the water column, resulting in inaccurate depth recordings
8	Action Required	Profile was reviewed and requires additional corrections
9	Suspect Climo/ Neighbor Mismatch	Profile either does not match climatology for a given region and time or year, or is surprisingly different from other profiles launched nearby

Figure 4-3: File formats for (a) FIN/NVO and (b) JJVV files. Note that the FIN file used in (a) is truncated below 48m, but the same temperature-depth format is continued for the remainder of the file.

JJVV files (format specified at `www.dfo-mpo.gc.ca/science/data-donnees/gts-smt/codes/new-nouv-bath-eng.html`) include only the subsampled points from the quality-controlled profile (these typically include only inflection points and critical levels in order to reduce file size). Like FIN/NVO files, JJVV files include a header with drop metadata, but in JJVV files each profile datapoint is encoded as five digits: the first two represent the depth in meters, and the last three represent the temperature to the tenths of a degree Celsius (for example, an observation of 28.3^{o}C at 54 meters would be encoded as '54283'). Temperature-depth observations are listed with increasing depth, and every time depth increases into the next hundreds place, this is denoted by another five digit code: '999', followed by the new hundreds of meters (for example, once depth exceeds 100m, '99901' would be appended to the file, and a subsequent listing of '54283' would now indicate a measurement of 28.3^{o}C at 154m). An example of the JJVV file format is provided in Fig. 4-3b.

BUFR files are binary files that follow a format outlined by the World Meteorological Organization (WMO) using a large series of tables to specify a common method of encoding large amounts of data from a wide range of sources (e.g. satellites, radiosondes, oceanographic observations) while minimizing file size. BUFR files are divided into six sections, labelled zero through five. Sections 0,1, and 5 contain primarily static data to confirm that the file is formatted correctly, and some basic information such as date and time. Section 2 is optional, used to encode any necessary information without a proper entry in the WMO BUFR tables (this section is unused in ARES-generated BUFR files). Section 3 contains F-X-Y descriptors, used to identify the type of data being transmitted (which is encoded in Section 4). In ARES, these descriptors denote date and time, latitude and longitude to the hundredth of a degree, and temperature and depth to the tenths place in Kelvin and meters (respectively). A guide to the BUFR version used by ARES (WMO FM94 BUFR Edition 4) is available at `https://community.wmo.int/activity-areas/wmo-codes/manual-codes/latest-version`.
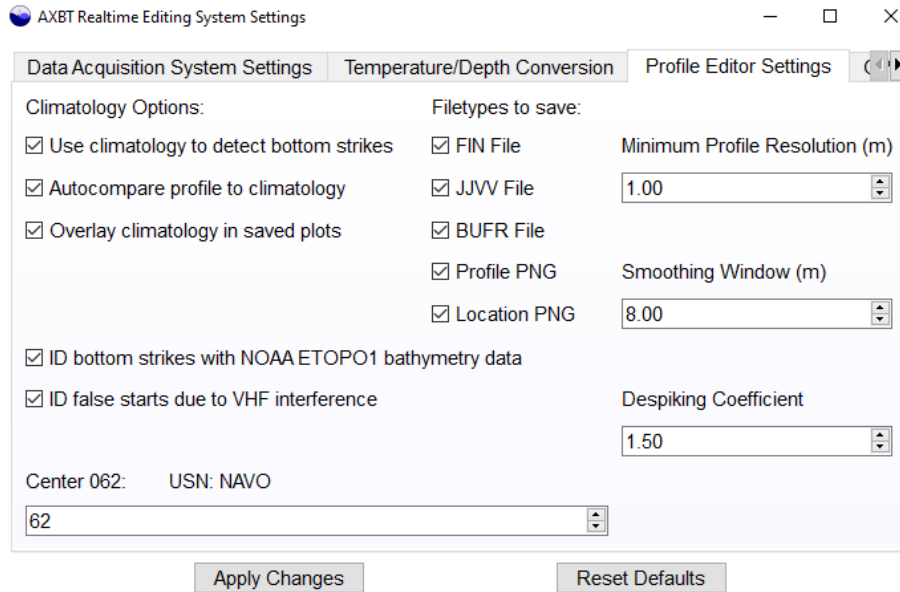
Figure 4-4: The ARES profile editor settings window.

4.1.6 Settings

The ARES profile editor settings (Fig. 4-4) are divided into four sections: climatology options, file save formats, autoQC thresholds, and additional settings. Climatology settings are listed in the top right corner of the settings window. When checked, the "Use climatology to detect bottom strikes" option compares the slope of the quality-controlled profile to the slope of the climatological profile to identify erroneous isothermal or warming features caused by a bottom strike and automatically truncates the profile (this correction is discussed further in Section 4.2). When "Autocompare profile to climatology" is checked, a red "Climatology Mismatch" will appear if less than 50% of the points in the quality-controlled profile are within the blue-shaded climatology region. Finally, "Overlay climatology in saved plots" controls whether the blue climatology shading is included in the output profile plots.

The "Filetypes to Save" column lists five different output file formats: two ASCII formats (FIN and JJVV), one binary format (BUFR) and two images (profile and location plots identical to those in the Profile Editor interface). These output files are discussed in Section 4.1.5. The two checkboxes in the bottom left corner of the window control automatic corrections applied by the autoQC algorithm. The

69

"ID bottom strikes with NOAA ETOPO1 bathymetry data" checkbox determines if bathymetry data is used to automatically truncate profiles whose maximum depths are greater than the ETOPO1-indicated ocean depth at the profile location, and the "ID false starts due to VHF interference" checkbox controls whether gaps in data in the upper ocean are identified as signs of VHF interference causing false starts and used to vertically shift the profile. Both of these corrections are discussed further in Section 4.2. Finally, the spinbox in the bottom left is used to identify the center through which data is uploaded to the Global Telecommunications System (GTS), if applicable. This identification is a necessary field for the WMO-formatted BUFR message, discussed further in Section 4.1.5.

The rightmost column of settings includes three spinboxes that control thresholds used by the autoQC algorithm. The minimum profile resolution is the resolution (in meters) at which points are downsampled following smoothing and despiking. The smoothing window is the length (in meters) of the running smooth filter used to remove high frequency variability before downsampling. Finally, the despiking coefficient adjusts how sensitive the autoQC algorithm is for removing spikes (a higher coefficient results in the inclusion of more data and a lower coefficient results in the exclusion of more data). The mathematical definition of this coefficient and its role in the autoQC algorithm are more clearly defined in Section 4.2.1.

4.2   autoQC Algorithm Overview

The automated quality control (autoQC) algorithm corrects raw temperature-depth profiles in two manners, described separately in the following subsections. First, the profile is corrected for common modes of VHF interference (e.g. spikes and false starts), and is smoothed and subsampled to a lower resolution (default 10 meter smooth, 1 meter sampling resolution). Then, the profile is compared to climatology and bathymetry data to identify (and correct when possible) bottom strikes, climatology mismatches, and other discrepancies before displaying the resulting quality-controlled profile to the user for additional edits and/or data export.
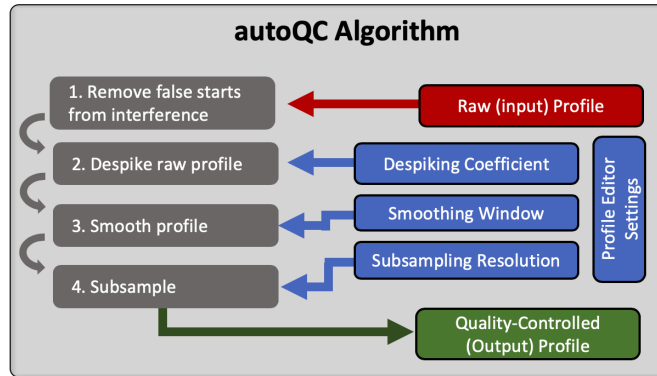
Figure 4-5: Data flow through the VHF interference correction component of the autoQC algorithm.

4.2.1 VHF Interference Correction

The following VHF interference corrections are applied to raw profiles:

1. Gaps in data due to profile false starts are identified and corrected

2. Spikes from interference are identified using a depth-based running standard deviation filter and removed

3. Profiles are smoothed using a depth-based box filter

4. Profiles are subsampled

Gap detection identifies "false starts" due to strong, typically external signals from a source other than the observed AXBT that trigger profile collection before the AXBT begins transmitting data. These false starts are usually characterized by a brief signal at or near the surface, followed by a several-second gap in data, and then an otherwise valid AXBT profile that is depth-shifted (so the actual surface temperatures from the AXBT are recorded at some subsurface depth). These are identified by searching for a break in data exceeding 10 m in the upper 50 m of the profile (as data gaps due to signal loss are less common near the surface when the AXBT signal is stronger as the aircraft is still close to the float). If such a break exists, the profile is shifted upwards so the surface observation is the first datapoint after the gap, and the gap check is repeated until no gaps exist in the upper 50 m.

The second step uses a running filter to identify spikes due to VHF interference. This is accomplished by calculating the mean and standard deviation for all profiles within a 10-m window of a point (± 5 m, corresponding to $D_\sigma = 5$ m in Eq. 4.1). If an individual point deviates from the mean by more than the product of a user-specified coefficient (β) and the standard deviation about the mean (such that at any index i, $|T[i] - \hat{T}[i]| \geq \beta \sigma_T[i]$, where $\hat{T}[i]$ and $\sigma_T[i]$ are the mean and standard deviation of the temperatures within the 10-m window of the point $z[i]$ as defined in Eq. 4.1), the datapoint is considered a spike and discarded (for the ARES Profile Editing System evaluation, $\beta = 1$).

$$\hat{T}[i] = \frac{1}{N} \sum_{j=1}^{N} \hat{T}_n, \quad j \in (z[i] - D_\sigma) \leq z[j] \leq (z[i] + D_\sigma)$$

$$\sigma_T[i] = \left(\frac{1}{N} \sum_{j=1}^{N} \left(T_j - \hat{T}[i] \right)^2 \right)^{1/2}$$

(4.1)

After the two primary modes of VHF interference (false starts and profile spikes) are corrected if necessary, the temperature-depth profile is passed through a depth-based smoother using a simple box filter described by Eq. 4.2, where the depth window is determined by D_M, and $\bar{T}$ is the smoothed temperature profile.

$$\bar{T}[i] = \frac{1}{N} \sum_{j=1}^{N} T_j, \quad j \in (z[i] - D_M) \leq z[j] \leq (z[i] + D_M)$$

(4.2)

Finally, profiles are subsampled to a lower resolution before being displayed to the user for additional edits as necessary and data export. The smoothing window should be at least twice the subsampled resolution in order to prevent aliasing (noting that box smoothers have a relatively poor frequency response compared to more advanced low pass filters). However, the default subsampling resolution is 1 m, and in practice it was observed that an 8-m smoother ($D_M = 4$ m) was necessary to remove high-frequency variability due to remaining VHF interference.

4.2.2 Comparison to Climatology and Bathymetry

After being corrected for VHF interference, smoothed, and subsampled, the temperature profile is compared to bathymetry and climatology data given the AXBT launch position and date. Ocean depth at the AXBT launch position is interpolated from the NOAA ETOPO1 Global Relief Dataset [20] which provides global topography and bathymetry at a 1-arcminute resolution. The autoQC algorithm automatically truncates profiles at the bathymetry-indicated ocean depth.

Climatological monthly ocean temperature means and standard deviations are from subset of data from the Generalized Digital Environmental Model (GDEM) climatology [19] that includes values at a 0.25^o horizontal resolution for 29 vertical levels in the upper 1,000 m. The profile climatology comparison has two steps: (1) comparing profile slopes to identify bottom strikes, and (2) comparing the observed profile to climatology to detect possible climatology mismatches (e.g. slow falling probes, late starts, and otherwise faulty AXBTs). Because bottom strikes typically appear as temperature observations that are erroneously isothermal or warming with depth at the base of the profile, the slopes of the observed and climatological profiles are calculated and compared. To minimize incorrect identifications due to higher-frequency variability, a fifty-point running mean of the differences between the climatology and observed profile slopes is examined. If this running mean exceeds $0.1^oC/m$, it is considered a bottom strike, and all data below the depth at which this threshold is exceeded are discarded. The observed profile is then compared to the climatology to identify potential profile discrepancies. If less than 50% of the observed profile falls within the uncertainty range for the climatology (currently one standard deviation), then the profile is flagged as a climatology mismatch.

If either a bathymetry-based or climatology-based bottom strike is identified, the profile is truncated at that depth (priority is given to the climatology-based strike if both are identified). After a profile has been corrected for both VHF interference and bathymetry- and climatology-indicated discrepancies, it is plotted on the Profile Editor interface. Original and quality-controlled profiles are overlaid in black and red

(respectively) on top of blue-shaded climatology (showing ± one standard deviation). When applicable, a climatology mismatch is noted with red text in the bottom-right corner of the profile window.

4.3   Profile Quality Control Evaluation

4.3.1   Methodology

To examine the accuracy of the ARES Profile Editor interface, 1,215 uncorrected AXBT profiles (raw data recorded with a MK-21) from the TROPIC field campaign (collected between 2011-2019) were reprocessed with ARES. For each observation, it was noted whether additional edits (beyond autoQC algorithm corrections) were required, if any quality control flags were applicable, and if the final profile was of satisfactory quality for transmission to the GTS for further use in numerical models (as agreed upon by four independent users). Additional profile comments and any required manual adjustments were also noted. Reprocessed profiles were evaluated against the original TROPIC dataset of profiles that were manually quality-controlled with the System for At-Sea Environmental Analysis (SASEA; [13]), as in [4].

4.3.2   Profile Quality Distribution

Of the 1,215 AXBT profiles processed, 1,177 (97%) were correctable to a sufficient quality for use in numerical models (light and dark green wedges, Fig. 4-6a). The most common discrepancies for the remaining 38 profiles were late starts (the profile failed to start transmitting data until the probe was no longer at the surface) and isothermal profiles (typically the thermistor probe failed to physically release from the AXBT surface float, and therefore recorded only surface data rather than descending through the water column) (Fig. 4-6c). Of those 1,177 good AXBT profiles, 156 profiles required manual user edits, leaving 1,021 (87%) for which the autoQC algorithm applied sufficient profile corrections without manual intervention.

