Utilizing JANUS for Very High Frequency Underwater Acoustic Modem

Jinfeng Li, Joseph Hall, and Y. Rosa Zheng
Dept. of ECE, Lehigh University, Bethlehem, PA 18015-3084, USA

***Abstract*—JANUS is a physical layer communication standard for underwater acoustic communications published by North Atlantic Treaty Organization (NATO) in 2017. Instead of the nominal frequency band of 9440 – 13600 Hz specified in the standard, we adopt the JANUS packet for a high frequency band spanning from 96 kHz to 134 kHz. We also add cargo packets in the same frequency band using JANUS fast mode with a symbol rate of 23 ksps. Experiments were conducted in a swimming pool and the JANUS 3.0.5 Matlab version of the example receiver program was used to process the JANUS packets. We found that the example receiver program uses many fix(), round() and floor() functions which lead to synchronization errors. After modifying the simple rx code and fixing the error, our JANUS decoding results show that the adopted JANUS fast mode successfully achieves carrier and frame synchronization in all cases despite some bit errors remaining in the JANUS packet in severe multipath scenarios.**

***Index Terms*—JANUS, Underwater Wireless Communications, Acoustic Communications.**

I. Introduction

JANUS is a physical-layer standard [1] defined by the North Atlantic Treaty Organization (NATO) for underwater acoustic communications. JANUS has been used in several underwater scenarios, such as a first contact and language switching, underwater automatic identification system (AIS), underwater meteorology and oceanography (METOC), and distressed submarine (DISSUB) operations [2].

To achieve robustness against the harsh underwater acoustic propagation environment, JANUS selects Frequency-Hopping (FH) Binary Frequency Shift Keying (BFSK) as the modulation and coding schemes (MCS). The original JANUS example divides the frequency band of 9440 - 13600 Hz 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 determined by user, CRC bits, and other indication flags. 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 cargo payload packets append after the baseline JANUS packet. The structure of a JANUS packet with cargo is shown in Fig. 1.

The typical chip duration defined in the standard is 62.5 ms resulting in a bit rate of 80 bps. For a higher bit rate in cargo packets, the JANUS fast mode has been suggested in [3] as a potential standard evolution, where MCSs for the cargo packet are selected from some high-order MCSs, such as BPSK, QPSK, or 8PSK. These modulation schemes have been tested within the frequency bands around 10 kHz and without coding schemes in [4], where the maximum bit rate achieved is 12 kbps with 8PSK. The work in [5] also tested JANUS fast mode with the 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 large bit errors when the bandwidth exceeded 8 kHz. However, it is unclear how the bandwidth was changed and what parameter modifications were made to the example tx-rx codes.

TABLE I
JANUS PARAMETERS

Parameter	*JANUS standard*	*Parameter in this paper*
center frequency (Hz)	11,520	115,000
bandwidth (Hz)	4,160	38,012
chip duration (ms)	6.25	0.67826
bit rate (bps)	80	737

In this paper, we use transducers operating in a frequency band of 96 kHz to 134 kHz and adopt this frequency band for both JANUS header and cargo packets. We set the center frequency to 115 kHz and bandwidth to 38 kHz. All other parameters in the FH-BFSK are unchanged. Therefore, the bit rate of the JANUS header of the FH-BFSK scheme rises to 737 bps. A comparison between the JANUS standard and the parameters used in this paper is listed in Table I. Either BPSK or QPSK is selected for cargo packets to increase the data transmission rate. For QPSK, the achieved bit rate of cargo

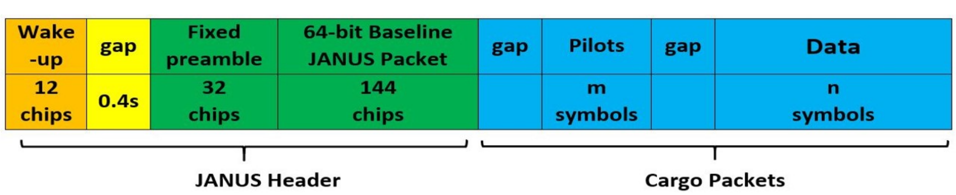

Fig. 1. The signaling structure of a JANUS packet followed by cargo packets.

packets is 23 kbps; while for BPSK, the bit rate is 11.5 kbps. The structure of cargo packets is also shown in Fig. 1, where pilot symbols are added before the data and used for channel estimation and equalization. Guard gaps are added between JANUS header and cargo packets, and between the pilots and data payload.

