Experimental Evaluation of JANUS Fast Modes in Very High Acoustic Frequency Bands

Jinfeng Li and Y. Rosa Zheng, *Fellow, IEEE,*

Abstract

This paper presents extensive field tests of a fast mode operation of the JANUS acoustic communication standard where the signals occupy high frequency bands of 38 kHz spanning from 96 kHz to 134 kHz instead of the nominal frequency band of 4.1 kHz spanning 9.44 kHz to 13.6 kHz specified in the standard. The fixed 32-chip preamble and the 144-chip baseline JANUS packet utilize the Frequency-Hopped (FH) Binary Frequency Shift Keying (BFSK) with 13 frequency pairs as defined in the standard, while the cargo packets use the single-carrier M-ary Phase Shift Keying (M-PSK) modulation with a center frequency of 115 kHz and a symbol rate of 23 ksps or up to 34.5 kbps information data rate with high-order PSK and rate-1/2 forward error correction codes. The original JANUS receiver algorithm is modified to improve the frame/symbol synchronization for the fast mode and increase the decoding success rate of the baseline JANUS packet in difficult multipath channels. More than ten experiments were conducted using a Field Programmable Gate Array (FPGA)-based hardware platform consisting of a single transmit projector and a single receive hydrophone. The experiment results show that the JANUS fast mode worked well with both the original JANUS receiver algorithm and the modified receiver algorithm, yielding zero-bit error in most of the baseline JANUS packets. The modified receiver algorithm is able to reduce 30% of error packets to zero error in the experiment experiencing difficult multipath channels where the original JANUS receiver algorithm suffers from large Bit Error Rates (BER). Meanwhile, the cargo packets utilize the Linear Minimum Mean Square Error (LMMSE) Turbo equalizer and achieve a BER around 10^{-3}.

Index Terms

JANUS, Fast Mode, M-PSK, Underwater Acoustic Communications, FPGA, Field Experiments.

I. Introduction

JANUS is a physical-layer standard [1], [2] defined by the North Atlantic Treaty Organization (NATO) for underwater acoustic (UWA) communications. It aims to offer the inter-operation capability between modems from different manufacturers. JANUS has been used in many underwater scenarios, for example, first contact and language switching, underwater automatic identification system (AIS), underwater meteorology and oceanography (METOC), and distressed submarine (DISSUB) operations [3]. Several field experiments have been conducted to test the JANUS interoperability [4]–[7]. JANUS has also been studied in [8] for its feasibility of improving the operational safety of Autonomous Underwater Vehicles (AUVs) and in [9] for the co-existence with underwater telephony.

To achieve robustness against the harsh underwater acoustic environment, JANUS selects Frequency-Hopping (FH) Binary Frequency Shift Keying (BFSK) and a rate 1/2 convolutional code as the modulation and coding scheme (MCS). The original JANUS standard [1] divides the frequency band of 9.44 kHz – 13.6 kHz into 13 pairs of evenly-spaced frequency tones and maps binary data bits into one of the tone pairs in a pseudo-random fashion. The 64-bit baseline JANUS packet consists of a user class identifier, an application data block (ADB) determined by the user, CRC bits, and other indication flags. The 64-bit baseline JANUS packet maps into 144 chips by a 1/2 rate convolutional encoder with constraint length 9. Three optional wake-up tones and a fixed 32-chip preamble are added before the baseline JANUS packet for wake-up and synchronization purposes, respectively. The optional cargo packets using the same MCS are appended after the baseline JANUS packet with or without a gap.

The authors are with the Department of Electrical & Computer Engineering, Lehigh University, Bethlehem, PA 18015, USA, Email: {jili18,yrz218}@lehigh.edu.

However, the typical chip duration defined in the standard is 62.5 ms resulting in a low information bit rate of 80 bps. To increase data rate without losing the inter-operation capability, the JANUS fast mode has been suggested in [10] as a potential standard evolution, where a wider frequency band can be adopted and some high-order modulation schemes, such as M-ary Phase Shift Keying (M-PSK), may be selected for the cargo packets. To the best knowledge of the authors, very limited works have been focused on the JANUS fast mode, and an extensive literature search has found a handful of publications. For example, the work in [11], [12] tested a higher center frequency set at 49 kHz and with different bandwidth settings such as 4 kHz, 8 kHz, 12 kHz, 16 kHz, and 20 kHz. The test results show that the system experienced significant bit errors when the bandwidth exceeded 8 kHz. Another work from NATO [13] tested several high-order modulation schemes with frequency bands around 10 kHz but without FEC coding schemes. The achieved maximum bit rate in [13] was 12 kbps with 8-PSK modulation. In [14], an adaptive selection among JANUS, BPSK, QPSK, and OFDM based on the environmental information was proposed for long-range underwater acoustic communication. The frequency range set in the work was from around 900 to 1500 Hz, and the maximum data rate was 466 bps with QPSK. In addition, the work in [15] investigated the performance of FH-BFSK and M-PSK in the 300 - 400 Hz acoustic band. A maximum data rate of 200 bps was demonstrated in the field experiment at a distance of 33 km. The different modulation schemes were separately implemented on different transceivers. In [16], both JANUS and OFDM were supported by the proposed modem that operates in the frequency range of 18 kHz – 34 kHz. The achieved data rate was 20 kbps with OFDM. A recent work by a commercial modem adopts the M-PSK for cargo packets and uses the original JANUS parameters for the baseline JANUS packet [17].

This work uses transducers operating in a wide frequency band of 38 kHz spanning from 96 kHz to 134 kHz and adopts the frequency band for both the JANUS header (consisting of optional wake-up tones, the fixed 32-chip preamble, and 144-chip baseline JANUS packet) and the cargo packets. The center frequency is set to 115 kHz, and different bandwidths are selected from 22 kHz to 38 kHz. All other parameters are the same as those in the original JANUS sample code. Therefore, the information bit rate of the baseline JANUS packet rises from 80 bps to 423 bps – 730 bps, respectively. A comparison between the JANUS standard and the parameters used in the fast modes is listed in Table I, where the sampling frequency for the standard is 44.1 kHz and for the fast mode in this work is 460 kHz.

TABLE I
PARAMETERS OF BASELINE JANUS PACKETS

Parameter	***Standard***	***Fast Mode Parameters (this paper)***				
center frequency (kHz)	11.520	115				
bandwidth (kHz)	4.160	22	26	30	34	38
sample per chip	276	544	460	399	352	315
chip duration (ms)	6.2585	1.1826	1	0.8674	0.7652	0.6848
bit rate (bps)	80	423	500	576	653	730

In addition, single-carrier M-PSK modulation is adopted for the cargo packets in the fast mode. Instead of appending directly to the baseline JANUS packet, a guard gap is added between the baseline JANUS packet and cargo packets. The signaling structure of a packet in the JANUS fast mode is shown in Fig. 1. The wake-up tone defined in the standard was omitted in this work. The cargo packet consists of pilot symbols for channel estimation and data payload. The symbol rate of the cargo packets is designed at 23 kilo-symbol-per-second (ksps) in this work. Therefore, the maximum bit rates of the cargo packets are 11.5 kbps for BPSK, 23 kbps for QPSK, and 34.5 kbps for 8-PSK.

A Field Programmable Gate Array (FPGA)-based hardware platform has been developed to test the proposed JANUS fast mode, and a series of field experiments have been conducted in two fresh-water lakes and a saltwater ocean field station. The results show that the original JANUS receiver algorithm worked well in easy channels but suffered from high bit error rates in difficult multipath channels. A modified

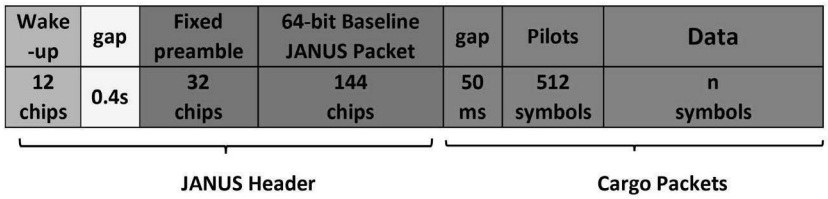


Fig. 1. The signaling structure of a packet in the JANUS fast mode.

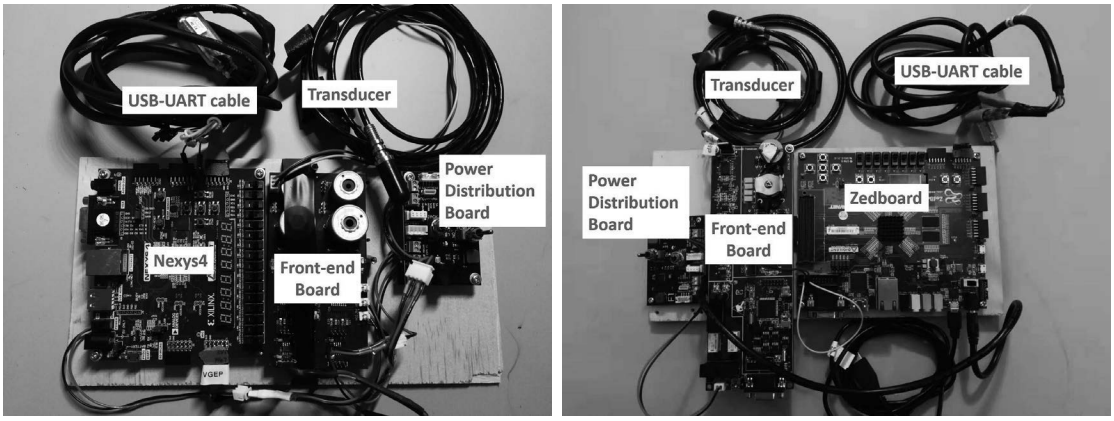
receiver algorithm is proposed, utilizing the information from the fixed 32-chip preamble to improve the frame/symbol synchronization in difficult multipath channels and increase the decoding success rate. In the experiments experiencing difficult multipath channels, the modified receiver algorithm reduces 30% of error packets to zero error. Meanwhile, for cargo packets, the LMMSE Turbo equalizer takes advantage of the synchronization and the Doppler spread estimated by the JANUS header, and achieves 10^{-3} BER in most of the experiments.

