Sensors and Materials, Vol. 36, No. 2 (2024) 701–728
MYU Tokyo

S & M 3556

Frequency-shift Keying-based Underwater Acoustic Sonar Sensor Communication System with Robust Data Processing Technology

Chih-Ta Yen[1*] and Guo-Chang Wang[2]

[1]Department of Electrical Engineering, National Taiwan Ocean University, Keelung City 202301, Taiwan
[2]Department of Electrical Engineering, National Formosa University, Yunlin County 632, Taiwan

(Received August 10, 2023; accepted January 30, 2024)

Keywords: frequency-shift keying (FSK), data robustness, sonar, software-defined radio, underwater acoustic modem, transceiver design

Marine communication has attracted widespread attention in the wake of global warming and national defense awareness in recent years, where acoustic systems enabling smooth underwater communication have become a popular research topic. In this study, we proposed an underwater communication system equipped with modulators and demodulators (modems). Specifically, the wireless communication software GNU's Not Unix Radio (GNU Radio) was adopted to develop communication formats and architectures applied to a Raspberry Pi computer, on which we developed a communication format available for signal synchronization to increase system robustness against underwater channel interferences. We also developed hardware and software modems for the system. The hardware device was developed using the XR2206 function generator and XR2211 equipment for frequency-shift keying (FSK)-based modulation and demodulation, respectively. The software FSK modem was developed using GNU Radio. To increase the robustness of the transmitted data, we added a cyclic redundancy check through the Async CRC32 block, after which the power distribution unit (PDU) to the Tagged Stream block placed a tag to the start of data combined with effective loads and CRC32, facilitating the attachment of header generators. Four experiments with different underwater scenarios were then conducted to verify the proposed system. The hardware modem had a 0 bit error rate (BER) during experiments implemented in a laboratory sink, a 20-m-long indoor swimming pool, and a 50-m-long outdoor swimming pool, compared with BERs of 2.8×10^{-3} and 6.6×10^{-3} during the breeding water tank experiment under transmission rates of 1 and 2 kbps, respectively. Similarly, the software modem delivered 0-BER performance in the laboratory sink as well as in the indoor and outdoor swimming pools, with its BERs being 2.3×10^{-4} and 2.98×10^{-4} under 1 and 2 kbps transmission rates, respectively. The proposed system comprises preambles, headers, and effective load communication formats for signal protection. In particular, its directional ultrasonic cylindrical piezoelectric transducer can mitigate multipath effects and interferences, thereby enhancing system robustness.

*Corresponding author: e-mail: chihtayen@gmail.com
https://doi.org/10.18494/SAM4693

ISSN 0914-4935
https://myukk.org/

1. Introduction

Earth's surface is predominantly covered by the ocean, occupying over two-thirds of its area. This highlights its vital role in regulating temperature, supporting biodiversity, and affecting various complex weather patterns on our planet. Furthermore, the ocean holds significant importance for human existence. Despite centuries of seafaring history, the vast body of saltwater still retains an air of mystery, emphasizing the need for us to carefully navigate and optimize our utilization of this limited resource.

In the exploration of underwater environments and resources, underwater wireless communication has attracted significant attention. Over the years, relevant technologies have evolved rapidly, progressing from the challenging underwater radio frequency communication of the 1960s to modern underwater acoustic communication methods. Sonar systems have been widely adopted to collect diverse data in marine environments, serving purposes such as monitoring environmental pollution, facilitating deep-sea exploration, detecting underwater seismic and volcanic activities associated with climate change, and enabling long-range underwater image transmission.

The use of sonar systems for underwater communication has developed for more than a hundred years. Acoustic technologies are simpler than their electromagnetic radiation and optical communication counterparts. With low-frequency signals that attenuate slowly in water, sonar systems are capable of long-range transmission from hundreds of meters to several kilometers. Thus, sonar-based technologies have been widely applied by submarines and vessels to underwater object detection or marine navigation.

Underwater acoustic communication is currently the major and most mature underwater communication technique, but acoustic channel signals often face multiple constraints due to the variability of underwater channels. Moreover, commercial modulators/demodulators (modems) are costly, prompting researchers to design high-efficiency modems with a low cost. For example, Li *et al.* proposed an energy-efficient frequency-shift keying (FSK) modem for short-range transmission based on a field-programmable gate array (FPGA).[1] The proposed design not only enabled accurate symbol synchronization but consumed only 54% of look-up tables in the adopted Spartan 3 xc3s2000 FPGA. Similarly, Krivić and Štimac[2] developed an FPGA-based binary phase-shift keying (PSK) modem for scenarios with high noise levels. They analyzed the properties and tradeoffs of two common methods for binary PSK demodulation and discovered that the two methods are suitable for FPGA realization. They also synchronized relevant filters and integrators, with simulation results revealing that the FPGA enabled the design of phase modulation with symbol synchronization and the establishment of an efficient physical layer based on BPSK. Specifically, they showed the shift of an acceptable bit error rate (BER) to a lower signal-to-noise ratio (SNR), where the BER ranged between 10^{-4} and 10^{-7} under communication conditions with 10–20 symbol length and 0 SNR.[2] Although the use of FPGA can ensure system stability, it lacks flexibility to modify system parameters for experiments in different sea areas once the system design is completed. Therefore, we initially designed the modem to have the capability of adjusting parameters at any given time, allowing for convenient experimentation in various sea areas in the future. Indriyanto and Edward

developed an FSK-based underwater modem using microcontrollers, amplifiers, and sensors.[3] The modem successfully received data at 1200 bps and captured the gradient relationship between range increment and BER, indicating a 0% BER at distances of 100 and 130 cm. The rate was 6.2% at 150 cm, compared with the largest bit error of 35% at 170 cm.[3] Su *et al.* proposed an embedded system compatible with multiple-carrier multiple FSK (MC-MFSK) and MFSK and conducted an experiment on the South China Sea.[4] Their results showed that the system had a BER of 5×10^{-2} at a distance of 2.5 km under a transmission rate of 1.07 kbps when MC-MFSK was adopted. When SC-MFSK was used under 125 bps, BERs were 3×10^{-2}, 1×10^{-4}, and 1×10^{-5} at distances of 8, 5.3, and 4.8 km, respectively. Another study adopted M-ary orthogonal code keying to implement BPSK modulation and completed 99.5% of data transmission at acoustic sensor nodes in the North Sea under various sea states.[5] While their transmission may be longer, they often suffer from a relatively low data transfer rate and a higher BER.

The properties of underwater communication channels are highly complex, causing the phase and frequency of transmitted signals to severely distort. In this study, FSK was adopted for frequency modulation as this approach is simple and unlikely to be affected by phase distortion. To develop an energy-efficient and cost-effective underwater acoustic modem, researchers have proposed several designs for different hardware platforms. For example, van Kleunen *et al.* applied the dual-core ARM Cortex-M4 processor to modem control and networking protocols.[6] Specifically, the modem achieved data transmission within a 140 m range at a rate of 150 bps in a small artificial lake using 20–22 kHz FSK.[6] Regarding modem development, DelPreto *et al.* assembled a Raspberry Pi B+, HiFiBerry DAC+, amplifier, and output transformer into a transmitter, while the receiver comprised the ARM Cortex-M3 processor and a microcontroller.[7] The modem, set to 30–36 kHz FSK, was capable of error-free communication at a rate of 20 bps and a distance of 0.5 m within an experimental fish tank.[7] Using a low-cost acoustic modem developed by AppliCon, Cario *et al.* implemented MFSK at a frequency ranging between 25 and 35 kHz and successfully transmitted data along a 100-m-long pier.[8] Renner *et al.* applied the ATMEL AVR32UC3 microcontroller to demodulation using 14–30 kHz BFSK.[9] In an underwater experiment, the microcontroller achieved 2 kbps data transmission at a distance of 9 m. On the basis of the high-performance multiband FSK, Jeong *et al.* completed data transmission at a depth of 50 m below a lake by setting the center frequency and data rate to 16 kHz and 20 bps, respectively.[10] Subsequently, a hydrophone and transducer were installed at the lakebed and a horizontal level of 300–500 m, respectively, along with a receiver with a sampling frequency of 192 kHz.[10] Zia *et al.* applied stationary data streams stored in Raspberry Pi to an FSK modem using microcontrollers and set the frequency range at 30.5–32.5 kHz; the modem achieved errorless transmission at a rate of up to 300 bps in a water tank experiment testing its performance at 100–1200 bps.[11] On the basis of the findings of the aforementioned studies, the primary focus of this research has been on the development of hardware for underwater communication systems. This includes designing embedded devices, optimizing amplifiers, and reducing the cost of the hardware setup. The objective of this study is to leverage the advantages highlighted in previous reference research to realize and enhance the underwater communication system, aiming to achieve affordable and reliable communication capabilities.

Considering the high variability of acoustic channels in water, the parameter adjustment of acoustic modems during operations is critical. GNU Radio, an open-source toolkit for implementing software-defined radios, allows users to adjust system parameters at each layer without the need to use exclusive hardware to tackle the variability problem of underwater acoustic channels. By developing a BPSK-based modem using GNU Radio, Zhu completed signal transmission in a multipath underwater scenario with a BER of 1.7×10^{-3}.[12] bin Abbas *et al.* also applied GNU Radio modulation and demodulation;[13] the adopted modem with FSK of 14–16 kHz frequencies delivered packet receiving rates of 80% at a distance of 24 m and 70% at 25 m in a swimming pool and a lake, respectively. Demirors *et al.* accomplished transmission at a rate of 104 kbps in a refrigerator using orthogonal frequency-division multiplexing (OFDM) with a BER of 2×10^{-5}.[14] The research team conducted another experiment in a lake using the modem to transmit data from distances of 98 and 200 m at 208 and 260 kbps, respectively, with BERs of 10^{-3} and 10^{-2}.[14] Ahmad implemented underwater communication in a simulated arctic environment (with 0, 20, 40, and 80% of ice cover) and analyzed signals of BPSK, FSK, frequency hopping continuous phase frequency-shift keying (FH-CPFSK), and OFDM.[15] The results indicated that the BER of OFDM decreased with degraded performance when ice cover increased. Conversely, that of FSK remained unaffected despite changes in ice cover. Ahmad also discovered that with a constant symbol time, BPSK, OFDM, and FH-CPFSK were affected in long-range communication, except that FH-CPFSK was less affected, whereas FSK was the least affected. In brief, FSK and FH-CPFS outperformed BPSK and OFDM.[15]

On the basis of the previous literature, we developed communication protocols using GNU Radio and implemented them on Raspberry Pi. Signal modulation and demodulation were achieved using custom preambles, headers, effective load settings, and synchronization techniques. For hardware design, an integrated circuit (IC) XR2206 was utilized as the FSK modulator, and XR2211 was used as the FSK demodulator, serving as the modem in this study. The choice of these ICs was due to the convenience of designing an FSK modulator and demodulator with desired encoding and decoding frequencies based on their data sheets. However, note that the modem developed with these ICs is limited to specific frequency ranges and cannot be easily switched to other frequencies. Consequently, if a different transducer is employed, the circuit needs to be redesigned accordingly. In contrast, by leveraging GNU Radio as a versatile software toolkit for modems, signal modulation and demodulation can be achieved without the need for additional modem circuits. In experiments conducted in sea environments without internet access, GNU Radio allows researchers to modify parameters and switch to different modulation techniques in real time. However, it is important to mention that as with other open-source toolkits, when GNU Radio undergoes upgrades, certain library functions may be revised. For this particular study, we utilized GNU Radio version 3.10.