We designed a hardware platform and tested the proposed JANUS fast mode in a swimming pool. The results show that the adopted JANUS packet can successfully achieve carrier and frame synchronization in all cases despite some bit errors remaining in the JANUS packet after we fixed the synchronization errors. When the bandwidth and sampling rate are different from the original settings, more careful debugging and parameters adjustment are needed for the example rx Matlab code.

II. Experiment Platform

The experiment platform used in the testing is shown in Fig. 2. It consists of a transmitter chain and a receiver chain.

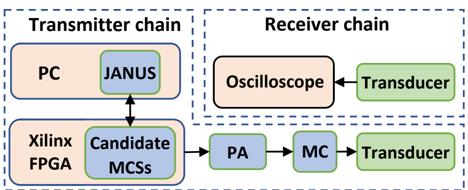

Fig. 2. Experiment platform used in the swimming pool.

A. Transmitter chain

The transmitter chain includes a host PC, a Xilinx FPGA board, Power Amplifier (PA), Matching Circuit (MC), and an omnidirectional acoustic transducer. The JANUS waveform including the 32-bit fixed preamble and baseline JANUS packet is generated based on the simple example sources (the C code or MATLAB code) in the JANUS code repository [6]. A GUI running on the host PC takes charge of loading JANUS waveform as well as cargo information bits into the FPGA through an USB to UART cable. The block structure of signal processing in the FPGA is depicted in Fig. 3. The JANUS waveform is directly transformed by a PWM (pulse-width modulation) module in the FPGA and is fed to a class-D power amplifier (PA).

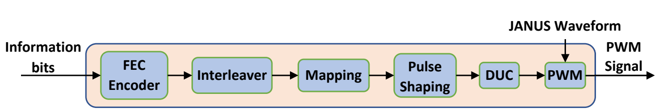

Fig. 3. Block structure of transmit signal processing in FPGA.

The Nexys 4 FPGA board is adopted to process the cargo information bits. The cargo information bits are coded by a rate-1/2 non-systematic convolutional encoder with generator polynomial $[G_1, G_2] = [17, 13]_{oct}$. A 32×32 block interleaver is adopted to combat burst errors encountered in UWA fading channels. A ping-pong interleaver structure is used to support continuous data transmission. The interleaved bits are mapped to either BPSK or QPSK symbols, and then up-sampled and pulse shaped via a square-root-raised cosine filter. The digital up-converter (DUC) module is to modulate the baseband signal to the passband signal. The passband signal is converted to a PWM output which drives the transmitter power amplifier. The FPGA board has an Artix-7 FPGA XC7A100T-1CSG324C with 15,850 Slices, 135 36Kb BRAMs, and 240 DSP48Es. The resource utilization of the transmitter is as follows: 17 BRAM, 2 DSP48E, 3% LUT, and 2% FF.

We modified the center frequency and bandwidth of JANUS standard to 115,000 Hz and 38,000 Hz respectively in the "parameter_sets.csv" file. We also added the sampling frequency 460,000 at the end of the array "COMMON_FS" in the "defaults.m" file. The passband signal of JANUS packet is then generated in MATLAB by the "simple_tx.m" function. The signal is read in the GUI and loaded into the FPGA board at the transmitter end. The guard gap between JANUS packet and pilots is set to 52 ms, and guard between pilots and cargo packets is set to 60 ms. The guard gaps can be adjusted flexibly through the GUI. The wake-up tone defined in the JANUS standard was omitted in this experiment.

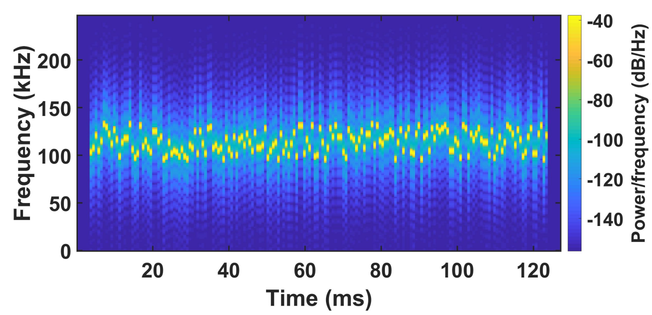

Fig. 4. The spectrogram of transmitted JANUS packet.