The main contributions of this paper are summarized as follows:

- A JANUS fast mode operating on a wide bandwidth of up to 38 kHz is proposed, where the baseline JANUS packet uses frequency hopping spread spectrum with 13 pairs of frequencies to achieve up to 730 bps, and the cargo packets adopt M-PSK (M=2, 4, and 8) single-carrier modulation and achieve a maximum information bit rate of 34.5 kbps.
- An FPGA-based hardware platform is developed to test the JANUS fast mode with flexible parameters and settings. Other than the power amplifier and the Low Noise Amplifier (LNA), most of the functions in both transmitter and receiver are implemented in digital forms, providing flexibility to test different physical layer signaling schemes.
- A series of field experiments have been conducted in two fresh-water lakes and an ocean coast, and the experimental data have been analyzed in detail. A modified receiver algorithm is proposed that utilizes multiple peaks in the cross-correlation between the fixed 32-chip preamble and the received signal to improve the frame/symbol synchronization in difficult multipath channels. Insights are provided on how the JANUS standard can be adopted to wide band systems for fast operation modes.

II. Experiment Platform

The hardware platform developed for the field experiments is shown in Fig. 2, where the transmitter consists of a Nexys-4 FPGA board, a front-end board with a Class-D Power Amplifier (PA) and impedance match circuit, a power distribution board, and a high-speed FTDI USB-UART cable. The receiver includes a data acquisition and transfer (DAT) system implemented on a Zedboard with a high-speed FTDI USB-UART cable, a power distribution board, and a front-end board consisting of a low-noise amplifier (LNA) and a band-pass filter. The bandwidth of the band-pass filtering is up to 38 kHz. The LNA supports single-ended or differential inputs and offers up to 46 dB amplification. The omnidirectional acoustic transducers are BT-1201 from B-Tech Acoustics for both the transmitter and the receiver. The Nexys-4 board has a Xilinx Artix-7 FPGA XC7A100T-1CSG324C with 15,850 Slices, 135 36Kb BRAMs, and 240 DSP48Es [18]. The Zedboard has a Xilinx Zynq 7000 XC7Z020-1CLG484CES All Programmable System on a Chip (AP SoC) FPGA with 85,000 Series-7 Programmable Logic (PL) cells and a dual Cortex A9 ARM Processing System (PS), a dual 16-bit Analog to Digital Converter (XADC) with up to 1 Mega-sample-per-second (MSPS) sampling rate [19], an I2S Audio Codec with 8 kHz to 96 kHz sampling rate, a USB-UART bridge, and many interface ports. In principle, the Zedboard is capable to implement both the transmitter and the receiver, while the Nexys-4 board is limited in adding ADC capability required for the receiver. Nevertheless, low cost, low power, and small form factor make the Nexys-4 board a good choice for the transmitter. In addition, the high-speed FTDI USB-UART cable C232HD-DDHSP-0 supports up to 12 Mbaud data transfer rate [20] and is used between a host computer and the transmitter or between a host computer and the receiver.



(a) (b)

Fig. 2. The hardware platform used in the field experiments. (a) transmitter and (b) receiver.

The block diagram of the transmitter is shown in Fig. 3, where the Digilent peripheral modules (PMOD) on Nexys 4 are used to interface the FPGA with the laptop computer and with the front-end power amplifier. The JANUS waveform [1] is pre-generated based on the sample MATLAB code examples in the JANUS code repository [21] and is written into a data file. Some parameters are modified for the high carrier frequency and wide bandwidth, as listed in Table II. For example, the "#Id", "Center Frequency", and "Bandwidth" terms in the "parameter_sets.csv" file, and the sampling frequency "COMMON_FS" in the "defaults.m" file is changed accordingly. Bits 31 and 32 in the ADB of the baseline JANUS packet are used to identify the modulation schemes used for the cargo packet.

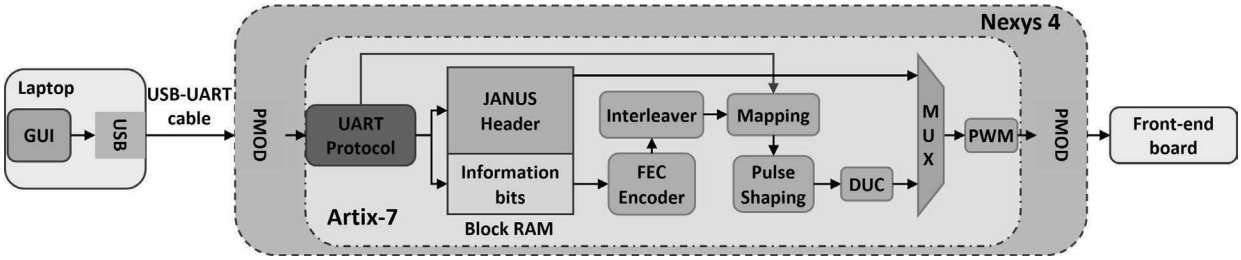


Fig. 3. The block diagram of the FPGA-based transmitter.

A GUI (Graphic User Interface) running on the host PC takes charge of reading the JANUS waveform and cargo information bits from the data files and loading them into the block RAM instantiated in the FPGA via the UART (Universal Asynchronous Receiver Transmitter) protocol. In addition, the duration of the guard gap between the baseline JANUS packet and cargo packet and the selection of modulation scheme are configured in the GUI through the high-speed USB-UART cable. Implemented in the FPGA, the multiplexer (MUX) first selects the JANUS waveform from the block RAM and sends it to the PWM (Pulse Width Modulation) module which drives the transmitter power amplifier on the front-end board. Meanwhile, the cargo information bits are coded by a rate-1/2 non-systematic convolutional encoder with the generator polynomial $[G_1, G_2] = [17, 13]_{oct}$. A 32×32 block interleaver is implemented by a ping-pong buffer to combat burst errors encountered in Underwater acoustic fading channels. The interleaved bits are mapped to BPSK, QPSK, or 8-PSK symbols, and then the symbols are up-sampled and pulse-shaped via a square-root raised-cosine filter. The digital up-converter (DUC) module modulates the baseband signal to the passband carrier, which is then converted to a PWM signal before sending it to the front-end power amplifier. The specifications for a cargo packet are listed in Table II. The FPGA resource utilization for this implementation is as follows: 17 BRAM, 2 DSP48E, 3% LUT, and 2% Flip Flop.

In contrast, the main functionality of the receiver in this work is signal capture and storage. The block

diagram of the receiver is shown in Fig. 4, where the DAT solution is implemented by two ADC options. One is the built-in dual 16-bit ADC (XADC) on the Zynq 7000 with up to 1 Mega sample per second (MSPS) sampling rate. Another option is to use an external ADC via the FMC (FPGA Mezzanine Card) connector on the Zedboard. The external ADC option provides more flexibility in parameter selection such as sampling rate, reference input, dynamic range, resolution, etc. Both ADC options support the unipolar and bipolar modes and may be shared with multiple external analog inputs, thus enabling multiple receive elements.

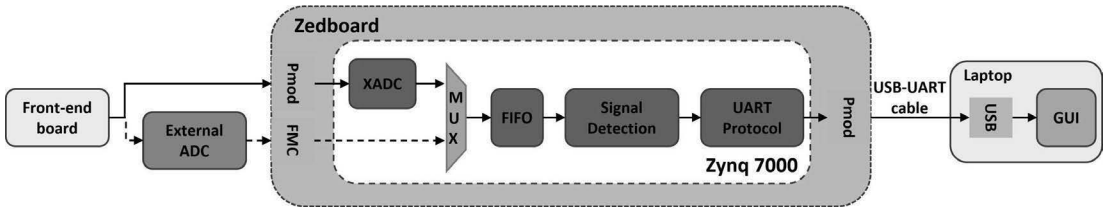


Fig. 4. The block diagram of the receiver.

The MUX in the receiver selects one of the ADC input sources, and a small number of samples N_{samp} (such as 4000) are first buffered in the FIFO, then the signal detection module compares the input signal power to a threshold. If the input signal power is smaller than the threshold, the FIFO will update with new samples so that the amount of the samples stored in the FIFO remains at N_{samp}. If the signal power exceeds the threshold, the FIFO will keep the input samples and continuously store input samples until the FIFO is full. The size of the FIFO is set to capture a large frame of transmitted signals. Next, the UART protocol reads the samples from the FIFO and uploads them to the host PC through the USB-UART cable. Finally, a receiver GUI running on the host PC manages the received samples in the memory and writes them into files.