The rest of the paper is organized as follows. In Sect. 2, we describe the selected hardware and designs of signal processing blocks provided by GNU Radio. In Sect. 3, we detail the experimental environment, data measurements, and multipath analysis. In Sect. 4, we present our experimental results and discussion. In Sect. 5, we show our conclusions.

2. Software and Hardware Designs for Underwater Acoustic System

2.1 Software design

2.1.1 GNU Radio for software-defined radio development

Software-defined radio (SDR) is a microcontroller system supporting digital signal processing, various wireless communication interfaces and protocols, and configuration resetting. The adopted GNU Radio is an open-source toolkit for SDR development comprising flow charts and signal processing blocks. Programming was completed using C++ and Python. Signal processing was achieved using C++ for efficiency concerns, while flow charts were established using Python. Specifically, the created flow graphs were connected to transmit data from one module to another, ultimately achieving the processing and application of various signals. In addition to the default flow graphs, GNU Radio allows users to self-design modules to control signals in different ways as per their needs, hence its suitability for SDR development.

2.1.2 GNU Radio modulation/demodulation architectures as part of the software system

2.1.2.1 Software FSK transmitter architecture

To generate the payload, we employed a pseudo random binary sequence (PRBS). Additionally, we included a CRC-32 check code to verify the correctness of the payload. To ensure the integrity of the packet, we added a header code for verification purposes. Furthermore, we incorporated a guard interval to safeguard signal integrity. Once connected in series, these components undergo constellation modulation to convert them into base frequency signals. Subsequently, these base frequency signals drive a voltage-controlled oscillator to oscillate and produce two distinct output frequencies.

The Vector Source produced preambles and postambles to ensure signal integrity. Next, tags were added to non-tagged preambles through the Stream to Tagged Stream block. A Random PDU Generator was used to generate pseudorandom sequences regarded as effective loads. To verify the data, we added a cyclic redundancy check through the Async CRC32 block, after which the PDU to Tagged Stream block placed a tag to the start of data combined with effective loads and CRC32, facilitating the attachment of header generators. ProtocolFormatter created headers for each data packet, while Tagged Stream Mux incorporated headers and data into packets and connected the aforementioned preambles to the start and end of each packet. Considering that the data were processed by bit rather than by byte, we converted data in bytes into those in bits using the Constellation Modulator block to achieve constellation modulation. Next, we resampled the output bits, each containing 4 samples per symbol originally, into 192 samples per symbol using the Rational Resampler with an interpolation rate of 48. The sample rate was set to 192k in this study; as 192k/192 is equal to 1000, the data transfer rate was 1 kbps. Another data rate setting was 2 kbps, for which the interpolation rate was set to 24. As the

complex output of two constellation points generated in this study was 0, we converted the complex signals into real ones through Complex To Real. Subsequently, Binary Slicer was used to perform one's decision to reduce errors, thereby facilitating desirable frequency selection. Chunks to Symbols then mapped 0 and 1 as the selected sine wave frequencies, the different frequencies of sine waves of which were formed through VCO. Finally, Audio Source was used to select a signal transmitter. Figure 1 illustrates the architecture of the GNU Radio FSK transmitter as part of the software system. Considering that the software/hardware modem systems were developed, the same communication format was applied to the hardware/software FSK modem, as indicated by the red box in Fig. 1.

2.1.2.2 Software FSK receiver architecture

In terms of reception, the signal is first recovered and then mixed to transform it into an orthogonal form. Subsequently, the signal is down-sampled to reduce the complexity of data processing. After orthogonal demodulation, the number of symbols in the base frequency signal is reduced to achieve clock synchronization and form the constellation. Next, the signal is demodulated, and the protection, header code, and CRC-32 are removed to extract the effective attached signal.

The Audio Source was adopted to set up the selected equipment and acquire the inputs from an analog-to-digital converter (ADC). Next, the Throttle block limited the amount of messages to prevent the receiver from failing to process signals owing to an excessively high data transfer rate. Fixed values were then input into the signals using the Constant Source, after which the

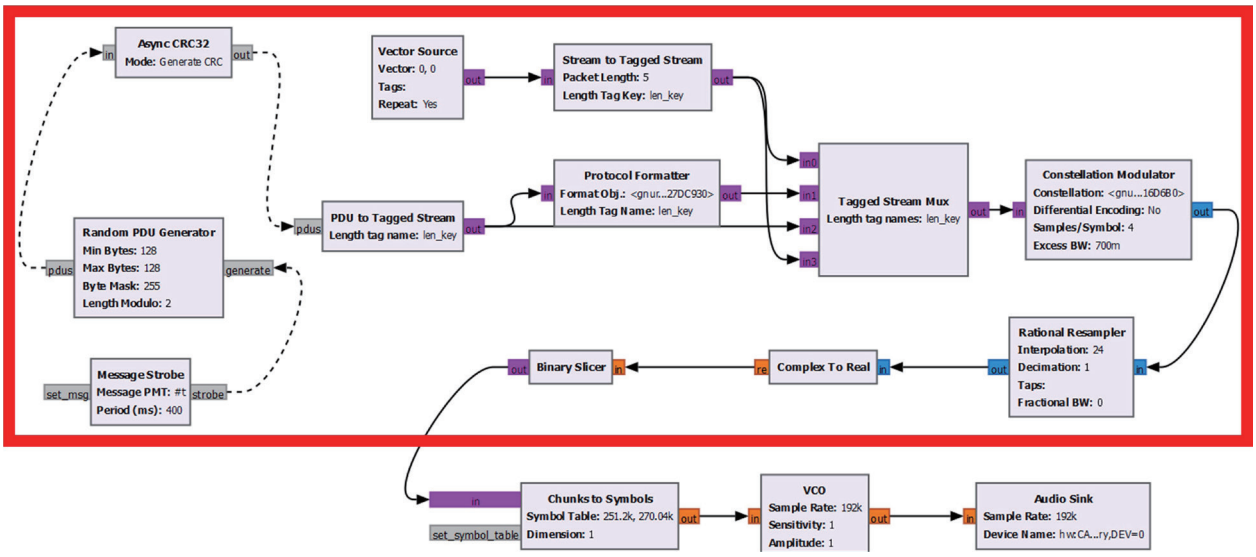


Fig. 1. (Color online) Architecture of GNU Radio FSK transmitter as part of the software system.

signals were converted into complex numbers through the Float To Complex. Subsequently, center frequencies with a 90° difference were multiplied through the Signal Source and Multiply blocks to achieve frequency down-conversion and quadrature demodulation. Rational Resampler divided the data rate by 8 to achieve down-sampling and reduce data processing complexity. Next, the Power Squelch module was used to avoid demodulation when signals are not received. Quadrature Demod was performed to demodulate FSK signals, after which the demodulated signals were filtered using the low pass filter to obtain more desirable signal waveforms. The researchers then converted the signals into their complex form by clicking the Float To Complex and maintained the signal intensity between −1 and 1 by clicking the Feed Forward AGC. The Symbol Sync module was adopted to synchronize the demodulated signals. Later, the constellation decoding of these signals was performed using the Constellation Decoder. The Map block mapped the signals to the values in the flow graph. To obtain effective loads, the block "Correlate Access Code - Tag Stream" synchronized them and extracted effective loads. The Repack Bits and Tagged Stream to PDU blocks then restored the data back to their original form in bytes. We then used the Async CRC32 module to verify the effectiveness of the obtained data and displayed effective loads through the Message Debug. Figure 2 depicts the architecture of the GNU Radio FSK receiver as part of the software system.

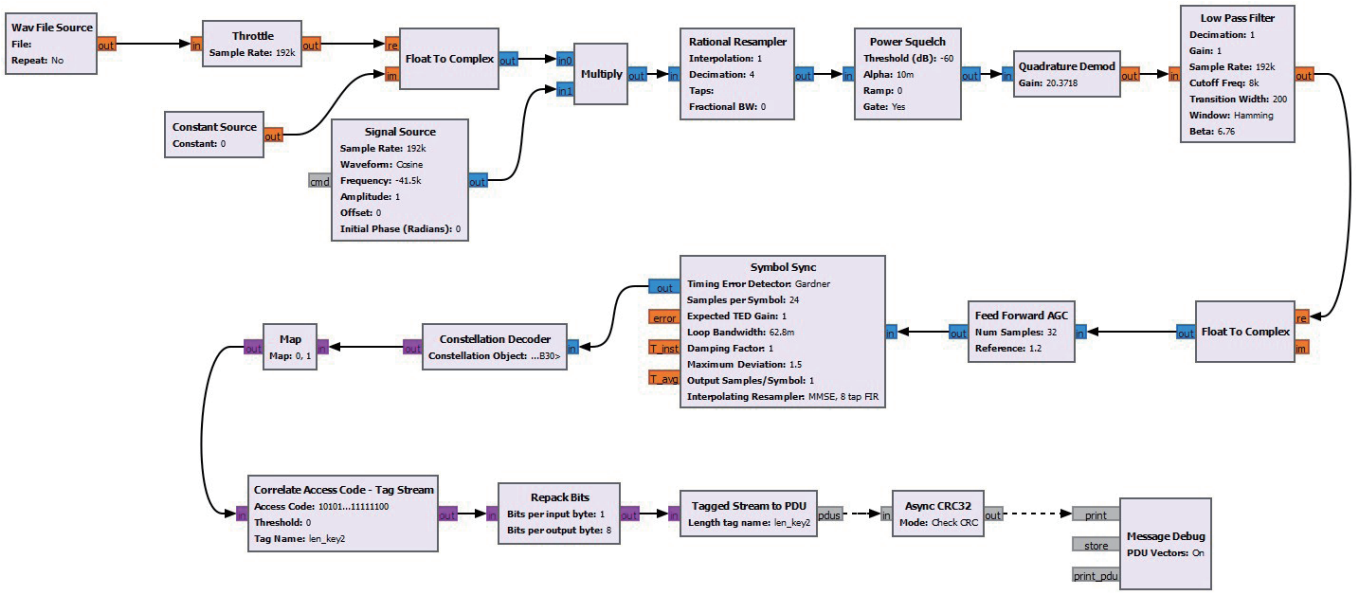


Fig. 2. (Color online) Architecture of GNU Radio FSK receiver as part of the software system.

