

## Article

# Accurate Channel Estimation and Adaptive Underwater Acoustic Communications Based on Gaussian Likelihood and Constellation Aggregation

Liang Wang<sup>1</sup>, Peiyue Qiao<sup>2,\*</sup>, Junyan Liang<sup>2</sup>, Tong Chen<sup>3</sup>, Xinjie Wang<sup>2</sup>  and Guang Yang<sup>2,4</sup><sup>1</sup> College of Marine Technology, Ocean University of China, Qingdao 266100, China; wanglianger@ouc.edu.cn<sup>2</sup> School of Information and Control Engineering, Qingdao University of Technology, Qingdao 266525, China; ljy15290436608@163.com (J.L.); wangxinjie@qut.edu.cn (X.W.); edit231@163.com (G.Y.)<sup>3</sup> School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710129, China; ct8690@163.com<sup>4</sup> School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798, Singapore

\* Correspondence: qpy17852830692@163.com

**Abstract:** Achieving accurate channel estimation and adaptive communications with moving transceivers is challenging due to rapid changes in the underwater acoustic channels. We achieve an accurate channel estimation of fast time-varying underwater acoustic channels by using the superimposed training scheme with a powerful channel estimation algorithm and turbo equalization, where the training sequence and the symbol sequence are linearly superimposed. To realize this, we develop a ‘global’ channel estimation algorithm based on Gaussian likelihood, where the channel correlation between (among) the segments is fully exploited by using the product of the Gaussian probability-density functions of the segments, thereby realizing an ideal channel estimation of each segment. Moreover, the Gaussian-likelihood-based channel estimation is embedded in turbo equalization, where the information exchange between the equalizer and the decoder is carried out in an iterative manner to achieve an accurate channel estimation of each segment. In addition, an adaptive communication algorithm based on constellation aggregation is proposed to resist the severe fast time-varying multi-path interference and environmental noise, where the encoding rate is automatically determined for reliable underwater acoustic communications according to the constellation aggregation degree of equalization results. Field experiments with moving transceivers (the communication distance was approximately 5.5 km) were carried out in the Yellow Sea in 2021, and the experimental results verify the effectiveness of the two proposed algorithms.

**Keywords:** time-varying underwater acoustic channels; superimposed training; Gaussian likelihood; constellation aggregation; turbo equalization



**Citation:** Wang, L.; Qiao, P.; Liang, J.; Chen, T.; Wang, X.; Yang, G. Accurate Channel Estimation and Adaptive Underwater Acoustic Communications Based on Gaussian Likelihood and Constellation Aggregation. *Sensors* **2022**, *22*, 2142. <https://doi.org/10.3390/s22062142>

Academic Editor: Juan V. Capella

Received: 11 January 2022

Accepted: 7 March 2022

Published: 10 March 2022

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



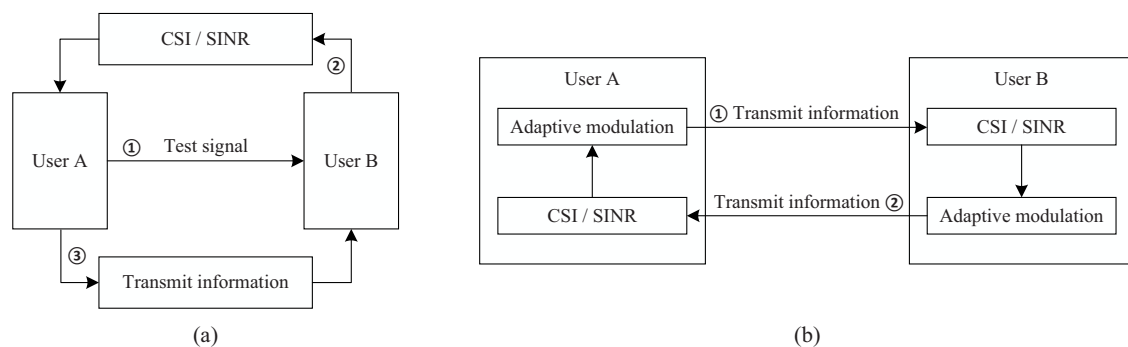
**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Underwater acoustic communication technology can be widely applied in many fields, such as marine pollution monitoring, underwater rescue, underwater autonomous underwater vehicle (AUV) positioning and navigation. However, underwater acoustic channels are characterized by a time-varying multipath. In particular, when there is relative motion between the transceivers, the channel will change rapidly, resulting in fast time-varying multi-path interference, which distorts the received signal waveform and leads to a reduction in or even failure of the decoding performance of the underwater acoustic communication system [1–3].

To solve the issues of time-varying underwater acoustic channels and environmental noise, an adaptive communication scheme was proposed, where the transmitter automatically selected an appropriate modulation according to the instantaneous channel state

information (CSI) and signal noise ratio (SNR). The adaptive communication scheme can be mainly divided into two categories, including feedback adaptive communications and direct adaptive communications, as shown in Figure 1a,b, respectively. For feedback adaptive communications, as shown in Figure 1a, User A sends a test signal to User B. User B estimates CSI and SNR based on the test signal, and then feeds them back to User A. User A selects a modulation according to the feedback CSI and SNR, and then transmits the data information to User B by using the selected modulation [4,5]. For direct adaptive communications in Figure 1b, User A initially selects a modulation, such as the direct sequence spread spectrum (DSSS), then transmits the data information to User B by using DSSS. User B identifies the modulation (i.e., identifies DSSS), demodulates and decodes, and estimates CSI and SNR, such as a simple channel and SNR = 20 dB. According to the estimated CSI and SNR, User B selects a new modulation, such as orthogonal frequency division multiplexing (OFDM), and then feeds data information back to User A by using OFDM. Similarly, User A identifies, demodulates and decodes, and estimates CSI and SNR. Then, according to the estimated CSI and SNR, User A selects an original or new modulation, and then transmits the data information to User B by using the selected modulation [6,7]. The biggest difference in the two adaptive communications is that there is no need to send a test signal for the second scheme. Therefore, for a specified amount of data information, the second scheme saves communication time, thereby reducing or even avoiding time-variation of the channel during communications. Therefore, the second scheme is more suitable for fast time-varying channels incurred by underwater acoustic communications with moving transceivers than the first scheme.



**Figure 1.** (a) Feedback adaptive communications; (b) direct adaptive communications.

For adaptive communications, there are mainly four selected modulations and demodulations: multiple frequency shift keying (MFSK), spread spectrum, orthogonal frequency division multiplexing (OFDM) and single carrier. The transmission rate of MFSK is low; spread spectrum technology always uses high-order spread spectrum code, which has a low communication efficiency [8]; OFDM has a poor anti-frequency-offset performance [9–16]. Therefore, with a high transmission rate and good anti-frequency-offset characteristics [17–19], the single carrier technology is adopted in this paper. It can be used with a variety of encoding rates to realize adaptive underwater acoustic communications with moving transceivers.

Channel estimation is one of the key factors to realize reliable adaptive communications. At present, there are mainly three kinds of underwater acoustic channel estimation algorithms, such as channel estimation algorithms based on reference signal, blind estimation algorithms and semi-blind estimation algorithms [20–28]. Among the three kinds of algorithms, the channel estimation capability and channel tracking capability based on the reference signal are the strongest. Much research on them has been conducted by some teams, such as the team of the University of Connecticut, the team of the Massachusetts Institute of Technology, the team of Institute of Acoustics, Chinese Academy of Sciences, and the team of Harbin Engineering University. So far, all of the above channel estimation

algorithms based on reference signals have adopted the traditional time-multiplexing training sequence scheme. In order to further improve the tracking capability of time-varying channels, the joint team of Qingdao University of Technology, the University of Wollongong and the University of Western Australia [29] proposed a superimposed training scheme for underwater acoustic communications, where the training sequence and the symbol sequence are linearly superimposed in order to make the channel information of the training sequence and the symbol sequence completely consistent.

Same as literatures [29,30], the superimposed training (ST) scheme and the segment strategy are used in this paper to enhance the estimation and tracking capability of fast time-varying channels. To realize the full potential of the ST scheme and the segment strategy, a channel estimation algorithm based on Gaussian likelihood (GL) is proposed. The product of the Gaussian probability-density functions of the segments is still the Gaussian probability-density function, which can be parameterized by the mean and the variance, where the mean is the channel estimate and the variance is the deviation of the channel estimate. The variance of the Gaussian probability-density function after the product is less than the variance of the Gaussian probability-density function of each segment, which means that the estimated channel for the segment after the product is more accurate than the estimated channel for the segment itself. This is equivalent to estimating the channel information of the segment by using the 'whole' data block [29,30], thereby leading to the ideal channel estimation of the segment.

It is important to note that the proposed GL algorithm can achieve the same channel estimation and tracking performance as literatures [29,30] in a 'novel' Gaussian product way, because it can be seen as a message-passing method in the Gaussian scenario [29,30]. The message-passing thought was first proposed in literature [30] to improve the channel estimation capability; then, it was applied in underwater acoustic communications with a communication distance of approximately 1 km [29]. Different from literatures [29,30], in this paper, the same thought as message passing [29,30] is realized in a 'novel' Gaussian product way; in particular, the proposed algorithm is applied in actual underwater acoustic communication machines, and the effective communication distance is extended from 1 km to 5.5 km.

In addition, an adaptive communication algorithm based on constellation aggregation (CA) is proposed. The encoding rate (such as rate-1/2, rate-1/4, rate-1/8, or rate-1/16) is automatically selected based on the aggregation degree of the constellation points after linear minimum mean square error (LMMSE) equalization. The working principles of the proposed direct adaptive communications and the traditional direct adaptive communications are different. The traditional direct adaptive communications select the modulation based on CSI and SNR. However, the proposed direct adaptive communications select the encoding rate based on the constellation aggregation. The proposed algorithm based on the constellation aggregation is more accurate in making a selection than the traditional algorithm based on CSI and SNR. In order to fully realize the potential of the GL algorithm and the CA algorithm, the single-carrier communication system and turbo equalization are adopted. The channel estimator (GL), constellation aggregation decision maker (CA), equalizer and decoder are combined together, and they are performed jointly in an iterative manner (turbo equalization) to realize an accurate estimation of fast time-varying channels and reliable communications by using the information exchange between the equalizer and the decoder (turbo equalization). Field experiments with moving transceivers (the communication distance was approximately 5.5 km) were carried out in the Yellow Sea in 2021 to verify the effectiveness of the proposed algorithms. The major contributions of this paper are summarized as follows:

- (1) A channel estimation algorithm, named GL, is proposed, realizing the same performance as the bidirectional channel estimation algorithm [29,30] in a novel product way of probability-density functions;
- (2) An adaptive communication algorithm based on constellation aggregation is proposed to improve the applicability of the system for different environments;

- (3) GL-based channel estimation, LMMSE equalization and decoding are iteratively performed (turbo equalization), leading to a significant performance improvement of the whole system;
- (4) The proposed algorithms are applied in actual underwater acoustic communication machines to verify their effectiveness.