Applying the autoQC algorithm and integrating climatology and bathymetry in-
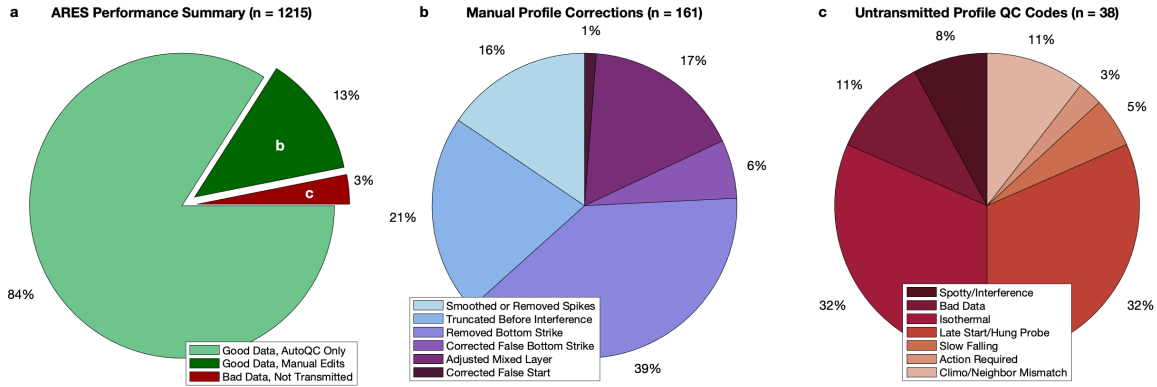
Figure 4-6: Distributions of (a) autoQC algorithm performance, (b) operator corrections for profiles that required manual edits, and (c) quality control codes for bad profiles. Note that (b) and (c) show the distributions of profiles from the dark green and red wedges (respectively) in (a).

creased operator efficiency while minimizing potential for operator error, compared to the original manual quality control method. Profiles reprocessed with ARES required less than a minute on average to quality control, with 87% of those profiles requiring no manual user intervention. This contrasts with the previous (manual) method, which typically took about ten minutes to quality control per profile and required manual intervention for 100% of profiles. Additionally, the integrated bathymetry and climatology enabled operators to make more informed decisions when distinguishing profile discrepancies from actual features. Together, these resulted in a noteworthy increase in the number of profiles corrected to a quality sufficient for transmission to the GTS and assimilation in numerical models. During TROPIC, 1,073 of 1,215 (88%) profiles were transmitted to the GTS. Thus, the 1,177 profiles (97%) successfully reprocessed with ARES yielded a 9% increase (104 profiles).

Root mean square differences between quality-controlled ARES- and SASEA-processed profiles remained below 0.1^oC over the full range of profile depths (Fig. 4-7, solid line). The standard deviation of these differences exceeds 0.5^oC between approximately 200 m and 400 m (Fig. 4-7, shading), corresponding to greater temperature variability in the thermocline. Additionally, there was an increase in mean profile differences and corresponding standard deviations near the surface (shallower than approximately 30 m). This suggests that either quality-controlled surface tem-
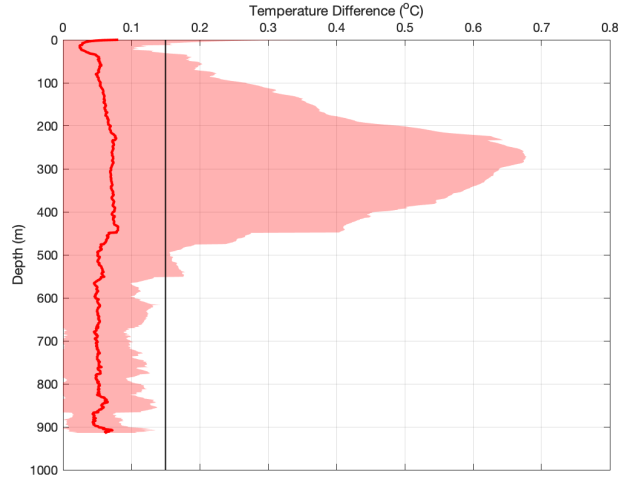
Figure 4-7: Root mean square temperature differences (^oC, solid red line) for ARES- and SASEA-processed profiles. Shading denotes mean temperature differences plus and minus one standard deviation. The solid black line denotes rated temperature accuracy for Sippican AXBTs [2].

peratures are much more sensitive to the quality control method applied, more manual corrections were applied during data reprocessing at the surface, or some combination thereof.

4.3.3   Common autoQC Failure Points

In 13% of cases (Fig. 4-6a, dark green wedge), the autoQC algorithm was not able to sufficiently correct profile discrepancies. Features that required manual corrections fell into one of several categories (Fig. 4-6b):

1. Bottom strikes (n = 73)
2. Excessive variability at depth (n = 34)
3. Erroneous mixed layer features (n = 27)
4. Spikes due to noise and interference (n = 25)
5. False starts (n = 2)

Each of these conditions are discussed separately below, with examples in Fig. 4-8. Because some profiles required multiple types of corrections, the sum of all edits above (n = 161) is greater than the number of manually edited profiles (n = 156).
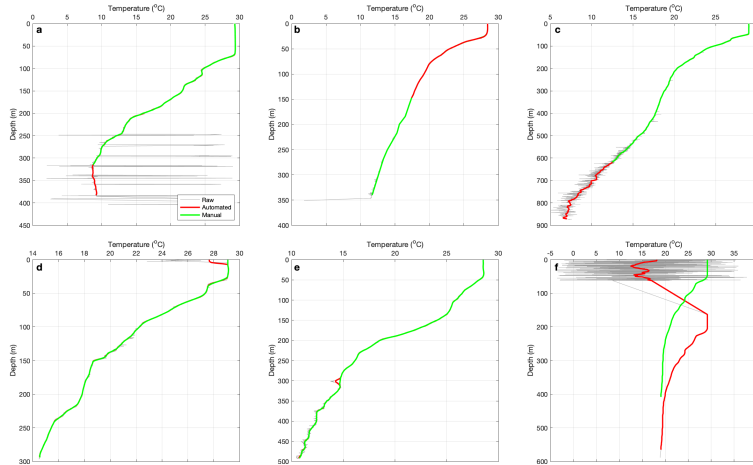
Figure 4-8: Examples of profiles that required manual edits, for: (a) missed bottom strike, (b) false positive bottom strike, (c) excessive interference at depth, (d) erroneous mixed layer feature, (e) temperature spike, and (f) missed profile false start. Unedited (raw) profiles are plotted as grey lines, and incorrectly (autoQC) and correctly (manually) quality-controlled profiles are overlaid in red and green, respectively.

Bottom strikes

ARES uses high resolution (1-arcminute) global bathymetry to assist in bottom strike detection. However, this check can still fail by either missing actual bottom strikes (Fig. 4-8a) or unnecessarily truncating good data (Fig. 4-8b). Bottom strike correction failures are most common in regions characterized by large bathymetric variability over small scales or when an AXBT's launch and splash points differ significantly due to launch conditions or high winds. Although the autoQC algorithm also uses climatology to assist in bottom strike detection by comparing profile slopes to detect erroneous warming, this check often failed to identify bottom strikes in the observed profiles. Routines could be implemented to compare profile slopes to predetermined thresholds, but this would be region-specific and therefore violate the objective of developing AXBT-editing software usable in any ocean basin. Regardless, bottom strike under or overcorrection can be easily identified during manual quality control, and increasing the robustness of the climatology-assisted component of the autoQC algorithm remains a priority for future work.

Excessive variability at depth

Occasionally, due to a combination of AXBT source signal strength and increasing distance between AXBT and receiver, some AXBT profiles become excessively spiky at depth so the autoQC algorithm's despiking and smoothing features fail to correct for these issues (Fig. 4-8c). Currently there are two manual solutions: (1) adjust the despiking coefficient and smoothing filter length to exclude a greater amount of datapoints and increase smoothing, or (2) truncate the profile above the depth at which excessive spiking begins to occur. An objective for future work is to implement an autoQC check that automatically performs one of the above corrections when a profile's sliding standard deviation exceeds and remains above a preset threshold.

Erroneous mixed layer features

Accurately quality-controlling mixed layer features is both critical and difficult for several reasons. Features such as sea surface and mixed layer temperatures and mixed layer depths are important for air-sea interactions, particularly for observations in tropical cyclones, the purpose for which this program was developed. However, dynamic surface conditions can be difficult to distinguish (qualitatively or programmatically) from probe equilibration with ambient ocean temperature, requiring climatology-informed decisions by the operator. Additionally, boundary conditions can reduce the effectiveness of smoothing filters at removing noise, making it difficult to implement programmatic solutions for erroneous mixed layer features (Fig. 4-8d). No additional checks have been implemented for mixed layer errors, which require manual quality control after autoQC algorithm application.

Spikes due to noise and interference

Although the autoQC algorithm is designed to identify and correct erroneous spikes from VHF interference, a few profiles had spikes that the algorithm failed to fully remove (Fig. 4-8e). The autoQC algorithm fails when spikes occur on a length scale greater than the despiking filter window (default 8 m), or when the spike skews

the entire distribution of temperature points in a given window so it is no longer centered approximately on the ideally corrected value. No corrections have been implemented to address this shortcoming, which is one reason that manual quality control is still necessary following application of the autoQC algorithm.

False starts

The autoQC algorithm was generally extremely effective at identifying and correcting false starts due to VHF interference. In the two profiles where false starts were not detected by the autoQC algorithm, the algorithm failed because either the gap between interference and good data was too small (less than 5 m), or the interference extended without gaps to too great a depth (greater than 50 m, e.g. Fig. 4-8f). This is because the autoQC false start detection check works by identifying (and correcting for) lapses in data exceeding a 5 m interval in the upper 50 m of the ocean. These thresholds are sufficient for correcting more subtle interference, whereas interference that the autoQC algorithm fails to detect (e.g. Fig. 4-8f) is much more obvious and easily detected during manual quality control checks.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Case Study: Hurricane Dorian (2019)

5.1 Overview

In 2019, Hurricane Dorian (Fig. 5-1) churned through the Atlantic basin, making landfall first over the Bahamas as a category 5 storm and again over Cape Hatteras as a category 1 before recurving out to sea south of New England. For much of Hurricane Dorian's life, including its time as a major hurricane, sustained aerial reconnaissance missions were conducted by both NOAA and U.S. Air Force Hurricane Hunter aircraft.

These missions marked the first opportunity to test and evaluate a fully functional version of ARES in realtime in an operational setting. From 31AUG to 03SEP, the TROPIC field program deployed AXBTs on seven hurricane fix missions with the USAF 53rd Weather Reconnaissance Squadron, including three missions for which ARES was tested in parallel with the MOOS-SASEA system for AXBT data acquisition and quality control.

As a major hurricane east of Florida, two features of Hurricane Dorian brought about the potential for a unique set of ocean observations. First, while approaching sustained winds of 160 kts, Hurricane Dorian slowed to a near-stall, crawling over the Bahamas at less than 2 mph for a nearly 24-hour period [21]. This combination of catastrophic intensity and low translation speed enhanced the integrated stress imparted on the ocean surface, resulting in a correspondingly enhanced upper-oceanic response. Second, Hurricane Dorian translated across a relatively large surface tem-
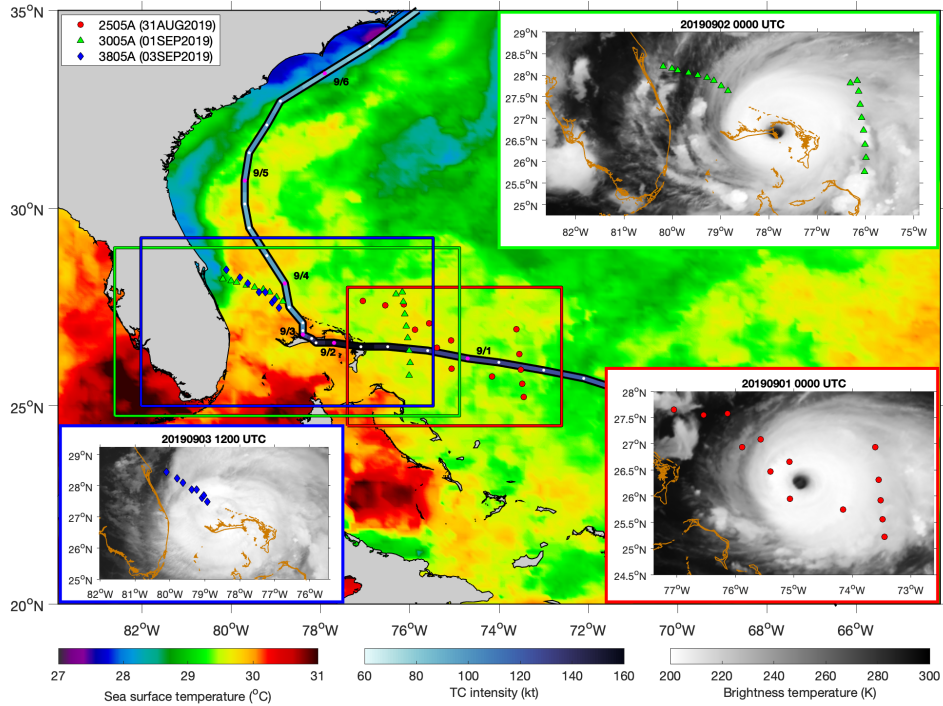
Figure 5-1: Hurricane Dorian (2019) track and intensity, environmental sea surface temperatures on 30AUG, and deployment locations (by mission) for AXBTs processed with ARES in realtime. Inset satellite imagery show storm-relative float deployment locations for each mission.

perature gradient associated with the Gulf Stream, yielding an opportunity to observe the upper-oceanic response in the presence of such a gradient. AXBT observations in Hurricane Dorian fell into three categories:

1. Transects across the Florida Current region of the Gulf Stream
2. Transects across Hurricane Dorian's cold wake
3. Individual observations just ahead of Hurricane Dorian's track or within the storm environment

During the three missions on which ARES was evaluated, 44 profiles were collected with ARES (including 43 AXBTs which successfully transmitted signal to the aircraft). The remainder of this chapter is divided into two sections: evaluating ARES performance processing and quality controlling profiles collected in realtime with a WiNRADIO receiver array (Section 5.2) and using the information gathered by sustained observations with ARES to develop a better understanding of the oceanic response beneath Hurricane Dorian (Section 5.3).