2.1.3 GNU Radio modulation/demodulation architectures as part of the hardware system

2.1.3.1 Hardware FSK transmitter architecture

We generated the payload using PRBS and included a CRC-32 check code to verify its correctness. We added a header code to ensure the accuracy of the packet and included a protection interval to maintain the signal integrity. We proceeded with constellation modulation to convert the signal into a base frequency signal. Then, we converted the base frequency signal into a square wave, allowing the FSK modulation IC to output the desired result.

The adopted IC XR2206 FSK modulator transmitted FSK signals by emitting two transmission frequencies according to the high- and low-voltage levels of a square wave running through the device. The signal formats involved were the same as those presented in Fig. 1. Next, the Char To Float block converted the signals into floats to transform them into square waves. Next, the Audio Sink inputs those waves into the FSK modulator, so that FSK signals can be emitted consistently through square waves in the same format. Figure 3 shows the architecture of the GNU Radio FSK transmitter as part of the hardware system.

2.1.3.2 Hardware FSK receiver architecture

In the reception process, the FSK demodulation IC retrieves the signal, enabling clock synchronization. By removing the guard interval, header code, and CRC-32, the effective attached signal can be obtained.

After the received signals were processed using the XR2211 FSK demodulator, the Audio Source was used to collect the demodulated hardware signals. Because GNU Radio cannot directly perform an effective load (actual transmission signals) analysis on the received waveform, we developed a self-defined module to obtain its square wave. However, the module, concerning its computing power, might cause missing data during microcontroller real-time processing. To avoid this problem, the Audio Source and Wave File Sink were adopted as modules solely for signal reception to process the signals before using the module, with the results being stored as a WAV file [Fig. 4(a)]. Next, the block diagram was opened. On the other hand, the Wave File Source was used to read the aforementioned WAV file, and the Throttle, considered a buffer zone, limited the amount of messages to facilitate smooth signal processing. At this point, the self-defined module was adopted. When the demodulated waveform signals are input into an ADC, they tend to deform upon the detection of square waves. To avoid this situation, we restored the waveform using the 127&255 module, after which the Blok module analyzed data by capturing the waveform level every 192k points to restore the waveform until it can be read by GNU Radio. Finally, square waves were converted into a range of 0 and 1 using the 0&1 block for subsequent use. Figure 4(b) presents the flow graph. The red waveform in Fig. 4(c) denotes the obtained hardware-demodulated signals, while the blue waveform represents the waveform of signals after performing the processing procedures specified in Fig. 4(b).

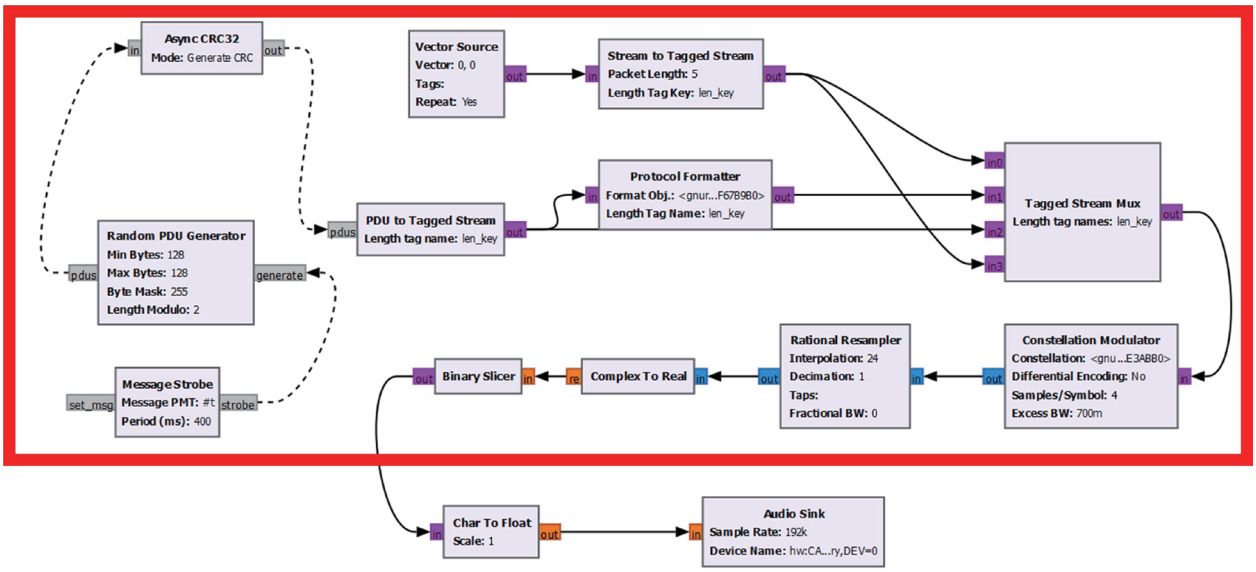
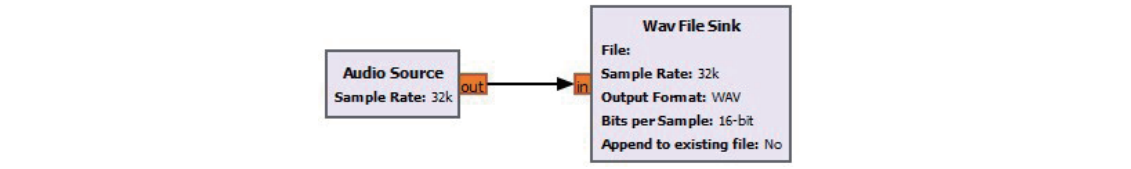
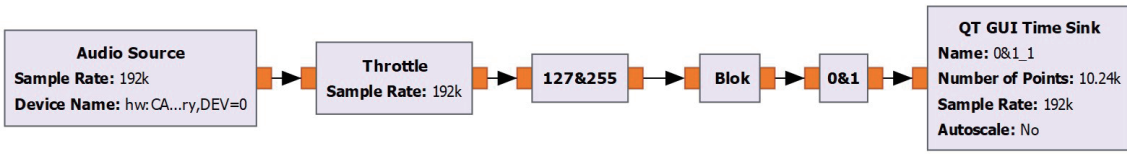


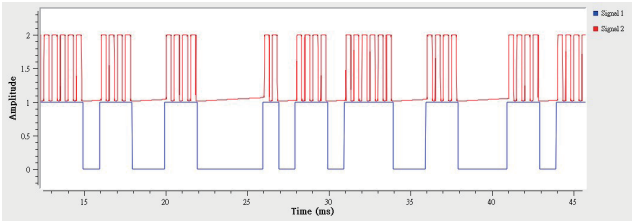
Fig. 3. (Color online) Architecture of GNU Radio FSK transmitter as part of the hardware system.



(a)



(b)



(c)

Fig. 4. (Color online) Signal process units: (a) blocks that solely receive signals, (b) self-developed GNU Radio modules, and (c) obtained waveform from output of Fig. 4(b).

The Gardner of the Symbol Sync block completed signal synchronization. We then used the Float To Char to convert streams of floats into chars, after which the obtained signals were decoded by regarding those with 43 kHz as 0 and 40 kHz as 1. However, the system developed for this study works in the opposite way (i.e., 43 kHz = 1; 40 kHz = 0). Therefore, we adopted the Not block to invert the waveform to provide one that can be detected by the system. Next, the block Correlate Access Code - Tag Stream obtained effective loads from the decoded waveform and synchronize them; the Repack Bits and Tagged Stream to PDU restored the data back to their original form in bytes, which were verified by the Async CRC32. Finally, effective loads were displayed through the Message Debug block. Figure 5 shows the architecture of the GNU Radio FSK receiver as part of the hardware system.

2.2 Hardware design

The following hardware devices were adopted in this study:

2.2.1 Microcontroller

Raspberry Pi 3 is a single-board computer equipped with a quad core 1.2 GHz processor and 40 pins. In this study, the powerhouse was regarded as the center processing unit of the transmitter that emits modulated signals. For data processing concerns, we then used Raspberry Pi 4, which has 10% faster processing performance than the Pi 3 model, to retract transmitted acoustic signals, demodulate them, and compute the BER.

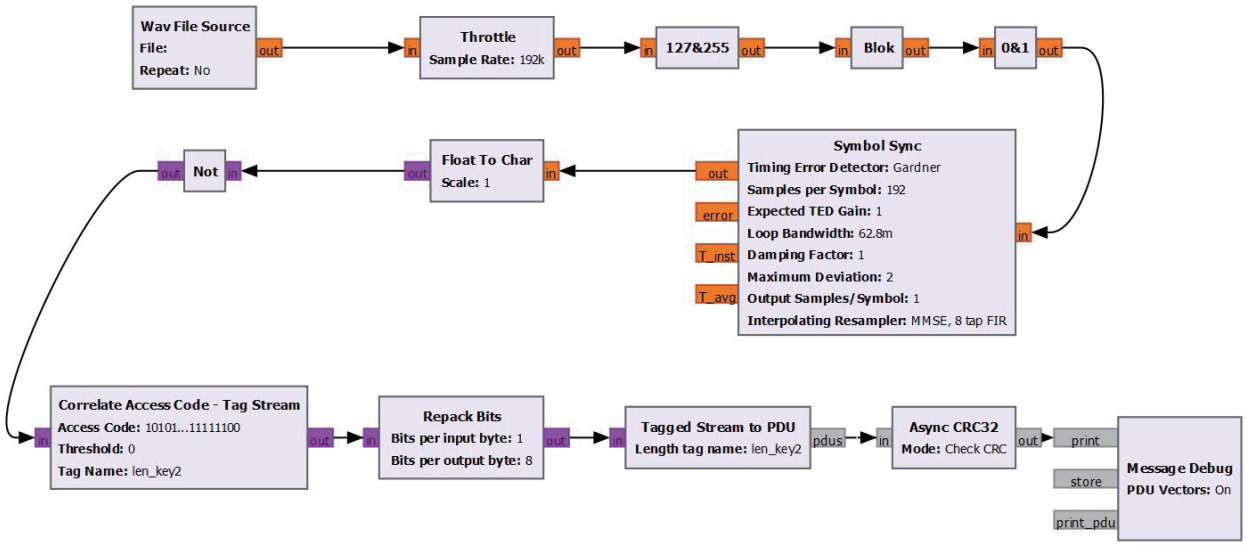


Fig. 5. (Color online) Architecture of GNU Radio FSK receiver as part of the hardware system.

2.2.2 Transducer

Because the experiments in this study would be conducted in a water tank and pool, where multipath effects are strong, we selected a piezoelectric cylinder transducer (model: DYW-40/200-NA) as the transmitter/receiver to mitigate the impact exerted by such effects as suggested in Ref. 16. The transducer offers two frequency ranges, namely, 200 kHz ± 5% and 40 kHz ± 5%, and we chose the 40 kHz ± 5% range for the underwater communication hardware system in this study. Considering that a transducer produces the optimal output under real power, we determined the device's characteristics using the impedance and phase angle with an impedance analyzer (Fig. 6). Specifically, pure resistance was detected at 43 and 44 kHz with phase angles being the closest to 0. However, only the frequencies 40 and 43 kHz were adopted for FSK signal transmission because the XR2211 demodulator could not decode adjacent frequencies.