For the center frequency of 115 kHz used in this work, the sampling rate was configured as 460 ksps. Therefore, the resulting data rate after the ADC is $460 \times 16 = 7.3$ Mbps. Although an onboard USB-UART bridge CY7C64225 is available on the Zedboard, it can only be accessed from the PS (Processing System) rather than the PL. Therefore, the XADC instantiated in the PL (Programming Logic) has no direct access to the onboard USB-UART bridge and has to go through the UART protocol implemented on the PL. In addition, the high-speed USB-UART cable C232HD-DDHSP-0 can transfer data to the host PC faster than the onboard USB-UART bridge which only supports 230400 baud rate. The Zynq AP SoC XC7Z020-1CLG484 has 85,000 programmable logic cell, 140 36Kb BRAMs, and 220 DSP48Es [22]. The resource utilization of the DAT system is as follows: 87% BRAM, 0 DSP48E, 2% LUT, and 1% Flip Flop. In summary, the parameters of the receiver hardware platform are also listed in Table II.

TABLE II
SPECIFICATION OF THE CARGO PACKET AND THE RX HARDWARE PARAMETERS

Cargo packet specifications	***Values***	**Rx hardware parameters**	***Values***
Encoder	$[17, 13]_{oct}$	ADC resolution (bits)	16
Block interleaver	32×32	Analog input	single-ended
BPSK data rate (kbps)	11.5	ADC input range (mV)	± 500
QPSK data rate (kbps)	23	ADC input mode	unipolar
8-PSK data rate (kbps)	34.5	ADC sampling rate (kHz)	460
Cargo Packet size (bits)	512	Data log speed (Mbps)	2
Channel estimation	IP-NLMS	Data length (samples)	262144
Decoder	MAX log-MAP	Amplification Gain (dB)	46
Equalizer	LMMSE Turbo	Maximum BW (kHz)	38

It is worth noting that most of the parameters on the hardware platform are custom adjustable to meet different testing needs. The design is also efficient in capturing and transferring a large among of data for

field experiments. The received baseline JANUS packet and cargo packet are post-processed to evaluate the performance. The receiver algorithms are detailed in Section III.

III. Improved receiver algorithms

The receiver data processing algorithms are depicted in Fig. 5, where the top row shows the JANUS Rx MATLAB sample code [21], and the bottom rows show the Turbo equalizer algorithm for cargo packet. First, the digitized passband signal is down-converted and down-sampled to the baseband complex signal in the digital down-converter (DDC) module. The baseband JANUS packet is then correlated with the fixed 32-chip preamble in the Chip Alignment module, and the frame start τ_b of the baseband signal is detected. Next, the first 32 chips are extracted from the baseband signal and are used to calculate the Doppler spread. After Doppler compensation, the 144 chips of the baseline JANUS packet are demodulated and converted into soft symbols fed into the de-interleaving module. Finally, the soft Viterbi decoder decodes the 144 chips into the 64-bit baseline information packet.

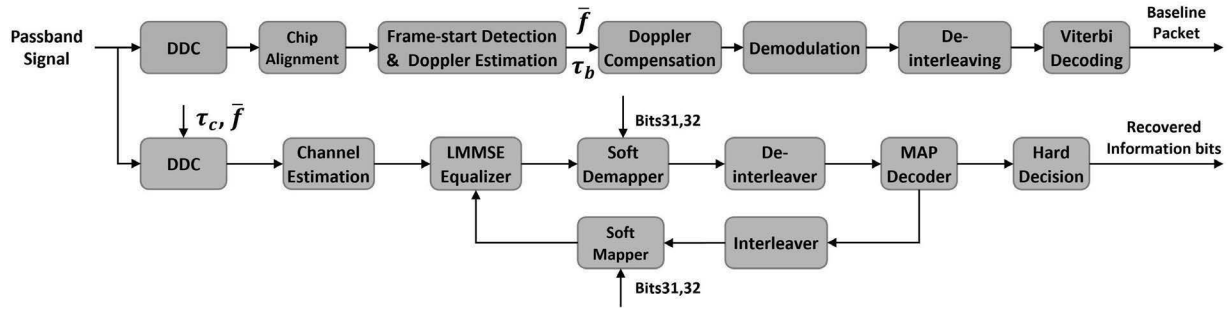
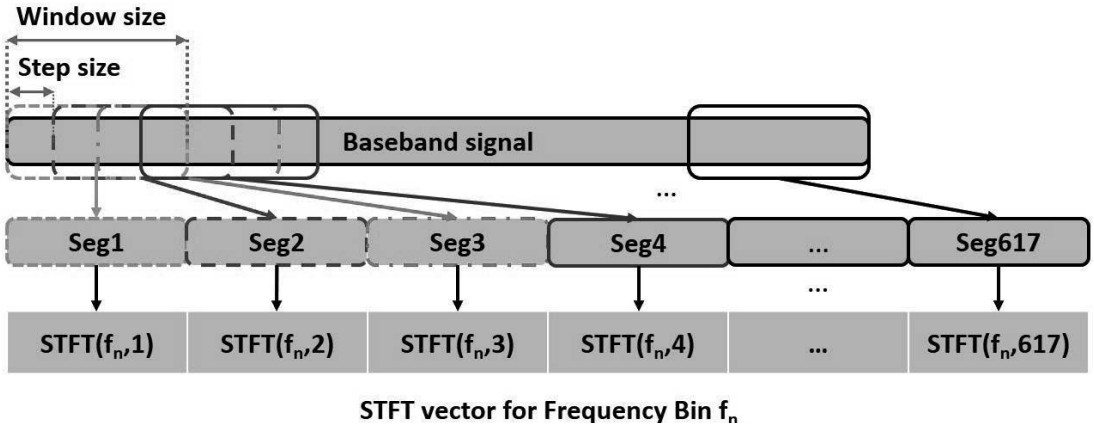


Fig. 5. Receiver data processing algorithms: the top row is for the baseline JANUS packet; the bottom rows are for cargo payload.

For the cargo packet, the Doppler shift $\bar{f}$ and symbol start information $\tau_c = \tau_b + t_{baseline} + t_{gap}$ estimated by the baseline JANUS packet are used to demodulate the passband cargo packet to the baseband in DDC, where $t_{baseline}$ is the time duration of the JANUS waveform, t_{gap} is the guard gap. The first 512 symbols are extracted from the baseband signal as the pilot and used for channel estimation by the improved proportionate NLMS (IPNLMS) algorithm [23]. The Linear Minimum Mean Square Error (LMMSE)-based turbo equalizer is used for symbol equalization [24]. The bits 31 and 32 from the baseline JANUS packet represent the modulation scheme and determine the mapping from the equalized symbols to the extrinsic Log-Likelihood Ratio (LLR) of encoded bits and from the *a priori* LLR of encoded bits to the soft symbols. The *a priori* LLR is provided from the MAX log-MAP decoder.

The LMMSE-based turbo equalizer is widely used in underwater acoustic communication [24] and an FPGA implementation is also discussed in [25]. However, the real-time implementation of the JANUS fast-mode receiver still requires effort. In this work, we analyze several modules in the JANUS Rx sample code and improve it for the fast mode.

In the Chip Alignment module, the received baseband signal $\mathbf{r}$ is separated into segments by an overlapped sliding window as shown in Fig. 6(a), where Seg1, Seg2, ..., and Seg617 represent the segments extracted from the baseband signal. The window size, denoted as N_{win}, equals to the number of samples per chip whose duration is modified by a maximum Doppler speed of 5 m/s. The step size N_{step} is a quarter of the number of samples per chip in the baseband N_{chip} with an oversampling ratio $r = 4$, thus $N_{step} = N_{chip}/r$. The overlap between one segment and the next equals the window size minus the step size. The Goertzel algorithm is then used to calculate the Short-Time Fourier Transform (STFT) for each segment with respect to the 22 frequency bins used by the 32-chip preamble. The 22 frequency bins are taken from the 13 pairs of FH frequencies with the frequency bin numbers as $\{5, 8, 16, 6, 12, 22, 20, 17, 10, 19, 13, 3, 11, 14, 14, 22, 7, 2, 4, 21, 9, 12, 8, 5, 18, 18, 12, 15, 1, 3, 11, 4\}$. Note that some of the 32 chips use the same frequencies.



(a)

Chip	*Frequency Bin*	*Xcorr(1)*	*Xcorr(2)*	*Xcorr(3)*	*...*	*Xcorr(493)*
1	5	STFT(5,1)	STFT(5,2)	STFT(5,3)	...	STFT(5,493)
2	8	STFT(8,5)	STFT(8,6)	STFT(8,7)	...	STFT(8,497)
3	16	STFT(16,9)	STFT(16,10)	SFTF(16,11)	...	STFT(16,501)
...	...	...	...	...	...	
24	5	STFT(5,93)	STFT(5,94)	STFT(5,95)		STFT(5,585)
...	...	...	...	...	...	...
31	11	STFT(11,121)	STFT(11,122)	STFT(11,123)	...	STFT(11,613)
32	4	STFT(4,125)	STFT(4,126)	STFT(4,127)	...	STFT(4,617)

(b)

Fig. 6. Cross-correlation of the JANUS header. (a) Segments separated from the baseband signal. (b) The sliding window used to compute the correlation based on the STFT vectors.