5.2 ARES Performance

5.2.1 Signal Processing

Signal level and SNR distributions for profiles demodulated with WiNRADIO receivers differ from those collected with the MOOS system in several ways. First, profile peak signal levels for valid AXBT profiles (n = 43) are entirely above 65 dB, with a peak signal level of 56 dB for the single AXBT that failed to transmit valid data (Fig. 5-2a). Second, the signal level distribution is much more continuous for the WiNRADIO-received dataset (and therefore somewhat more difficult to distinguish valid signals from noise), with values ranging between 45 and 73 dB (Fig. 5-2b). Third, the SNR distribution includes a larger proportion of observations with signal ratios below 0.3, but still has a secondary maximum of observations with ratios above approximately 0.4 but less than unity (Fig. 5-2c).
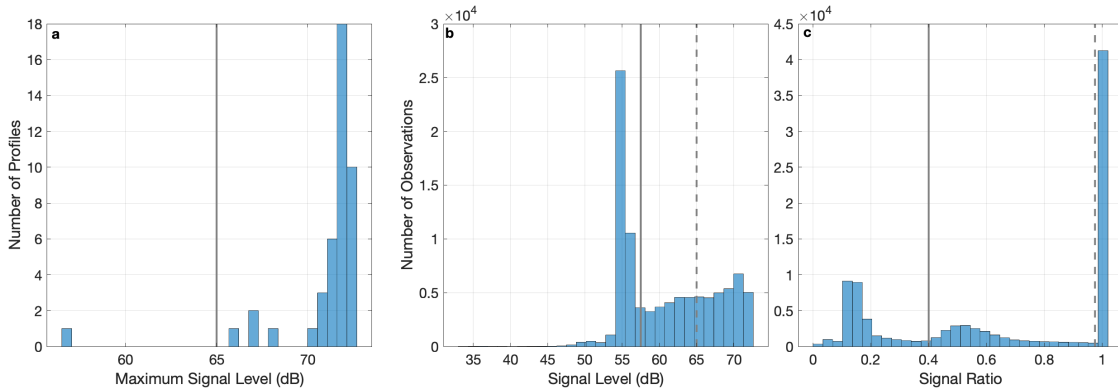
Figure 5-2: (a) Distribution of maximum signal levels (dB) for Hurricane Dorian. The vertical gray line denotes the required maximum peak signal level for all profiles analyzed ($S_{P_{MAX}} = 65dB$). (b and c) Distributions of (b) all signal levels (dB) and (c) all signal-to-noise ratios (%) for all observations in Hurricane Dorian. Vertical lines represent minimum signal level and ratio thresholds (solid and dashed lines correspond to general and trigger thresholds, respectively) applied when evaluating data acquisition system performance.

Two-dimensional histograms for signal level and ratio distribution over time (Fig. 5-3) provide further justification for optimal thresholds. As in Fig. 5-2a, peak signal levels all begin above 70 dB, marking it as an ideal triggering threshold. Beyond approximately 400 seconds post-trigger (corresponding to 600+ m observation depths),
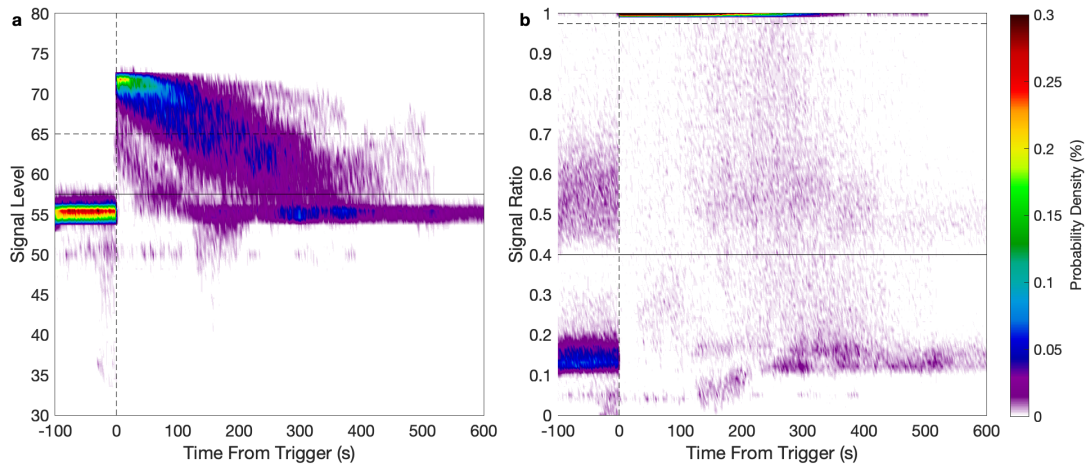
Figure 5-3: Distributions of observations in Hurricane Dorian for all observation times versus (a) signal levels and (b) signal ratios. Profiles were standardized so zero seconds (vertical lines) denotes the first observation that satisfied the minimum triggering thresholds. Triggering and general thresholds are denoted by horizontal dashed and solid lines, respectively.

signal levels decay to their pre-trigger values centered at approximately 55 dB (corresponding to the peak in Fig. 5-2b). Signal-to-noise ratios for observations from Hurricane Dorian are generally similar to those from reanalyzed audio files in Section 3.3 with one exception: SNRs in Hurricane Dorian include an additional maximum beneath 0.2 before AXBT signal transmission that return approximately 300 seconds post-trigger, corresponding to increased profile termination at approximately 450 m. Given this information, trigger and minimum signal levels were set to 65 dB and 57.5 dB (respectively), and trigger and minimum signal ratios were set to 0.975 and 0.4, respectively.

All AXBT profiles collected in Hurricane Dorian were reprocessed with the aforementioned thresholds to test their suitability for WiNRADIO receiver arrays. The mean 10-m standard deviations remain below 0.1^oC to 800 m (Fig. 5-4; depth of the deepest collected profile), performing similarly to the 389 reprocessed audio files in the Data Acquisition System evaluation, Section 3.3. This supports the use of these thresholds for AXBTs processed in realtime with a WiNRADIO receiver array similar to the one used in this study.
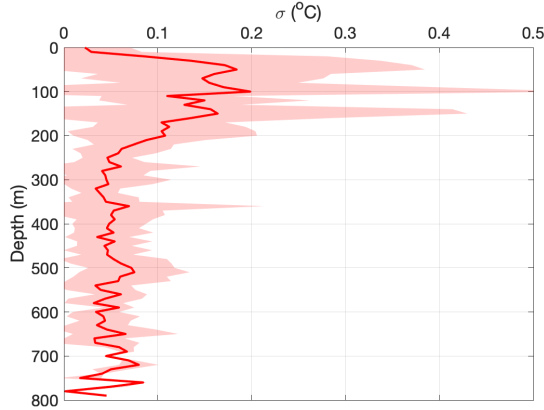
Figure 5-4: Mean 10-m standard deviations (solid line) and $\pm$ one standard deviation of the 10-m standard deviations (shading) for the 43 AXBT profiles received with WiNRADIO receivers during Hurricane Dorian, processed with the signal level and ratio thresholds shown in Figs. 5-2 and 5-3.

5.2.2 autoQC Algorithm

Of the 44 AXBT profiles that were launched, 39 were good profiles (this number includes profiles with correctable bottom strikes), 2 failed to transmit, 1 was isothermal, and 2 started transmitting after the probe had already begun descending through the water column (late starts; Table 5.1). All 42 profiles that transmitted data were edited using the ARES Profile Editor. Only 4 of the 42 profiles (9.5%) required manual edits, all of which were for bottom strikes (three profiles were not terminated until after the bottom strike, and one profile was terminated about 30 m too early). All four of these profiles were located along a bathymetry gradient associated with the continental slope east of Florida.

Table 5.1: AXBT profile breakdown by quality control codes. Quantities in parentheses include AXBTs that were not collected with ARES but are not included in the totals.

Mission	Good	No Signal	Isothermal	Late Start	Total
31AUG	13	1	0	1	15
01SEP	18 (20)	0 (1)	1 (2)	1 (1)	20
03SEP	8	1	0	0	9
Total	39	2	1	2	44
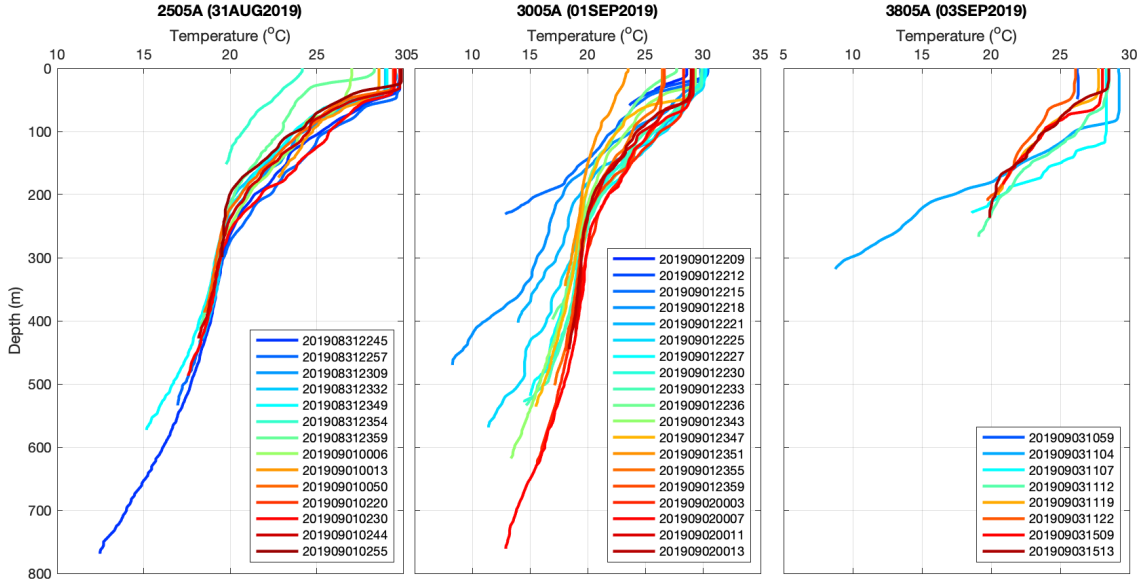
Figure 5-5: AXBT profiles collected with ARES in Hurricane Dorian, organized by mission.

All received profiles except for the isothermal profile (41 total) are shown in Fig. 5-5. Comparing the profiles in this figure, one late start on the first mission (201908312354) is easily distinguished. However, the late start on the second mission (201909012236) is difficult to distinguish from the other profiles, and an accurate AXBT profile (201909012351) appears to be a late start when compared to other profiles. This highlights a challenge with AXBT observations beneath tropical cyclones and in other extremely dynamic regions with high variability: determining whether profile features that deviate from climatology are accurate or erroneous. In this case, the pronounced cold wake Hurricane Dorian produced as a category 5 storm resulted in extremely cool (approximately 23^oC) sea surface temperatures.

Three AXBT profile characteristics that can distinguish late starts from accurate, cool profiles are presented in this section: (1) thermal structure, (2) initial signal levels and SNRs, and (3) AXBT fall time (time from launch to initial AXBT signal receipt). Even in environments with large horizontal temperature gradients, nearby AXBTs launched at similar times should share some features that may distinguish accurate or erroneous profiles. For example, three sets of adjacent questionable (dashed lines) and good (solid lines) profiles are shown with like colors in Fig. 5-6. Here, the
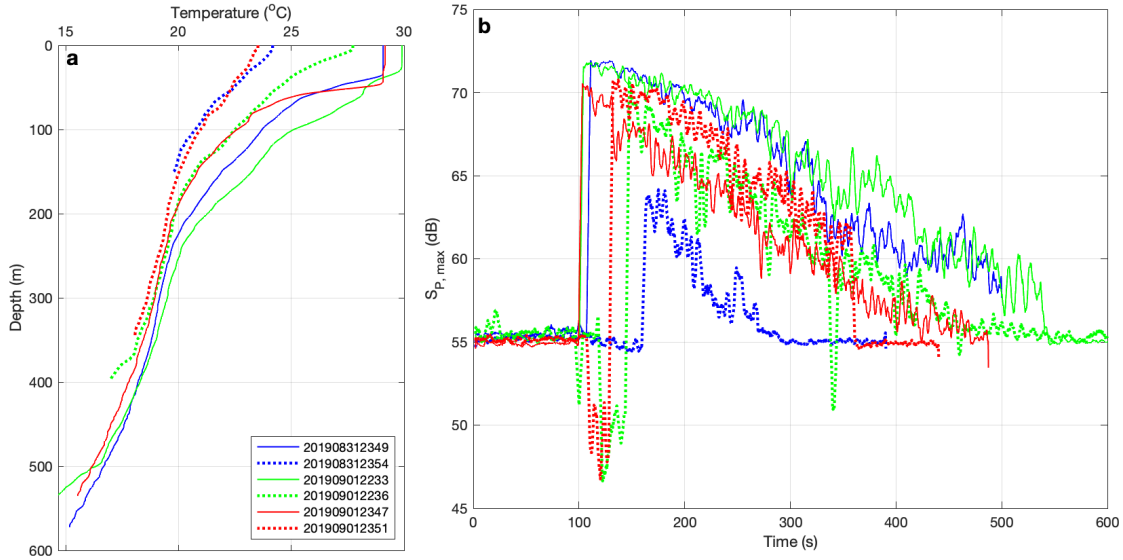
Figure 5-6: (a) Temperature-depth profiles and (b) signal level (S_P; Eq. 3.3) versus time for (dashed lines) possible late starts and (solid lines) adjacent profiles.

solid and dashed red lines (adjacent profiles) converge at depth (below approximately 200 m) whereas the other two sets of profiles (blue and green) do not (Fig. 5-6a). Additionally, in the absence of a large wind gradient or change in altitude between drops, adjacent AXBTs should have similar fall times. In Fig. 5-6b, the red, green, and blue pairs have start times differing by 30, 45, and 52 seconds, respectively. Finally, low initial signal levels (S_P, Eq. 3.3) compared to adjacent profiles can indicate a bottom strike. In Fig. 5-6, only with the red pair of profiles does the possible late start have signal levels that exceed those of the adjacent profile (by 3 dB). For the green and blue pairs, the peak signal levels of the questionable profiles are lower than those of their adjacent profiles by 3 dB and 8 dB, respectively.

As a final check, drastic surface cooling was expected for the 201909012351 AXBT (which was launched directly behind Hurricane Dorian in an area where modeled surface cooling approached 5-6°C) whereas the other two questionable AXBTs were not launched in regions where excessive cooling was expected. Thus, for the analyses in the next section, data from 201909012351 is included whereas 201908312354 and 201909012236 are excluded, leaving 39 profiles for subsequent analyses.