2.2.3. ADC/DAC

Selecting a proper ADC/digital-to-analog converter (DAC) is critical for communication. An inappropriate ADC/DAC would lead to inaccurate signal transmission and reception. HiFiBerry DAC+ ADC Pro is a sound card suitable for Raspberry Pi and supports sample rates up to 192 kHz. To receive signals in real time, we considered it as a signal transmitter and receiver in this study.

2.2.4. FSK modulation IC

The XR2206 IC is a function generator capable of generating sine, square, triangle, ramp, and pulse waveforms. The desired amplitude modulation frequency or FSK modulation can be customized using a provided circuit. We performed FSK by referring to the IC design proposed in Ref. 17. Two desired frequencies for FSK output were determined by using the equations $F_1 = 1/R_1C$ and $F_2 = 1/R_2C$, where R_1 and R_2 denote Pin 7 and Pin 8, respectively, and C refers to the external capacitor of Pin 5 and Pin 6, from which the generated FSK signals at 40 and 43 kHz were adopted in this study. Table 1 shows the corresponding components on the XR2206 circuit.

2.2.5. FSK demodulation IC

The XR2211 IC was regarded as an FSK demodulator in this study. The detectable voltage for signal detection was set to 10 mV and 3 V, with the IC being able to decode signals ranging from 0.01 to 300 kHz. We adopted the FSK demodulation circuit proposed in Ref. 18. Specifically, FSK signals between 40 and 43 kHz were decoded. As the hardware regarded decoded signals at 40 kHz as 1 and 43 kHz as 0, the waveform after decoding was inverted as per the GNU Radio blocks. Components corresponding to the XR2211 IC are detailed in Table 2.

The impedance of the DYW-40/200-NA transducer varies between different frequencies. To ensure successful impedance matching, the impedance of the transducer should be as close as

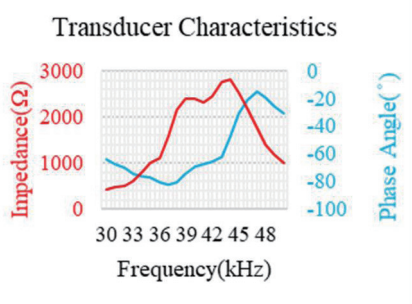


Fig. 6. (Color online) Transducer characteristics by impedance and phase angle.

Table 1
Corresponding components on the XR2206 circuit.

Component value	Component in use
R_1 = 25 kΩ	24 kΩ * 1, 1 kΩ * 1
R_0 = 23 kΩ	22 kΩ * 1, 1 kΩ * 1
C = 1 nF	1 nF * 1

Table 2
Corresponding components on the XR2211 circuit.

Component value	Component in use
R_0 = 15.56 kΩ	12 kΩ * 1, 4.7 kΩ * 1
C_0 = 1.446 nF	1.5 nF * 1
R_1 = 461 kΩ	360 kΩ * 1, 100 kΩ * 1
C_1 = 15.69 pF	15 pF * 1
R_F = 2.304 MΩ	2 MΩ * 1, 300 kΩ * 1
R_B = 11.52 MΩ	10 MΩ * 1, 1 MΩ * 1, 510 kΩ * 1, 20 kΩ * 1
C_F = 46.7 pF	47 pF * 1
R_D = 470 kΩ	470 kΩ * 1
R_I = 5.1 kΩ	5.1 kΩ * 1
C_D > 5.3 mF	5.6 mF * 1

possible to the actual resistance while minimizing their differences at different frequencies. To do so, we set the impedance at 43 and 40 kHz to 1132 $\angle$ −7.5 Ω and 948.7 $\angle$ −20.3 Ω, respectively. R_2 was set to 1 kΩ, resulting in a turn ratio of N_1:N_2 = 1:11. The AB amplifier LM386 was adopted considering its design for low voltage applications, high amplification rate, low distortion, and low noise. The instrument was regarded as a receiver amplifying received signals.

2.3. Physical design and structure of the proposed GNU Radio modem

By referring to Refs. 12 and 19, we designed an FSK modem with hardware and software systems for underwater acoustic communication on the GNU Radio platform and developed the modem by connecting transmitters and receivers physically. According to the aforementioned

software/hardware designs, the FSK modem with transmitter and receiver systems was completed. Figures 7 and 8 illustrate the physical wiring designs of the software/hardware FSK transmitter and receiver, respectively.

2.3.1. Physical layout of software FSK transmitter

The transmitter comprises the Raspberry Pi 3, HiFiBerry DAC+ADC Pro, toroidal transformer, TDA7293 power amplifier, self-developed transformer, and DYW-40/200-NA transducer. To determine its power output, we measured the voltage by separately placing an oscilloscope probe and a current probe at the transducer port and the current by hooking the output wires. An advantage of the transmitter is that its system enables users to adjust power settings between 0.003 and 0.048 W by turning the default knobs in different sizes at the power amplification board. Figure 9 depicts the physical layout of the software FSK transmitter.

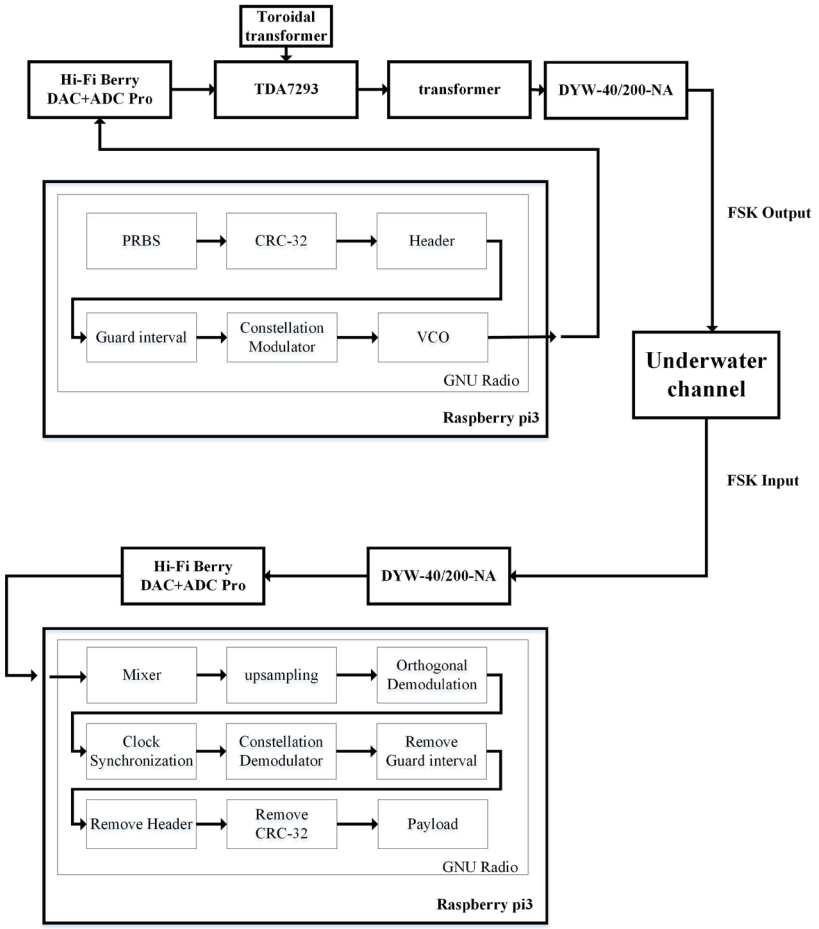


Fig. 7. Physical wiring design of software FSK transceiver.

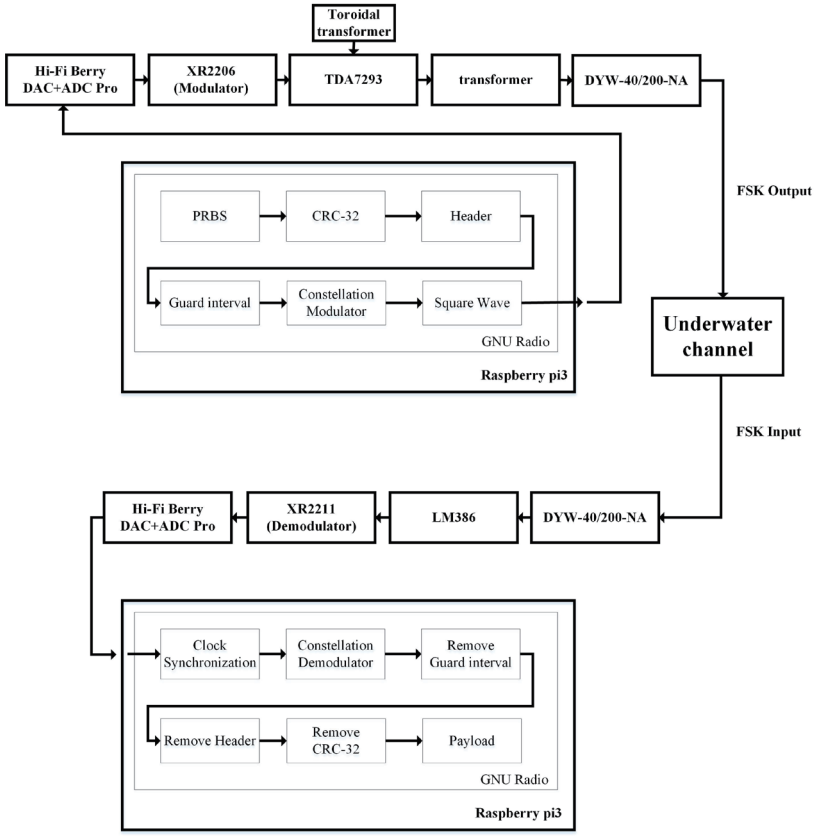


Fig. 8. Physical wiring design of hardware FSK transceiver.

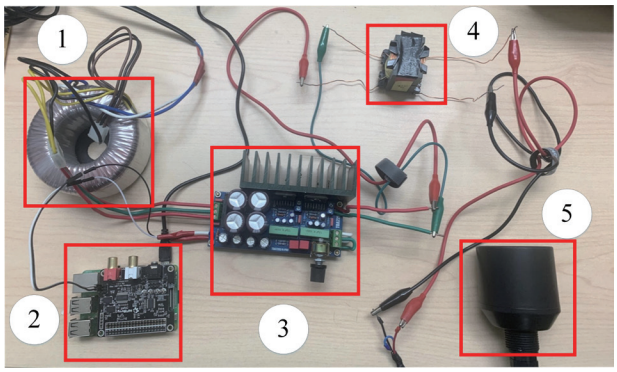


Fig. 9. (Color online) Physical layout of software FSK transmitter: (1) Toroidal transformer. (2) Raspberry Pi 3 and HiFiBerry DAC+ADC Pro. (3) TDA7293 power amplifier. (4) Self-developed transformer. (5) DYW-40/200-NA transducer.