The remainder of the paper is organized as follows. The system structure is provided in Section 2. Then, a channel estimation algorithm based on Gaussian likelihood and an adaptive communication algorithm based on constellation aggregation are shown in Section 3. Simulations, experiments and the conclusion are presented in Sections 4–6, respectively. Throughout the paper, superscripts  $[\cdot]^{Tr}$  and  $[\cdot]^H$  represent transpose and conjugate transpose, respectively.

## 2. System Structure

The system structure of underwater acoustic communications is shown in Figure 2. At the transmitter, the information bit sequence is encoded, interleaved and mapped by using quadrature phase shift keying (QPSK) to symbols. The training sequence and the symbol sequence are linearly superimposed, the resultant sequence is partitioned into multiple segments and then each segment is appended a cyclic prefix (CP) to avoid inter-segment interference and to facilitate low-complexity equalization. The in-phase quadrature (IQ) modulation is used for each CP plus segment. Hyperbolic frequency modulation (HFM) signals with negative and positive modulation rates are used as the head and the tail of the signal frame, respectively. Then, the resultant signals are transmitted by a transducer.

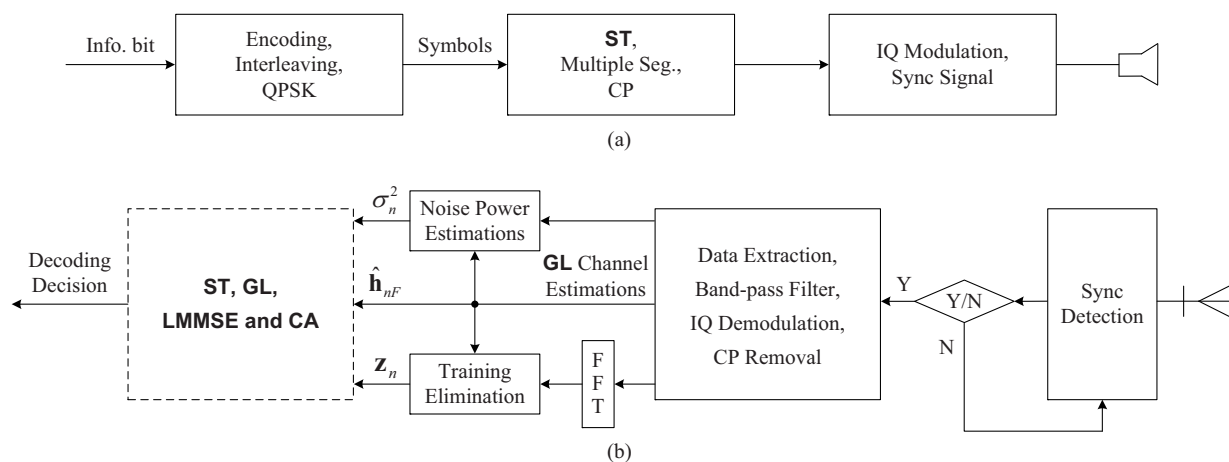


Figure 2. System structure. (a) Transmitter; (b) receiver.

The HFM signals are used to estimate and eliminate the average frequency offset and to synchronize the received signals [31]. Then, the transmitted signals are extracted, band-pass filtering and IQ demodulation are carried out and CPs are removed. With the resultant signals, we estimate initial channels  $\hat{h}_{nF}$  of all segments based on the GL algorithm and noise powers  $\hat{\sigma}_n$ , and obtain a ‘clean’ signal  $z_n$  after training elimination for data equalization. Then, LMMSE equalization, CA decision and decoding are carried out based on  $\hat{h}_{nF}$ ,  $\hat{\sigma}_n$  and  $z_n$ , as shown in Figure 3. They, on both sides of equalization, represent the same things. Please note that, on the right side, they represent the initial values, and, on the left side, they represent the iterative values. The iterative process proceeds until a pre-set number of iterations is reached, and difficult decisions are made on each information bit in the last iteration.



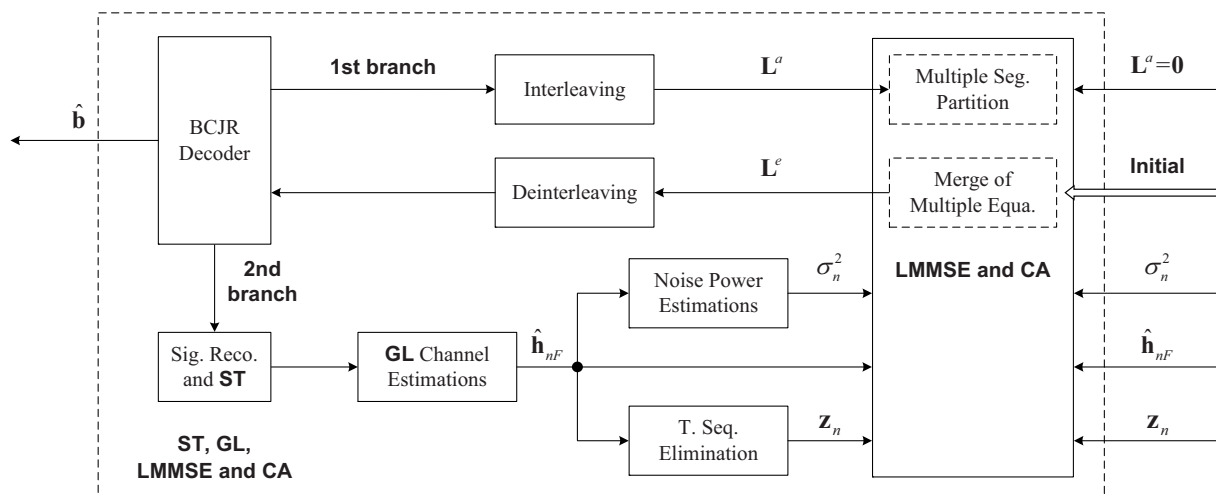


Figure 3. Turbo equalization.

An illustration of the CA decision is shown in Figure 4. We can ensure the return encoding rate by comparing the pre-set values  $\zeta_{in}$  and  $\zeta_{ex}$ . When the constellation aggregation degree  $\zeta_m < \zeta_{in}$ , the return encoding rate is increased. When the constellation aggregation degree  $\zeta_m > \zeta_{ex}$ , the return encoding rate is reduced. When the constellation aggregation degree  $\zeta_{in} \leq \zeta_m \leq \zeta_{ex}$ , the return encoding rate is kept the same.

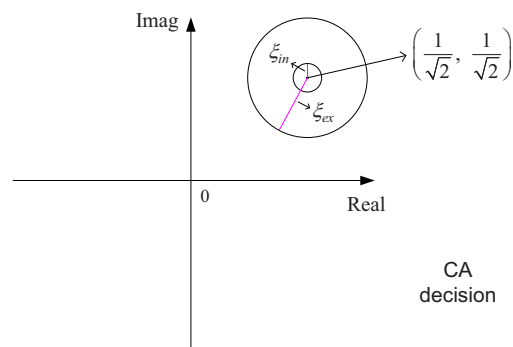


Figure 4. Constellation aggregation decision.

The turbo equalization is shown in Figure 3. Based on  $\hat{h}_{nF}$ ,  $\hat{p}_n$  and  $z_n$ , the LMMSE equalization and decoding for each segment are carried out, where the LMMSE equalization can be efficiently implemented with fast Fourier transform (FFT), where the initial a priori logarithm likelihood ratios (LLRs) of the interleaved encoded bits are set to zeros, i.e.,  $L^a = \mathbf{0}$ . The soft detection outputs for multiple segments are collected to make up extrinsic LLRs  $L^e$ , and then deinterleaving and decoding are carried out. The output of the decoder is used by the equalizer and the channel estimator, so there are two branches from the decoder. Both branches use the latest decoding results, i.e., the LLRs of encoded bits of the decoder, and they are updated in each iteration. In the first branch, the LLRs of encoded bits are interleaved and input to the equalizer. In the second branch, difficult decisions on the encoded bits are executed, followed by interleaving and QPSK mapping to obtain the (estimated) symbol sequence. They, together with the training sequence, are used for accurate channel (re)estimation. After that, based on  $L^a$  from the first branch and  $\hat{h}_{nF}$ ,  $\hat{p}_n$  and  $z_n$  from the second branch, LMMSE equalization is performed to obtain  $L^e$ , which is input into the decoder for the next round of iteration (turbo equalization).

### 3. Accurate Channel Estimation and Adaptive Communications

A block of an information bit sequence denoted by  $\mathbf{b} = [b_1, \dots, b_{L_b}]^{Tr}$  is encoded and interleaved, yielding an interleaved coded bit sequence denoted by  $\mathbf{c} = [\mathbf{c}_1, \dots, \mathbf{c}_{L_i}]^{Tr}$ , where  $\mathbf{c}_i = [c_i^1, c_i^2]^{Tr}$ . Then,  $\mathbf{c}$  is mapped into a symbol sequence denoted by  $\mathbf{f}_{L_f} = [f_1, \dots, f_{L_f}]^{Tr}$ , where each  $f_i$  corresponds to a  $\mathbf{c}_i$ . Denote the periodic training sequence as  $\mathbf{t}_{L_f} = [t_1, \dots, t_{L_f}]^{Tr}$  with a period of  $T$ , where  $t_k = e^{j\frac{\pi}{T}(k-1)^2}$ ,  $k = 1, \dots, T$  [32]. The training sequence and the symbol sequence are linearly superimposed with a power ratio  $r$ , yielding the transmitted signal  $\mathbf{s}$  with length  $L_f$ , where  $L_f$  is an integer multiple of  $T$ .