5.3 Application of Observations

Using ARES for AXBT data acquisition and quality control in an operational setting enabled two advances from the MOOS-SASEA system:

1. Easier rapid deployments of high spatial resolution AXBT transects

2. Deployment of a greater number of AXBTs with a reduced operator quality control workload

Further analyses of AXBT data collected with ARES in Hurricane Dorian are divided into two sections:

1. Two cross-track transects observing Dorian's cold wake, and two cross-Gulf stream transects collected before and during passage of Hurricane Dorian

2. Storm-relative changes in SST, mixed layer depth, and ocean heat content

5.3.1 AXBT Transects

Two transects collected just 24 hours apart across Hurricane Dorian's cold wake depict quite different upper ocean responses (Fig. 5-7). Both transects include a cooler, deeper, mixed layer right of track and a sea surface temperature minimum just right of storm center (approximately 26°N on 01SEP2019 and 26.5^{o}N at 02SEP2019 UTC). However, the magnitude of the response is much greater for the 02SEP2019 transect. On 01SEP2019, sea surface temperatures range from 29.1^{o}C at the southernmost AXBT to 27.0^{o}C closest to the cold wake center. On 02SEP2019, Hurricane Dorian translated to regions of warmer pre-storm SSTs (Fig. 5-1), but the SST minimum in the cold wake cooled further. Specifically, the southernmost AXBT observed an SST of 29.3^{o}C, and an AXBT just north of Dorian's track measured an SST of 23.3^{o}C, a difference of 6°C over less than 70 km. Additionally, observed cross-track differences in depth of the 26°C isotherm more than doubled from 52 m (as shallow as 22 m left of track and as deep as 74 m right of track) on 01SEP to 107 m (0 m in the cold wake maximum and 107 m right of track) on 02SEP.
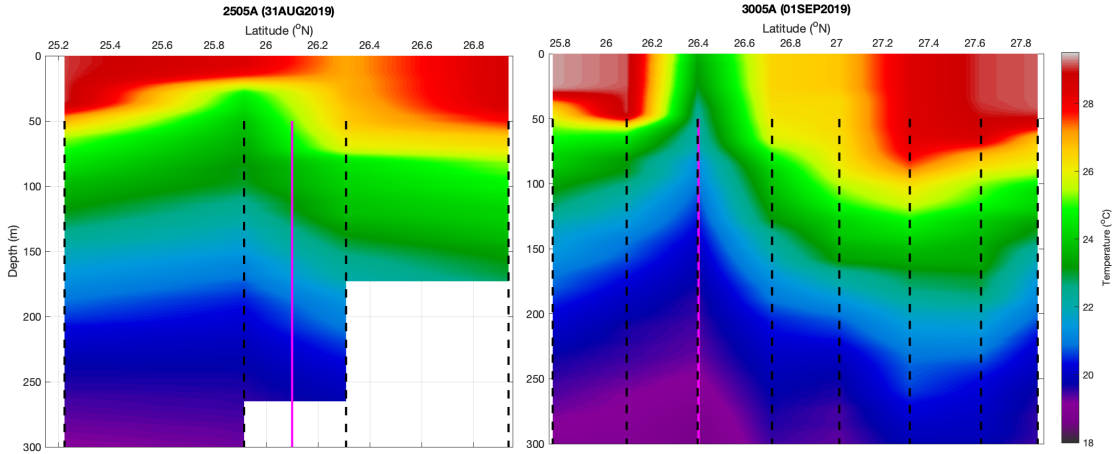
Figure 5-7: Cross-track transects of AXBT observations for Hurricane Dorian collected at (left) 01SEP2019 0000 UTC, 73°W and (right) 02SEP2019 0000 UTC, 76°W. Dashed black lines mark AXBT observation locations and magenta lines denote Hurricane Dorian's center latitude on crossing each transect.

It is worth noting that the magnitude of both cold wakes may have been greater than observed if the sea surface temperature minimum occurred somewhere between AXBT observation locations, and this is particularly true for the 01SEP2019 transect as it was only composed of four profiles. Additionally, pre-storm conditions varied across the transect locations- satellite-measured SSTs were warmer at the location of the second transect on 30AUG (Fig. 5-1). However, two co-located transects across the Gulf Stream, collected approximately 36 hours apart (02SEP2019 0000 UTC-green triangles and 03SEP2019 1200 UTC- blue diamonds; Fig. 5-1) provide an opportunity to identify changes in upper ocean conditions with storm passage. These transects are both entirely left (west) of Dorian's track, and thus do not capture the full cross-track response to Dorian's passage; rather, these observations show the effects of Hurricane Dorian's passage on a major ocean feature that could be expected to have feedbacks on both Dorian's intensity and other ocean processes.

The pre-storm AXBT transect (left transect, Fig. 5-8; 02SEP2019 0000 UTC) across the Gulf Stream has two noteworthy features:

1. Warm (29-30.5°C) SSTs across the entire transect

2. Deepening isotherms moving offshore, with the 26°C reaching a maximum at approximately 79.4°W
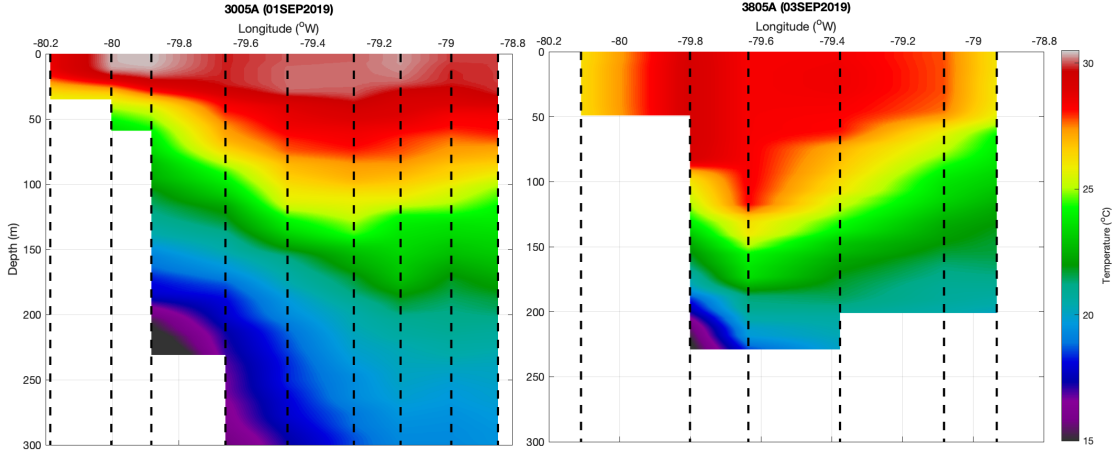
Figure 5-8: Cross-Gulf Stream transects (approximately 28.3°N, 80.2°W to 27.6°N, 78.8°W) of AXBT observations for Hurricane Dorian collected at (left) 02SEP2019 0000 UTC and (right) 03SEP2019 1200 UTC. Dashed black lines mark AXBT observation locations for each transect.

As Hurricane Dorian passed, several changes in the upper ocean structure across the Gulf Stream became apparent, even at the time of the second transect (03SEP2019 1200 UTC) when the region was still in the storm's environment. First, mixed layer temperatures cooled across the entire transect. Specifically, SSTs cooled from 28.6-30.4°C in the pre-storm environment to 26.2-29.2°C during Dorian's passage. Additionally, the 26°C isotherm deepened in the western half of the transect (west of approximately 79.4°W) and shallowed in the eastern half. This response is consistent with enhanced mixing driven by surface stress in the storm environment combined with mixed layer cooling and upwelling increasing towards the storm center.

5.3.2   Storm-Relative Characteristics

In order to examine storm-relative, time-independent features of the upper ocean response to Hurricane Dorian's passage, SST, tropical cyclone heat potential (TCHP), and 26°C isotherm depth ($D_{26°C}$) were plotted on a storm-relative (determined with NHC best track data; [21]) polar grid for all 43 AXBT profiles (Fig. 5-9). For each AXBT, values are expressed as anomalies relative to the mean value of each respective parameter for all AXBT profiles that were: (1) at least 100 km ahead of the storm, (2) between 300°R and 60°R to the storm, and (3) launched within 12
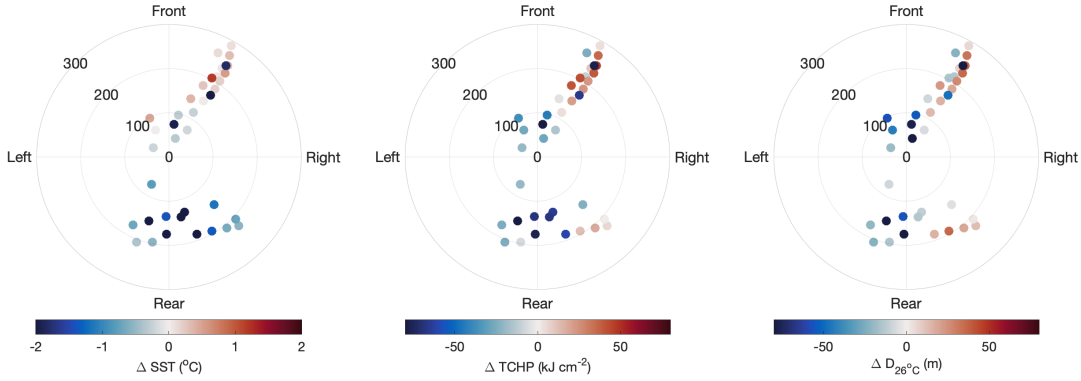
Figure 5-9: Storm-relative (distances in km) anomalies of SST, TCHP, and $D_{26^\circ C}$.

hours of the AXBT whose anomaly is being calculated. Storm-relative distance and bearing were determined relative to Hurricane Dorian's best track identified storm center interpolated linearly to the time of launch for each AXBT.

These storm-relative plots capture the spatial distribution of the upper ocean response. Anomalies of SST, TCHP, and $D_{26^\circ C}$ are generally positive (with some exceptions due to spatial variability in the Gulf Stream) for all profiles launched more than 100 km ahead of Dorian. Within 100 km, these anomalies are slightly negative, indicative of a combination of an upward heat flux and subsurface mixing within the storm environment. Behind Hurricane Dorian's path, both SST and TCHP anomalies are generally negative with the greatest anomalies just right of track, colocated with the expected maximum cold wake based on previous observations of upper ocean response to tropical cyclone passage [22]. Positive TCHP anomalies in the farther reaches of Hurricane Dorian's right-rear quadrant are likely due to some combination of (1) advection of higher heat content water into the cold wake during the inertial response to Hurricane Dorian's surface forcing or (2) residual differences in heat content due to temperature gradients that existed prior to Hurricane Dorian's passage.

However, the $D_{26^\circ C}$ minimum is centered slightly right of track directly behind the tropical cyclone, whereas negative SST and TCHP anomalies extend at least 50 km right of track. Additionally, $D_{26^\circ C}$ anomalies are very positive right of track in Hurricane Dorian's cold wake compared to colocated negative SST anomalies and small positive TCHP anomalies. This highlights the mixed layer deepening and cooling as-

sociated with the cold wake response: SST anomalies capture the surface cooling, and TCHP anomalies are weakly positive because the inertial response advects higher heat content waters from further right of track. The D_{26^oC} anomalies are comparably more positive than the TCHP anomalies because in addition to advection, vertical mixing results in an increase in isotherm depth accompanied with a decrease in TCHP (from a Lagrangian perspective) due to mixing across the 26^oC isotherm (because TCHP is calculated relative to 26^oC).

These results can be summarized together with information from the cross-wake transects. Relatively cooler sea surface temperatures, decreased heat content, and shallower isotherm depths are observed within the storm environment. Surface cooling is observed across the entire track, but the magnitude of this cooling is greatest just right of track. However, observed heat content increases right of track may be due to advection of warmer water from farther away from the tropical cyclone, and accompanying 26^oC isotherm deepening is likely due to a combination of that advection and cross-isotherm mixing (which also cools the mixed layer). These AXBT observations thus provide a rather comprehensive picture of the upper ocean temperature response behind Hurricane Dorian, partially due to the high deployment rate enabled by using ARES for data acquisition and quality control.

Chapter 6

Conclusions and Future Work

6.1 ARES Summary

The AXBT Realtime Editing System is a software solution that enables the simultaneous processing of multiple AXBTs in real time, with a seamless transition to a profile quality control interface. This system integrates often hardware-defined tasks (signal processing and audio recording) as software-defined functions completed on the processing computer in order to minimize necessary external hardware. This system also integrates connected GPS receivers, and supports exporting raw and quality-controlled temperature-depth profiles in several binary and ASCII formats. A comprehensive diagram of the data flow for the AXBT Realtime Editing System, linking the Data Acquisition and Profile Editing Systems, is shown in Fig. 6-1.