2.3.2. Physical layout of software FSK receiver

The Raspberry Pi 3, HiFiBerry DAC+ADC Pro, and DYW-40/200-NA constitute a receiver for the modem. Because the GNU Radio block diagrams will automatically amplify signals, signal amplification and retracting functions provided by the LM386 preamplifier were not used. Figure 10 illustrates the layout of the software FSK receiver.

2.3.3 Physical layout of hardware FSK transmitter

We developed the hardware FSK transmitter and receiver on the basis of the modem circuit design proposed in Ref. 20. To apply the same communication format as that of the software system, GNU Radio was used to generate and synchronize transmitter signals and drive the XR2206-generated FSK signals along with the use of the XR2211 demodulator.

The Raspberry Pi 3, HiFiBerry DAC+ADC Pro, toroidal transformer, TDA7293 power amplifier, XR2206 FSK demodulator, self-developed transformer, and DYW-40/200-NA transducer constitute the hardware FSK transmitter, the layout of which is depicted in Fig. 11.

2.3.4 Physical layout of hardware FSK receiver

The hardware reception architecture comprises the Raspberry Pi 4, HiFiBerry DAC+ADC Pro, XR2211 FSK demodulator, LM386 preamplifier, and DYW-40/200-NA transducer; Fig. 12 shows the physical layout of the FSK receiver.

3. Experiments and System Settings

To explore underwater channel environments and decoding after signal transmission, the research team analyzed the ultrasonic field, sinusoidal pulse, eye pattern, and constellation diagram of the underwater communication system as well as its overall performance.

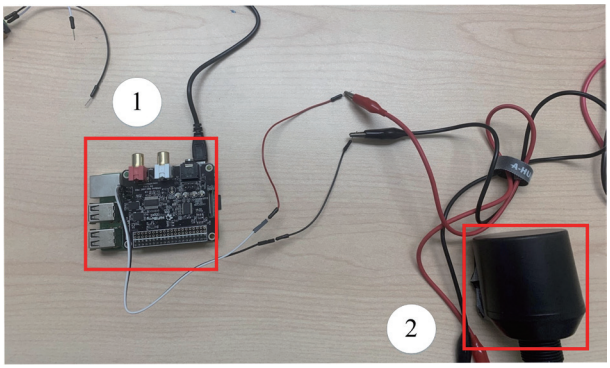


Fig. 10. (Color online) Physical layout of software FSK receiver: (1) Raspberry Pi 4 and HiFiBerry DAC+ADC Pro. (2) DYW-40/200-NA transducer.

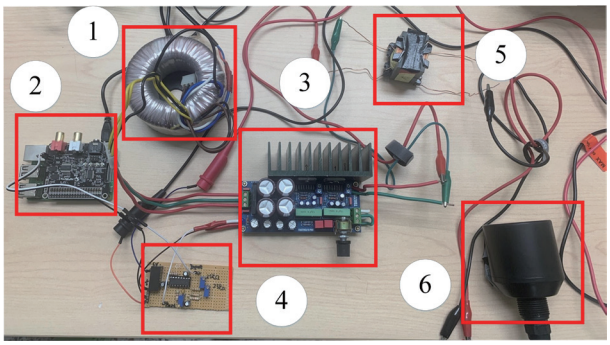


Fig. 11. (Color online) Physical layout of hardware FSK transmitter: (1) Toroidal transformer. (2) Raspberry Pi 3 and HiFiBerry DAC+ADC Pro. (3) TDA7293 power amplifier. (4) XR2206 FSK demodulator. (5) Self-developed transformer. (6) DYW-40/200-Nat transducer.

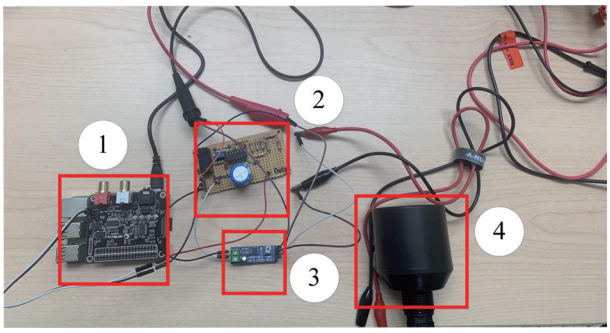


Fig. 12. (Color online) Physical layout of hardware FSK receiver: (1) Raspberry Pi 4+ HiFi Berry DAC+ADC Pro. (2) XR2211 FSK demodulator. (3) LM386 preamplifier. (4) DYW-40/200-NA transducer.

Acoustic transmission in a multipath environment is difficult because multiple wave reflections exist in addition to the direct wave; the resultant interference between reflections is likely to cause inaccurate measurement and serious errors. To determine the properties of the underwater acoustic channels to be tested, we generated a sinusoidal pulse as suggested in Ref. 21; in this way, noise properties and multipath effects in water can be identified. Specifically, three sine waves were emitted in 1 s within the experimental environment at a frequency of 40 kHz.

4. Experiments

4.1. Experimental scenario 1

The performance of the proposed underwater acoustic system in signal transmission was analyzed in four different underwater channel environments. To determine the resultant transmission power, voltage was measured using an oscilloscope probe and a current probe that were separately connected to the ultrasonic module port, while current was measured by

hooking the output wires. By turning the default knobs on the power amplification board, we adjusted signal levels and obtained V_{rms} and I_{rms}, after which three power levels (i.e., 0.003, 0.01, and 0.048 W) for this study were calculated.

$$P = V_{rms} \times I_{rms} \tag{1}$$

The first experiment was conducted in a 100 × 100 × 100 cm^3 sink at the smart IoT laboratory in National Taiwan Ocean University. The sink facilitates real-time software and hardware adjustment and correction, as shown in Fig. 13. Transceivers were placed on both sides of the tank at a depth of 28 cm; moreover, the bottom was kept empty for signal transmission. Considering that underwater communication is affected by various ambient noises, we placed a submersible motor between the two transducers to simulate underwater ocean currents and an electric fan blowing air towards the water surface to simulate ocean waves.

We analyzed the channel response of the four underwater environments by emitting 40 kHz sinusoidal pulse signals with an effort to determine their property differences. Figure 14 depicts signals received in the experimental sink as sinusoidal waves passed through and indicates extremely intense direct and numerous reflected waves as a result of signal transmission within a narrow space. Because the amplitude of direct waves was greater than that of reflected ones, the modulated signals were barely interfered.

(a) (b)

Fig. 13. (Color online) Sink for the first experiment: (a) photo of the tank and (b) experimental layout.

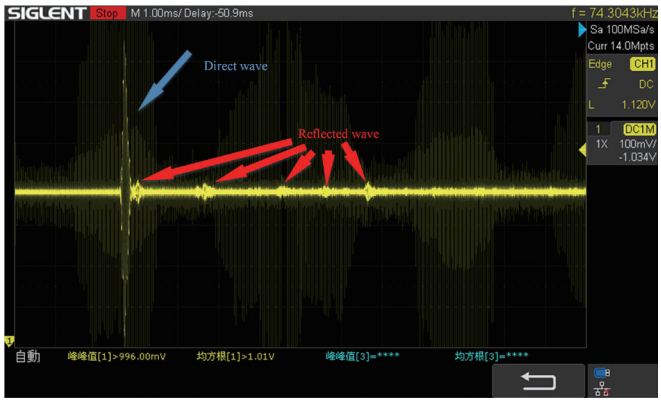


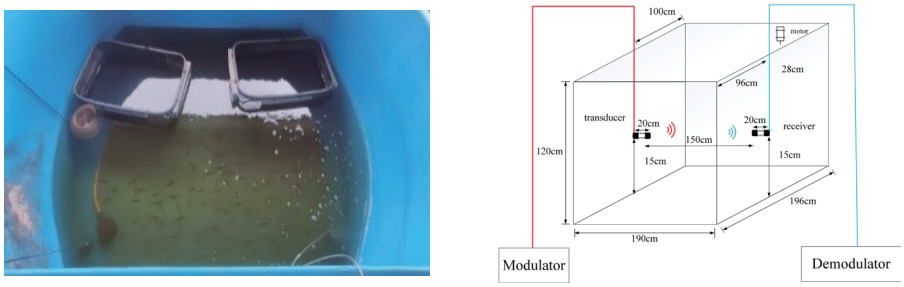
Fig. 14. (Color online) Signals received as sinusoidal waves passed through the experimental sink.

4.2. Experimental scenario 2

The second experiment was conducted in an outdoor tank measuring 200 × 196 × 120 cm^3 (Fig. 15), in which transceivers were placed on both sides at a depth of 15 cm. The tank comprised a pump and fish to simulate a complex environment with noise, and the transceiver performance was examined in the presence of interference caused by animal movements. The received signals in the outdoor tank are presented in Fig. 16, indicating little multipath interference with the environment, where signals were successfully received. However, the amplitude of signal disturbance was present owing to fish movements near the emitter and transceiver, resulting in a high BER.

4.3. Experimental scenario 3

The third experiment was conducted in an indoor swimming pool measuring 20 × 6 × 1.03 m^3 (Fig. 17), in which transceivers were placed on both sides at a depth of 50 cm. The pool comprises a large reflection surface and shallow water, where noise caused by pumps and filters was greater than those in the previous two scenarios; the third experiment simulated a shallow water marine environment and its surface. Figure 18 illustrates sinusoidal signals received in the swimming pool as sinusoidal waves passed through. Despite the presence of multipath effects



(a) (b)

Fig. 15. (Color online) Outdoor breeding tank for the second experiment: (a) photo of the tank and (b) experimental layout.

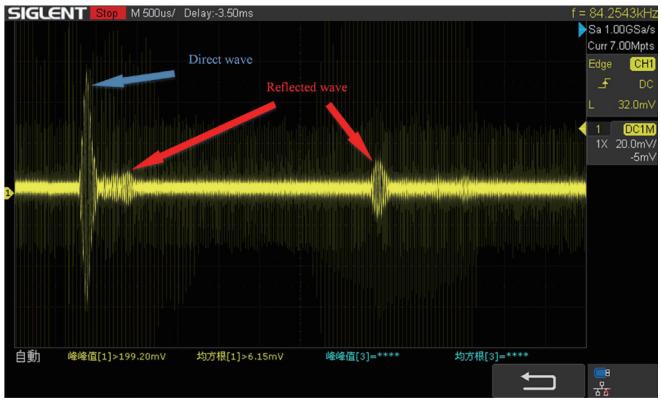
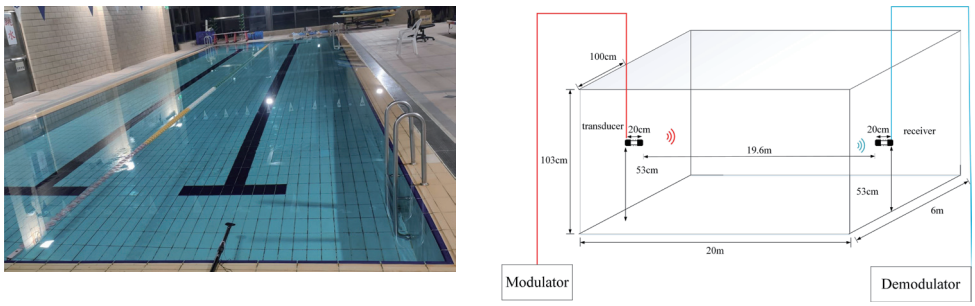


Fig. 16. (Color online) Signals received as sinusoidal waves passed through the simulated breeding environment.