Divide  $\mathbf{s}$  into  $N_y$  segments, i.e.,  $\mathbf{s} = [\mathbf{s}_1, \dots, \mathbf{s}_{N_y}]^{Tr}$ , and the length of each segment is  $L_s$ , where  $L_f = N_y \times L_s$ . Taking  $\mathbf{s}_n$  as an example, the corresponding symbol sequence is  $\mathbf{f}_n$ , and the corresponding training sequence is  $\mathbf{t}_{L_s}$ . CP is added to each segment, yielding the channel circulant matrix denoted by  $\mathbf{H}_n$ . Denote the white Gaussian noise as  $\mathbf{w}$ .

Denote a segment of the received signal after CP removal as  $\mathbf{y}_n$ , and its length is an integer multiple of  $T$ , i.e.,  $L_s = pT$ . Then, we can represent  $\mathbf{y}_n$  as  $\mathbf{y}_n = [\mathbf{y}_{1T}, \dots, \mathbf{y}_{pT}]^{Tr}$ . The received signal  $\mathbf{y}_n$  can be written as

$$\begin{aligned} \mathbf{y}_n &= \mathbf{H}_n \mathbf{s}_n + \mathbf{w} = \mathbf{H}_n (r\mathbf{t}_{L_s} + \mathbf{f}_n) + \mathbf{w} \\ &= \mathbf{H}_n \mathbf{f}_n + r\mathbf{H}_n \mathbf{t}_{L_s} + \mathbf{w} \end{aligned} \tag{1}$$

Define  $L_c$  as the channel order, where  $T \geq L_c$ ; then, the Toeplitz matrix formed by the training sequence can be represented as

$$\mathbf{A} = \begin{bmatrix} t_1 & t_T & \cdots & t_{T-L_c+2} \\ t_2 & t_1 & \cdots & t_{T-L_c+3} \\ \vdots & \vdots & \ddots & \vdots \\ t_T & t_{T-1} & \cdots & t_{T-L_c+1} \end{bmatrix}_{T \times L_c} \tag{2}$$

From Appendix A, based on the least squares (LS) algorithm, the channel estimate of a segment can be computed as

$$\hat{\mathbf{h}}_n = \left[ (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \left( \frac{1}{p} \sum_{i=1}^p \mathbf{y}_{iT} \right) \right]_{L_c \times 1} \tag{3}$$

#### 3.1. Accurate Channel Estimation Based on Gaussian Likelihood

Channel estimates of two consecutive segments can be expressed as two independent and identically distributed probability-density functions in the Gaussian scenario. Denote two independent and identically distributed probability-density functions as  $p_{n-1}(x)$  and  $p_n(x)$ . Denote  $\mu_{\hat{\mathbf{h}}_{n-1}}$  and  $\sigma_{\hat{\mathbf{h}}_{n-1}}^2$  as the mean value and variance of the channel estimate  $\hat{\mathbf{h}}_{n-1}$  of the  $(n-1)$ -th segment, respectively, and denote  $\mu_{\hat{\mathbf{h}}_n}$  and  $\sigma_{\hat{\mathbf{h}}_n}^2$  as the mean value and variance of the channel estimate  $\hat{\mathbf{h}}_n$  of the  $n$ -th segment, respectively. Then, we can obtain

$$\begin{cases} p_{n-1}(x) = \frac{1}{\sqrt{2\pi\sigma_{\hat{\mathbf{h}}_{n-1}}^2}} e^{-\frac{(x-\mu_{\hat{\mathbf{h}}_{n-1}})^2}{2\sigma_{\hat{\mathbf{h}}_{n-1}}^2}} \\ p_n(x) = \frac{1}{\sqrt{2\pi\sigma_{\hat{\mathbf{h}}_n}^2}} e^{-\frac{(x-\mu_{\hat{\mathbf{h}}_n})^2}{2\sigma_{\hat{\mathbf{h}}_n}^2}} \end{cases} \tag{4}$$

Denote  $\hat{\mathbf{h}}_{nF}$  as the channel estimate after information fusion of the channel estimate  $\hat{\mathbf{h}}_{n-1}$  of the  $(n-1)$ -th segment and the channel estimate  $\hat{\mathbf{h}}_n$  of the  $n$ -th segment. Denote  $\mu_{\hat{\mathbf{h}}_{nF}}$

and  $\sigma_{\hat{\mathbf{h}}_{nF}}^2$  as the mean value and variance of the channel estimate  $\hat{\mathbf{h}}_{nF}$  after information fusion, respectively. Then, the product of the two probability-density functions can be expressed as

$$\begin{aligned}
 p_{n-1}(x)p_n(x) &= \frac{1}{2\pi\sigma_{\hat{\mathbf{h}}_n}\sigma_{\hat{\mathbf{h}}_{n-1}}} e^{-\left[\frac{(x-\mu_{\hat{\mathbf{h}}_n})^2}{2\sigma_{\hat{\mathbf{h}}_n}^2} + \frac{(x-\mu_{\hat{\mathbf{h}}_{n-1}})^2}{2\sigma_{\hat{\mathbf{h}}_{n-1}}^2}\right]} \\
 &= \left(\frac{1}{\sqrt{2\pi(\sigma_{\hat{\mathbf{h}}_n}^2 + \sigma_{\hat{\mathbf{h}}_{n-1}}^2)}} e^{-\frac{(\mu_{\hat{\mathbf{h}}_n} - \mu_{\hat{\mathbf{h}}_{n-1}})^2}{2(\sigma_{\hat{\mathbf{h}}_n}^2 + \sigma_{\hat{\mathbf{h}}_{n-1}}^2)}}\right) \frac{1}{\sqrt{2\pi\sigma_{\hat{\mathbf{h}}_{nF}}^2}} e^{-\frac{(x-\mu_{\hat{\mathbf{h}}_{nF}})^2}{2\sigma_{\hat{\mathbf{h}}_{nF}}^2}} \quad (5) \\
 &= C_A \frac{1}{\sqrt{2\pi\sigma_{\hat{\mathbf{h}}_{nF}}^2}} e^{-\frac{(x-\mu_{\hat{\mathbf{h}}_{nF}})^2}{2\sigma_{\hat{\mathbf{h}}_{nF}}^2}},
 \end{aligned}$$

where

$$\mu_{\hat{\mathbf{h}}_{nF}} = \frac{\mu_{\hat{\mathbf{h}}_n}\sigma_{\hat{\mathbf{h}}_{n-1}}^2 + \mu_{\hat{\mathbf{h}}_{n-1}}\sigma_{\hat{\mathbf{h}}_n}^2}{\sigma_{\hat{\mathbf{h}}_n}^2 + \sigma_{\hat{\mathbf{h}}_{n-1}}^2}, \quad (6)$$

and

$$\sigma_{\hat{\mathbf{h}}_{nF}}^2 = \frac{\sigma_{\hat{\mathbf{h}}_n}^2\sigma_{\hat{\mathbf{h}}_{n-1}}^2}{\sigma_{\hat{\mathbf{h}}_n}^2 + \sigma_{\hat{\mathbf{h}}_{n-1}}^2}. \quad (7)$$

It is important to note that

$$\begin{cases} \sigma_{\hat{\mathbf{h}}_{nF}}^2 - \sigma_{\hat{\mathbf{h}}_{n-1}}^2 = \frac{\sigma_{\hat{\mathbf{h}}_n}^2\sigma_{\hat{\mathbf{h}}_{n-1}}^2}{\sigma_{\hat{\mathbf{h}}_n}^2 + \sigma_{\hat{\mathbf{h}}_{n-1}}^2} - \sigma_{\hat{\mathbf{h}}_{n-1}}^2 = \frac{-\sigma_{\hat{\mathbf{h}}_{n-1}}^4}{\sigma_{\hat{\mathbf{h}}_n}^2 + \sigma_{\hat{\mathbf{h}}_{n-1}}^2} < 0 \\ \sigma_{\hat{\mathbf{h}}_{nF}}^2 - \sigma_{\hat{\mathbf{h}}_n}^2 = \frac{\sigma_{\hat{\mathbf{h}}_n}^2\sigma_{\hat{\mathbf{h}}_{n-1}}^2}{\sigma_{\hat{\mathbf{h}}_n}^2 + \sigma_{\hat{\mathbf{h}}_{n-1}}^2} - \sigma_{\hat{\mathbf{h}}_n}^2 = \frac{-\sigma_{\hat{\mathbf{h}}_n}^4}{\sigma_{\hat{\mathbf{h}}_n}^2 + \sigma_{\hat{\mathbf{h}}_{n-1}}^2} < 0 \end{cases}, \quad (8)$$

which means that the variance  $\sigma_{\hat{\mathbf{h}}_{nF}}^2$  after the product becomes smaller than  $\sigma_{\hat{\mathbf{h}}_{n-1}}^2$  and  $\sigma_{\hat{\mathbf{h}}_n}^2$ , i.e., the fused channel estimate  $\hat{\mathbf{h}}_{nF}$  becomes more accurate, which is more close to the real channel than  $\hat{\mathbf{h}}_{n-1}$  and  $\hat{\mathbf{h}}_n$ .  $C_A$  is the scale factor of the Gaussian distribution, and it is not a variable, which can be normalized. Therefore, we can obtain the Gaussian distribution  $p_{nF}(x)$  after the product, i.e.,

$$p_{n-1}(x)p_n(x) = p_{nF}(x) \sim N(\mu_{\hat{\mathbf{h}}_{nF}}, \sigma_{\hat{\mathbf{h}}_{nF}}^2), \quad (9)$$

where  $N()$  represents Gaussian distribution. From (7), we can acquire

$$\frac{1}{\sigma_{\hat{\mathbf{h}}_{nF}}^2} = \frac{\sigma_{\hat{\mathbf{h}}_n}^2 + \sigma_{\hat{\mathbf{h}}_{n-1}}^2}{\sigma_{\hat{\mathbf{h}}_n}^2\sigma_{\hat{\mathbf{h}}_{n-1}}^2} = \frac{1}{\sigma_{\hat{\mathbf{h}}_{n-1}}^2} + \frac{1}{\sigma_{\hat{\mathbf{h}}_n}^2}. \quad (10)$$

From (6) and (7), i.e.,  $\mu_{\hat{\mathbf{h}}_{nF}}$  is divided by  $\sigma_{\hat{\mathbf{h}}_{nF}}^2$ , we can obtain

$$\frac{\mu_{\hat{\mathbf{h}}_{nF}}}{\sigma_{\hat{\mathbf{h}}_{nF}}^2} = \frac{\mu_{\hat{\mathbf{h}}_n} \sigma_{\hat{\mathbf{h}}_{n-1}}^2 + \mu_{\hat{\mathbf{h}}_{n-1}} \sigma_{\hat{\mathbf{h}}_n}^2}{\sigma_{\hat{\mathbf{h}}_{n-1}}^2 \sigma_{\hat{\mathbf{h}}_n}^2} = \frac{\mu_{\hat{\mathbf{h}}_{n-1}}}{\sigma_{\hat{\mathbf{h}}_{n-1}}^2} + \frac{\mu_{\hat{\mathbf{h}}_n}}{\sigma_{\hat{\mathbf{h}}_n}^2}, \tag{11}$$

i.e.,

$$\mu_{\hat{\mathbf{h}}_{nF}} = \sigma_{\hat{\mathbf{h}}_{nF}}^2 \left( \frac{\mu_{\hat{\mathbf{h}}_{n-1}}}{\sigma_{\hat{\mathbf{h}}_{n-1}}^2} + \frac{\mu_{\hat{\mathbf{h}}_n}}{\sigma_{\hat{\mathbf{h}}_n}^2} \right). \tag{12}$$

The message fusion Formulas (10) and (12) are equivalent to the message fusion Formulas (18) and (19) of literature [29], i.e., the proposed GL algorithm using a ‘novel’ Gaussian product can achieve the same performance as the bidirectional channel estimation algorithm in literature [29]. They have the same computational complexity.