The ARES Data Acquisition System provides an optional high level of control system settings when processing incoming data, such as minimum signal and signal-to-noise ratio thresholds, window size, and window tapering. This high level of control enables users to maximize the amount of AXBT signal recovered while optimizing the balance between computational expense and temperature precision for a given computer and AXBT deployment frequency. Additionally, the optional configuration to autopopulate drop position and time reduces the margin for operator input error, and the ability to switch directly to a Profile Editing tab enables a seamless transition from data acquisition to profile quality control and file export.
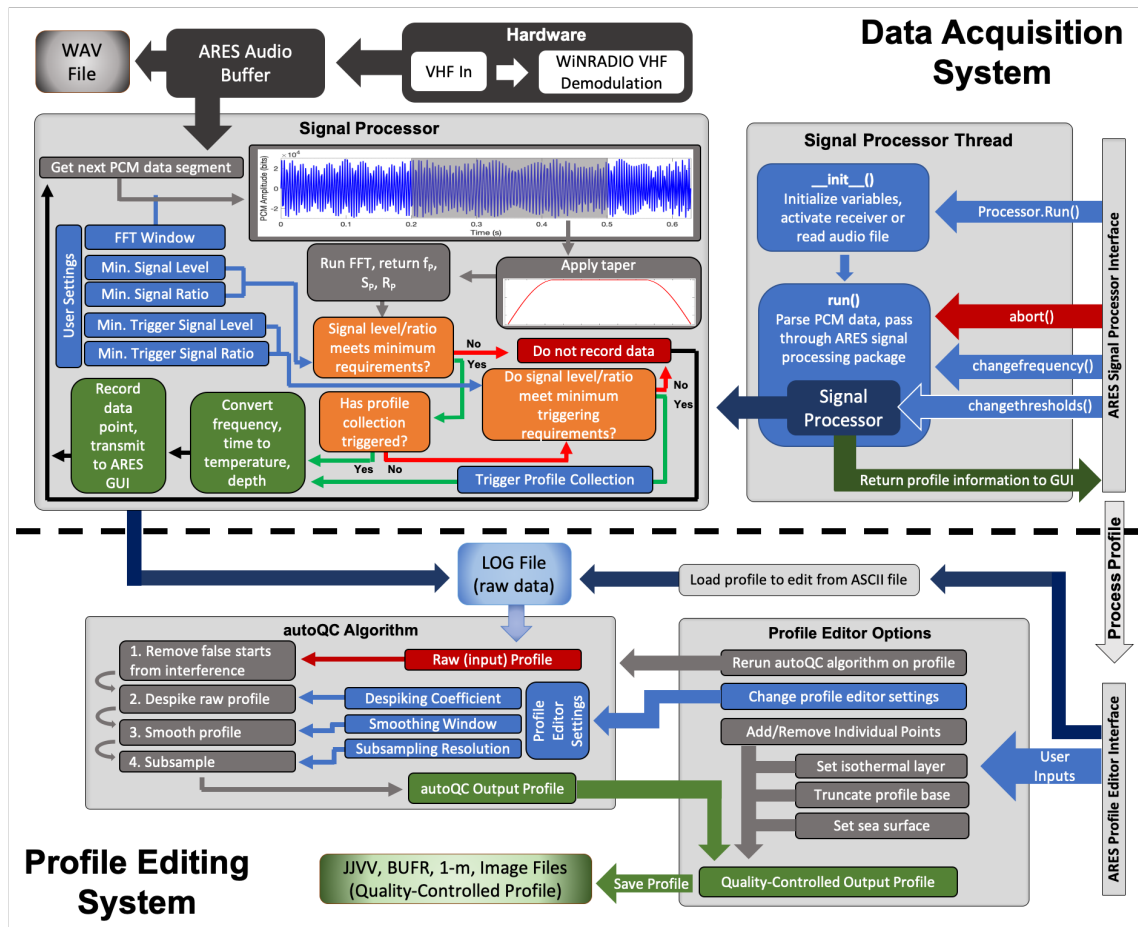
Figure 6-1: Comprehensive data flow for the AXBT Realtime Editor System, including both the Data Acquisition System (top) and Profile Editing System (bottom) separated by a horizontal dashed line.

The ARES Profile Editing System automates the vast majority of necessary profile corrections while providing a simple interface for users to make additional manual edits as necessary. The integrated autoQC algorithm corrects for common modes of VHF interference and integrates bathymetry and climatology to identify and correct additional profile discrepancies, and the interactive interface enables users to make additional edits as necessary and export quality-controlled profiles as a variety of ASCII and binary file formats. This system drastically reduces the amount of operator input necessary to quality control an AXBT profile and provides an overlay with climatological profiles, reducing both the margin for operator error and required oceanographic knowledge necessary to quality control AXBT profiles.

6.2   ARES Evaluation

6.2.1   Evaluation Against 2011-2019 TROPIC Data

The ARES Data Acquisition System performed well when evaluated against AXBT profiles generated with a MK-21 processor. For tropical Atlantic and Pacific ocean temperature-depth profiles collected during the TROPIC field program, the signal processing techniques incorporated by ARES were demonstrated to reduce spiking due to VHF interference, particularly near the surface and below 350 meters, when evaluated against the industry standard for AXBT data acquisition. This evaluation also supported the use of a 0.3 second default window length as an optimal balance between temperature resolution, computational expense, and smoothing in time.

The ARES autoQC algorithm successfully completed all necessary profile corrections for 87% of 1,177 temperature-depth profiles collected during the TROPIC field program. Additionally, using the ARES profile editing interface enabled a 9% increase in the number of profiles of sufficient quality for transmission to the GTS while reducing per-profile processing time five- to ten-fold.

6.2.2 Operational Performance

When tested operationally during TROPIC with a WiNRADIO receiver array on aerial weather reconnaissance missions through Hurricane Dorian, ARES performed exceptionally well. The Data Acquisition System successfully distinguished valid AXBT signals from noise and interference. The system's performance with the WiNRADIO receivers was similar to that with the reprocessed audio files (originally collected with a MK-10A Receiver), further justifying the capability of a WiNRADIO receiver array. Additionally, the Profile Editing System successfully quality controlled 90% of the AXBT profiles without any manual intervention necessary and enabled correction of all 41 good AXBT profiles to a level sufficient for transmission to the GTS and possible ingestion into coupled numerical weather prediction models.

Operational testing of ARES demonstrated the manpower benefits it provides while deploying AXBTs in a fast-paced environment. Integrating signal processing tasks as a software-defined function enabled simultaneous processing of multiple AXBTs, limited only by the number of available VHF radio receivers. This notably simplified the data acquisition process for collecting high resolution transects of upper ocean observations and reduced the operator workload for this task so a single user could launch, process, and quality control all AXBTs for the transect. Additionally, quality control took less than a minute per profile (comparable to approximately ten minutes per profile with the previous manual MOOS-SASEA system employed during TROPIC), reducing the work tempo and associated margin for error.

6.3 Future Work

ARES performed exceptionally well when evaluated against profiles collected during the TROPIC field program. However, room for improvement remains. The Data Acquisition System would benefit from further testing with a wider range of AXBT profiles reprocessed with the same temperature and depth conversion equations used initially. Additionally, signal processing routines can be implemented for other airborne expendable probes (e.g. AXCTDs, AXCPs).

There are two primary areas where the Profile Editing System performance can be improved and additional testing is needed. The autoQC algorithm can be improved to better identify and correct (either by more aggressive spike removal or profile truncation) regions with high profile interference. Additionally, the autoQC algorithm should be evaluated against a global dataset of temperature profiles to identify any shortcomings in quality controlling profiles with unique regional characteristics.

With these improvements, ARES can be further developed as a comprehensive, customizable system with seamless integration of data acquisition and quality control to collect ocean observations from air-launched expendable floats.

6.4 Data and Source Code Availability

Quality controlled data collected during the TROPIC field program, used to evaluate ARES, are available online at `data.nodc.noaa.gov/cgi-bin/iso?id=gov.noaa.nodc:0209221`. Individual AXBT profiles or spatial and temporal subsets, as well as additional file types, are accessible at `mmmfire.whoi.edu/tropic/`.

The AXBT Realtime Editing System (ARES) source code is available in a Github repository located at `www.github.com/cdens/ARES`. Data files not included in the source code but required for ARES to function, as well as a self-contained executable installer (for Windows 10 x64), user manual, and additional resources, are available at `mmmfire.whoi.edu/ares`.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

ARES Installation

A.1   Installing ARES

ARES can be installed by one of two methods: using the executable installer with all supporting libraries bundled together (recommended) or using the Python base and supporting all required Python modules. The steps necessary for each method are described separately in the following subsections.

A.1.1   Installing Via Executable (Recommended)

The executable (.exe) installer file can install ARES with minimal work on the part of the user. This file installs a version of ARES that has been bundled with the required Python dependencies so the computer does not require a functional Python installation (and ARES will operate independently of any Python installations). Run the installer by double clicking and enable administrator access if prompted.

Next, follow the user prompts and enter the password if prompted (if required, this should have been distributed with the file). If necessary, select where to install ARES or leave the default selection (C:\\Program Files\ARES). Follow the remaining prompts (shown left to right, top to bottom below) to install ARES. It may be necessary to install hardware drivers for ARES to recognize connected VHF receivers, following the instructions in Appendix A.2.
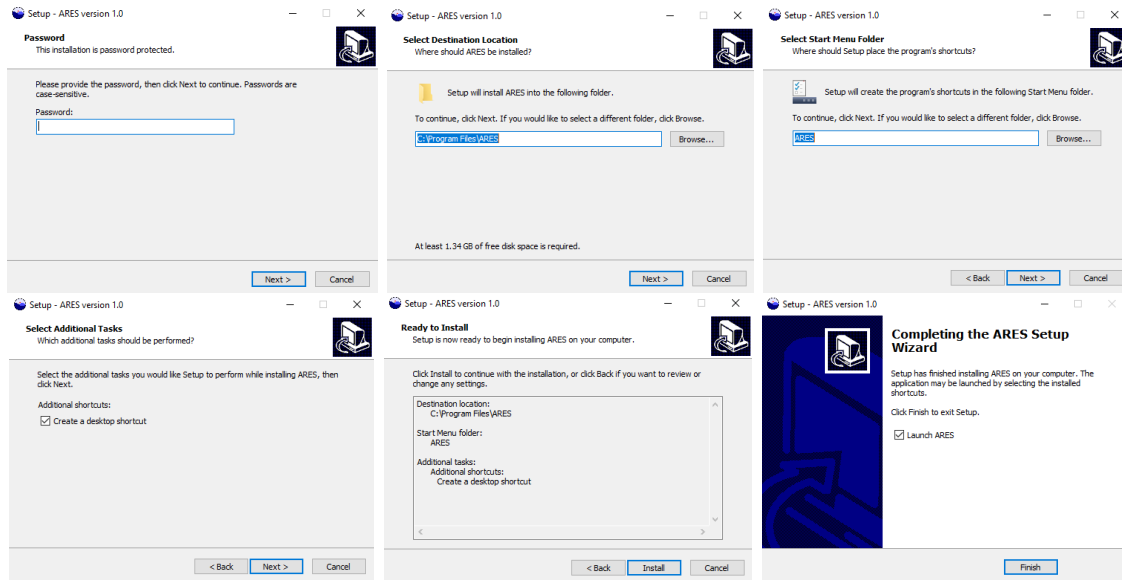
A.1.2 Installing To Use Python Base and Dependencies

ARES can be installed to operate with Python base and required modules as follows:

1. Install Python Base (version $>= 3.6.0$)
2. Extract the ARES source code, drivers, and data dependencies
3. Use pip to install supporting Python modules

Find and download the base module from Python (select a version greater than or equal to 3.5) from `www.python.org/downloads/` (the simplest method is to download and install from an executable). Next, clone the ARES repository (`git clone` ↘ `https://github.com/cdens/ARES`) to your directory. ARES requires source data files that are not included in the repository due to size constraints. These files can be downloaded as a compressed folder from `mmmfire.whoi.edu/ares` and extracted to the ARES directory as shown below.

Check to confirm that the unzipped folder contains all of the necessary files as shown in the directory tree on the next page. For brevity, climatology and bathymetry data are abridged to show the general formats of files in each directory. Additionally, WiNRADIO driver files are shown for the Windows 10 folder (Win10) only, but files with identical names exist in the Win7 and Win8 folders for use with the Windows 7 and Windows 8 operating systems, respectively).

Finally, install all dependencies for ARES using pip. This can be accomplished using the included requirements.txt file (via command line) with `pip install -r` ↘ `requirements.txt`. On Windows systems it may be necessary to download the appropriate Shapely and Cartopy wheels (for your computer's architecture and Python version) from `www.lfd.uci.edu/~gohlke/pythonlibs/` and install them separately. Install the wheel module (`pip install wheel`), navigate to the directory with the downloaded files, and install both modules from their wheel files (`pip install` ↘ `WheelFile.whl`). On MacOS and Linux systems, installing the Shapely and Cartopy modules requires the GEOS and Proj libraries (on MacOS, both libraries can be installed with Homebrew).

A.2   Installing Necessary Drivers

WiNRADIO drivers are necessary for a computer to recognize and communicate with the WiNRADIO receivers. If these are not automatically installed during the executable installation process, they must be installed manually. Currently, the necessary drivers are only available for windows and thus these instructions are Windows-specific. To check if the necessary drivers are installed, first connect at least one receiver to the computer and then open the "Device Manager" from the Start menu. The receivers will appear in the Device Manager in one of two ways (shown below). If the drivers are already installed, the receiver will appear as in the left image: under the "Radio Receivers" category, named "WiNRADIO G39WSB USB radio receiver". If the drivers are not installed, the receiver will appear as in the right image: under the "Other devices" category, named "G3USBv1.6" (in both of the screenshots below, three radio receivers were connected).
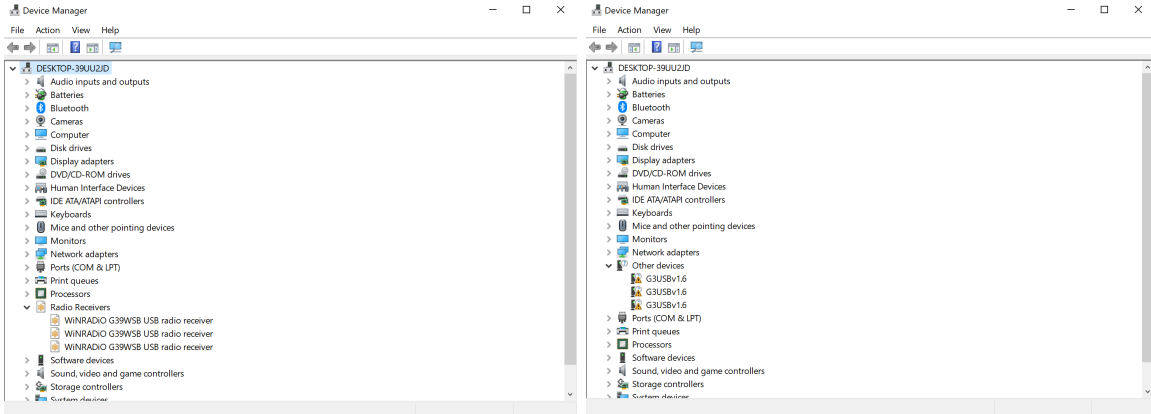
If the drivers are not installed (connected receivers appear as in the right image), complete the following steps:

1. Right-click on the radio receiver ("G3USBv1.6") and select "Update Driver"

2. Select "Browse my computer for driver software" (left image, below)

3. Select "Browse" and navigate to the ARES program folder (right image, below). If ARES was installed from the executable installer with default settings, this folder will be C:\\Program Files\ARES

4. Select "Next" and allow the computer to automatically find and install the necessary drivers

5. Upon completion, disconnect and reconnect all receivers. All connected receivers should now appear in the Device Manager as in the above image on the left (this process only needs to be completed once, not individually for each receiver).
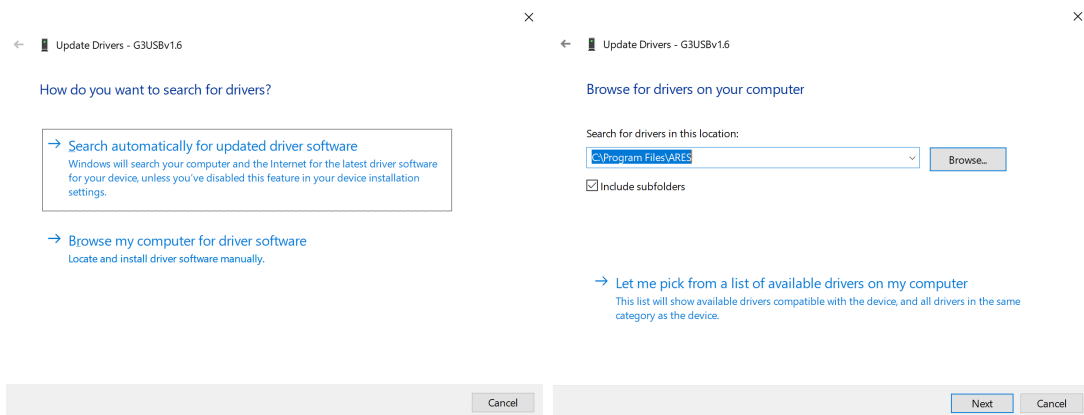
Appendix B

ARES Troubleshooting

B.1 Data Acquisition System

Common ARES Data Acquisition System (DAS) errors can be subdivided into three categories: (1) errors associated with WiNRADIO VHF receiver integration, (2) errors associated with user-defined settings for the DAS, and (3) errors associated with user inputs. These three categories of errors (and accompanying solutions) are described separately in the following subsections.