(a) (b)

Fig. 17. (Color online) Indoor swimming pool for the third experiment: (a) photo of the pool and (b) experimental layout.

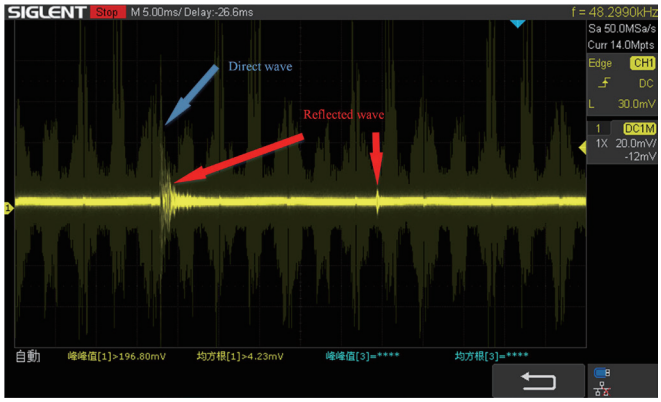
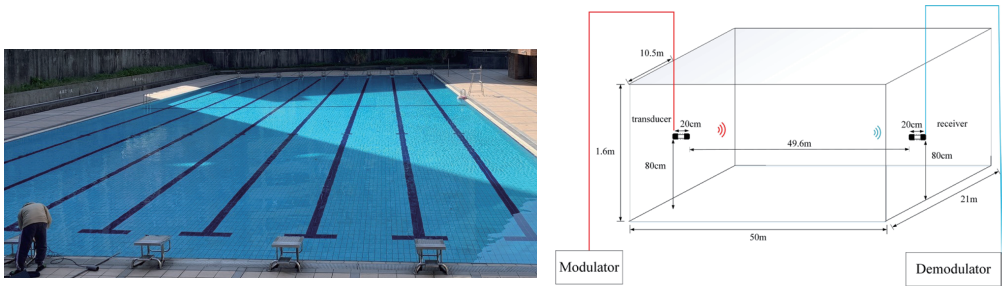


Fig. 18. (Color online) Signals received as sinusoidal waves passed through the indoor swimming pool.

within an environment surrounded by four falls, we increased the power output and adjusted the transducer angle to enhance the main beam power; therefore, signals were successfully transmitted without bit errors.

4.4. Experimental scenario 4

The fourth experiment was conducted in an outdoor swimming pool measurement (Fig. 19), in which transceivers were placed on both sides at a depth of 80 cm. The experiment was designed to explore the effects of long-range underwater communication and strong multipath effects on system performance. Figure 20 shows the signals received in the pool as sinusoidal waves passed through. Considering an underwater environment with severe multipath effects, we increased the output power to strengthen the main beam power transfer, after which the BER remained at 0, indicating errorless transmission. The results demonstrated that the proposed hardware/software modems remained robust under different transmission circumstances. Owing to constraints of transducer and amplifier specifications, we set the long-range communication distance to 50 m to ensure that the proposed system can receive acoustic signals in water and accomplish demodulation and decoding under different circumstances.



(a) (b)

Fig. 19. (Color online) Outdoor swimming pool for the fourth experiment: (a) photo of the pool and (b) experimental layout.

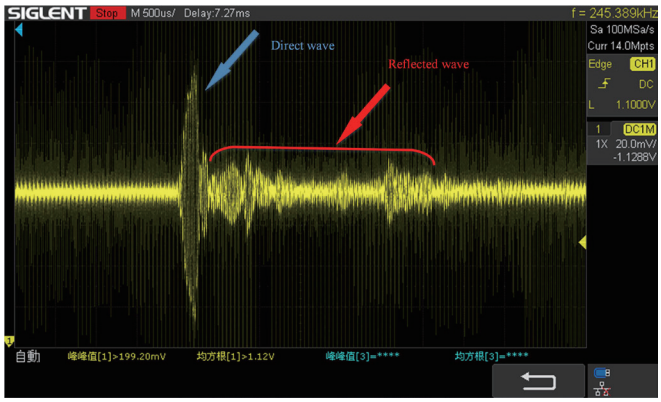
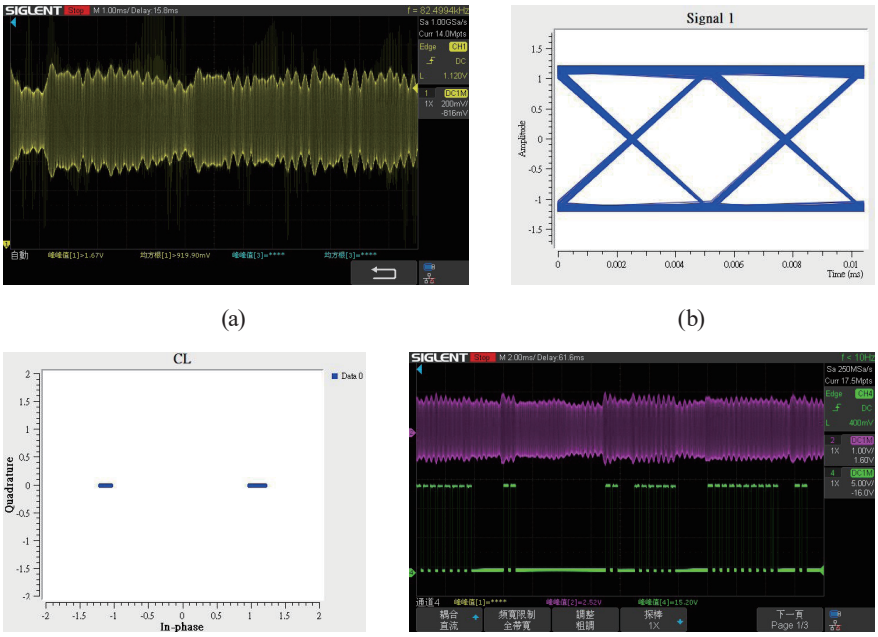


Fig. 20. (Color online) Signals received as sinusoidal waves passed through the outdoor swimming pool.

5. Results and Discussion

The underwater communication performance of the proposed system was analyzed through four experiments with different underwater channels, with the output power being set to 0.003, 0.01, and 0.048 W according to the required transmission distance. The output power for the first and second experiments was set to 0.003 W considering that multipath effects were weak according to the emitted pulse signal response, indicating that a low power level might suffice. Conversely, the output power was enhanced to 0.01 and 0.048 W in the third and fourth experiments, respectively, owing to the long-range transmission involved.

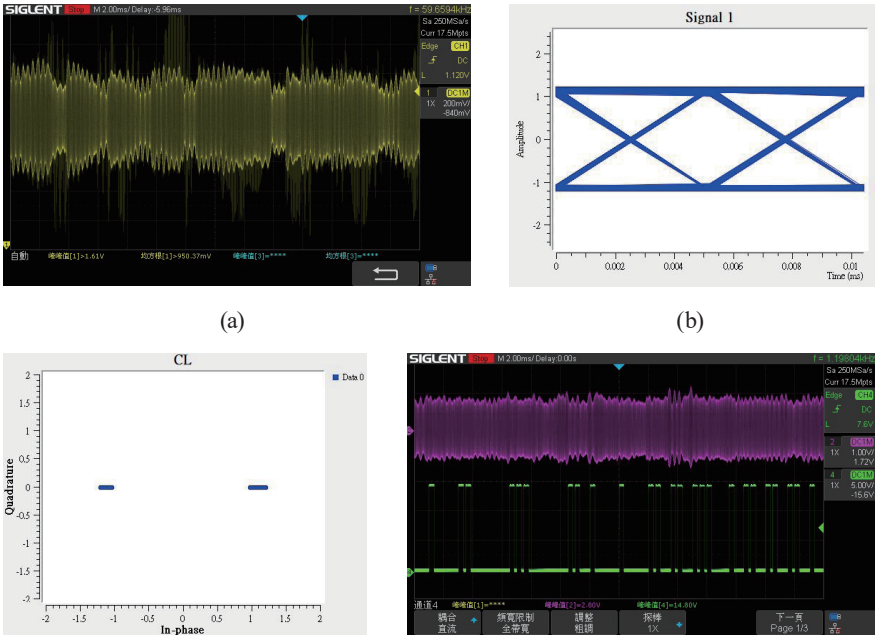
Waveforms derived from the first experiment are depicted in Figs. 21 and 22 with transmission rate being set to 1 and 2 kbps, respectively, and output power for both being 0.003 W. Specifically, the figures provide the received waveforms, eye patterns, constellation diagrams, and preamplifier-processed and hardware-decoded waveforms. Figures 23 and 24 present the aforementioned data collected during the second experiment (i.e., breeding tank scenario). Figures 25 and 26 illustrate those obtained in the third experiment (indoor swimming pool scenario), except that the output power was 0.01 W, while Figs. 27 and 28 offer those derived from the fourth experiment (outdoor swimming pool scenario) with output power being 0.048 W.



(a) (b)

(c) (d)

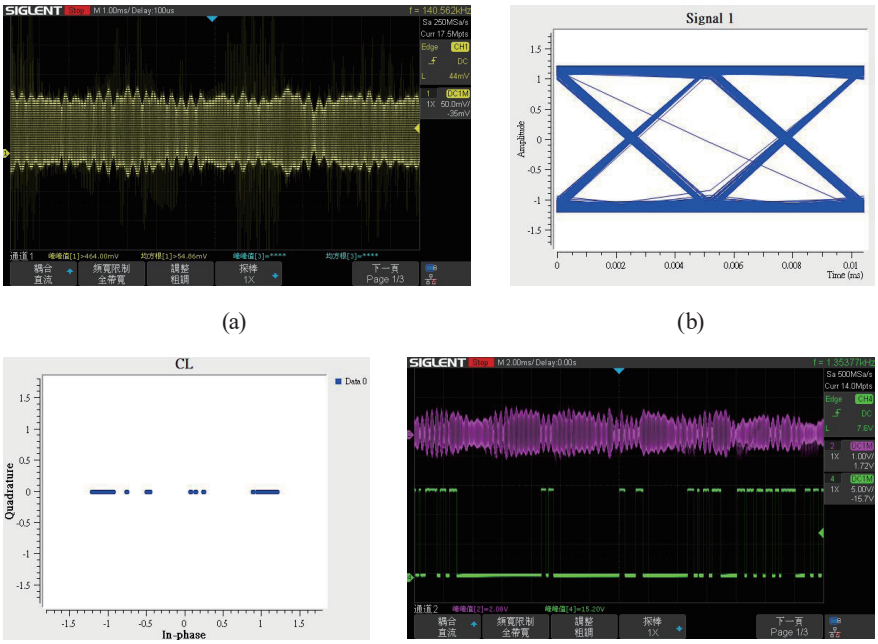
Fig. 21. (Color online) Lab sink experiment with transmission rate of 1 kbps and output power of 0.003 W: (a) waveform derived from underwater channel transmission, (b) eye pattern, (c) constellation diagram, and (d) preamplifier-processed waveform (purple lines) and hardware-decoded waveform (green lines).