The formulas of the forward passing and the backward passing are as follows [33]:

$$\begin{cases} \hat{\mathbf{h}}_n = \alpha_p \hat{\mathbf{h}}_{n-1} + n_p \\ \sigma_{\hat{\mathbf{h}}_n}^2 = \alpha_p^2 \sigma_{\hat{\mathbf{h}}_{n-1}}^2 + \beta \mathbf{I} \end{cases}, \tag{13}$$

and

$$\begin{cases} \hat{\mathbf{h}}_{n-1} = \alpha_p^{-1} (\hat{\mathbf{h}}_n + n_p) \\ \sigma_{\hat{\mathbf{h}}_{n-1}}^2 = \alpha_p^{-2} (\sigma_{\hat{\mathbf{h}}_n}^2 + \beta \mathbf{I}), \end{cases} \tag{14}$$

where  $\alpha_p$  is the channel correlation coefficient of the consecutive segments,  $n_p$  is Gaussian white noise (the mean is 0) and  $\beta$  is the noise power.

Take the  $n$ -th segment as an example to show the flow of the ‘global’ channel estimation, and the flow diagram is shown in Figure 5. For forward message passing, the local channel estimation  $\hat{\mathbf{h}}_1$  of the first segment is fused with the local channel estimation  $\hat{\mathbf{h}}_2$  of the second segment to obtain a fused channel estimation  $\hat{\mathbf{h}}_{2f}$  by using (10) and (12). Then, the message update  $\hat{\mathbf{h}}_{2a}$  can be obtained by using (13) until the fused channel estimation  $\hat{\mathbf{h}}_{nf}$  is acquired. For backward message passing, the local channel estimation  $\hat{\mathbf{h}}_{N_y}$  of the last segment is fused with the local channel estimation  $\hat{\mathbf{h}}_{N_y-1}$  of the  $(N_y-1)$ -th segment to obtain a fused channel estimation  $\hat{\mathbf{h}}_{(N_y-1)f}$  by using (10) and (12). Then, the message update  $\hat{\mathbf{h}}_{(N_y-1)b}$  can be obtained by using (14) until the fused channel estimation  $\hat{\mathbf{h}}_{(n+1)b}$  is acquired. Finally,  $\hat{\mathbf{h}}_{nf}$  and  $\hat{\mathbf{h}}_{(n+1)b}$  can be fused to obtain a ‘global’ channel estimation  $\hat{\mathbf{h}}_{nF}$  of the  $n$ -th segment. Appending a proper number of zeros to  $\hat{\mathbf{h}}_{nF}$  to form a length- $L_s$  vector, i.e.,  $\hat{\mathbf{h}}_{nF} = [\hat{\mathbf{h}}_{nF}, \mathbf{0}]_{L_s \times 1}$ .

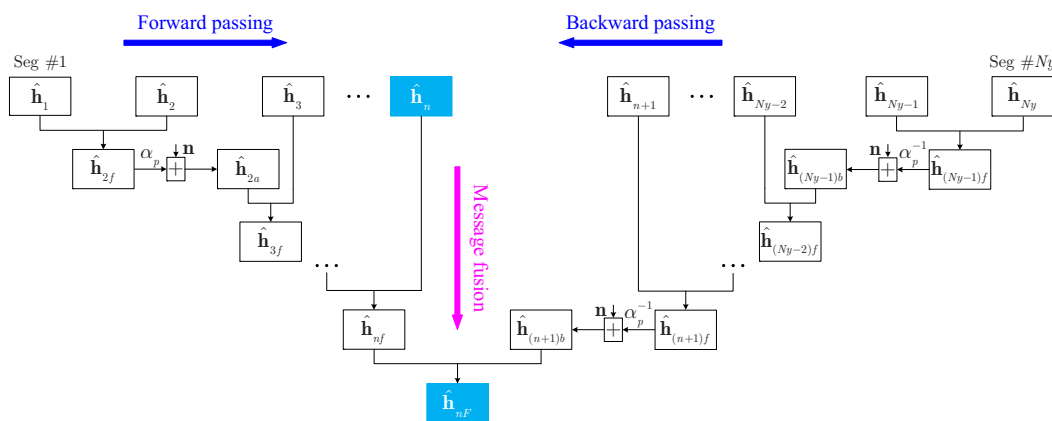


Figure 5. Accurate channel estimation of the  $n$ -th segment.

### 3.2. Training Interference Elimination, Estimation of Noise Power and Turbo Equalization

We use  $\mathbf{F}$  to denote a normalized discrete Fourier transform (DFT) matrix, i.e., the  $(m, n)$ th element is given as  $L_s^{-1/2} e^{-j2\pi mn/L_s}$  with  $j = \sqrt{-1}$ . Take the  $n$ -th segment as an example. The circulant matrix  $\mathbf{H}_n$  can be diagonalized by a DFT matrix, i.e.,  $\mathbf{H}_n = \mathbf{F}^H \mathbf{D}_n \mathbf{F}$ , where  $\mathbf{D}_n$  is a diagonal matrix. After the training interference elimination, the frequency-domain received signal can be written as

$$\begin{aligned} \mathbf{z}_n &= \mathbf{z}_{nf} - \mathbf{F}\hat{\mathbf{h}}_{nF} * \mathbf{F}\mathbf{t}_{L_s} = \mathbf{F}\mathbf{y}_n - \mathbf{F}\hat{\mathbf{h}}_{nF} * \mathbf{F}\mathbf{t}_{L_s} \\ &= \mathbf{D}_n \mathbf{F}\mathbf{s}_n + \left[ \mathbf{D}_n (r\mathbf{F}\mathbf{t}_{L_s}) - \mathbf{F}\hat{\mathbf{h}}_{nF} * \mathbf{F}\mathbf{t}_{L_s} + \mathbf{F}\mathbf{w}_n \right]. \\ &= \mathbf{D}_n \mathbf{F}\mathbf{s}_n + \mathbf{w}'_n \end{aligned} \quad (15)$$

Based on the estimated channel, the diagonal elements of the diagonal matrix  $\hat{\mathbf{D}}_n$  can be acquired as follows:

$$\left[ \hat{d}_{n1}^1, \hat{d}_{n1}^2, \dots, \hat{d}_{n1}^{L_s} \right]^T = \sqrt{L_s} \mathbf{F}\hat{\mathbf{h}}_{nF}, \quad n = 1, \dots, N_y. \quad (16)$$

As the power of the transmitted symbol sequence is set to 1, the noise power  $\sigma_n^2$  for the  $n$ -th segment is the difference between the power  $P\mathbf{y}_n$  for the received segment and the corresponding channel energy  $E\hat{\mathbf{h}}_{nF}$ , i.e.,

$$\sigma_n^2 = P\mathbf{y}_n - E\hat{\mathbf{h}}_{nF}. \quad (17)$$

Take the  $n$ -th segment as an example of LMMSE equalization. Following literatures [30,33,34], the a priori mean and variance of the symbol  $f_i$  (the symbol sequence  $\mathbf{f}_n$ ) are as follows:

$$\begin{cases} m_i^a = \frac{1}{\sqrt{2}} \tanh\left(\frac{L_n^a(c_i^1)}{2}\right) + j \frac{1}{\sqrt{2}} \tanh\left(\frac{L_n^a(c_i^2)}{2}\right), \\ v_i^a = 1 - |m_i^a|^2 \end{cases}, \quad (18)$$

where both the initial values of  $L_n^a(c_i^1)$  and  $L_n^a(c_i^2)$  (the initial a priori LLRs of the interleaved encoded bits) are set to 0. The estimated interleaved bit sequence is converted to the symbol sequence  $\mathbf{m}^a = [m_1^a, \dots, m_{L_s}^a]$  by using (18). The a posteriori mean and variance of the symbol  $f_i$  (the symbol sequence  $\mathbf{f}_n$ ) are as follows:

$$\begin{cases} v_1^p = v_2^p = \dots = v_{L_s}^p = \frac{1}{L_s} \sum_{k=1}^{L_s} \left( \frac{1}{\bar{v}} + \frac{|a_n^k|^2}{\sigma_n^2} \right)^{-1}, \\ \mathbf{m}^p = \mathbf{m}^a + \mathbf{F}^H \hat{\mathbf{D}}_n^H \left( \frac{\sigma_n^2}{\bar{v}} \mathbf{I} + \hat{\mathbf{D}}_n \hat{\mathbf{D}}_n^H \right)^{-1} (\mathbf{z}_n - \hat{\mathbf{D}}_n \mathbf{F}\mathbf{m}^a) \end{cases}, \quad (19)$$

where  $\bar{v} = \frac{1}{L_s} \sum_{i=1}^{L_s} v_i^a$  and  $\mathbf{m}^p = [m_1^p, \dots, m_{L_s}^p]$ . The a posteriori mean sequence  $\mathbf{m}^p$  is the estimated symbol sequence after LMMSE equalization. It is noted that the computational complexity of the LMMSE equalizer is dominated by (19), and its computational complexity is only in the order of  $\log(L_s)$  per symbol. In addition, the extrinsic mean and variance of the symbol  $f_i$  (the symbol sequence  $\mathbf{f}_n$ ) are as follows:

$$\begin{cases} v_i^e = \left( \frac{1}{v_i^p} - \frac{1}{v_i^a} \right)^{-1} \\ m_i^e = v_i^e \left( \frac{m_i^p}{v_i^p} - \frac{m_i^a}{v_i^a} \right) \end{cases}. \quad (20)$$

As QPSK mapping is used, the extrinsic LLRs of the interleaved encoded bits  $c_i^1$  and  $c_i^2$  can be expressed as



$$\begin{cases} L_n^e(c_i^1) = 2\sqrt{2}\text{Re}[m_i^e/v_i^e] \\ L_n^e(c_i^2) = 2\sqrt{2}\text{Im}[m_i^e/v_i^e] \end{cases} \quad (21)$$

The estimated symbol sequence is converted to the extrinsic LLRs (i.e., the estimated interleaved bit sequence  $L_n^e$ ) by using (19)–(21). The extrinsic LLRs of the segments are denoted collectively as  $L^e$  and then input into the decoder for the next round of iteration (turbo equalization).