Each chip has 314.64 samples at the 460 ksps sampling rate. To show the spectrogram of the JANUS transmitted signal, we up-sampled the generated passband signal to 1,462,000 sps so that each chip has 1000 samples. The spectrogram of the up-sampled JANUS transmit signal using Blackman window is shown in Fig. 4, where the Blackman window size is set to 1000, the overlap is specified to be 0, and the number of FFT points is chosen as 1024. The spectrogram clearly shows frequencies hopping in the band from 96 kHz to 134 kHz.

B. Receiver chain

An oscilloscope Tektronix MDO3104 is used to sample the signal captured from the transducer at 0.5 or 1.0 Msps. The saved signals are processed by MATLAB. The example rx code first transforms the received passband signal into the baseband signal with the "ddc" function. The baseband signal is used to compute the chip correlation in the "chips_alignment" function using the Goertzel algorithm. Based on the correlation, the start of the signal is detected and then converted back to its corresponding index in the baseband signal. From which, the following 32 chips are used to compute the Doppler spread. The start of the signal and the Doppler spread are used together to sample the 144 chips baseline packet. The resulted samples are then demodulated, de-interleaved, and finally decoded using the soft Viterbi algorithm. Note, since the chip duration is changed to around 0.68 ms, we also modified the variable "cfar_mov_avg_time" in the "rx.m" file from 0.15 to 0.015.

For the cargo packets, the carrier phase and symbol start are extracted form the JANUS detection. The Doppler computation offered by JANUS is used in the digital down-converter (DDC) to demodulate the received passband cargo packets to the baseband. The baseband processing algorithms include channel estimation, channel equalization, and MAP decoder. The UWA channels are often sparse with the most energy of the channel impulse response (CIR) concentrated on a few taps but the CIR length is rather long. Therefore, we adopt the improved proportionate NLMS (IPNLMS) algorithm for channel estimation. The LMMSE-based turbo equalizer is adopted for the symbol equalization. The MAX-log-MAP decoder is used to decode the equalized symbols.

III. Experiments and Data Processing

We conducted experiments in a swimming pool with a size of 75 ft in length and 25 ft in width. The depth of the pool changes from 4 ft at the shallow end to 8.5 ft at the deep end. In the experiment, the transmitter is located at the shallow end about 6 ft away from the walls, and the receiver was located towards the deep end at some fixed distances away from the transmitter, such as 5, 15, 25, or 35 ft. The experiment setup is shown in Fig. 5. The parameters of the cargo packets in the experiment are summarized in Table II.

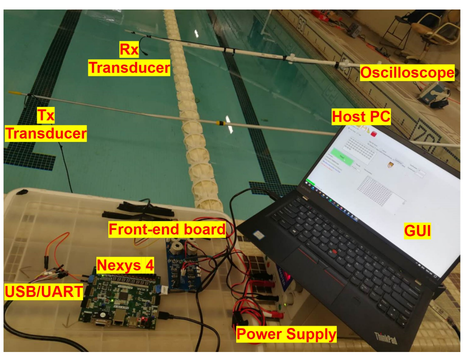

Fig. 5. Experiment setup in a swimming pool.

Fig. 6(a) shows the passband signal captured by the oscilloscope with the receive hydrophone located at 5 ft away from the transmitter projector. The 176 JANUS chips consist of 32 fixed chips plus 144-chip baseline packet, lasting around 120 ms. The 256-symbol pilot and the 1024-symbol BPSK packet last around 11 ms and 44.5 ms, respectively. Strong echoes are clearly present in the gaps between the signals. The spectrogram of the JANUS packet is depicted in Fig. 6(b), where the frequency bins are quite blurry indicating that the strong echoes lead to severe inter-carrier and intersymbol interference. As the sampling rate of the oscilloscope is set

TABLE II
Specification of the Cargo Packets

Specifications	***Values***
Encoding	$[17, 13]_{oct}$
Interleaver	32×32
Channel Estimation	IP-NLMS
Decoder	MAX log-MAP
Equalizer	LMMSE Turbo
BPSK data rate (bps)	11,500
QPSK data rate (bps)	23,000
Tx sampling rate (SPS)	460,000
Pilot Block length (bits)	256
Rx sampling rate (Hz)	1,000,000

to 1 Msps, the number of samples for each chip is 683.995. This introduces 1/2 sample shift in every 100 chips if we set the size of Blackman window to 684, which is negligible in the spectrogram. Therefore, the blurry spectrum is mainly the effect of the underwater channel rather than the sampling error.