B.1.1 WiNRADIO Communication Errors

Errors associated with WiNRADIO receiver communication typically fall into one of two categories: missing hardware drivers or loose connections/loss of power. The former can be resolved by following the driver installation instructions in Section A.2. To ensure that no connections are loose, check that: (1) the receiver is powered on and receiving power (the blue light is flashing on the receiver) and (2) the serial-to-USB cable is plugged securely into the receiver and the other end is plugged into the computer (if using a USB hub with multiple devices, connect the receiver directly to the computer to confirm that there are no connection or communication issues with the hub). Common error messages or issues connecting to receivers are listed below:

Warning message: "WiNRADIO communications only supported with Windows! Processing and editing from audio/ASCII files is still available.": This message is displayed on non-Windows (e.g. MacOS, Linux) systems when first opening the program. As of this manual's writing, hardware drivers for WiNRADIO receivers are available for Windows only, and therefore receiver functionality is unavailable for MacOS or Linux. The "Test" case and audio file reprocessing, as well as profile quality control, are still fully functional on MacOS or Linux. In order to process AXBT data in real-time from a WiNRADIO with ARES on MacOS or Linux, install ARES on a Windows 10 virtual machine (realtime processing with ARES has been tested successfully in both VirtualBox and VMWare) running within the host operating system.

A radio receiver is connected to the computer and powered on, but is not appearing in the dropdown menu within an ARES Data Acquisition System tab: First, click "Refresh" to force ARES check again for any connected devices. Devices plugged into the computer after the tab was opened will not appear unless "Refresh" is selected. If the receiver is still not listed, follow the instructions for driver installation in Section A.2 to ensure that the computer can properly communicate with the receiver.

Error message: e.g. "Failed to connect to the specified receiver" or "Contact lost with WiNRADIO receiver": This message is displayed when ARES recognizes and is able to communicate with a receiver (so the drivers are most likely installed correctly) but fails to begin processing data from that receiver for some reason. The most common cause is intermittent contact with the receiver due to a loose connection between the receiver and the computer or to the receiver's power supply. Check that the WiNRADIO is receiving a constant power supply, and that the serial-to-USB connection between the WiNRADIO and the computer is secure. If the receiver is connected to the computer through a USB hub or similar, try connecting the receiver and computer directly (with a single cable) to ensure that the USB hub was not the cause of the connection issues. Finally, if alternate receivers are available, try connecting to and processing with a different receiver to ensure that the errors are not the result of a hardware issue.

Error message: "The selected WiNRADIO appears to be in use. Please select a differ-
ent device before proceeding" or similar: Each receiver is only capable of demodulating
signal from an AXBT on one channel at a time. ARES prevents users from select-
ing a receiver from the dropdown menu if that receiver is actively processing a VHF
channel in a different tab. To resolve this, find the tab currently processing with the
active receiver and either (1) continue using that tab or (2) select "Stop" to stop
processing in that tab and free the desired receiver for processing in another tab.

B.1.2   User-Defined Setting Related Errors

ARES offers a high level of control over Data Acquisition System settings, but
some combinations of these settings can result in suboptimal performance depending
on a number of factors including the computer in use, operational conditions, and the
ocean features being observed. The following issues can be resolved by adjusting the
user settings described in Section 3.1.2. Profiles with any of the following discrepancies
can be resolved by:

1. Saving the profile as an audio (WAV) file

2. Opening the preferences window (from the options menu at the top of the
   screen), selecting the "Data Acquisition System Settings" tab, and making any
   necessary changes

3. Reprocessing the profile from the WAV file by choosing "Audio" in the "Data
   Source" drop down menu, selecting "Start", and browsing to the saved WAV
   file when prompted

The Data Acquisition System begins processing but no valid AXBT signal is detected
(only blue rows appear on the table and no data appears on the plot): This occurs when
the received signal is not strong enough to trigger profile collection (the minimum and
trigger signal level and signal ratio settings are set too high). The table on the lower
right hand side of the Data Acquisition System interface lists the received signal levels
and signal-to-noise ratios under the columns labeled S_P and R_P. Identify the signal

levels and ratios for valid AXBT data by scrolling through the table and distinguishing where the profile started and finished (there should be a clear increase and decrease in both when the AXBT starts and finishes transmitting, respectively). Adjust the trigger and minimum thresholds below these values (the trigger thresholds must be satisfied for the first observation to be identified, but the minimum thresholds are required for all observations) and reprocess the profile.

The observed profile begins too soon due to interference, and the entire profile is depth shifted (e.g. the surface temperatures appear at 200m): This occurs when profile collection is incorrectly initiated ("triggered") by interference or noise. To correct for this, adjust the trigger signal level and trigger signal ratios in the settings to higher values somewhere between the observed signal levels and signal ratios of the interference and the first valid AXBT signal. Observed values of signal level and signal ratios are listed in the S_P and R_P columns (respectively) of the table in the lower right corner of the Data Acquisition System interface. It is worth noting that if the initial interference is more powerful than the AXBT signal, this will not work. In that case (or if reprocessing isn't feasible due to time or operational constraints), false starts can also be corrected using the "Depth Delay" feature in the Profile Editor interface.

The observed profile begins at the correct depth, but excessive interference (erroneous spiking) appears in the profile: This occurs when the minimum signal level and ratio (vice trigger signal level and ratio) are set too low to distinguish noise from valid AXBT signal. To correct for this, adjust the minimum signal level and ratio in the settings to higher values somewhere between the observed signal levels ratios of the interference and valid AXBT signal. Observed values of signal level and signal ratios are listed in the S_P and R_P columns (respectively) of the table in the lower right corner of the Data Acquisition System interface. If the signal level and ratio of the noise are both greater than that of the valid AXBT signal, then no settings will be able to parse valid signal from the noise. In that case (or if reprocessing isn't feasible due to time or operational delays), the "autoQC" algorithm is designed to

automatically identify and remove spikes, and the Profile Editor interface provides user options to remove individual observations or vertical ranges of observations.

The computer is lagging and/or processing extremely slowly and recording observations at too low of a vertical resolution: This occurs when the signal processing by the Data Acquisition System becomes too computationally expensive, consumes too much of the computer's resources, and takes too long to process each observation, reducing the frequency with which it can collect observations. While signal processing is inherently a more computationally expensive task, the load on the computer can be reduced by decreasing the value of the "FFT Window" setting. This setting determines the length of the signal (in seconds, ranging from 0.1 s to 1.0 s) the ARES Data Acquisition System uses to identify the peak frequency (and corresponding temperature) being transmitted from the AXBT. Shorter windows enable the Data Acquisition System to process faster but will also reduce the temperature resolution in the resulting profile.

The collected temperature-depth profile appears unnaturally discontinuous (e.g. a "stair-stepping" pattern): This can occur when the FFT Window setting, which determines the length of the signal (in seconds, ranging from 0.1 s to 1.0 s) the ARES Data Acquisition System uses to identify the peak frequency (and corresponding temperature) being transmitted from the AXBT, is too short (although this is also a valid feature common in some regions, e.g. the tropical Atlantic Ocean east of the Lesser Antilles). Low FFT Window settings have a correspondingly low temperature resolution, causing the profile to 'jump' from one temperature to the next. Increasing the window size will increase the temperature resolution but also cause the Data Acquisition System to consume more of the computer's resources and possibly reduce the frequency of observations (and corresponding vertical resolution) of the collected profile when processing in realtime. The tradeoff between temperature resolution, processing time, and window length is discussed further in Section 3.3.4.

B.1.3   Input-Associated Errors

ARES is configured to raise errors when users attempt to control the program in a way that could result in unexpected behavior. These errors are discussed below.

Error Message: "You cannot switch between Test, Audio, and RF data sources after starting processing.": To prevent multiple profiles from being concatenated together (e.g. starting with the "Test" sequence before processing from audio or in realtime with a WiNRADIO), ARES does not allow users to switch the datasource (other than switching between multiple receivers) for a tab after "Start" has been selected. In addition, ARES only allows users to select "Start" once when processing from audio, to prevent multiple audio files from being concatenated together. To start processing from a different data source, open a new Data Acquisition System tab by selecting "Options" > "New Data Acquisition System Tab".

Error Message: "Unable to read audio file": This error can be raised in a Data Acquisition System tab where the datasource is set to either Test or Audio (as both of these sources actually reprocess profiles from raw audio). If the datasource is set to test, this is because the required audio file located in the ARES program folder under 'testdata/MZ000006.WAV' is either missing or corrupted. When processing from an audio file, this can occur if the selected audio file is missing, the program does not have permission to access it, it is corrupted, or it is not a WAV file (the Data Acquisition System is only configured to reprocess audio from WAV files).

Error Message: "Selected audio file is too large!": ARES is configured to prevent users from accessing audio files that exceed more than a certain number of datapoints (the current configuration is 2.5×10^8 frames, corresponding to 0.5 GB for 16-bit PCM data and approximately 1 hour and 34 minutes of audio collected at the standard rate of 44.1 kHz) to prevent ARES from accessing too much of a computer's memory. An AXBT profile should be approximately one tenth of this size, so to resolve this make a copy of the audio file to be processed and trim to only include the audio for the profile of interest.

Warning Message: "ARES has stopped audio recording as the WAV file has exceeded maximum allowed length.": To prevent ARES from accessing too much of a computer's random access memory (RAM), the Data Acquisition System is configured to stop recording audio from a receiver after it has processed 2.5×10^8 frames of data, corresponding to slightly over one hour of recording at the standard 64 kHz sampling frequency for WiNRADIO receivers. To start recording again, select "Stop" on the tab that is recording (so the receiver can be accessed by a different tab), open a new Data Acquisition System tab and start processing with that tab.

B.2   Profile Editor

B.2.1   Error loading a profile

Error Message: "Invalid data file format.": This error is raised when the user attempts to load a file containing profile data with an unrecognized extension. The currently supported file types are: log files (.DTA), edf files (.edf), NVO and FIN files (.nvo or .fin) and JJVV (bathy) files, (.jjvv). Formats for NVO, FIN, and JJVV files are described in in 4.1.5).

Error Message: "Failed to read selected file.": This error is raised when the user attempts to load a file with a recognized extension containing a data format that does not match the extension. This can occur if the file is named with the incorrect extension or incorrectly formatted. Formatting for FIN, NVO, and JJVV files is discussed in Section 4.1.5.

Warning Message: "Profile does not contain any valid data.": This message is provided when the profile editor attempts to load a profile that does not contain any valid data, either from an ASCII file or from the Data Acquisition System. If receiving this message when attempting to transition to Profile Editor mode from the Data Acquisition System, save the raw data as an audio file, lower the minimum signal thresholds, and reprocess from that audio file as described in Section B.1.2.

Warning Message: "Invalid (latitude, longitude, date, time, etc.) entered": ARES is configured to only accept specific formats for date, time, and position. Date should be entered in year-month-day (YYYYMMDD) format and time in hour-minute (HHMM) format (both entries should contain only digits, no hyphens, colons, slashes, letters, or other characters). Latitude and longitude should also be entered as numbers (e.g. 23, 23.0, or 23.56). Latitude and longitude should be entered in decimal format or comma-delimited (with commas separating degrees, minutes, and seconds). Northern and eastern hemispheres are considered positive, and southern and western hemispheres are considered negative. For example, the longitude 55^0 33' 12" W would be entered as either -55.553 or -55,33,12. Additional checks are applied to ensure input is realistic: latitude is constrained between -90 and 90, longitude between -180 and 180, year after 1938, month between 1 and 12, and day between 1 and 31.

Error Message: "Unable to find/load (climatology or bathymetry) data": This error is raised when ARES is unable to read a supporting climatology or bathymetry data file. If the message is raised when ARES is first opened, then the file containing climatology and/or bathymetry positions (named indices.mat) failed to load. If the error appears when attempting to quality control a profile, then ARES failed to read the climatology and/or bathymetry data for that profile. In either case, this can be resolved by ensuring that the necessary data files are stored in the ARES program folder's qcdata/bathy and qcdata/climo subdirectories, following the layout shown in Appendix C.1.2.

B.2.2   Profile editing issues

Quality-controlled profile does not appear on plot (black profile exists but red profile does not): This occurs when the automated quality control checks apply changes that remove all profile data. To correct this, first check that the "Maximum Depth" field is set to zero: drops near land or where the climatology does not match the profile can trigger ARES to incorrectly identify a bottom strike. If that does not work, open the Preferences menu (by selecting Options > Preferences), and in the Profile Editor

Settings tab, uncheck the fields "Use climatology to detect bottom strikes", "Autocompare profile to climatology", "ID bottom strikes with NOAA ETOPO1 bathymetry data", and "ID false starts due to VHF interference", and then select "Re-QC Profile (Reset)" on the Profile Editor tab to reprocess the profile without any of those quality control checks.

Quality controlled (red) profile either has too many spikes or is too smoothed: These features can be corrected by adjusting the Smoothing Window and Despiking Coefficient features in the Profile Editor Settings (found by opening ARES preferences menu by selecting Options > Preferences) and then selecting "Re-QC Profile (Reset)". Decreasing the despiking coefficient causes ARES to identify more points as spikes and exclude them from the quality-controlled profile, and conversely increasing it causes ARES to include more points. Increasing the smoothing window causes ARES to apply a more intense smooth to the profile and filters out more small-scale variability, whereas increasing the window reduces the magnitude of the smooth and results in a profile with more variability.