Denote the STFT vector elements as $STFT(f_n, m)$ where f_n is the frequency bin of the nth chip and m is the segment index. Since the oversampling rate is $r = 4$, the elements $STFT(f_n, m + 4(n - 10)$ of the 32 STFT vectors are aligned in columns, as shown in Fig. 6(b). Therefore, each column is summed to yield the correlation function $Xcorr(m)$ as

$$Xcorr(m) = \sum_{n=1}^{32} STFT(f_n, m + 4(n - 1)), \tag{1}$$

for $m = 1, 2, \cdots, 493$. Based on the correlation function $Xcorr(m)$, the Greatest Of Constant False Alarm Rate (GO-CFAR) detector is adopted to detect the frame start of the signal. The detected frame-start in terms of segment index m is then converted to the baseband signal sample index by multiplying the step size N_{step} used in the chip alignment module. The first 32 chips are extracted from the baseband signal to estimate the Doppler spread [26], which is used to compensate for the 144 chips following the preamble.

In the Demodulation module, the soft chips of the 144-chip baseline JANUS packet are computed by correlating each received chip with the corresponding pair of transmitted frequency signals and normalizing the peaks of the correlation functions. Denote the pair of transmitted frequency signals (cosine) of the tth chip $chip_t$ as $f_0(t, k)$ and $f_1(t, k)$, respectively, where $k = 1, \cdots, N_{chip}$ is the sample index. The received chip is denoted $y(t, k)$. The peaks of the correlation functions are computed as

$$R_0(t) = \sum_{k=1}^{N_{chip}} y(t, k) f_0(t, k), \quad R_1(t) = \sum_{k=1}^{N_{chip}} y(t, k) f_1(t, k),$$

Therefore, the normalized soft chip estimate P_t is

$$P_t = P(chip_t = 1) = \frac{R_1(t)}{R_0(t) + R_1(t)}. \tag{2}$$

for $t = 1, \cdots, 144$. The normalized soft chip estimates P_t are fed to the de-interleaver and then soft Viterbi decoder which yields the 64 hard bits.

The JANUS Rx sample code works well with the example signals using the defined bandwidth in the standard. However, we found in a previous work [27] that the MATLAB code caused synchronization issues in the fast mode because of the inconsistent usage of the step size N_{step} in the Chip Alignment module and the process of converting the segment index of the frame-start to the sample index of the baseband signal. This inconsistency results in a significant shift in the frame synchronization when applied to the fast-mode operation. The problem was resolved in [27] and the performance after correcting the frame-start detection error is significantly improved, but some packets still perform badly. We discovered recently that the main reason is due to the difficult multipath channel where two or more channel impulses are located close to each other resulting in multiple peaks in the correlation $Xcorr$. Selecting different peaks can result in different performances in decoding the baseline JANUS packet. We thus proposed a modified frame-start detection and Doppler estimation algorithm, as shown in Fig. 7, where multiple peaks in the correlation function are considered. Compared with the original receiver algorithm, three more steps are added, including the First 32-chip demodulation, error bit calculation, and Frame-Start & Doppler Selection. The modified algorithm is detailed in Algorithm 1.

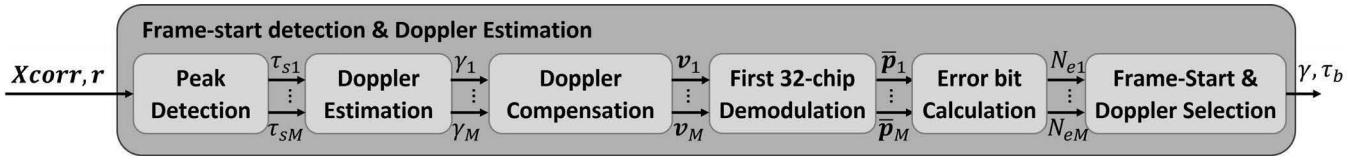


Fig. 7. The modified algorithm for frame-start detection and Doppler estimation using the fixed 32-chip preamble.

All the peaks above the threshold within twice the length of the 32-chip preamble are considered as the candidates of the frame start $\tau_{\mathbf{s}}$ in terms of segment index in the Peak Detection module (lines 4). Each peak $\tau_{si}, i = 1, \ldots M$ is then used to calculate a frame-start τ_{bi} in baseband and estimate a Doppler spread γ_i (lines 6-9). After Doppler compensation, the first 32 chips $\mathbf{v}_i$ are demodulated from the baseband signal as $\bar{\mathbf{p}}_i$ (lines 10). The error bits N_{ei} are calculated by comparing $\bar{\mathbf{p}}_i$ with the fixed 32-bit preamble $\mathbf{p}$ (lines 11). Only the peak resulting in the least error bits (lines 12-19) is treated as the frame start of the signal τ_b and is selected to decode the following 144 chips with the corresponding estimated Doppler spread γ_i. If several peaks yield the same number of error bits in the first 32 chips, then the one that occurs earlier is preferred.

IV. Summary of Underwater Experiments

A group of experiments was conducted in two fresh-water lakes in 2020 and 2021: Lake Nockamixon (denoted as P1) and Green Lane Lake (denoted as P2) in Pennsylvania, USA. In addition, another experiment was undertaken in 2021 at the Rutgers University Marine Field Station (RUMFS) near the ocean coast at Tuckerton, NJ (denoted as P3). The three experimental sites and node placements are shown in Fig. 8. Lake Nockamixon is the largest lake in Bucks County, southeastern Pennsylvania. The satellite view of the marina of Lake Nockamixon is shown in Fig. 8(a) from Google Maps, where two short docks at the launching ramp are 15 m apart. There are 14 long docks equally spaced along the shore on the left of the launching ramp, which are marked Dock A to Dock N from right to left. On the right to the launching ramp, a small peninsula is connected to the shore by a narrow pedestrian path. Six locations denoted as Node 1 – 6 are selected to place the transmitter or the receiver. Node 1 and Node 2 are on the short docks at the launching ramp; Node 3 and Node 4 are 4 m from the shore near the peninsula;

Algorithm 1 The modified Frame-start detection and Doppler Estimation algorithm

Function Frame-start detection and Doppler Estimation ($\mathbf{r}, \mathbf{Xcorr}$);
Input: Baseband signal $\mathbf{r}$, Correlation vector $\mathbf{Xcorr}$, the fixed 32-bit preamble $\mathbf{p} = 0xAEC7CD20$
Output: The frame start of the signal in the baseband τ_b, the Doppler spread γ
Initialize the frame-start vector in terms of segment index as $\tau_\mathbf{s} \leftarrow 0$
Initialize the frame-start vector in baseband as $\tau_\mathbf{b} \leftarrow 0$, a 32-chip vector as $\mathbf{v} \leftarrow 0$, a 32-bit vector as $\bar{\mathbf{p}} \leftarrow 0$, the number of error bits in 32-bit preamble $N_e \leftarrow 0$
$\tau_\mathbf{s} \leftarrow Peak\_detection(\mathbf{Xcorr}), \quad \tau_\mathbf{s} = [\tau_{s1}, \tau_{s1}, ..., \tau_{sM}]$ /*M frame-start candidates*/
for $i = 1 : M$ **do**
  $\tau_{bi} = \tau_{si} \times N_{step} + D_{align}$ /*convert frame-start in terms of segment index back to the baseband signal index, D_{align} is the delay introduced by the Chip Alignment module*/
  Extract the first 32 chips $\mathbf{v}_i$ from baseband signal $\mathbf{r}$ starting at τ_{bi}
  $\gamma_i \leftarrow Doppler\_Estimation(\mathbf{v}_i)$
  $\mathbf{p}_i \leftarrow Doppler\_Compensation(\mathbf{v}_i, \gamma_i)$
  Demodulate the 32 chips $\mathbf{v}_i$ as $\bar{\mathbf{p}}_i$
  Calculate the number of error bits N_{ei} in $\bar{\mathbf{p}}_i$ compared with $\mathbf{p}$
  if $i = 1$ **then**
    $\tau_b \leftarrow \tau_{bi}; \quad f \leftarrow fi$
  else
    if $N_{ei} < N_{e(i-1)}$ **then**
      $\tau_b \leftarrow \tau_{bi}; \quad f \leftarrow fi$
    end if
  end if
end for

Node 5 and Node 6 are on Dock E and N, respectively. The distances among these nodes are noted in Fig. 8(a). The depths of the experiment locations vary from 1.5 m to 6 m.

(a) (b) (c)

Fig. 8. The satellite maps of experimental sites and node placements: (a) Lake Nockamixon P1, (b) Green Lane Lake P2, and (c) Ocean coast at RUMFS P3.

The P2 experiment site is the boat rental area in Green Lane Lake, Montgomery County, Pennsylvania. It has a 37 m long loading dock parallel to the bank. The depth of the water is around 10 m. Three places, denoted as GNode 1 – 3, are chosen for the experiment. The satellite view from Google Maps and the node locations are shown in Fig. 8(b). The two furthest docks at the P3 experiment site, as shown in Fig. 8(c), are separately selected to place TX and RX. The equipment setup in the field experiments is shown in Fig. 9. A snow tube floating on the water and long rods were used to hold the transmitter node or the receiver hydrophone in place.

Ten experiments were conducted from 10/17/2020 to 11/25/2020, and two experiments were conducted in fall 2021. A summary of the field experiments is shown in Table III, where the packet error rate (PER) is computed for the 64-bit baseline JANUS packet. During each experiment, the transmitter was stopped

(a) (b) (c) (d)

Fig. 9. The equipment setup in the field experiments. (a) Tx placed at Node1, (b) Rx placed at Node 4, (c) Tx placed at GNode2, (d) Rx placed at GNode1.