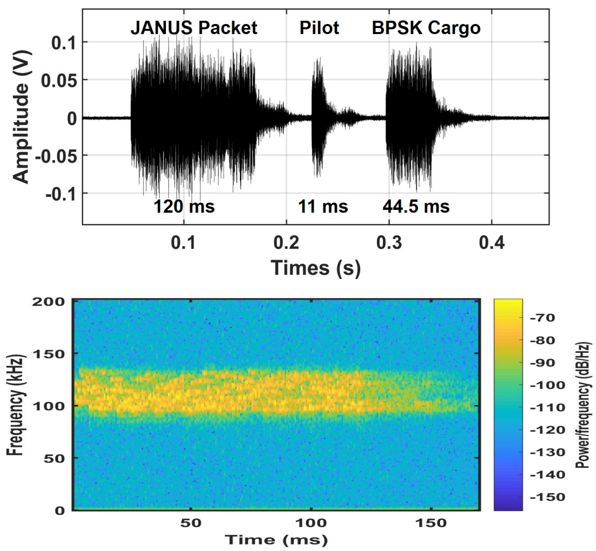

Fig. 6. Received passband signal with Rx at 5 ft. (a) Time-domain waveform consisting of the JANUS packet and BPSK cargo packet; (b) spectrogram of received JANUS packet.

We tested at different locations with the receiver located 5 ft to 35 ft apart from the transmitter. A total of 20 JANUS packets were recorded at each location. As data processing on the received signals at 25 ft exhibit strange behaviors, we plotted the spectrograms of several packets received at 25 ft and at 35 ft, respectively, as shown in Fig. 7. Compared with the packets received at 5 ft and 35 ft, the spectrograms of the packets received at 25 ft are pretty similar to signals received at other locations. We then repeated experiments at 15 ft and 25 ft again. The Bit Error Rate (BER) results of the signals at 15 ft remain similar and the two experiments at 25 ft were different, as shown in Table III, where the two experiments at 25 ft are denoted as 25_1 and 25_2, respectively.

First, we use the example rx.m Matlab code to process the received signal directly with the only changes on two

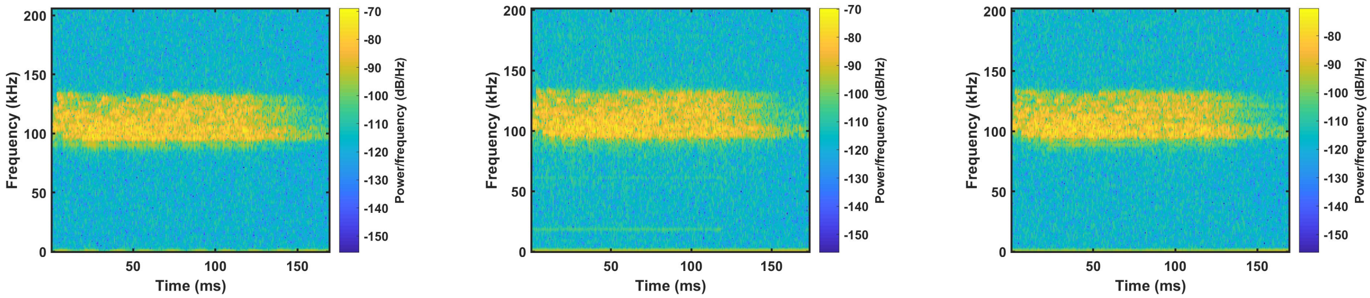

Fig. 7. The spectrogram of received JANUS packet at (a) 25 ft, BER=0.3809 (b) 25 ft, BER=0.0938 (c) 35 ft.

TABLE III
DECODE PERFORMANCE OF THE 144-CHIP JANUS BASELINE PACKETS IN THE SWIMMING POOL EXPERIMENTS

Location (ft)	5	15	25_1	25_2	35
SNR (dB)	56.9286	48.1135	42.8529	40.4107	38.7527
BER	0.4297	0.4105	0.4409	0.4843	0.3103
BER_{mod1}	0	0.1238	0.3101	0.294	0.2254
BER_{mod2}	0	0	0.3809	0.0938	0.03571