### 3.3. Adaptive Underwater Acoustic Communications Based on Constellation Aggregation

We set a certain iteration number, such as one iteration, to obtain the aggregation degree after LMMSE equalization, i.e., (22), where  $\hat{f}_i$  is the a posteriori mean of a symbol  $f_i$ , and its real part and imaginary part are denoted by  $\hat{f}_i^{\text{Re}}$  and  $\hat{f}_i^{\text{Im}}$ , respectively.

$$\begin{aligned} \zeta_i &= \left\| \begin{pmatrix} \hat{f}_i^{\text{Re}}, \hat{f}_i^{\text{Im}} \end{pmatrix} - 1/\sqrt{2}\{(1, -1), (1, 1), (-1, -1), (-1, 1)\} \right\| \\ &= \left\| \begin{pmatrix} \text{abs}(\hat{f}_i^{\text{Re}}) + j \times \text{abs}(\hat{f}_i^{\text{Im}}) \end{pmatrix} - (1/\sqrt{2} + j \times 1/\sqrt{2}) \right\| \\ &= \sqrt{[\text{abs}(\hat{f}_i^{\text{Re}}) - 1/\sqrt{2}]^2 + [\text{abs}(\hat{f}_i^{\text{Im}}) - 1/\sqrt{2}]^2} \end{aligned} \quad (22)$$

We compute the mean of all  $\zeta_i$  for a frame of information bits, i.e.,  $\zeta_m = \frac{1}{L_f} \sum_{i=1}^{L_f} \zeta_i$ .

Denote  $\zeta_{in}$  and  $\zeta_{ex}$  as the inner boundary and the outer boundary, respectively. As shown in Figure 4, when  $\zeta_m < \zeta_{in}$ , the encoding rate will be increased automatically; when  $\zeta_m > \zeta_{ex}$ , the encoding rate will be reduced automatically; when  $\zeta_{in} \leq \zeta_m \leq \zeta_{ex}$ , the encoding rate will be kept.

## 4. Simulation Results

The simulation parameters are shown in Table 1. Rate-1/2, rate-1/4, rate-1/8 and rate-1/16 convolutional codes and QPSK mapping are used. A variety of power ratios of the training sequence and the symbol sequence, such as 0.15:1, 0.2:1, 0.25:1 and 0.3:1, are used. The standard block with 1024 symbols is divided into a number of segments with a variety of lengths, including 128 symbols, 256 symbols, 512 symbols and 1024 symbols. The corresponding cases are denoted by S128, S256, S512 and W1024, respectively, where the prefix W means that the standard block is treated as a segment. W1024 is used as the benchmark turbo system. S128, S256 and S512 are used in the proposed GL turbo system. The CP is set to 128 symbols. One frame includes 100 blocks, and one block includes 1024 information bits. Assume that a 4 kHz bandwidth is provided. For S256, with rate-1/2, rate-1/4, rate-1/8 and rate-1/16 convolutional codes, the transmission rates are 2667 bits/s, 1333 bits/s, 667 bits/s and 333 bits/s, respectively, and the corresponding bandwidth efficiencies are 0.67 bps/Hz, 0.33 bps/Hz, 0.17 bps/Hz and 0.08 bps/Hz, respectively. The SNR is from −4 dB to 13 dB. A static channel, as shown in Figure 6, and the white Gaussian noise are used.

The BER performance for S256 is shown in Figure 7. It can be seen from the results that both the ST scheme and the channel estimate fusion based on Gaussian likelihood are effective. The lower the encoding rate, the better the BER performance that the system can achieve. Taking Figure 7b, with SNR = 7 dB and rate-1/4 convolutional code, after three iterations, 100 blocks of information bits are correctly decoded. From Figure 7, after two iterations, all information bits with rate-1/2 convolutional code and SNR = 13 dB (Figure 7a), with rate-1/4 convolutional code and SNR = 8 dB (Figure 7b), with rate-1/8 convolutional code and SNR = 3 dB (Figure 7c) and with rate-1/16 convolutional code and SNR = 0 dB (Figure 7d) are correctly decoded.

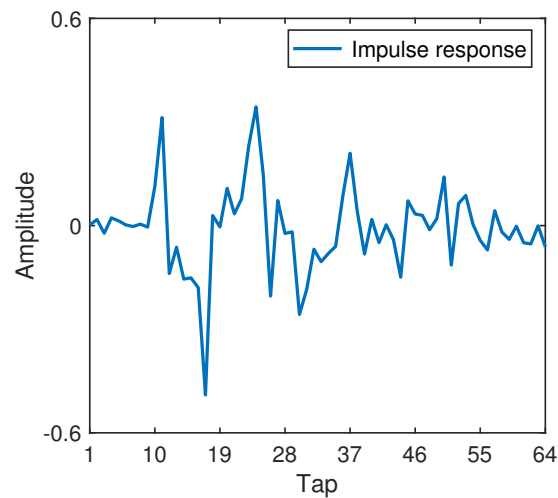
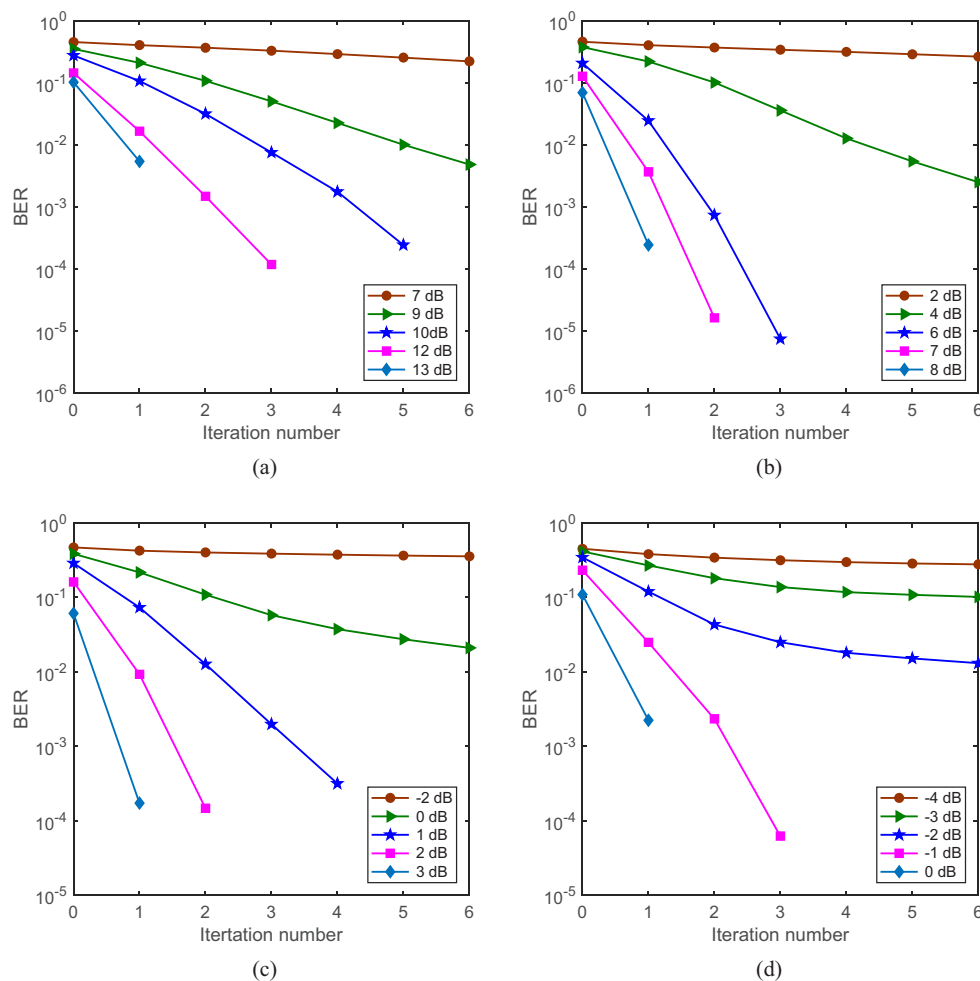


Figure 6. Impulse response of a channel.

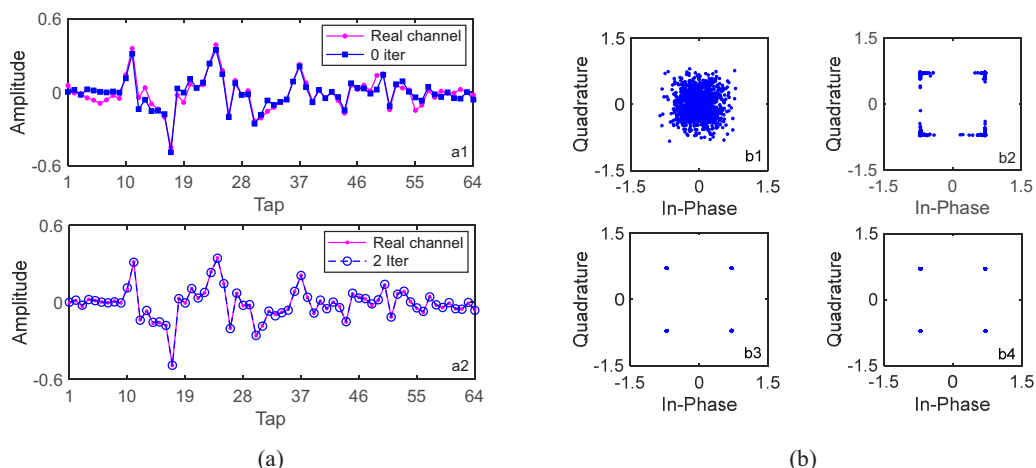
Table 1. Parameters of simulations and experiments.