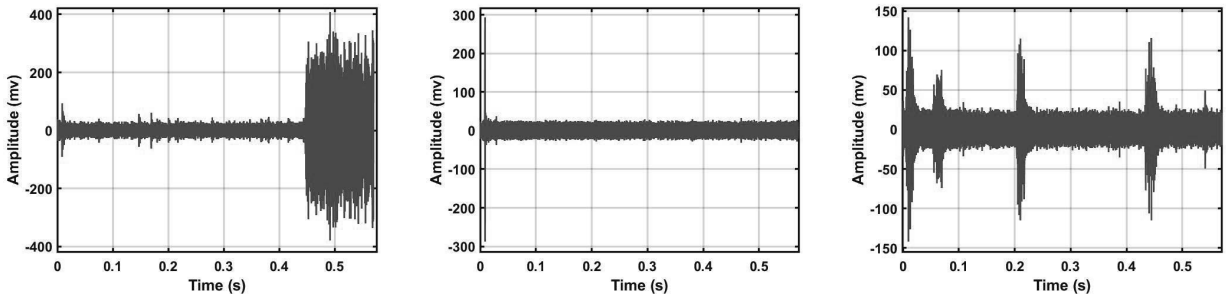
after transmitting for one bandwidth, and then the waveform for the next bandwidth was loaded and transmitted. Normally, the Xilinx Chipscope was used at the receiver during the first experiment in a new location to test the system and observe the signal strength. Later on, the DAT solution was used to record a large group of data for BER performance analysis. In all field experiments, the transmitter and receiver transducers were 0.5 m below the water surface. The transmit power was fixed at around 3 W, and the estimated sound pressure level (SPL) was 153 dB re μPa @ 1 m.

TABLE III
SUMMARY OF THE FIELD EXPERIMENTS AND PACKET ERROR RATES (PER) OF THE BASELINE JANUS PACKETS

Date	Time/Loc.	BW (kHz)	Tx-Rx Distance	Tx/Rx Position	Rx Tool	No. of files	PER%[1]	PER%[2]
11/09/2021	8:31-9:03/P3	38	35 m	TX/RX	Chipscope	88	3.4	3.4
10/07/2021	8:31-9:03/P1	38	150 m	Node 2/Node 5	Chipscope	159	0.6	0.6
11/25/2020	9:42-11:25/P1	$22 \sim 38$	415 m	Node 2/Node 6	DAT	693	0.3	0.3
11/24/2020	16:12-17:04/P1	$22 \sim 34$	150 m	Node 2/Node 5	DAT	371	0.5	0.5
11/20/2020	7:39-8:10/P1	22, 26	80m	Node 1/Node 4	DAT	229	0	0
11/17/2020	10:11-11:57/P1	38	$65 \sim 415$ m	Node 2/Each Dock	Chipscope	74	NA	NA
11/07/2020	7:14-9:06/P1	$22 \sim 38$	80 m	Node 1/Node 4	DAT	664	0.3	0.3
11/06/2020	7:30-8:23/P1	$22 \sim 30$, 38	70 m	Node 1/Node 3	DAT	360	5.71	1.67
11/04/2020	8:41-10:25/P1	$22 \sim 38$	70 m	Node 3/Node 1	DAT	704	52.93	50.19
10/27/2020	17:25-18:27/P1	$22 \sim 38$	70 m	Node 3/Node 1	DAT	525	67.05	38.46
10/23/2020	19:49-21:08/P1	38	15 m	Node 1/Node 2	Chipscope	11	0	0
10/17/2020	8:31-9:03/P2	22, 38	11 m, 23 m	GNode 2,3/GNode 1	Chipscope	33	0	0

[1] Processed by the original receiver algorithm.
[2] Processed by the modified receiver algorithm.

It is worth noting that some of the recordings contain invalid data, noise, or interference only, some of which are shown in Fig. 10. For example, Fig. 10(a) shows a recording that contains only part of the baseline JANUS packet. Fig. 10(b) shows that a strong interference triggered the recording, no signal but the noise was recorded. Fig. 10(c) recorded several strong interfering signals uncorrelated with the 32-chip preamble. All these cases are account for around 1% of the total recordings and are excluded in the data analysis. Significant experience has been gained through the field experiments, such as logistics, robustness of testing equipment, environmental effects, data recording and storage, etc. One significant learning with the wideband transmission at 115 kHz is that the significant interference sources around this carrier frequency exists in switching power supplies and power inverters. The key step to eliminate the interference is to use batteries to power laptops and all experimental devices.



(a) (b) (c)

Fig. 10. Different cases of the interference in the received packets.

V. Experimental Results and Analysis

A. The baseline JANUS Packets

As listed in Table III, the received baseline JANUS packets were processed by both the original sample MATLAB code and the modified fast-mode JANUS algorithm. The PER of most days is less than 1% using both the algorithms, for example, on days from 11/07/2020 to 10/07/2021. On the other hand, some days, the PER is slightly higher using the original sample MATLAB code, in the range of 3% to 6%, such as on days 11/06/2020 and 11/09/2021. However, the modified fast-mode JANUS algorithm is able to reduce the PER of the baseline JANUS packets recorded on 11/06/2020 from 5.71% to 1.67%. While a couple of days, such as on 11/04/2020 and 10/27/2020, the PER is worse than 30%, even though the modified fast-mode JANUS algorithm corrected around 30% of error packets.

The exact reason why the error performances on these two days are bad is unknown, as the signal strength and power spectrum plots appear to be similar to those of other days. We guess one reason is that the multipath effects are different when the nodes are placed at different locations. Another reason could be the strong interference from boat engines because we did notice more boat activities presented at the experiment sites on those days. For example, the number of boats leaving or arriving at the docks was significantly higher during the experimental time slots on 11/04/2020 and 10/27/2020, while other days saw very small numbers of launching and docking activities during the experiments.

The detailed decoding performance of the original sample MATLAB code for the baseline JANUS packets recorded on days 10/27/2020, 11/04/2020, and 11/06/2020 is listed in Table IV. The results demonstrate that the error rates on different dates are quite different, but the error rates with different bandwidths of the fast mode are very similar. Therefore, all the results of different bandwidths on the same day of the experiment are grouped for further analysis. The time of the day when the experimental data was recorded also has little effect on the error performance.

TABLE IV
Decoding performance of the original MATLAB sample code for fast-mode baseline JANUS packets in the field experiments.

Date	Metrics	38 kHz	34 kHz	30 kHz	26 kHz	22 kHz
11/06/2020	Packet Error Rate	0.0282 (4/142)	NA	0.0354 (4/113)	0.0769 (3/51)	0.1509 (8/53)
	Bit Error Rate	0.0051	NA	0.0162	0.0282	0.0607
	Time	7:30 - 7:47	NA	8:00 - 8:14	8:17 - 8:23	7:51 - 7:57
11/04/2020	Packet Error Rate	0.1964 (33/168)	0.9359 (73/78)	0.8804 (81/92)	0.5652 (52/92)	0.3958 (38/96)
	Bit Error Rate	0.0673	0.3015	0.3528	0.1663	0.0955
	Time	8:41 - 9:02	10:04 - 10:14	9:46 - 9:58	9:28 - 9:42	9:10 - 9:22
10/27/2020	Packet Error Rate	0.5747 (50/87)	0.7901 (64/81)	0.8265 (81/98)	0.78 (78/100)	0.4902 (75/153)
	Bit Error Rate	0.2128	0.3255	0.289	0.25	0.143
	Time	17:25 - 17:30	17:41 - 17:45	17:49 - 17:55	17:57 - 18:05	18:10 - 18:20

The comparison of the PER processed by the original JANUS receiver algorithm and the modified fast-mode JANUS algorithm is shown in Fig. 11, where the percentage of packets whose number of error bits in the detected 64-bit baseline JANUS packet falls in the specified ranges, such as 0, (0,10], (10,32], and (32,64], are plotted. Note that the results for days from 11/07/2020 to 11/25/2020 and two days in 2021 are excluded as the error rates in those days are close to zero using the original JANUS receiver algorithm. Taking the result for day 10/27/2020 as an example, the modified fast-mode JANUS algorithm reduced the percentage of the packets containing more than 32 error bits from 10.4% to 4.14%; reduced the percentage of the packets containing (10,32] error bits from 43.93% to 25.84%; reduced the percentage of the packets containing (0,10] error bits from 12.72% to 8.48%. Thus, around 30% more packets are decoded correctly using the modified fast-mode JANUS algorithm.

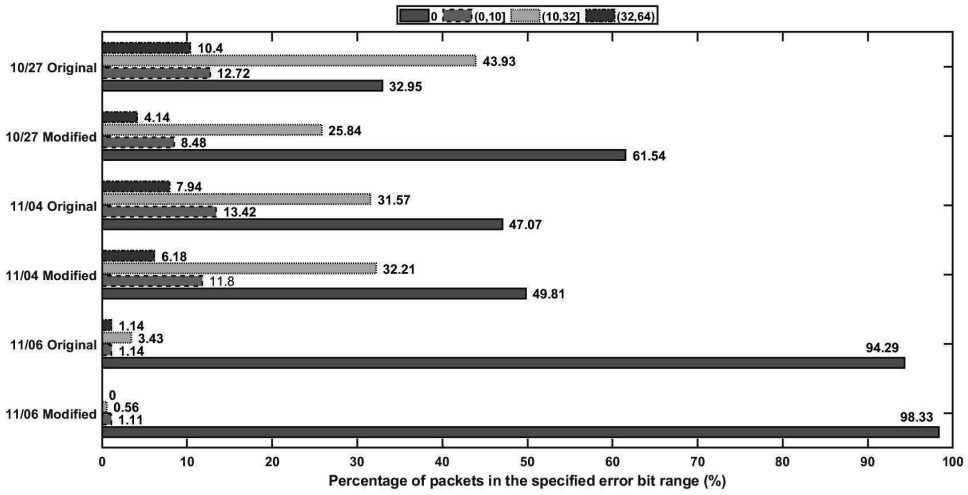


Fig. 11. The percentage of packets in the specified range of the number of error bits: comparison of the JANUS error performance between the original algorithm and the modified algorithm.