No climatology is overlaid or the climatology does not extend as deep as (or extends deeper than) the profile: The ocean climatology is stored at a 1/4 degree horizontal resolution, and therefore does not account for sharp changes in ocean depth. Particularly when near land or a sharp change in ocean depth, the climatology can fail to include those changes in ocean depth, and in some cases no climatology data exists for profiles near land or extremely shallow areas.

B.3 Miscellaneous

Error Message: "Unable to communicate with specified COM port" This message occurs when the serial (COM) port selected as the optional GPS receiver is unreachable. Typically, either the device has become physically disconnected or lost power so it is completely unresponsive to the computer. To resolve this, open the "GPS COM Selection" tab of the ARES Preferences Menu (Options > Preferences), and select

"Update COM Port List" to refresh the dropdown menu of available devices. If the device is not listed, check to ensure it is connected and powered on. If it is, select it and choose "Refresh GPS Info" to test if the computer is able to access the devices measured latitude and longitude.

Error Message: "GPS request timed out" This message occurs when a selected device is connected and responsive but not transmitting valid GPS (NMEA) data to the computer. Typically this is because either the selected device is not a GPS receiver or the receiver is not streaming data. The GPS COM Selection settings tab lists all available serial connections, so it may require some trial and error to select the port associated with a connected receiver (ideally the port's details in the dropdown menu will contain a familiar company name or detail associated with the receiver) by checking all available ports and identifying which successfully reads position data after selecting "Refresh GPS Info". If a selected GPS receiver is causing timeout issues, it may be not streaming data because it is unable to establish a communication link with GPS satellites. To correct this, try moving the receiver (while connected to a computer) outside to a location with an unobscured view of the sky. If all else fails, check the receivers manual to ensure that any necessary drivers or additional supporting software for the receiver was installed and configured correctly.

Appendix C

ARES Source Code Description

C.1   Organization and Dependencies

C.1.1   ARES Dependencies

Python

The AXBT Realtime Editing System (ARES) was developed using Python version 3.6.8 but is viable with any Python 3.6-3.8 releases in combination with the correct module dependencies. In addition to Python base, the following modules are required as ARES dependencies: Cartopy=0.18.0, geopy=2.0.0, matplotlib=3.1.1, numpy=1.16.4, pynmea2=1.15.0, PyQt5=5.9.2, pyserial=3.4, pyshp=2.1.0, scipy=1.3.0, Shapely=1.6.4.post2

Climatology and Bathymetry Data Files

ARES uses global climatology and bathymetry data to assist with the profile quality control process. Both the climatology (Generalized Digital Environmental Model (GDEM) climatology; [19]) and bathymetry (NOAA ETOPO1 Global Relief Dataset; [20]) are integrated into ARES using MATLAB- (.mat) formatted data files. To optimize memory consumption, data are stored a large number of files subdivided by latitude and longitude (for bathymetry) and month (for climatology) as either 8- or 16-bit integers, so only relevant data are transferred to random access memory

when each AXBT profile is quality controlled. Both bathymetry and climatology use a standard naming convention so ARES can access the relevant file(s) given the month and position of interest.

The integrated GDEM climatology data includes monthly ocean temperature means and standard deviations at a 0.25^o horizontal resolution for 29 vertical levels in the upper 1,000 meters. The data are subdivided into 7,776 subfiles, each of which includes ocean temperatures and standard deviations for all vertical levels in a 10° by 10° grid for a single month. A single file is briefly loaded into the computer's memory during each AXBT profile's initial automated quality control sequence and cleared after the AXBT profile's corresponding climatology profile has been interpolated.

Global bathymetry and relief data from the NOAA ETOPO1 dataset are stored at a 1-arcminute horizontal resolution. The data are divided into 64,800 subfiles, with each file containing a 1° by 1° grid of data. When an AXBT profile is quality controlled, individual bathymetry files are loaded into random access memory as "tiles" and concatenated into a 9° longitude by 6° latitude grid surrounding the location of interest. Ocean depth at the location is interpolated, and the grid of bathymetry data are subsampled by a factor of four and contoured on the UI location plot before being cleared from memory.

Additional Data Files

ARES also requires several additional data files to operate. The International Hydrological Organization (IHO) World Seas shape file set is used to identify the region in which an AXBT was deployed, to be included in AXBT location plots for better context. This requires World_Seas_IHO_v3.xxx files in the following formats: .cpg, .dbf, .prj, .qpj, .shp, and .shx). NaN values in this dataset introduced errors that were corrected using an ASCII log of regional outlines stored in IHO_seas.txt.

Dynamically-linked library (DLL) files are also integrated to enable WiNRADIO receiver control in 32- and 64-bit Windows (Linux and MacOS are not currently supported for this feature). The application-programmer interfaces (APIs) for 32- and 64-bit Windows are stored in WRG39WSBAPI_32.dll and WRG39WSBAPI_64.dll,

respectively. Additionally, WiNRADIO receiver drivers (including .sys, .cat, and .inf files) for Windows 7, 8, and 10, are also included in the source data but are not explicitly called by ARES. Rather, the appropriate driver for the operating version of Windows must be installed before connecting to a WiNRADIO receiver.

A directory named testdata stores sample AXBT data files in various formats. A WAV file (MZ000006.WAV) is required to run the "Test" processing option in the Data Acquisition system. Additional files (e.g. ASCII files with raw profile data) are not required for ARES to run but useful for system testing and training. A colormap for bathymetry contours (generated from the cmocean modules "topo" colormap) is provided in topocolors.txt, using comma-delimited, RGB triplets with values ranging from zero to unity, with the deepest contour colors at the start and most shallow contour colors at the end of the file. Finally, the ARES icon is stored in both .png and .ico formats for use in the ARES GUI window as well as the host operating system's user interface.

Python Source Files

The ARES source code is subdivided among several files, in two different folders:

- main.py: Initializes the splash screen (Windows only) and starts the application
- ARESgui/__init__.py: Contains the Qt QMainWindow class that initializes the ARES user interface
- ARESgui/_DASfunctions.py: Contains all functions to build and control user events for the ARES Data Acquisition System
- ARESgui/_PEfunctions.py: Contains all functions to build and control user events for the ARES Profile Editor
- ARESgui/_GUIfunctions.py: Contains functions to handle menu creation, window formatting, and other generic features of the user interface
- ARESgui/_globalfunctions.py: Contains generic functions called by other functions within the above files, including tab access and file saving
- ARESgui/settingswindow.py: Contains a separate QMainWindow class to build the settings interface and handle reading and saving ARES preferences
- qclib/autoQC.py: Contains the autoQC algorithm for interference correction
- qclib/geoplotfunctions.py: Functions to handle mapping (cosine contraction corrections and axes labelling)

- qclib/GPS_COM_interaction.py: Functions to identify and interact with GPS NMEA streams through serial (RS-232) interfaces

- qclib/makeAXBTplots.py: Functions to handle AXBT profile and location plots used in the profile editor

- qclib/ocean_climo_interaction.py: Functions to handle bathymetry and climatology retrieval and comparison

- qclib/tropicfileinteraction.py: Functions to read and write to several common file formats for AXBT data

- qclib/VHFsignalprocessor.py: Contains signal processor thread class and functions to handle WiNRADIO communication and control, signal processing, and temperature/depth conversion

C.1.2   ARES Source Code File Tree

ARES required file organization is shown on the next page. For brevity, climatology and bathymetry data are abridged to show the general formats of files in each directory. Additionally, WiNRADIO driver files are shown for the Windows 10 folder (Win10) only, but files with identical names exist in the Win7 and Win8 folders for use with the Windows 7 and Windows 8 operating systems, respectively).

- ARES Program Folder
  - main.py
  - requirements.txt
  - License_GNU_GPL_v3.txt
  - qclib
    - autoQC.py
    - geoplotfunctions.py
    - GPS_COM_interactions.py
    - makeAXBTplots.py
    - ocean_climo_interaction.py
    - tropicfileinteraction.py
    - VHFsignalprocessor.py
    - topocolors.txt
    - dropicon.ico
    - dropicon.png
  - ARESgui
    - __init__.py
    - _DASfunctions.py
    - _PEfunctions.py
    - _GUIfunctions.py
    - _globalfunctions.py
    - settingswindow.py
  - testdata
    - 201508261140.DTA
    - MZ000006.WAV
  - qcdata
    - driver
      - Win7
      - Win8
      - Win10
        - WRG39WSB_XP32.sys
        - WRG39WSB_XP64.sys
        - WRG39WSB.cat
        - WRG39WSB.inf
    - bathy
      - incides.mat
      - b_N23_E89.mat
      - b_N(lat)_E(lon).mat
    - climo
      - incides.mat
      - c_M3_N10_E30.mat
      - c_M(month)_N(lat)_E(lon).mat
    - regions
      - IHO_seas.txt
      - World_Seas_IHO_v3.shp
      - World_Seas_IHO_v3.cpg
      - World_Seas_IHO_v3.dbf
      - World_Seas_IHO_v3.prj
      - World_Seas_IHO_v3.shx
    - WRG39WSBAPI_32.dll
    - WRG39WSBAPI_64.dll

C.2   Compilation and Installer Generation

C.2.1   Compiling Source Code with PyInstaller

The ARES source python code was compiled into an executable using PyInstaller (version=3.5, with Python 3.6.8). PyInstaller collects the necessary compiled python (.pyc) files for all Python module dependencies, as well as any other binaries or other program dependencies (this does not include referenced data files, such as bathymetry or climatology in this case). The output is a directory containing a primary executable (in this case, main.exe, generated from main.py), with additional dependencies (e.g. compiled Python and other binaries) in that directory and subdirectories. Any additional Python source files (e.g. the additional files in ARES), as well as any other files (e.g. bathymetry and climatology data files) must be moved into the same file path relative to main.exe (so a required file in the same directory as main.py must be moved into the same directory as main.exe, and so forth for files in subdirectories).

PyInstaller generates the final executable using a series of settings in a "spec" file (e.g. main.spec). If one does not exist, the spec file is autogenerated when PyInstaller is called (e.g. `pyinstaller main.py` will generate a spec file, main.spec). Future instances of PyInstaller can be run from that same spec file with `pyinstaller↘ main.spec`). The spec file contains specific settings for executable generation (e.g. whether running the executable will open a console and whether to bundle all code into a single executable). The contents of the spec file for ARES are listed below (filepath-specific information replaced):

```
1  # -*- mode: python ; coding: utf-8 -*-
2  block_cipher = None
3  a = Analysis(['main.py'],
4              pathex=['C:\\Users\\$USERNAME\\$FILEPATH\\AXBT'],
5              binaries=[],
6              datas=[],
7              hiddenimports=[],
8              hookspath=[],
9              runtime_hooks=[],
10             excludes=[],
11             win_no_prefer_redirects=False,
12             win_private_assemblies=False,
```

```
13            cipher=block_cipher,
14            noarchive=False)
15  pyz = PYZ(a.pure, a.zipped_data, cipher=block_cipher)
16  exe = EXE(pyz,
17            a.scripts,
18            [],
19            exclude_binaries=True,
20            name='main',
21            debug=False,
22            bootloader_ignore_signals=False,
23            strip=False,
24            upx=True,
25            console=False )
26  coll = COLLECT(exe,
27               a.binaries,
28               a.zipfiles,
29               a.datas,
30               strip=False,
31               upx=True,
32               upx_exclude=[],
33               name='main')
```

C.2.2   Generating Installer with Inno Script Setup

The compiled source code was packaged into an executable installer using Inno Script Setup. Inno Script Setup uses Inno scripts (extension .iss) to specify information including source path, files to compile, and icons to integrate. More advanced capabilities are optional, such as encrypting the installer and requiring a PIN to access, and executing other commands (e.g. driver installation) during the installation process. An example of the Inno script used to generate an installer for ARES is provided below (file paths and PINs removed for security purposes).

```
 1  [Setup]
 2  AppId={{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}
 3  AppName=ARES
 4  AppVersion=1.0
 5  ;AppVerName=ARES 1.0
 6  AppPublisher=Casey Densmore
 7  DefaultDirName={pf}\ARES
 8  DefaultGroupName=ARES
 9  OutputBaseFilename=ARES_win64
10  SetupIconFile=C:\PATH\ARES\qclib\dropicon.ico
11  Password=NNNN
12  Compression=lzma
13  SolidCompression=yes
14  ArchitecturesInstallIn64BitMode=x64
15  [Tasks]
16  Name: "desktopicon"; Description: "{cm:CreateDesktopIcon}"; ↘
        GroupDescription: "{cm:AdditionalIcons}"; Flags: unchecked
17  Name: "quicklaunchicon"; Description: "{cm:↘
        CreateQuickLaunchIcon}"; GroupDescription: "{cm:↘
        AdditionalIcons}"; Flags: unchecked; OnlyBelowVersion: ↘
        0,6.1
18  [Files]
19  Source: "C:\PathToExecutable\ARES.exe"; DestDir: "{app}"; ↘
        Flags: ignoreversion
20  Source: "C:\PathToExecutable\*"; DestDir: "{app}"; Flags: ↘
        ignoreversion recursesubdirs createallsubdirs
21  [Icons]
22  Name: "{group}\ARES"; Filename: "{app}\ARES.exe";  ↘
        IconFilename: "{app}\qclib\dropicon.ico"
23  Name: "{group}\{cm:UninstallProgram,ARES}"; Filename: "{↘
        uninstallexe}";  IconFilename: "{app}\qclib\dropicon.ico"
24  Name: "{commondesktop}\ARES"; Filename: "{app}\ARES.exe"; ↘
        Tasks: desktopicon; IconFilename: "{app}\qclib\dropicon.↘
        ico"
25  Name: "{userappdata}\Microsoft\Internet Explorer\Quick Launch↘
        \ARES"; Filename: "{app}\ARES.exe"; Tasks: quicklaunchicon↘
        ; IconFilename: "{app}\qclib\dropicon.ico"
26  [Run]
27  Filename: "{app}\ARES.exe"; Description: "{cm:LaunchProgram,↘
        ARES}"; Flags: nowait postinstall skipifsilent;
```

C.3 Source Code Examples

C.3.1 Signal Processing

The following code segments are examples of the ARES signal processor source code. The signal processor package includes support for controlling and receiving data from WiNRADIO software-defined radio receivers (using the Python ctypes module and WiNRADIO API) and GPS NMEA feeds. Context for these examples is provided in Section 3.2.