	Simulation	Yellow Sea 1	Yellow Sea 2
Encoding rate	1/2, 1/4, 1/8, 1/16	1/4, 1/8, 1/16	1/2, 1/4
Power ratio $r$	0.15:1 to 0.3:1	0.25:1	0.25:1
Segment length	128, 256, 512, 1024 sym	256, 512, 1024 sym	256 sym
CP	128 sym	128 sym	16 sym
1 frame, 1 block	100 blocks, 1024 bits	16 blocks, 1024 bits	16 blocks, 1024 bits
Mapping, system	QPSK, Baseband	QPSK, Single carrier	QPSK, Single carrier
Time dura. of one bit	—	$5 \times 10^{-4}$ , $10 \times 10^{-4}$ , $20 \times 10^{-4}$ s	$2.5 \times 10^{-4}$ , $5 \times 10^{-4}$ s,
Time dura. of one symbol	—	$2.5 \times 10^{-4}$ s	$2.5 \times 10^{-4}$ s
Center frequency, Filter	—	12 kHz, Band pass	12 kHz, Band pass
Bandwidth	—	4 kHz	4 kHz
Sampling frequency	—	96 kHz	96 kHz
S256, Transmission rate	—	1333, 667, 333 bits/s	3765, 1882 bits/s
S256, Band. effi. (bps/Hz)	—	0.33 (1/4), 0.17, 0.08	0.94 (1/2), 0.47
Communication distance	—	5.5 km	5.5 km
Transducer depth	—	4 m	4 m
Hydrophone depth	—	4 m	5 m
Relative speed	—	0.5 m/s	0.5 m/s
SNR	−4 dB to 13 dB	approximately 9 dB	approximately 13 dB

Taking a block of 1024 information bits with rate-1/4 convolutional code as an example, where the noise is not added, the channel in Figure 6 is used, and the channel estimation and equalization results are shown in Figure 8. S256 and a static channel are used; therefore, the channels of the four consecutive segments are the same and their channels are perfectly correlated, i.e.,  $\alpha_p = 1$ . When turbo equalization is not used, i.e., with 0 iterations, the corresponding channel estimate and equalization results are shown in Figure 8(a1),(b1). The estimated channel in Figure 8(a1) is obviously different from the real channel in Figure 6, and the constellation points after LMMSE equalization in Figure 8(b1) are significantly scattered. When turbo equalization is performed once, i.e., after one iteration, the corresponding equalization results are shown in Figure 8(b2), where it is noted that the estimated channels have been updated before turbo equalization, and the aggregation degree of the constellation points after LMMSE equalization becomes significantly better. When turbo equalization is performed two times, i.e., after two iterations, the corresponding equalization results are shown in Figure 8(b3), where the constellation points after LMMSE equalization are ideally condensed together, and the corresponding estimated channel in Figure 8(a2) is exactly the same as the real channel in Figure 6, demonstrating the effectiveness of the ST scheme used for enhancing the channel estimation and tracking capability and the GL algorithm used for the channel information fusion of the segments.



**Figure 7.** BER performance for S256. (a) Rate-1/2 convolutional code; (b) rate-1/4 convolutional code; (c) rate-1/8 convolutional code; (d) rate-1/16 convolutional code.

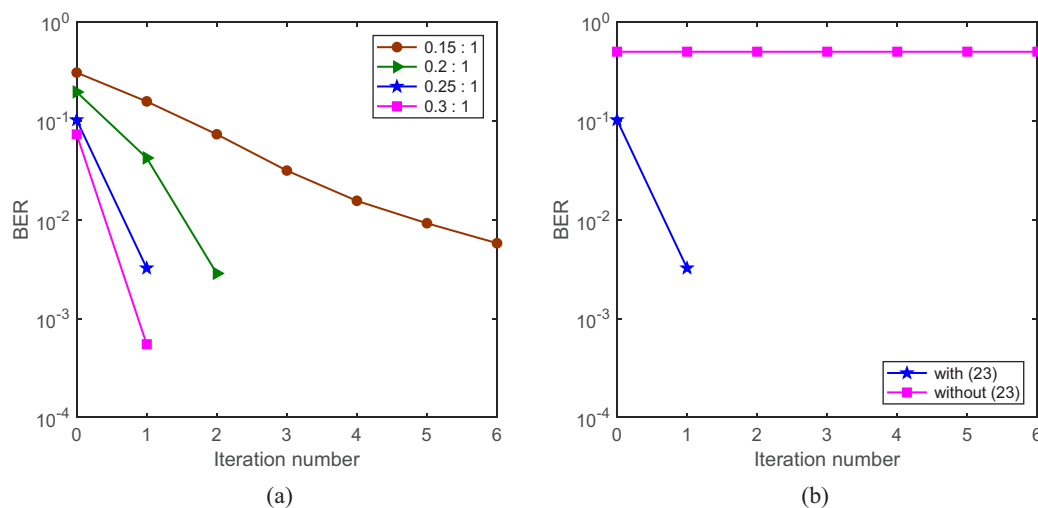


**Figure 8.** Estimated channels and constellations of a block of 1024 information bits with S256 in simulations at SNR = 13 dB. (a1) Channel estimate of one segment without iteration; (a2) channel estimate of one segment after 2 iterations; (b1) constellations after 0 iteration; (b2) constellations after 1 iteration; (b3) constellations after 2 iterations; (b4) constellations after 3 iterations.

From Figure 8b, it is clear that we can carry out adaptive communications according to the pre-set constellation aggregation degree threshold. It is important to note that we do

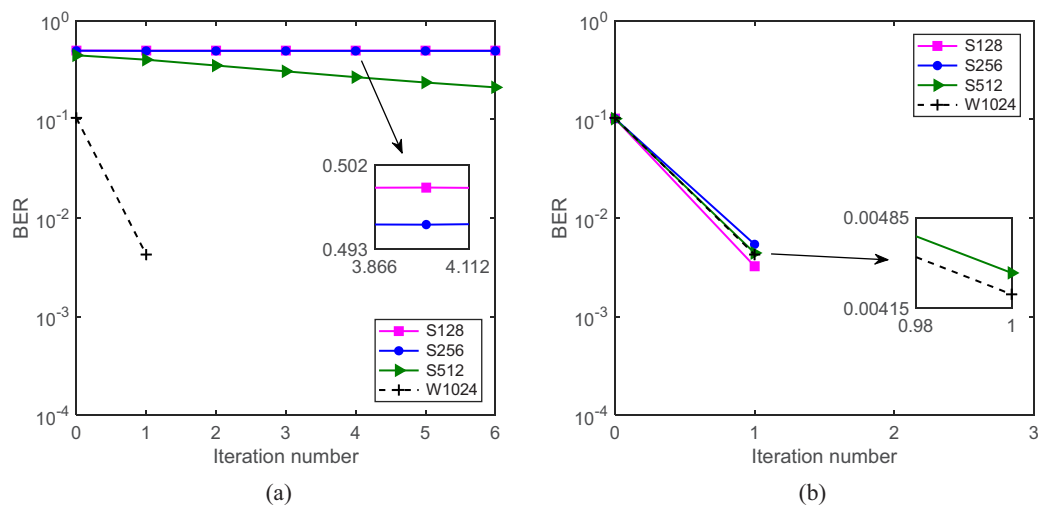
not show the adaptive communication performance, as we can see the results clearly from Figure 7.

Next, we test the BER performance with a variety of power ratios of the training sequence and the symbol sequence. Taking S128 with rate-1/2 convolutional code as an example, the BER performance of the system is shown in Figure 9a. The green triangle line represents the BER performance with a power ratio 0.2:1 and SNR = 13 dB; after three iterations, 100 blocks of information bits are correctly decoded. Considering the complexity and variability of underwater acoustic channels incurred by the moving transceivers, the power ratio 0.25:1 is used in the follow-up simulations and experiments. Assuming that the SNR = 13 dB and the power ratio is 0.25:1, the BER performance of the system is shown in Figure 9b. The blue star line represents the BER performance of the system with the training interference elimination; after two iterations, 100 blocks of information bits are correctly decoded, demonstrating the effectiveness of the training interference elimination. The pink square line represents the BER performance of the system without the training interference elimination, and it can be seen that, if we do not use the training interference elimination, the system simply does not work.



**Figure 9.** BER performance of the system with S128 and SNR = 13 dB. (a) A variety of power ratios of the training sequence and the symbol sequence; (b) with or without the training interference elimination. The power ratio is 0.25:1.

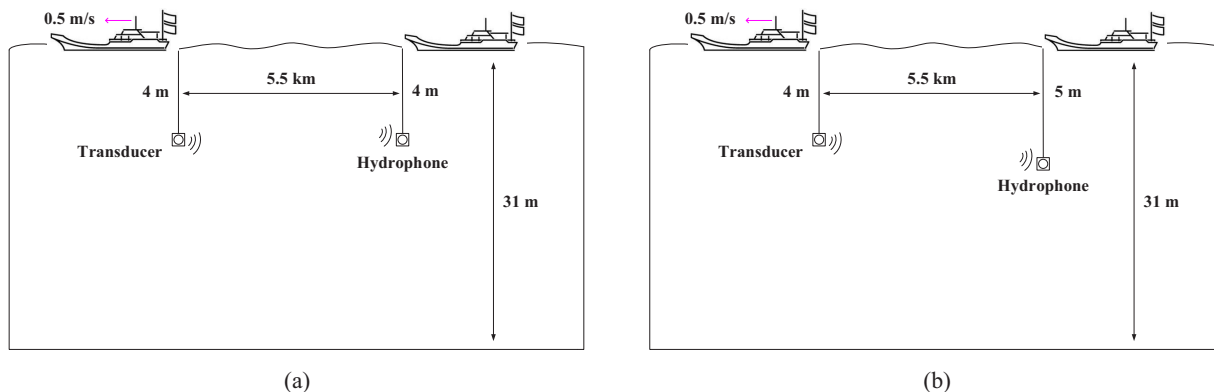
Then, we test the BER performance of the system by using the ST scheme and the GL algorithm, where W1024 is used as the benchmark turbo system. The BER performance comparison is shown in Figure 10. From Figure 10a, if the GL algorithm is not used, the system with a variety of segment lengths does not work. From Figure 10b, if the GL algorithm is used to fuse the local channel estimates to obtain global channel estimates, it can be seen that, no matter how long the segment is, the BER performances for S128, S256, S512 and W1024 are similar. This is because, regardless of the segment length, the ‘whole’ standard symbol block is used to acquire the global channel estimate for each segment. This demonstrates that the proposed GL turbo system (S128, S256 and S512) can achieve a similar performance as the benchmark turbo system (W1024).