Based on the performance difference between the original receive algorithm and the modified algorithm, the decode performance may be divided into three categories: Category 1 contains easy channels where the baseline JANUS packets are decoded without any error bits; Category 2 involves difficult channels where the packets decoded by the original algorithm exhibit some error bits but can be corrected by the modified algorithm; and Category 3 contains packets with a significant amount of error bits and are difficult to decode correctly by both the original and the modified algorithms.

1) Category 1 Easy channels: Most of the packets in the experiments experienced easy or mild channels, and both the original JANUS algorithm and the modified algorithm decoded the packets with no error bits. Some of the channels exhibit little Doppler spread or mild multipath fading so that the estimates of the soft chips were pretty close to the transmitted chips. For example, packet 695 was recorded at a Tx-Rx distance of 415 m on day 11/25/2020, whose waveform and spectrogram are shown in Fig. 12(a). Despite the signal strength being pretty low at ± 40 mV, the signal exhibited little multipath interference compared with the transmitted signal whose waveform and the spectrogram are shown in Fig. 12(b). Furthermore, the correlation of these kinds of packets has a single clear peak, as shown in Fig. 12(c) and (d). Therefore, the frame-start detection was accurate, and the 144-chip estimates had no errors or few errors. The soft Viterbi decoder was able to correct all errors and yield all correct bits.

2) Category 2 Difficult Channels: The packets in Category 2 experienced difficult multipath and Doppler spread channels whose spectrogram is shown in Fig. 13(a) exhibited significant multipath and frequency spread. Note that the correlation functions contain multiple similar peaks located close to each other, as shown in Fig. 13(b). The corresponding index for both peaks in the baseband signal is shown in Fig. 13(c), where the blue diamond marks the first peak, and the red circle indicates the second peak.

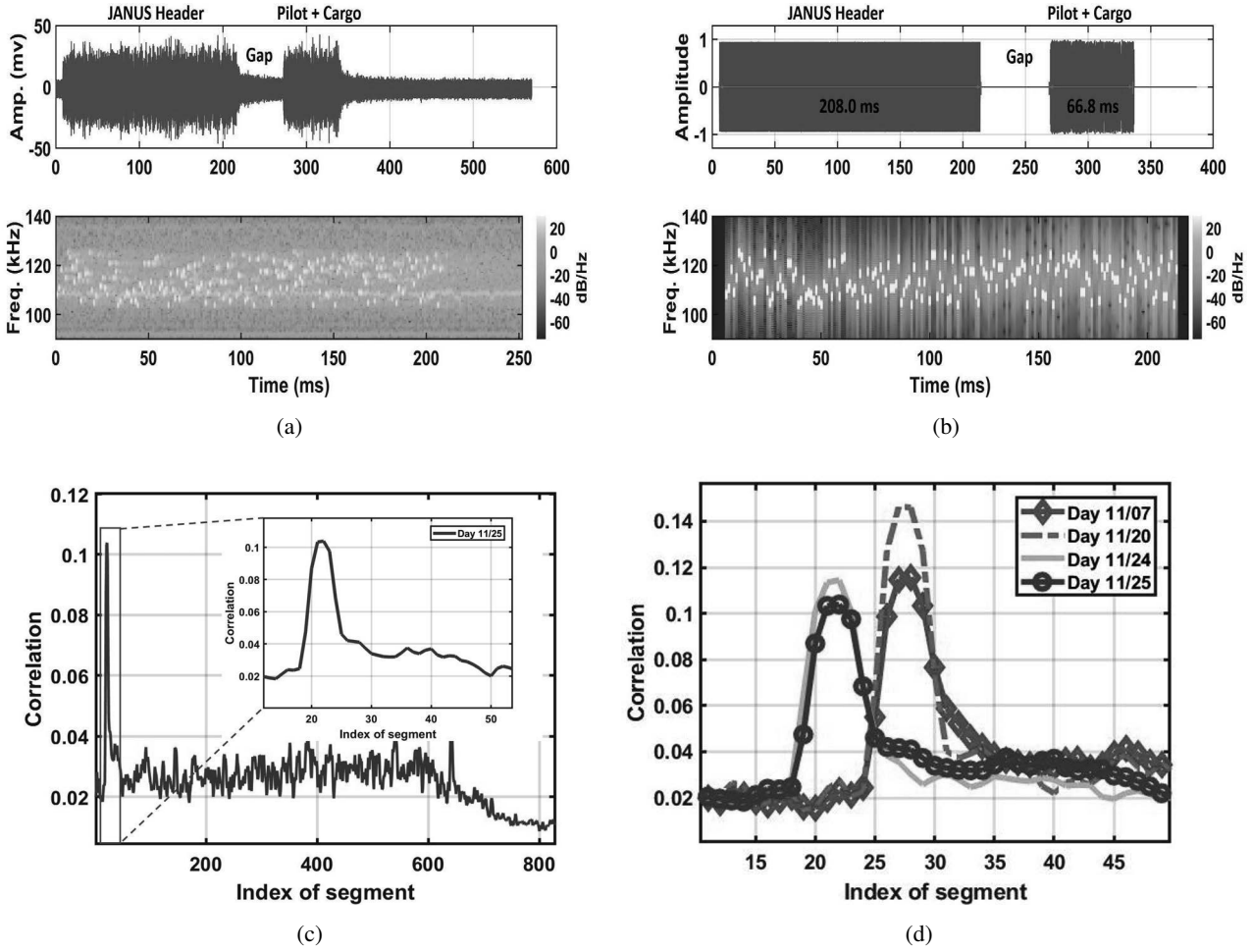


Fig. 12. (a) The passband signal and spectrogram of packet 695 with BW = 22 kHz recorded on Day 11/25/2020. (b) The passband signal and spectrogram of transmitted signal with BW = 22 kHz. (c) The normalized correlation function between the 32-chip preamble and the JANUS packets recorded on day 11/25/2020 with BW = 26 kHz. (d) The zoomed-in normalized correlation function between the 32-chip preamble and the JANUS packets recorded on different days with BW = 26 kHz.

The original JANUS algorithm would select the highest peak and its corresponding index marked by the red circle as the frame start of the signal. Therefore, part of the chips is missed, resulting in 49 error chips out of the 144 chips after demodulation and then 22 error bits out of 64-bit baseline JANUS packet. The error chip distributions after demodulation, de-interleaver, and Viterbi decoder are shown in Fig. 13(d).

In contrast, the modified algorithm would select multiple peaks as the potential candidates and test out both the frame starting at the blue diamond and red circle in the 32 chips decoding first. The frame starting at the blue diamond yielded no error chips in the 32 chips, while the red circle yielded 3 error chips. Therefore, the frame starting at the blue diamond was picked to decode the following 144 chips. The modified algorithm yielded 5 error chips out of the 144 chips after demodulation and no error bit after Viterbi decoding.

3) Category 3 Severe Channels: A small number of packets experienced severe multipath and Doppler channels; thus, both the original JANUS algorithm and the modified algorithm exhibited difficulties in detecting the baseline JANUS packets. One example of packet 51 recorded on 11/04/2020 is shown in Fig. 14, where the signal exhibited a single small peak in its correlation function and strong multipath components after the 32-chip preamble. Both the original JANUS algorithm and the modified algorithm yielded the same results in frame-start detection and Doppler estimation. After demodulation, the packet produced 38 error chips out of the 144 chips. The error chip distributions after demodulation, de-interleaver, and Viterbi decoder are shown in Fig. 14(b). As the baseline JANUS packet utilized the

