Underwater Diver Communication System Based on Speech/Text Conversion

Peng Chen and Yue Rong
School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Australia

***Abstract*—Underwater acoustic (UA) communication system has low data rate due to the limited bandwidth of the UA channel. This makes real-time speech communication challenging. In this paper, we present an underwater diver communication system which uses speech-to-text conversion at the transmitter to achieve speech compression. The text message is converted back to speech at the receiver end. A real-time speech-to-text and text-to-speech transceiver, utilizing Raspberry Pi, Wav2Vec 2.0, and PicoTTS is presented in the paper. The system design encompasses both the transmitting and receiving processes, with a focus on its real-time capabilities. We discuss the performance of this diver communication system, validating its efficacy through an experiment conducted in Fremantle, Western Australia.**

I. Introduction

Classical speech communication requires an 8 kHz sampling rate and 8-bit quantization, resulting in a data rate of 64 kbps. Such data rate is too high for real-time diver speech communication, due to the limited bandwidth of the underwater acoustic (UA) channel. Low-rate audio codecs can be applied to achieve speech compression. However, data rate of the state-of-the-art low-rate audio codecs such as Lyra and Opus is in the order of several thousand bits per second [1], which is still high for speech communication through the UA channel.

To solve this problem, we propose an underwater diver communication system, which uses speech-to-text conversion at the transmitter to achieve speech compression. The average speech rate is around 140 words per minute in normal conversation. In underwater diver scenarios, the speech rate could be even lower. Considering that the average word length is 5.5 characters, and ASCII (8 bit) is used for character encoding, the data rate of the speech text is around $140 \times 5.5 \times 8/60 = 103$ bps. Thus, the data rate after speech-to-text conversion is one order lower than the rate of the most advanced speech codec, making the converted text suitable for transmission through bandwidth limited UA channel.

We present a prototype diver communication system developed by our team. The core parts of the prototype are real-time offline speech-to-text and text-to-speech converters based on Wav2vec 2.0 [2] and PicoTTS [3], respectively, implemented on a Raspberry Pi board. The prototype has a small form factor, lightweight and is portable by divers. We discuss the hardware and software design at the transmitter end and the receiver end. We demonstrate the performance of this real-time system, validating its efficacy through a UA communication experiment conducted in the Success Boat Harbour in Fremantle, Western Australia. Initial trial results show promising outcomes of the proposed diver communication system. To the best of our knowledge, it is the first time such diver communication system is presented in open literature.

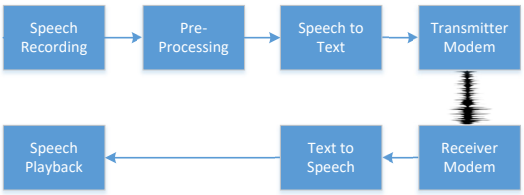

Fig. 1. System architecture.

II. System Design

The system design encompasses both the transmitting and receiving processes, with a focus on its real-time capabilities. The data transmission workflow of the system is illustrated in Fig. 1. The diver speech signal is first recorded and processed to filter out the bubble noise, breathing noise, and background noise. Then the pre-processed speech signal is passed to a speech recognition engine where the speech is converted to text. The transcription is passed to the transmitter modem as ASCII characters, which are sent through the UA channel to the receiver modem. At the receiver end, the ASCII text is converted to speech signal through a text-to-speech engine. Then, the speech signal is played back.

The flowchart of the system software is illustrated in Fig. 2. The system starts with two parallel processes, one for receiving signals while the other for transmitting signals. Note that the system has included the replay function, which is useful in case the incoming speech is not fully comprehended by the diver due to reasons such as lack of diver's attention or high background noise level. Alternatively, we can include warning sounds such as a beep to alert the diver of the incoming speech. After each receiving process, the target file for replay is updated.

The hardware configuration of the prototype system is shown in Fig. 3. The microphone and speaker from a diver mask are connected to a Raspberry Pi board, together with a push-to-talk button. This button can release two commands with short press for starting new speech and long press for playing back the last received message.

The system software is developed using Python. At the transmitter side, the Wav2vec2.0 speech-to-text conversion is adopted to transcribe speech to text. The resulting text message is transmitted under the binary terminal mode of the

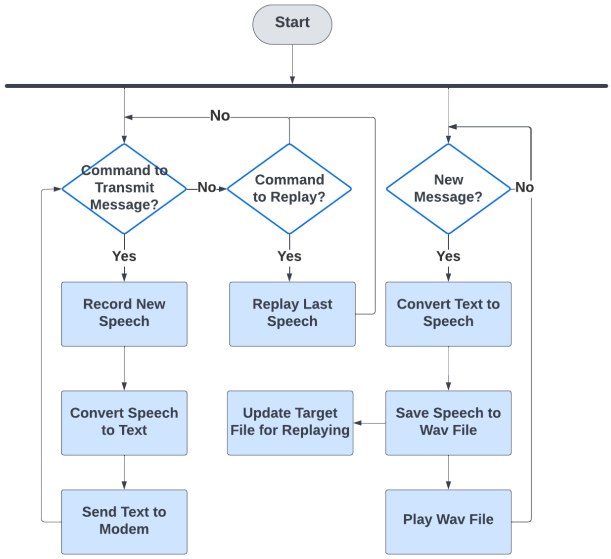

Fig. 2. System software flowchart.