**Figure 10.** BER performance of the system with SNR = 13 dB, rate-1/2 convolutional code and the power ratio 0.25:1. (a) The GL algorithm is not used for channel estimation; (b) the GL algorithm is used for channel estimation.

**5. Experimental Results**

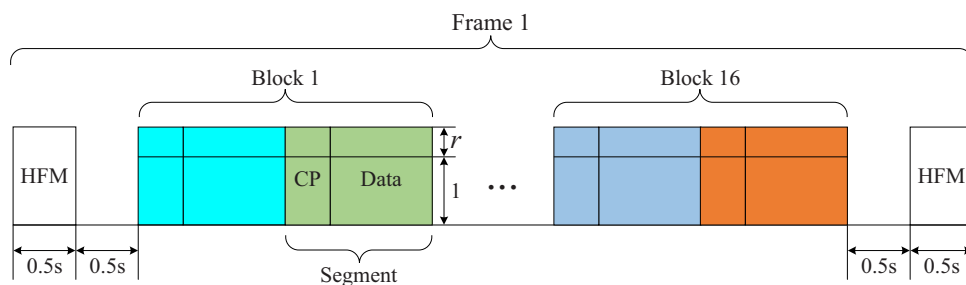
Two separate underwater acoustic communication experiments with moving transceivers were carried out in the Yellow Sea in 2021, named Yellow Sea 1 and Yellow Sea 2, respectively. Their deployments are shown in Figure 11a,b, respectively. We did not use a vertical array in the experiments. The two receiving hydrophones were completely independent, i.e., they had nothing to do with each other. Therefore, multiple receiver channels were not exploited in the system. The height of the sea waves was from 0.5 m to 1 m; the sea temperature was 5.6 °C; the south wind was from level 3 to level 4; the ship with the transducer floated away from the ship with the hydrophone at a speed of approximately 0.5 m/s.



**Figure 11.** Experimental deployment. (a) Yellow Sea 1; (b) Yellow Sea 2.

The detailed experimental parameters are shown in Table 1. For the two experiments, QPSK mapping and the power ratio of the training sequence and the symbol sequence 0.25:1 were used; both the communication distances of the transceivers were approximately 5.5 km; one frame included 16 blocks, and one block included 1024 information bits; the single-carrier communication system was used; the center frequency was 12 kHz with a bandwidth of 4 kHz; and the sampling frequency was 96 kHz. The signal structure for field experiments is shown in Figure 12.

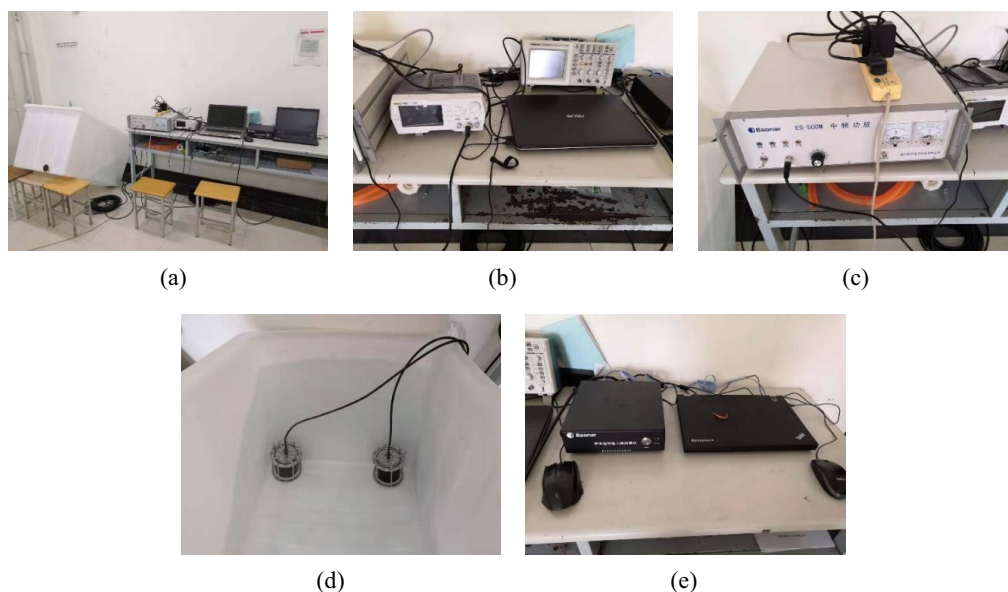




**Figure 12.** Signal structure for field experiments.

### 5.1. Adaptive Underwater Acoustic Communications with SNR = 9 dB

The experimental deployment and instruments for Yellow Sea 1 are shown in Figure 11a and Figure 13, respectively. Rate-1/4, rate-1/8 and rate-1/16 convolutional codes were adopted. S256, S512 and W1024 were used, and the CP was set to 128 symbols. Taking S256, for rate-1/4, rate-1/8 and rate-1/16 convolutional codes, the transmission rate were 1333 bits/s, 667 bits/s and 333 bits/s, respectively, and the corresponding bandwidth efficiencies were 0.33 bps/Hz, 0.17 bps/Hz and 0.08 bps/Hz, respectively. Both transceivers were deployed at a depth of 4 m.



**Figure 13.** Experimental instruments in the Yellow Sea. (a) The whole system; (b) transmitter; (c) power amplifier; (d) the transducer and the hydrophone; (e) receiver.

We firstly used rate-1/16 convolutional code, and the BER performance of 16 data blocks based on the GL algorithm is shown in Figure 14. By comparing the results of S256, S512 and the benchmark turbo system (W1024), it can be seen that S256 was much more effective than S512 and the benchmark turbo system for underwater acoustic communications with moving transceivers. After only one iteration, all information bits with S256 were correctly decoded. However, both S512 (pink square curve) and the benchmark turbo system (blue dotted curve) are completely invalid. This is because moving communications incur time-varying channels. The average channel estimate does not effectively represent the channel information of the 512 symbol block and 1024 symbol block. Taking the first block for S256 in Figure 14 as an example, as S256 was used, there were four consecutive segments for the first data block, and their channels were different due to floating transceivers, i.e.,  $\alpha_p \neq 1$ . It is important to note that  $\alpha_p$  can be obtained automatically, and can be calculated by using the estimated channels of the four segments, as  $\alpha_p$  was equal to the correlated coefficient of the estimated channels of the four segments, which was also

used in the initial channel estimation. When turbo equalization was not used, i.e., with 0 iterations, the corresponding channel equalization results are shown in Figure 15(b1), and the constellation points after LMMSE equalization were very scattered. Then, the automatically determined  $\alpha_p$  was recalculated by using the updated channel estimates of the four segments. When turbo equalization was performed once, i.e., after one iteration, the corresponding equalization results are shown in Figure 15(b2), where the constellation points after LMMSE equalization were ideally condensed together. The corresponding estimated channels of the four segments in Figure 15a were significantly different, where  $\alpha_p = 0.07$  after one iteration, demonstrating the time-variation of the channel and the effectiveness of the ST scheme and the GL algorithm.

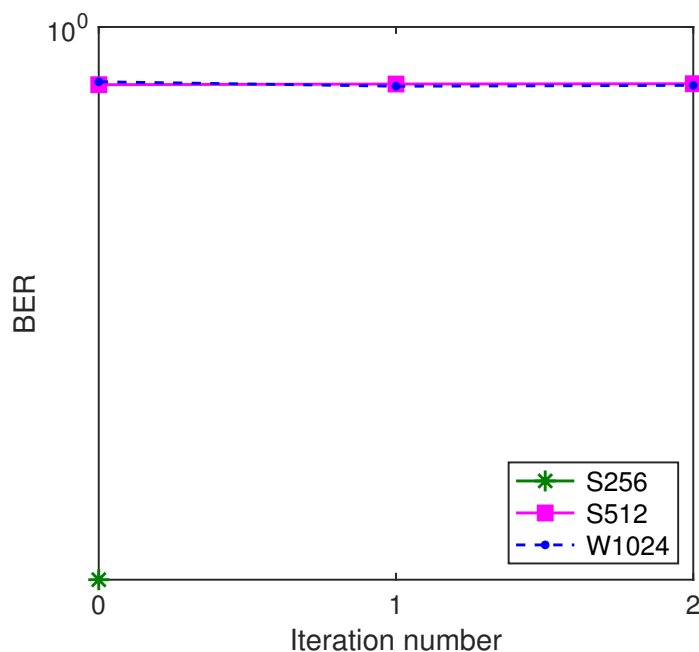


Figure 14. BER performance of the GL turbo system with rate-1/16 convolutional code for Yellow Sea 1.