TABLE IV
DECODE PERFORMANCE OF THE GENERATED TX PASSBAND SIGNAL

Number of zero samples padding	***BER***
800	0.4844
916	0.1094
917	0
1580 (5 chips)	0
39617	0
39618	0.1094
48980 (155 chips)	0.3594

parameters "cfar_mov_avg_time =0.015" and "threshold = 2.0". The resulting BER was mostly around 0.4 as shown in the row of BER in Table III, which seems to match the results reported in [5]. However, when we truncated the recorded noise segment preceding the JANUS signal, the BER of some signals became 0, and some improved slightly. This behavior prompted us to probe further into the rx.m code and made some modifications. The two ways of modification yield different improvements of the decoded JANUS packets, as shown in the rows BER_{mod1} and BER_{mod2} in Table III. Next, we illustrate how the modifications are made and how the BER performance are affected.

Before making any modifications to the matlab rx programs, we tested the JANUS example rx Matlab code with the generated tx packets passing a pure delay channel by padding different lengths of zeros preceding the JANUS packet. The resulting BER is shown in Table IV, where the decoder works when the number of padded zero samples is in the range from 917 to 39,499. That is, the longest time duration of the noise preceding the JANUS signal is about 85.87 ms. Observing the packets we recorded in the experiments, the noise interval ahead of the JANUS signal is up to 300 ms. If we truncate the noise samples such that the JANUS signal appears after 917 samples and before 39,499 samples, then the packets have a good chance of being decoded properly.

After looking into the code, we found that rx.m used Matlab functions fix(), round(), and floor() in many places and in both baseband and passband processing of chip correlation and carrier synchronization. These cause discrepancies in some variables such as "step_s" used in the "chips_alignment.m" function, "offs" and "chips_nsample" in "rx.m", etc. For example, if 2.048×10^5 zero samples is prepended to the JANUS packet, which is similar to the case in our experiments, the variable "offs_detector" offered by the "detect_first" function in "rx.m" is about 2.5907×10^3. Therefore, the small approximation error in fix() on line 77 of "rx.m" leads to $(79-78.6593)/4 \approx 0.0852$, which would cause $2.5907\times10^3 \times 0.0852 \approx 221$ samples of shift in the detected synchronization in the baseband signal. Since each chip has about 79 samples in baseband, this yields an offset of around 3 chips, as shown in Fig. 8. This synchronization error results in 27 error bits out of the 64 JANUS bits.

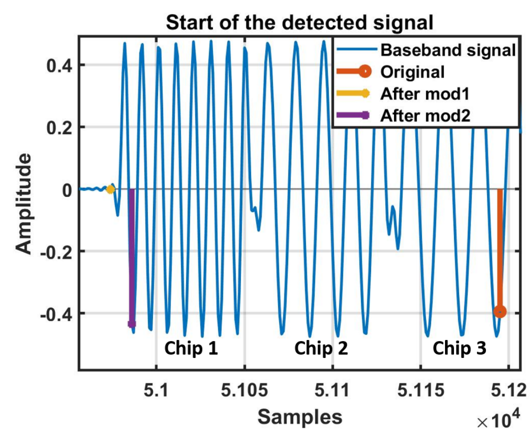

Fig. 8. The synchronization comparison in baseband for the generated tx passband signal padded with 2.048×10^5 zero samples.

We modified the "chip_nsmaple" on line 116 in "rx.m" to "(pset.chip_dur * bband_fs)" which is the same calculation as the variable "step_s" on line 52 in "chips_alignment.m". This modification is denoted as *mod1*:

```
% Converting oversampled chip in baseband ...
    index and time.
%original JANUS rx.m line 116
offs = fix(offs_detector * chip_nsample / ...
    chip_oversampling + align_delay * bband_fs);
% modification 1: replace chip_nsample
offs = fix(offs_detector * (pset.chip_dur * ...
    bband_fs) / chip_oversampling + ...
    align_delay * bband_fs);
```

With *mod1*, no matter how many zero samples precedes the JANUS packet in the generated tx passband signal, the synchronization error is drastically reduced for all received packets and the BER performance is improved significantly. The average BER of received JANUS baseline packets are listed in Table III. We can see that BER_{mod1} of the packets received at 5 ft went to 0, and the BER_{mod1} at other locations is also reduced.