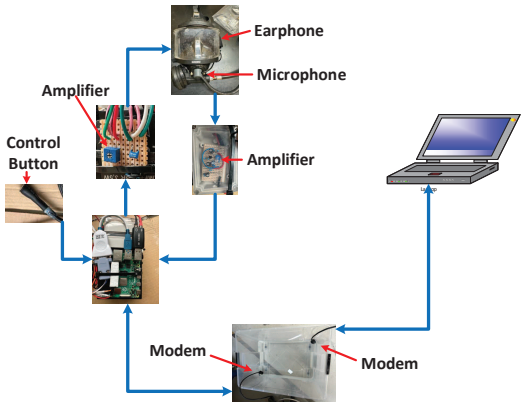

Fig. 3. Hardware configuration of the prototype system.

Delphis modem [4]. The Python pyserial library is used to communicate between the Raspberry Pi and the Delphis UA communication modem.

The receiver modem also works under the binary terminal mode and applies the Pico text-to-speech (TTS) engine to convert text to speech. Compared with other tools such as the Google text-to-speech (gTTS), PicoTTS provides a better speech quality. In addition, it can also be trained to generate personalized speech.

III. Experiment Results

An initial trial of the system was carried out in the Success Boat Harbour in Fremantle, Western Australia. During this trial, a diver talked through a microphone located in the AGA mask. The speech signal was converted to text by the Raspberry Pi board located in a watertight canister carried by the diver, and the text signal was transmitted by one Delphis modem to another Delphis modem. The received signal was converted to speech and played back to the speaker on a laptop computer located on a jetty. This setup simulates the scenario of communication between a diver and the topside.

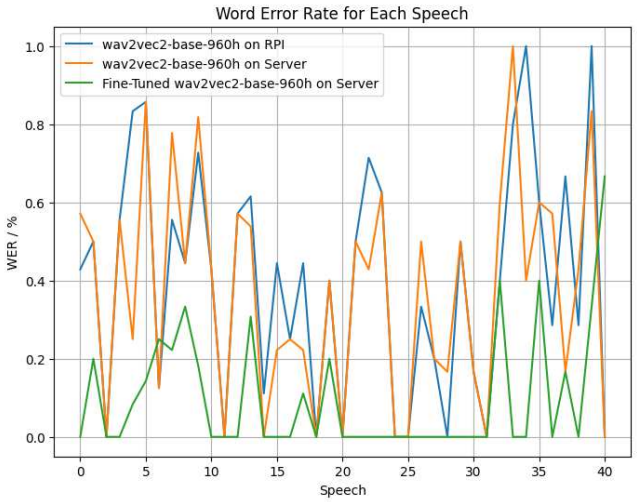

Fig. 4. WER of each speech.

In this underwater communication experiment, a total of 41 speech files were recorded. The word error rate (WER) of each speech is shown in Fig. 4. With the original Wav2ve2.0-960 hours model, the overall WER for these recordings is calculated to be 41.6%.

Additionally, the recorded speech underwent off-line evaluation using an improved Wav2vec 2.0 model, which was fine-tuned through a set of 100 sentences from the same diver. The WER for each speech using the fine-tuned Wav2vec2.0 model is shown in Fig. 4. It can be seen that through fine-tuning, the average WER is reduced to 16%.

During the trial, the laptop computer also sent speech to the diver. Noises generated by wind and boat activities were present during the trial. The experimental results indicate the successful functioning of the prototype system in a real underwater environment.

IV. Conclusions

We have presented a speech-to-text conversion-based diver underwater communication system. Initial trial results show promising outcomes of the proposed diver communication system. Future works include improving the reliability of the speech-to-text conversion algorithm and the integration of the system to diver equipment.

References

[1] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "SoundStream: An end-to-end neural audio codec," https://arxiv.org/abs/2107.03312, 2021.

[2] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," https://arxiv.org/abs/2006.11477, 2020.

[3] "Pico Text-to-Speech," [Online]. Available: https://github.com/ihuguet/picotts.

[4] "Succorfish Delphis," [Online]. Available: https://succorfish.com/products/delphis/.