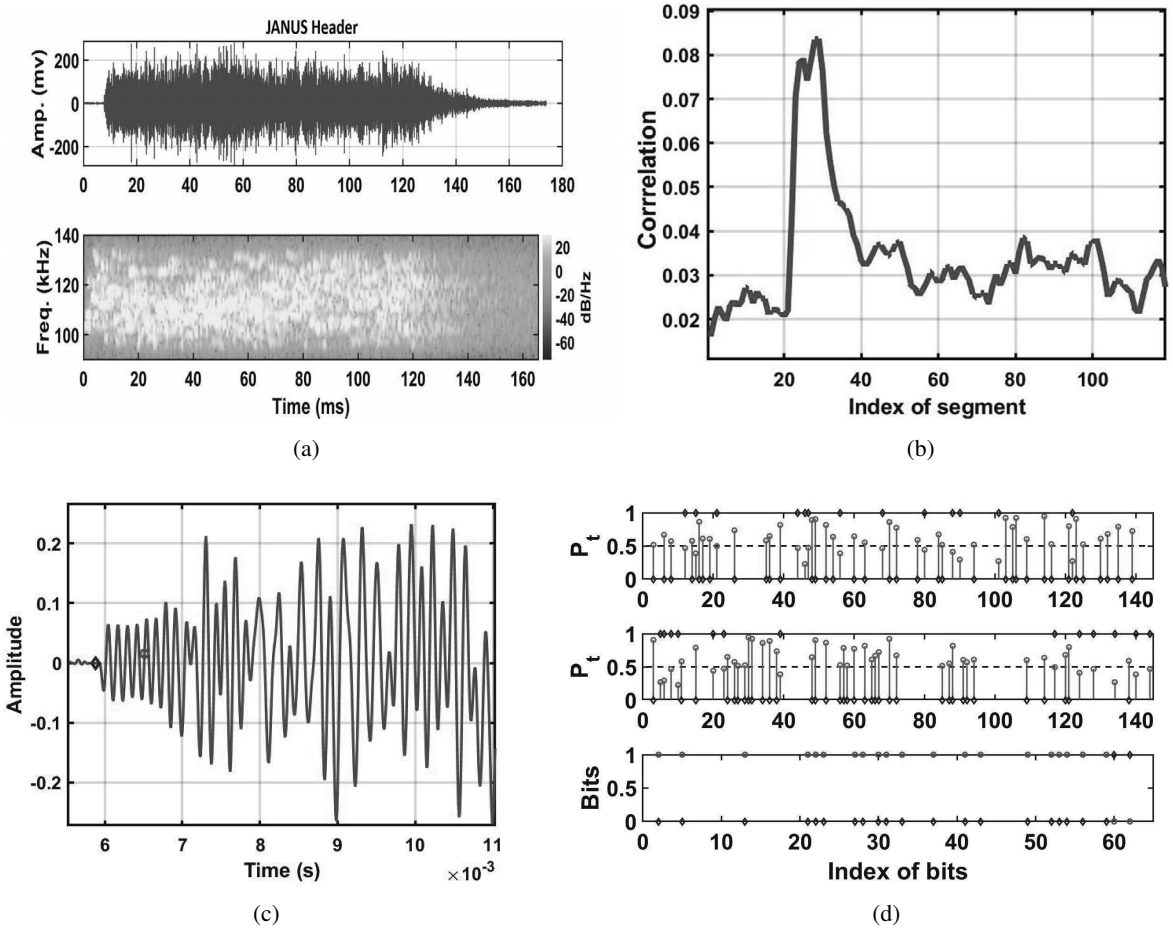


Fig. 13. Packet 314 recorded on Day 11/06/2020 with BW = 26 kHz. (a) The passband signal and spectrogram. (b) The zoomed-in correlation function between the received JANUS packets and the 32-chip preamble. (c) The frame start detection: the original JANUS algorithm chose the time instant marked by the red circle, while the modified algorithm chose the time marked by the blue diamond. (d) Error distributions at different stages in the processing of Packet 314 using the first peak in the correlation. Top: after demodulation; middle: after de-interleaver; bottom: after Viterbi decoding.

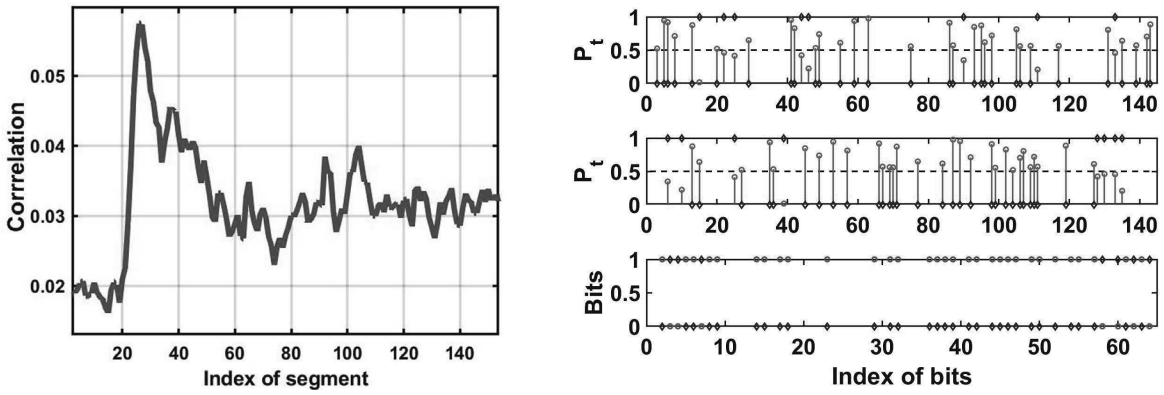


Fig. 14. JANUS fast-mode signals experienced severe channels: packet 51 with BW = 30 kHz recorded on Day 11/04/2020. (a) The zoomed-in correlation function between the received JANUS packets and the 32-chip preamble. (b) Error distribution of the JANUS baseline packets at different stages. Top: after demodulation; middle: after de-interleaver; bottom: after Viterbi decoding.

rate 1/2 convolutional code with constraint length 9, the Viterbi soft decoder was able to correct some of

the errors in the 64-bit baseline JANUS packet. However, when the number of error chips exceeds 30, as in the case of packet 51, the de-interleaver and the decoder are incapable of correcting the errors.

B. Cargo packets

The synchronization information and the estimated Doppler spread offered by the fixed 32-chip preamble are used to process the cargo packet in the received signals as shown in Fig. 5. Benefiting from the proposed frame-start detection algorithm, the symbol start of cargo packet τ_c is able to calculate correctly. However, since M-PSK modulation scheme is more sensitive to the Doppler shift than the FH-BFSK, a Bruce-search for an accurate frequency shift within the frequency shift range offered by the baseline JANUS packet is adopted in the cargo processing. The BER and the PER (dividing the 512 bits payload into 8 packets of 64 bits) results using BPSK are listed in Table V, where the performances of the cargo packets in the difficult multipath and severe channels are better than those of the baseline JANUS packets as the equalizer significantly improved the performance of the cargo packets at the cost of additional pilots and computational complexity.

TABLE V
DECODE PERFORMANCE OF BPSK CARGO PAYLOAD IN EXPERIMENTS AT LAKE NOCKAMIXON.

	11/25/2020	11/24/2020	11/20/2020	11/07/2020	11/06/2020	11/04/2020	10/27/2020
Total bits ($\times 512$)	598	298	156	472	216	197	328
Bit Error Rate	1.4632×10^{-3}	5.2×10^{-3}	2.2×10^{-3}	3.6×10^{-3}	0.0290	0.1472	0.1434
Packet Error Rate%	7.6	17	7.8	7.3	50	60	77

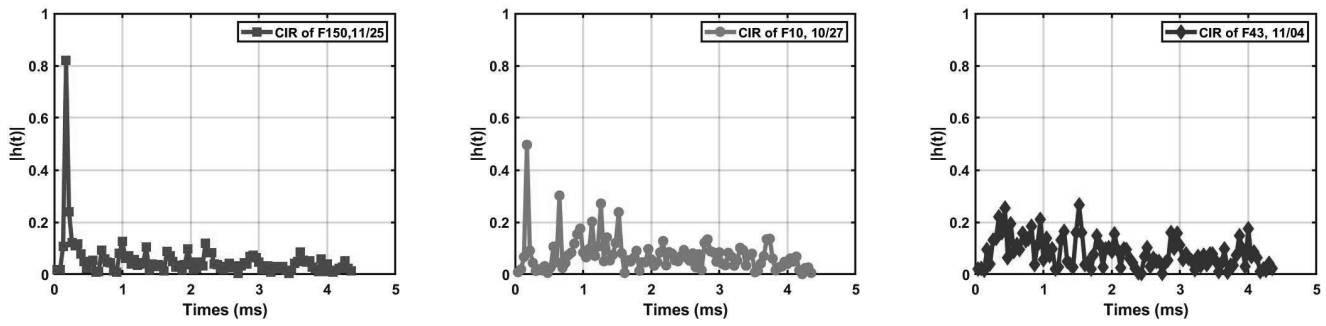
Moreover, two additional experiments were conducted in fall 2021 to evaluate the performance of QPSK and 8-PSK in fresh water and saltwater, respectively. The decoding performance of M-PSK is summarized in Table VI. Both experiments experienced easy channels and resulted in good BER performance that is around or better than 10^{-3} for M-PSK. As the received signals have similar SNR for all M-PSK cases, the BER performance of BPSK is the best; QPSK is a little worse, while 8-PSK is the worst.

TABLE VI
DECODING PERFORMANCE OF M-PSK CARGO PAYLOAD IN THE 2021 FIELD EXPERIMENTS.

Scenario	Metrics	BPSK	QPSK	8-PSK
Salt water (P3)	Total bits ($\times 512$)	30	28	30
	BER ($\times 10^{-3}$)	0.1302	1.0463	1.6927
	PER%	0.8	6.7	9
Fresh water (P1)	Total bits ($\times 512$)	50	50	59
	BER ($\times 10^{-3}$)	0.5361	1.8359	4.2
	PER%	3.4	11.3	16

For further analysis of the three categories described in Section V-A, the Channel Impulse Response (CIR) of a typical packet for each of the categories are depicted in Fig. 15, where the CIR of the easy channel has only one main path, the difficult channel has multiple paths, and the severe channel has no obvious paths whose frequency response exhibits deep fades.

In summary, the cargo packet utilizes the higher-order M-PSK single carrier modulation scheme and achieves high information data rates of up to 34.5 kbps. The receiver algorithm achieves BER on the order of 10^{-3} in most of the experiments, thanks to the carrier and frame synchronization estimated by the proposed frame-start detection algorithm, the channel estimation from the pilot block, and the Turbo equalization algorithm [24]. In contrast, the baseline JANUS packet uses the Frequency Hopping to combat the multipath channels and achieves up to 730 bps in the proposed fast mode operation. The PER is on the order of 10^{-2} thanks to the modified JANUS receiver algorithm. Both the baseline JANUS packet and the cargo packet experience some severe multipath fading channels where strong interference or deep fades of the UWA channels result in a large number of bit errors in a packet.