In addition, we applied the fix() function in the calculation of the variables "step_s" and "offs". This modification is denoted as *mod2*:

```
%original JANUS chips_alignment.m line 52
step_s = (chip_dur * bband_fs) / ...
    chip_oversampling;
% modification 2: use fix() function
step_s = fix(chip_dur * bband_fs) / ...
    chip_oversampling;
```

```
% Converting oversampled chip in baseband ...
    index and time.
% modification 2: replace chip_nsample
offs = fix(offs_detector * fix(pset.chip_dur ...
    * bband_fs) / chip_oversampling + ...
    align_delay * bband_fs);
```

The detected signal start point of the pure-delay channel is shown in Fig. 8, where the baseband received signal is a delayed version of the transmit signal and the detected synchronization points are compared for the original, *mod1* and *mod2* schemes. The synchronization point of mod2 is 12 samples behind the start based on *mod1*. The BER_{mod2} of received packets at *mod2* is better at all locations except at 25 ft of experiment 1. We guess that some other settings in the Matlab code may cause potential error and would need to be modified. Further debugging is needed to find the reason that caused the relatively higher BER at 25 ft in experiment 1.

Besides JANUS packets, we demodulated and down-sampled the pilot and cargo packets from the received signal based on the synchronization information detected by the 32 chips. The estimated baseband channel impulse responses (CIR) at different locations are depicted in Fig. 9, where large number of cargo samples are used as pilots for channel estimation. The BER results of the cargo packets are below 10^{-3} for all locations when Turbo equalization is utilized.

IV. Conclusion and Future Work

In this paper, we have implemented a JANUS fast mode operating in a very high frequency band spanning 38 kHz with a center frequency of 115 kHz. The 176-chip FH-BFSK JANUS packet is reduced to 120 ms and the cargo packets have a symbol rate of up to 23 ksps using either BPSK or QPSK. The experiments conducted in a swimming pool show that the adopted JANUS packet can successfully achieve carrier and frame synchronization in most cases after we modified the Matlab receiver program to reduce the synchronization errors. However, some received signals still exhibit large BER despite their high SNRs and good synchronization. More careful debugging and parameters adjustment are needed for the example rx Matlab code when the bandwidth and sampling rate are different from the original settings. In addition, if the bandwidth is increased to 20 kHz and higher, we may consider increasing the spreading gain beyond the 13 pairs of frequency tones to combat the severe multipath UWA channel.

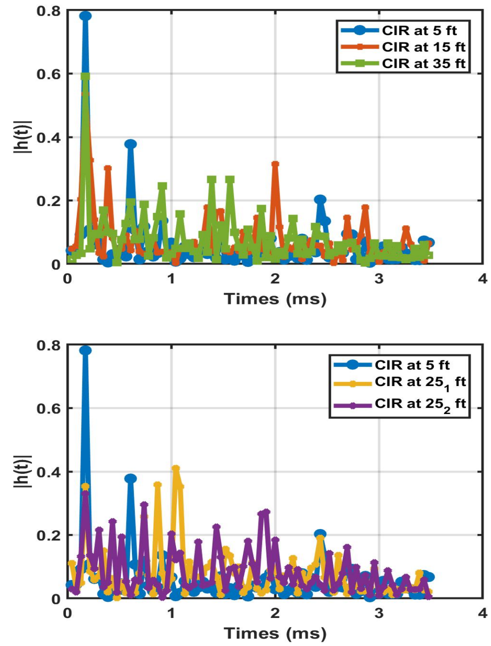

Fig. 9. The estimated channel impulse response of the receiver at different location (a) 5 ft, 15 ft, and 35 ft (b) 5 ft, 25_1 ft, and 25_2

References

[1] S. NATO, "4748 ed. a ver. 1:'digital underwater signalling standard for network node discovery & interoperability'," 2017.

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

[3] J. Alves, T. Furfaro, K. LePage, A. Munafò, K. Pelekanakis, R. Petroccia, and G. Zappa, "Moving janus forward: a look into the future of underwater communications interoperability," in *OCEANS 2016 MTS/IEEE Monterey*. IEEE, 2016, pp. 1–6.

[4] K. Pelekanakis, L. Cazzanti, G. Zappa, and J. 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.

[5] D. Sura, M. Pfetsch, N. Tran, D. Radosevic, P. Longhini, and B. Neuner III, "Implementation and testing of the janus standard with ssc pacific's software-defined acoustic modem," Space and Naval Warfare Systems Center Pacific San Diego United States, Tech. Rep., 2017.

[6] G. Zappa, "JANUS code repository," http://http://www.januswiki.com, 2018.