Connecting to a WiNRADIO Receiver

```
1  import ctypes
2  wrdll = ctypes.windll.LoadLibrary(pathtoDLLfile)
3  serialnum = 'H3QG7B'
4  serialnum_conv = ctypes.c_char_p(serialnum.encode('utf -8'))
5  hradio = wrdll.Open(serialnum_conv)
```

Using SetupStreams to Direct PCM Data Flow

```
1  from ctypes import (CFUNCTYPE, POINTER, c_void_p, c_ulong,
2      c_int16, cast)
3  global f_s, audiostream #alternatively attach to class (self.
       f_s, self.audiostream)
4  audiostream = []
5  #Declaring the callbuck function to update the audio buffer
6  @CFUNCTYPE(None, c_void_p, c_void_p, c_ulong, c_ulong)
7  def updateaudiobuffer(_,bufferpointer_int,size,samplerate):
8      bufferlength = int(size / 2)
9      bufferpointer = cast(bufferpointer_int,POINTER(c_int16 *
           bufferlength))
10     bufferdata = bufferpointer.contents
11     f_s = samplerate
12     audiostream.extend(bufferdata[:])
13     del audiostream[:bufferlength] #remove data from start
14 #setting up stream to radio opened in previous example
15 wrdll.SetupStreams(hradio,None,None,updateaudiobuffer,None)
```

Preallocating and Filling a Structure with Receiver Information

```python
import ctypes
class Features(ctypes.Structure):
    _pack_ = 1
    _fields_ = [("ExtRef", c_uint32), ("FMWEnabled", c_uint32)↘
        ,("Reserved", c_uint32)]
class RADIO_INFO2(ctypes.Structure):
    _pack_ = 1
    _fields_ = [("bLength", c_uint32), ("szSerNum", c_char*9), ↘
        ("szProdName", c_char*9), ("MinFreq", c_uint32),("↘
        MaxFreq", c_uint32), ("Features", Features)]
# creating array of RADIO_INFO2 structures
radiolistarray = (RADIO_INFO2 * 50)()
radiolistpointer = ctypes.pointer(radiolistarray)
radiolistsize = getsizeof(radiolistarray)
radiolistinfosize = ctypes.c_int(0)
radiolistinfosizepointer = pointer(radiolistinfosize)
#loading and calling GetRadioList function from API
wrdll = ctypes.windll.LoadLibrary($path-to-DLL-file)
numradios = wrdll.GetRadioList(radiolistpointer,radiolistsize↘
    ,radiolistinfosizepointer)
# getting list of all connected winradios
winradioserials = []
for i in range(numradios):
    currentserial = radiolistarray[i].szSerNum.decode('utf-8')
    winradioserials.append(currentserial)
```

Listing All Available Serial Connections

```python
from serial.tools import import list_ports
def listcomports():
    portnums = []
    portinfo = []
    ports = list_ports.comports()
    for pnum, sdesc, details in sorted(ports):
        portnums.append(pnum) #pnum = port ID, sdesc = ↘
            description
        portinfo.append("{}:{}".format(pnum, sdesc))
    return portnums,portinfo
```

Accessing and Parsing Position Data from a GPS NMEA Feed

```python
from serial import Serial
from pynmea2 import parse
def getcurrentposition(port, baudRate, numattempts):
  with Serial(port, baudRate, timeout=1) as ser:
    nmeaobj = parse(ser.readline().decode('ascii',errors='↘
        replace').strip())
    lat = nmeaobj.latitude
    lon = nmeaobj.longitude
    dt = nmeaobj.datetime
```

C.3.2 Profile Editing

The following code segments are examples of source code for the ARES profile editor. Specifically, examples here are components of the autoQC algorithm to identify and remove common modes of VHF interference from a raw temperature-depth profile before displaying the profile to the user for further quality control. Context for these code examples is provided in Section 4.2.

VHF Interference Gap Detection

```
1  #function to identify and remove gaps due to false starts ↘
       from interference
2  def removegaps(rawdepth,rawtemp):
3      donegapcheck = False
4      while not donegapcheck:
5          maxcheckdepth = 50 #only checks the upper 50m of the ↘
               profile
6          maxgapdiff = 10 #if gap is larger than this range (m)↘
               , correct profile
7          isgap = [0]
8          for i in range(1,len(rawdepth)):
9              if (rawdepth[i] >= rawdepth[i-1]+ maxgapdiff) and↘
                   (rawdepth[i-1] <= maxcheckdepth): #if there ↘
                   is a gap of sufficient size to correct
10                 isgap.append(1) #NOTE: the logical 1 is ↘
                       placed at the first depth AFTER the gap
11             else:
12                 isgap.append(0)
13
14         #if there are gaps, find the deepest one and correct ↘
               t/d profile with that depth as the surface (only ↘
               works with linear fall rate equation)
15         if np.sum(isgap) > 0:
16             lastgap = np.max(np.argwhere(isgap))
17             realstartdepth = rawdepth[lastgap]
18             rawtemp = rawtemp[lastgap:]
19             rawdepth = rawdepth[lastgap:]-realstartdepth
20         else: #otherwise, exit loop
21             donegapcheck = True
22
23     return rawdepth,rawtemp
```

VHF Interference Spike Removal

```python
#removes spikes from profile with depth-based standard \
    deviation filter
def rundespiker(rawdepth,rawtemp,maxdev):
    temp_despike = np.array([])
    depth_despike = np.array([])

    depthwin = 5 #range of spiker is +/- 5 meters
    maxdepth = np.max(rawdepth)

    for n,cdepth in enumerate(rawdepth):

        #assigning region for running standard deviation \
            filter
        if cdepth <= depthwin:
            goodindex = np.less_equal(rawdepth,depthwin)
        elif cdepth >= maxdepth - depthwin:
            goodindex = np.greater_equal(rawdepth, maxdepth-\
                depthwin)
        else:
            ge = np.greater_equal(rawdepth, cdepth - depthwin\
                ) #all depths above bottom threshold
            le = np.less_equal(rawdepth, cdepth + depthwin) #\
                all depths below top threshold
            goodindex = np.all([ge,le],axis=0) #all depths \
                with both requirements satisfied

        #pulling subset
        tempspike = rawtemp[goodindex]

        #mean and standard deviation of current range
        curmean = np.mean(tempspike)
        curstd = np.std(tempspike)

        #only retain values within +/- 1 standard deviation \
            of running mean or top 10 m
        if abs(rawtemp[n]-curmean) <= maxdev*curstd or \
            rawdepth[n] < 10:
            depth_despike = np.append(depth_despike,rawdepth[\
                n])
            temp_despike = np.append(temp_despike,rawtemp[n])

    return depth_despike,temp_despike
```

Profile Smoothing

```python
#run depth-based smoother- ensures that first and last ↘
    datapoints match original profile
def runsmoother(depth_despike,temp_despike,smoothlev):

    temp_smooth = np.array([])
    depth_smooth = depth_despike.copy()
    mindepth = np.min(depth_despike)
    maxdepth = np.max(depth_despike)

    for n,cdepth in enumerate(depth_despike):
        if cdepth == mindepth or cdepth == maxdepth: #if ↘
            first or last point in profile, append current ↘
            temperature
            temp_smooth = np.append(temp_smooth,temp_despike[↘
                n])

        else: #otherwise- average a range of points
            if cdepth <= smoothlev/2: #in top of profile
                cursmoothlev = 2*cdepth
            elif cdepth >= maxdepth - smoothlev/2: #in bottom↘
                of profile
                cursmoothlev = 2*(maxdepth - cdepth)
            else: #in middle of profile
                cursmoothlev = smoothlev

            ge = np.greater_equal(depth_despike, cdepth - ↘
                cursmoothlev/2)
            le = np.less_equal(depth_despike,cdepth + ↘
                cursmoothlev/2)
            goodindex = np.all([ge,le],axis=0)

            #append mean of all points in range as next ↘
                datapoint
            temp_smooth = np.append(temp_smooth,np.mean(↘
                temp_despike[goodindex]))

    return depth_smooth,temp_smooth
```

Profile Subsampling

```python
#subsample profile
def subsample_profile(depth_smooth,temp_smooth,profres):

    dtdz = [] #calculating profile slope (append 0 at start
        and end so length matches that of depth_smooth)
    dtdz.append(0)
    for i in range(1,len(temp_smooth)-1): #need to use range
        here because we are only interested in a subset of
        indices
        #dtdz = (t3 - t1)/(z3 - z1): centered on z2
        dtdz.append(((temp_smooth[i+1] - temp_smooth[i-1])/
            (depth_smooth[i+1]-depth_smooth[i-1])))
    dtdz.append(0)

    depth = []
    temperature = []
    lastdepth = -100000 #large enough value so subsampler
        will always grab first datapoint
    for i,cdepth in enumerate(depth_smooth):
        #constraint on first derivative of temp with depth (
            no unrealistic spikes), if the point is a critical
             value given the selected resolution level, append
             to output profile
        if dtdz[i] <= 0.5 and cdepth-lastdepth >= profres:
            depth.append(cdepth)
            temperature.append(temp_smooth[i])
            lastdepth = cdepth

    return depth,temperature
```

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] G. Rovira-Melendez, S.J. Sun, J.L. Drogowski, S.G. McAllister, C.R. Densmore, S.R. Jayne, and E.R. Sanabia. The TROPIC Data Archive: Ready access to 9 years of AXBT data collected during USAF 53rd Weather Reconnaissance Squadron missions. Poster Presentation, Ocean Sciences Meeting, 2020.

[2] J.D. Boyd and R.S. Linzell. Evaluation of the Sparton tight-tolerance AXBT. *J. Atmos. Oceanic Technol.*, 10:892–899, 1993, https://www.doi.org/10.1175/1520-0426(1993)010<0892:eotstt>2.0.co;2.

[3] D.M. Legler, H.J. Freeland, R. Lumpkin, G. Ball, M.J. McPhaden, S. North, R. Crowley, G.JH. Goni, U. Send, and M.A. Merrifield. The current status of the real-time in situ Global Ocean Observing System for operational oceanography. *J. Oper. Oceanogr.*, 8(S2):189–200, 2015, https://www.doi.org/10.1080/1755876x.2015.1049883.

[4] E.R. Sanabia, B.S. Barrett, P.G. Black, S. Chen, and J.A. Cummings. Real-time upper-ocean temperature observations from aircraft during operational hurricane reconnaissance missions: AXBT demonstration project year one results. *Wea. Forecasting*, 28:1404–1422, 2013, https://www.doi.org/10.1175/WAF-D-12-00107.1.

[5] J.M. Bane and M.H. Sessions. A field performance test of the Sippican deep aircraft-deployed expendable bathythermograph. *J. Geophys. Res.*, 89:3615–3621, 1984, https://www.doi.org/10.1029/jc089ic03p03615.

[6] J.D. Boyd. Improved depth and temperature conversion equations for Sippican T-5 XBTs. *J. Atmos. Oceanic Technol.*, 4:545–551, 1987, https://www.doi.org/10.1175/1520-0426(1987)004<0545:idatce>2.0.co;2.

[7] K.E. Grempler. An improved airborne ocean temperature acquisition display and analysis system. *Johns Hopkins APL Tech. Digest*, 14(3):253–258, 1993.

[8] R.H. Heinmiller, C.C. Ebbesmeyer, B.A. Taft, D.B. Olson, and O.P Nikitin. Systematic errors in expendable bathythermograph (XBT) profiles. *Deep Sea Res.*, 30A:1185–1197, 1983. https://doi.org/10.1016/0198-0149(83)90096-1.

[9] D.P. Alappattu and Q. Wang. Correction of depth bias in upper-ocean temperature and salinity profiling measurements from airborne expendable probes. *J.*

Atmos. Oceanic Technol., 32:247–255, 2015, https://www.doi.org/10.1175/jtech-d-14-00114.1.

[10] D. Roemmich and B. Cornuelle. Digitization and calibration of the expendable bathythermograph. *Deep Sea Res.*, 34A:299–307, 1987. https://doi.org/10.1016/0198-0149(87)90088-4.

[11] L. Cheng, J. Abraham, G. Goni, T. Boyer, S. Wijffels, R. Cowley, V. Gouretski, F. Reseghetti, S. Kizu, S. Dong, F. Bringas, M. Goes, L. Houpert, J. Sprintal, and J. Zhu. XBT science: Assesment of instrumental biases and errors. *Bull. Amer. Meteor. Soc.*, pages 923–933, 2016, https://www.doi.org/10.1175/bams-d-15-00031.1.

[12] A.E. Gent. An evaluation of the airborne expendable bathythermograph (AXBT, SSQ-36 BTS). Technical Report TR-277, Naval Oceano. Office, NSTL, MS, 1982.

[13] J.L. Hanson. An overview of the System for At-Sea Environmental Analysis (SASEA). *Inst. Electrical Electronics Eng.*, 5, 1989, https://www.doi.org/10.1109/oceans.1989.587139.

[14] R. Bailey, A. Gronell, H. Phillips, E. Tanner, and G. Meyers. Quality control cookbook for XBT data. Technical Report 221 v1.1, CSIRO Marine Laboratories, 1994.

[15] A. Fisher. Oceanographic analysis manual for on-scene prediction systems. Technical Report RP-20, Naval Oceanographic Office, 1978.

[16] G.K. Kloss. Automatic C library wrapping- Ctypes from the trenches. *Res. Lett. Inf. Math. Sci.*, 13, 2009.

[17] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

[18] M.H. Sessions, T.P. Barnett, and W.S Wilson. The airborne expendable bathythermograph. *Deep. Sea Res.*, 23:779–782, 1975, https://www.doi.org/10.1016/S0011-7471(76)80021-6.

[19] NRL. Generalized Digital Environmental Model (GDEM), 2009. Data retrieved from `https://nrlgodae1.nrlmry.navy.mil/ftp/outgoing/static/ocn/gdem/`.

[20] C. Amante and B.W. Eakins. ETOPO1 1 arc-minute global relief model: Procedures, data sources and analysis, 2004. NOAA Technical Memorandum NESDIS NGDC-24.

[21] L.A. Avila, S.R. Stewart, R. Berg, and A.B. Hagen. National Hurricane Center Tropical Cyclone Report: Hurricane Dorian (AL052019). Technical report, NOAA National Weather Service, 2020.

[22] E.R. Sanabia and S.R. Jayne. Ocean observations under two major hurricanes: Evolution of the response across the storm wakes. *AGU Adv.*, 2020. In Press.