(a) (b) (c)

Fig. 15. The CIR comparison among three channel categories, (a) Easy channel, (b) Difficult channel, (c) Severe channel.

Ideally, a practical application would prefer a better PER or BER performance in the JANUS header and cargo packet, as many applications of the fast-mode short-range UWA may occur in nearshore or other water bodies with high human activities. Further improvement of the baseline JANUS packet in fast mode operations may be obtained from a stronger FEC (Forward Error Correction) code, the use of more than 13 pairs of hopping frequencies, or reducing the information bit rate. Further improvement on the cargo packet may be obtained from advanced signal processing techniques, such as stronger FEC codes, Turbo decision feedback equalization, and decision-directed channel updating. These would be good topics for future research.

VI. Conclusion

This paper has presented twelve field experiments for testing a fast mode operation of the JANUS acoustic communication standard. An FPGA hardware platform has been developed to transmit and receive the JANUS header and cargo packets with different bandwidths, modulation schemes, and data rates. A modified JANUS receiver algorithm is proposed to combat the difficult multipath channels and improve the performance for the fast mode operation. The experiment results show that the JANUS fast modes work well with the original JANUS algorithm, yielding zero bit error in most of the baseline JANUS packets, while the modified algorithm has improved the PER performance by 30% in the experiment experiencing difficult multipath channels. In addition, the cargo packets have utilized the LMMSE equalizer and achieved around 10^{-3} BER in most of the experiments.

VII. Acknowledgment

The work was supported in part by NSF projects IIP-1853258 and CNS-1853257.

References

[1] STANAG NATO, "4748 Ed. A ver. 1: 'Digital underwater signalling standard for network node discovery & interoperability'," 2017.

[2] John Potter, João Alves, Dale Green, Giovanni Zappa, Ivor Nissen, and Kim McCoy, "The JANUS underwater communications standard," in *2014 Underwater Communications and Networking (UComms)*. IEEE, 2014, pp. 1–4.

[3] Roberto Petroccia, João Alves, and Giovanni Zappa, "JANUS-based services for operationally relevant underwater applications," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 994–1006, 2017.

[4] Roberto Petroccia, Gianni Cario, Marco Lupia, Vladimir Djapic, and Chiara Petrioli, "First in-field experiments with a "bilingual" underwater acoustic modem supporting the JANUS standard," in *OCEANS 2015-Genova*. IEEE, 2015, pp. 1–7.

[5] Roberto Petroccia, João Alves, and Giovanni Zappa, "Fostering the use of JANUS in operationally-relevant underwater applications," in *2016 IEEE Third Underwater Communications and Networking Conference (UComms)*. IEEE, 2016, pp. 1–5.

[6] João Alves, Roberto Petroccia, Alberto Grati, Nicolas Jourden, Gennaro Vitagliano, Paulo Santos Garcia, Jose D Nieves Prieto, and João Borges De Sousa, "A paradigm shift for interoperable submarine rescue operations: The usage of JANUS during the dynamic monarch 2017 exercise," in *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*. IEEE, 2018, pp. 1–7.

[7] João Alves, Bruno Cardeira, Giovanni Zappa, Fausto Ferreira, Roberto Petroccia, Vincenzo Manzari, Davide Buselli, Petrika Gjanci, Oleksiy Kebkal, Jean-Michel Passerieux, et al., "The first JANUS interoperability fest-a field report," in *OCEANS 2019 MTS/IEEE SEATTLE*. IEEE, 2019, pp. 1–6.

[8] Fausto Ferreira, Roberto Petroccia, and João Alves, "Increasing the operational safety of Autonomous Underwater Vehicles using the JANUS communication standard," in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*. IEEE, 2018, pp. 1–6.
[9] João Alves and Justus Ch Fricke, "Analysis of JANUS and underwater telephone capabilities and co-existence," in *2016 IEEE Third Underwater Communications and Networking Conference (UComms)*. IEEE, 2016, pp. 1–5.
[10] João Alves, Thomas Furfaro, Kevin LePage, Andrea Munafò, Konstantinos Pelekanakis, Roberto Petroccia, and Giovanni Zappa, "Moving JANUS forward: a look into the future of underwater communications interoperability," in *OCEANS 2016 MTS/IEEE Monterey*. IEEE, 2016, pp. 1–6.
[11] Daniel Sura, Michael Pfetsch, Nghia Tran, Dusan Radosevic, Patrick Longhini, and Burton Neuner III, "Implementation and testing of the JANUS standard with SSC pacific's software-defined acoustic modem," Tech. Rep., Space and Naval Warfare Systems Center Pacific San Diego United States, 2017.
[12] Dusan Radosevic, Randall Plate, Michael Pfetsch, Burton Neuner III, and Jose Chavez, "Implementation and testing of JANUS and AORUN acoustic communication algorithms," in *OCEANS 2018 MTS/IEEE Charleston*. IEEE, 2018, pp. 1–5.
[13] Konstantinos Pelekanakis, Luca Cazzanti, Giovanni Zappa, and João Alves, "Decision tree-based adaptive modulation for underwater acoustic communications," in *2016 IEEE Third Underwater Communications and Networking Conference (UComms)*. IEEE, 2016, pp. 1–5.
[14] Jianchun Huang and Roee Diamant, "Adaptive modulation for long-range underwater acoustic communication," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6844–6857, 2020.
[15] Konstantinos Pelekanakis, Stéphane Blouin, and Dale Green, "Performance analysis of underwater acoustic communications in Barrow Strait," *IEEE Journal of Oceanic Engineering*, 2021.
[16] Stefano Mangione, Giovanni Ettore Galioto, Daniele Croce, Ilenia Tinnirello, and Chiara Petrioli, "A channel-aware adaptive modem for underwater acoustic communications," *IEEE Access*, vol. 9, pp. 76340–76353, 2021.
[17] Dale Green, John Dellamorte, and Jim Dellamorte, "Enhancing JANUS signaling," in *2021 Fifth Underwater Communications and Networking Conference (UComms)*. 2021, pp. 1–4, IEEE.
[18] Xilinx, "7 Series FPGAs Data Sheet: Overview," https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf, 2020.
[19] Xilinx, "7 Series FPGAs and Zynq-7000 SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter," https://www.xilinx.com/support/documentation/user_guides/ug480_7Series_XADC.pdf, 2018.
[20] FTDI, "C232HD USB 2.0 HI-SPEED TO UART CABLE Datasheet ," https://ftdichip.com/wp-content/uploads/2020/07/DS_C232HD_UART_CABLE.pdf, 2019.
[21] CMRE, "JANUS code repository," http://www.januswiki.com, 2018.
[22] Xilinx, "Zynq-7000 SoC Data Sheet: Overview," https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf, 2018.
[23] Z. Yang and Y. Rosa Zheng, "Iterative channel estimation and turbo equalization for multiple-in multiple output underwatr acoustic communicaltions," *IEEE J. Oecan. Eng.*, vol. 41, no. 1, pp. 232–242, Jan. 2016.
[24] Y. R. Zheng, J. Wu, and C. Xiao, "Turbo equalization for single-carrier underwater acoustic communications," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 79–87, November 2015.
[25] Jinfeng Li and Y. Rosa Zheng, "Hardware implementation of single-carrier time-domain turbo equalization for severe multipath acoustic communication channels," in *Global Oceans 2020: Singapore–US Gulf Coast*. IEEE, 2020, pp. 1–7.
[26] Concetta Baldone, Giovanni Ettore Galioto, Daniele Croce, Ilenia Tinnirello, and Chiara Petrioli, "Doppler estimation and correction for JANUS underwater communications," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.
[27] Jinfeng Li, Joseph Hall, and Y Rosa Zheng, "Utilizing JANUS for very high frequency underwater acoustic modem," in *Global Oceans 2020: Singapore–US Gulf Coast*. IEEE, 2020, pp. 1–6.

Jinfeng Li received the M.S. degree in communication engineering from Beijing Institute of Technology University, Beijing, China, in 2012. He is currently working toward the Ph.D. degree in electrical engineering, Lehigh University, Bethlehem, PA, USA. From 2012 to 2017, he was an assistant researcher in Institute of Electronics, Chinese Academy of Science. His research interests include underwater acoustic communications and signal processing, modem development, and hardware acceleration.

Yahong Rosa Zheng received the Ph.D. degree from Carleton University, Ottawa, ONT, Canada, in 2002. She was an NSERC (Natural Sciences and Engineering Research Council of Canada) Postdoctoral Fellow for two years with the University of Missouri-Columbia. From 2005 to 2018, she was on the faculty of the Department of Electrical and Computer Engineering at the Missouri University of Science and Technology for 13 years. She joined Lehigh University in Aug. 2018 as a professor in the ECE department. Her research interests include underwater and underground IoT and robotics, compressive sensing, wireless communications, and wireless sensor networks. She has served as a Technical Program Committee (TPC) member for many IEEE international conferences and as Associate Editor for three IEEE journals. She is currently a senior editor for IEEE Vehicular Technology Magazine and an Associate Editor for the IEEE Journal of Oceanic Engineering. She is the recipient of an NSF (National Science Foundation) faculty CAREER award in 2009. She has been an IEEE fellow and a Distinguished Lecturer of IEEE Vehicular Technology Society since 2015.