(a) (b)

(c) (d)

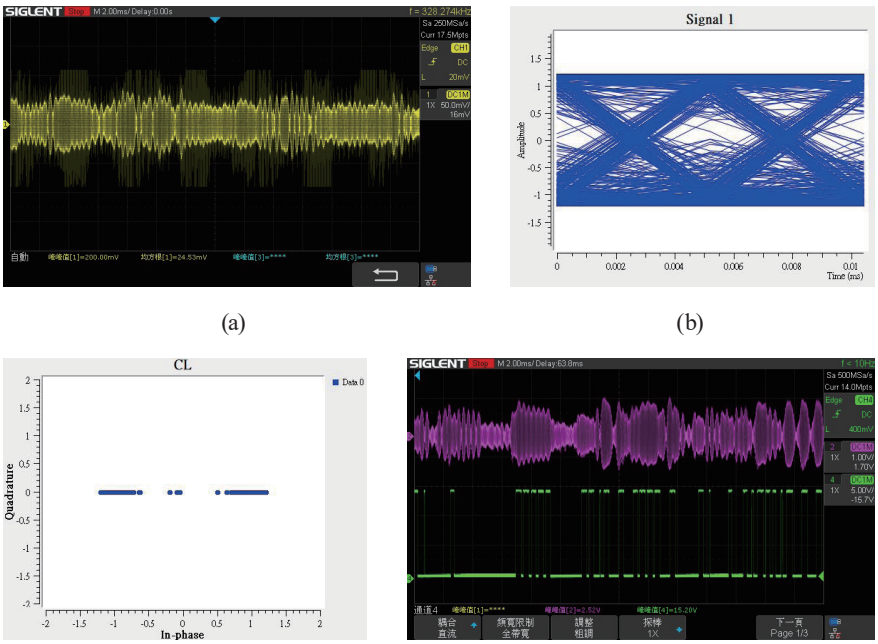
Fig. 22. (Color online) Lab sink experiment with transmission rate of 2 kbps and output power of 0.003 W: (a) waveform derived from underwater channel transmission, (b) eye pattern, (c) constellation diagram, and (d) preamplifier-processed waveform (purple lines) and hardware-decoded waveform (green lines).



(a) (b)

(c) (d)

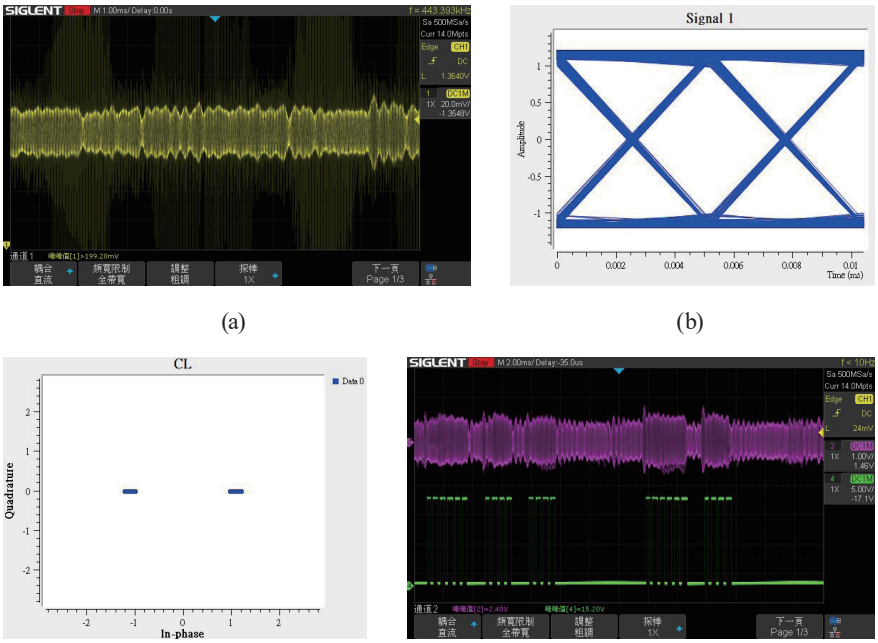
Fig. 23. (Color online) Breeding tank experiment with transmission rate of 1 kbps and output power of 0.003 W: (a) waveform derived from underwater channel transmission, (b) eye pattern, (c) constellation diagram, and (d) preamplifier-processed waveform (purple lines) and hardware-decoded waveform (green lines).



(a) (b)

(c) (d)

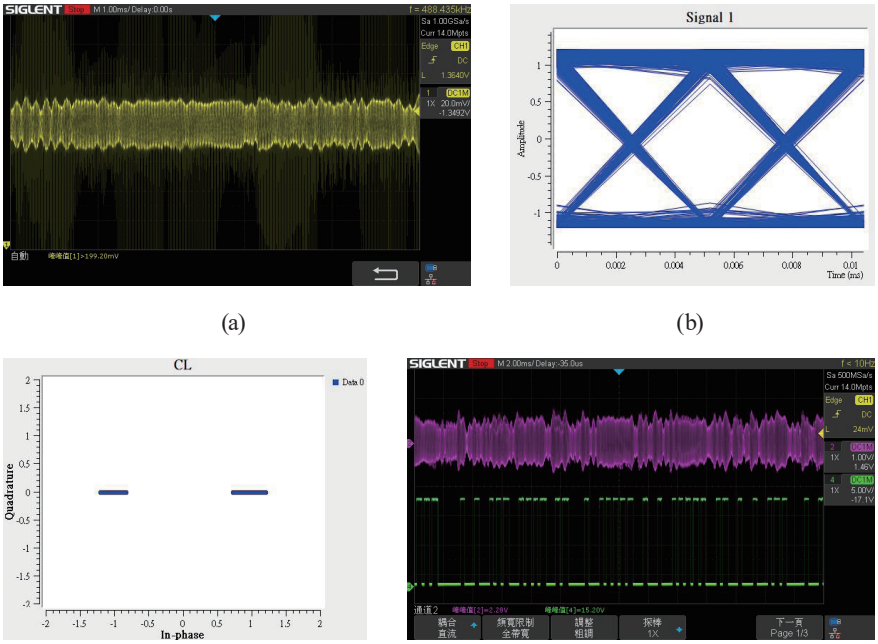
Fig. 24. (Color online) Breeding tank experiment with transmission rate of 2 kbps and output power of 0.003 W: (a) waveform derived from underwater channel transmission, (b) eye pattern, (c) constellation diagram, and (d) preamplifier-processed waveform (purple lines) and hardware-decoded waveform (green lines).



(a) (b)

(c) (d)

Fig. 25. (Color online) Indoor swimming pool experiment with transmission rate of 1 kbps and output power of 0.003 W: (a) waveform derived from underwater channel transmission, (b) eye pattern, (c) constellation diagram, and (d) preamplifier-processed waveform (purple lines) and hardware-decoded waveform (green lines).



(a) (b)

(c) (d)

Fig. 26. (Color online) Indoor swimming pool experiment with transmission rate of 2 kbps and output power of 0.003 W: (a) waveform derived from underwater channel transmission, (b) eye pattern, (c) constellation diagram, and (d) preamplifier-processed waveform (purple lines) and hardware-decoded waveform (green lines).

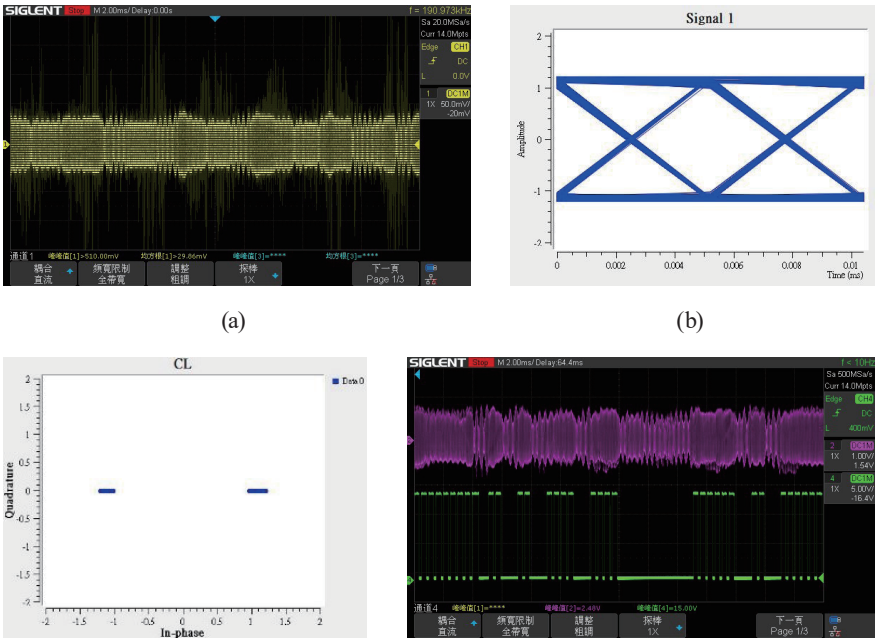


Fig. 27. (Color online) Outdoor swimming pool experiment with transmission rate of 1 kbps and output power of 0.048 W: (a) waveform derived from underwater channel transmission, (b) eye pattern, (c) constellation diagram, and (d) preamplifier-processed waveform (purple lines) and hardware-decoded waveform (green lines).

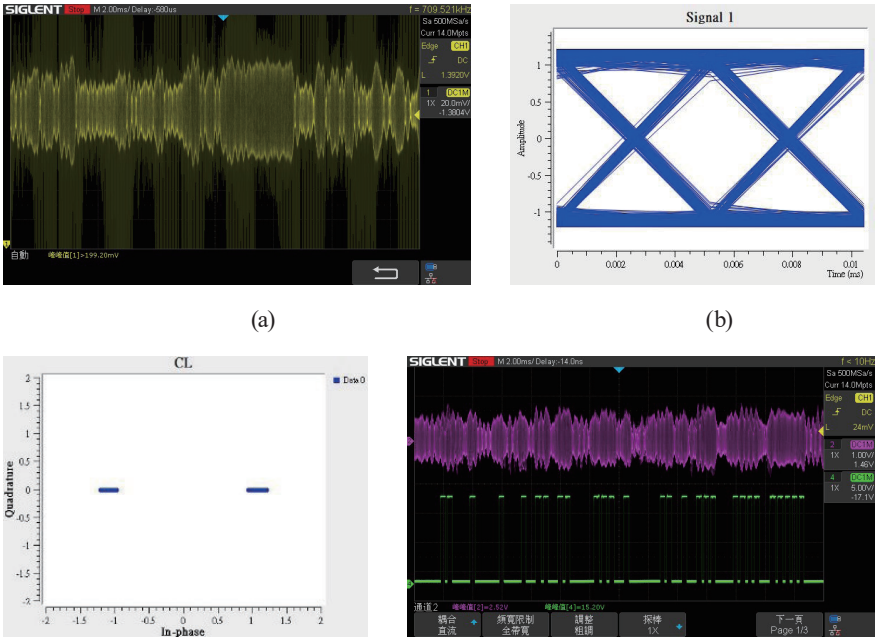


Fig. 28. (Color online) Outdoor swimming pool experiment with transmission rate of 2 kbps and output power of 0.048 W: (a) waveform derived from underwater channel transmission, (b) eye pattern, (c) constellation diagram, and (d) preamplifier-processed waveform (purple lines) and hardware-decoded waveform (green lines).

Table 3 shows the results of the four experiments. An analysis of sinusoidal pulse signals captured by the Raspberry Pi 4 during the first experiment demonstrated that multipath effects in the sink were weak, which could be considered as direct wave reception. The resultant eye pattern and constellation diagram also indicated excellent decoding performance, while waveforms were accurately obtained through hardware decoding. The BERs of software and hardware transceivers were both 0 under transmission rates of 1 and 2 kbps (output power = 0.003 W). Although the water volume was lower in the second experiment with a solid tank bottom, the transmission was shorter. Therefore, we originally expected the system performance to be similar to that in the first experiment. However, the actual results showed that fish movements affected the emitted FSK frequency and amplitude. Specifically, signals were disrupted when the fish swam through the main beam route of the transducer during transmission. The eye pattern and constellation diagram revealed an increase in error probability; the hardware-decoded waveform depicted an imbalanced amplitude of the received signals owing to interference with the underwater channel. Consequently, the BERs of software and hardware modems were higher at both transmission rates, implying a lower transmission quality than those in other experiments. Waveforms obtained during the third experiment conducted in a 20-m-long indoor swimming pool were slightly distorted owing to noise caused by the flat surface, pumps, and filters. Nonetheless, the main beam was sufficiently strong to deliver a satisfactory performance, as signals can be clearly identified in the eye pattern, constellation diagram, and hardware-decoded waveform, resulting in 0 BERs for both software and hardware modems. The results outperformed those shown in Ref. 20 using the same hardware system. In the fourth experiment implemented in a 50-m-long outdoor swimming pool, the transmission was longer than those in the other three experiments. Therefore, we enhanced the output power. However, simply increasing the output power could not help improve the transmission

Table 3
Results of the four experiments.

Experiment site	Transmission architecture	Transmission rate (kbps)	Transmission power (W)	Distance	BER
Experimental sink	Software transceiver	1	0.003	0.6 m	0
		2			0
	Hardware transceiver	1			0
		2			0
Breeding tank	Software transceiver	1	0.003	150 cm	2.3×10^{-4}
		2			2.98×10^{-4}
	Hardware transceiver	1			2.8×10^{-3}
		2			6.6×10^{-3}
Indoor swimming pool	Software transceiver	1	0.01	19.6 m	0
		2			0
	Hardware transceiver	1			0
		2			0
Outdoor swimming pool	Software transceiver	1	0.048	49.6 m	0
		2			0
	Hardware transceiver	1			0
		2			0

performance. Instead, the transmission was satisfactory mainly because the adopted directional transducers did not cause an excess of reflected beam power for signal reception on the premise of increased power of the main beam. The eye pattern and constellation diagram also demonstrated that an increase in power improved the decoding performance, thereby offsetting the distanced-caused transmission loss with the BERs of software and hardware modems being 0. The aforementioned results indicated that the proposed system would be capable of long-range communication without changing the existing specifications as long as high-power amplifiers and long-range transducers for business use are adopted.

6. Conclusion

Acoustic communication is currently the most advanced technology for underwater communication. To realize energy-efficient and cost-effective signal modulation and demodulation in water, we proposed a software/hardware modem system. The hardware modem comprises an XR2206 FSK modulator and an XR2211 FSK demodulator as signal transceivers, while the software modem was designed using GNU Radio. The system was verified through four experiments, among which the system achieved desirable signal transmission in an experimental sink, indoor swimming pool, and outdoor swimming pool, with BERs being 0 under rates of 1 and 2 kbps. However, BERs of the hardware modem were 2.8×10^{-3} at 1 kbps and 6.6×10^{-3} at 2 kbps, compared with 2.3×10^{-4} and 2.98×10^{-4} of its software counterpart in the breeding tank environment, where the movement of numerous fish disrupted underwater acoustic transmission. This experiment demonstrated that animal disturbance on acoustic transmission paths can largely affect the quality of FSK-based communication in water. Measuring BERs in different underwater environments enabled us to confirm that the designed system, featuring fault-tolerant signal processing and directional transducers, can achieve satisfactory transmission with a low error rate after its main beam intensity is increased. However, the angle settings of transceivers can change the multipath properties owing to the nature of the adopted transducers and indirectly affect the communication quality. In addition to reducing the transmission rate, we discovered that adding more signal protection systems through GNU Radio (e.g., the use of equalizers and coding techniques) helps reduce underwater noise and mitigate the multipath effect and that adopting transducers with superior field patterns and highly sensitive hydrophones can facilitate long-range signal reception. To further enhance the transmission performance in the future, exploring error control codes can be a promising avenue, for example, source information coding and channel coding techniques, among others.

Acknowledgments

We thank the National Science and Technology Council NSTC (grant nos. MOST 111-2221-E-019-074, NSTC 111-2634-F-019-001-, and NSTC 112-2221-E-019-023) and the National Taiwan Ocean University for supporting this research. We also thank the editor for the kind coordination. Moreover, we are grateful to the reviewers for constructive suggestions.

References

1 Y. Li, X. Zhang, B. Benson, and R. Kastner: Proc. 2010 IEEE Int. Conf. Sensor Networks, Ubiquitous, and Trustworthy Computing (2010 SUTC) 82–88. https://doi.org/10.1109/SUTC.2010.14
2 P. Krivić and G. Štimac: Proc. 34th Int. Convention MIPRO (2011) 1727–1731. https://ieeexplore.ieee.org/document/5967341
3 S. Indriyanto and I. Y. M. Edward: Proc. 2018 4th Int. Conf. Wireless and Telematics (2018 ICWT) 1–4. https://doi.org/10.1109/ICWT.2018.8527809
4 Y. Su, L. Dong, Z. Zhou, X. Liu, and X. Wei: IEEE Embedded Sys. Lett. **13** (2021) 3. https://doi.org/10.1109/LES.2020.3006838.
5 B. Sherlock N. Morozs, and J. Neasham: Electronics **11** (2022) 9. https://doi.org/10.3390/electronics11091446
6 W. A. P. van Kleunen, N. A. Moseley, P. J. M. Havinga, and N. Meratnia: Int. J. Distrib. Sens. Netw. **11** (2015) 10. https://doi.org/10.1155/2015/791046
7 J. DelPreto, R. Katzschmann, R. MacCurdy, and D. Rus: Proc. 10th Int. Conf. Underwater Networks & Systems (WUWNet '15) 1–7. https://doi.org/10.1145/2831296.2831337
8 G. Cario, A. Casavola, M. Lupia, and C. Rosace: SeaModem: Proc. OCEANS 2015 - Genova, Genova, Italy, (IEEE 2015) 1–6. https://doi.org/10.1109/OCEANS-Genova.2015.7271721
9 C. Renner, A. Gabrecht, B. Meyer, C. Osterloh, and E. Maehle: Quantitative Monitoring of the Underwater Environment. Ocean Eng. Oceanogr. **6** (2016) 59. https://doi.org/10.1007/978-3-319-32107-3_6
10 H.-W. Jeong, J.-E. Shin, and J.-W. Jung: Proc. 2021 Twelfth Int. Conf. Ubiquitous and Future Networks (2021 ICUFN) 141–144. https://doi.org/10.1109/ICUFN49451.2021.9528819
11 M. Y. Zia, P. Otero, and J. Poncela: Wirel. Pers. Commun. **101** (2018) 375. https://doi.org/10.1007/s11277-018-5694-5
12 Z. Zhu: Development of a Software-Defined Underwater Acoustic Communication System. M.S. thesis, Michigan Technological University, USA, 2020. https://doi.org/10.37099/mtu.dc.etdr/985
13 W. b. Abbas, N. Ahmed, C. Usama, and A. A. Syed: Ad Hoc Networks. **34** (2015) 34. https://doi.org/10.1016/j.adhoc.2014.10.007
14 E. Demirors, G. Sklivanitis, G. E. Santagati, T. Melodia, and S. N. Batalama: IEEE Access. **6** (2018) 18602. https://doi.org/10.1109/ACCESS.2018.2815026
15 T. Ahmad: Software-Defined Underwater Acoustic Modems for Arctic-Like Environments. M.S. thesis, Carleton University, Canada, 2016. https://doi.org/10.22215/etd/2016-11633
16 W. W. L. Au and M. C. Hastings: Principles of Marine Bioacoustics. Modern Acoustics and Signal Processing (Springer, New York, NY, 2008) pp. 87–120.
17 XR2206 datasheet: https://www.sparkfun.com/datasheets/Kits/XR2206_104_020808.pdf (June 1997).
18 XR2211 datasheet: https://www.maxlinear.com/Files/Documents/XR2211Av104.pdf (June 1997).
19 GNU Radio Companion - Audio modem loop back test FSK: https://aaronscher.com/GNU_Radio_Companion_Collection/Audio_Modem_loop_back_test_FSK.html (Sep. 2016).
20 S. Indriyanto, A. F. Isnawati, J. Hendry, and I. Y. M. Edward: Buletin Pos dan Telekomunikasi **18** (2020) 1. https://doi.org/10.17933/bpostel.2020.180102
21 J. Misiurewicz, K. Bruliński, W. Klembowski, and K. Kulpa: Proc. 2021 Signal Processing Symp. (2021 SPSympo) 197–201. https://doi.org/10.1109/SPSympo51155.2020.9593835

About the Authors

Chih-Ta Yen received his B.S. degree from the Department of Electrical Engineering, Tamkang University, Taiwan, in 1996, his M.S. degree from the Department of Electrical Engineering, National Taiwan Ocean University, Taiwan, in 2002, and his Ph.D. degree from the Department of Electrical Engineering, National Cheng Kung University, Taiwan, in 2008. He is currently a professor in the areas of communication, optical design, and artificial intelligence technologies at the Department of Electrical Engineering, National Formosa University, Yunlin, Taiwan. His major interests are in the areas of multiuser optical communications, machine learning applications, deep learning, image processing, and optical design. He has published over 60 journal papers in the field of electrical engineering. (chihtayen@gmail.com)

Guo-Chang Wang was born in Kaohsiung, Taiwan. He has already graduated with a master's degree at the Department of Electrical Engineering, National Formosa University, Taiwan. His major interests are in the areas of deep learning and transfer learning. (10965124@gm.nfu.edu.tw)