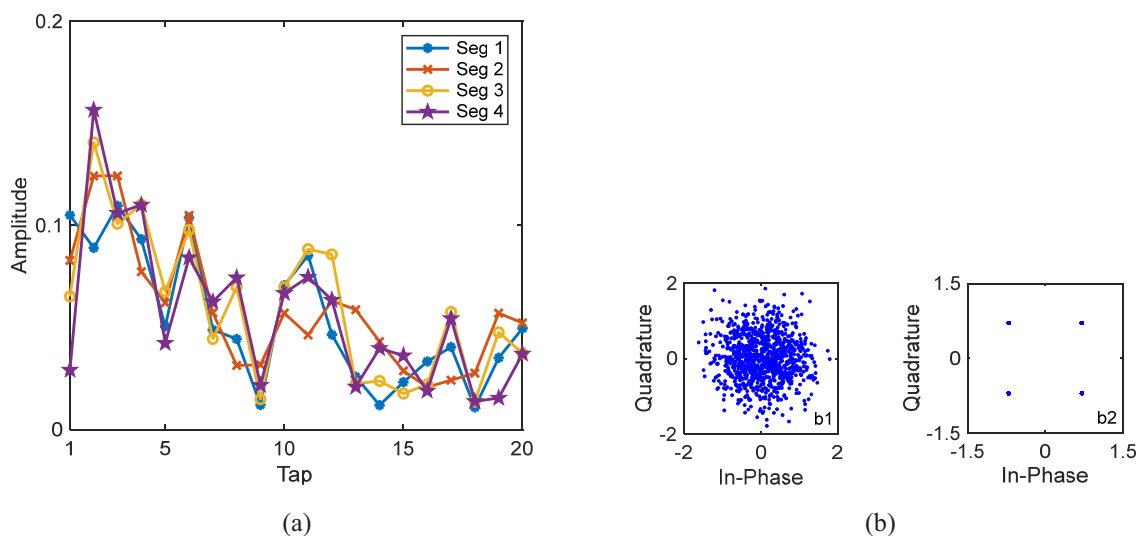


Figure 15. Estimated channels and equalization results of the first block with S256 and 1/16 code rate for Yellow Sea 1 in Figure 14. (a) The estimated channels of the consecutive four segments after 1 iteration; (b1) constellations after 0 iteration; (b2) constellations after 1 iteration.

Then, we carried out field experiments with a variety of convolutional codes to test the effectiveness of direct adaptive communications. The adaptive threshold setting is shown in Table 2. We used the mean aggregation degree after one iteration to compare the threshold, where the inner boundary is set to  $\zeta_{in} = 0.03$  and the outer boundary is set to  $\zeta_{ex} = 0.2$ . When the mean aggregation degree is  $\zeta_m < 0.03$ , the encoding (transmission) rate will be improved automatically; when the mean aggregation degree is  $\zeta_m > 0.2$ , the encoding (transmission) rate will be reduced automatically; when the mean aggregation degree is  $0.03 \leq \zeta_m \leq 0.2$ , the encoding (transmission) rate will be kept the same.

**Table 2.** Threshold setting of mean aggregation degree  $\zeta_m$  after one iteration for Yellow Sea 1 and Yellow Sea 2.

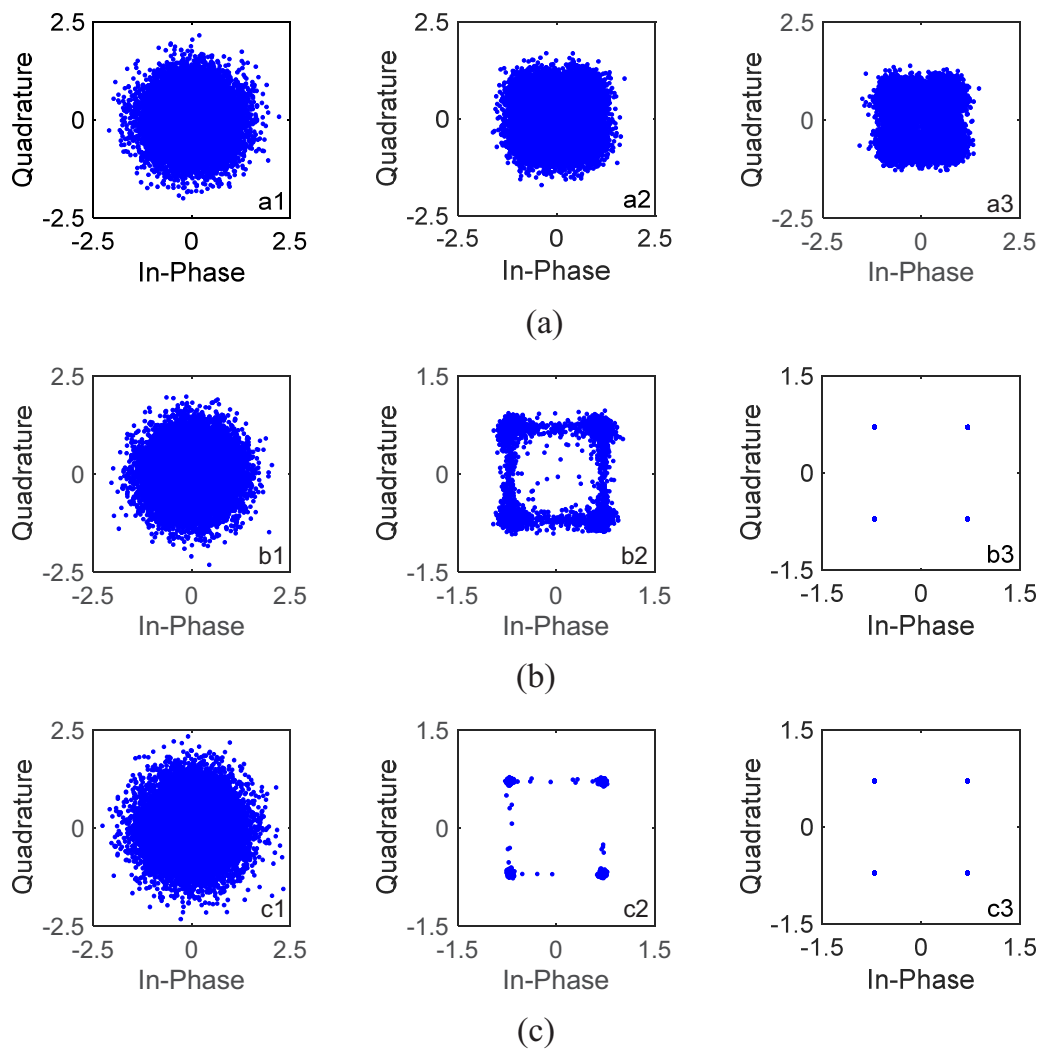
$\zeta_m < 0.03$	Improve the encoding rate
$\zeta_m > 0.2$	Reduce the encoding rate
$0.03 \leq \zeta_m \leq 0.2$	Keep the encoding rate

The calculation of the mean aggregation degree  $\zeta_m$  is shown in Table 3. For Yellow Sea 1, assuming that rate-1/16 convolutional code was used first, after one iteration, the mean aggregation degree was  $\zeta_m = 0.002$ , which was less than 0.03. Therefore, the encoding rate was improved automatically, i.e., the encoding rate was adjusted from rate-1/16 convolutional code to rate-1/8 convolutional code automatically. We can see that, after one iteration, with rate-1/8 convolutional code, the mean aggregation degree was  $\zeta_m = 0.0591$ , which belonged to  $[0.03, 0.2]$ . Therefore, the encoding rate was kept the same. Assuming that rate-1/4 convolutional code was used first, after one iteration, the mean aggregation degree was  $\zeta_m = 0.4442$ , which was more than 0.2. Therefore, the encoding rate was reduced automatically, i.e., the encoding rate was adjusted from rate-1/4 convolutional code to rate-1/8 convolutional code automatically. As after one iteration with rate-1/8 convolutional code, the mean aggregation degree was  $\zeta_m = 0.0591$ , which belonged to  $[0.03, 0.2]$ , the encoding rate was kept. The aggregation performance of the 16 blocks of information bits with S256 is shown in Figure 16. From Figure 16b, after one iteration with rate-1/8 convolutional code, the constellation points of the 16 blocks of information bits were obviously clustered.

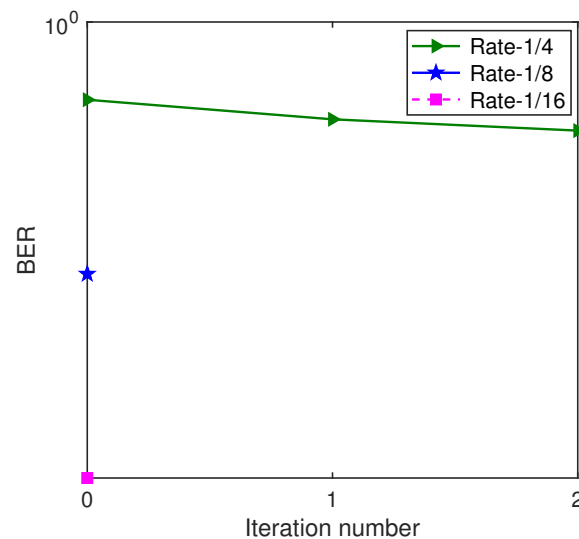
**Table 3.** Calculation of the mean aggregation degree  $\zeta_m$  for Yellow Sea 1 and Yellow Sea 2.

Iteration Number	Yellow Sea 1			Yellow Sea 2	
	Rate-1/4	Rate-1/8	Rate-1/16	Rate-1/2	Rate-1/4
0	0.5546	0.5505	0.5613	0.4719	0.5396
1	0.4442	0.0591	0.002	0.041	0.007
2	0.2953	$3.44 \times 10^{-6}$	$3.09 \times 10^{-6}$	0.0018	$1.51 \times 10^{-6}$

The BER performance based on the ST scheme and the GL algorithm with S256 for Yellow Sea 1 is shown in Figure 17. We can see that, after one iteration, the decoding with rate-1/4 convolutional code was invalid, and the decodings with rate-1/8 convolutional code and rate-1/16 convolutional code were valid. With rate-1/8 convolutional code, all information bits were correctly decoded after one iteration, and the BER performance was sufficient in meeting the needs for underwater acoustic communications. Therefore, the rate-1/8 convolutional code was kept, which was in keeping with the result from the mean aggregation degree, demonstrating the effectiveness of the GL algorithm and the CA algorithm.



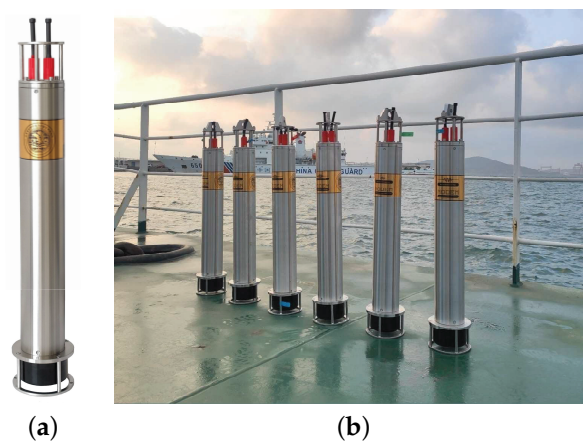
**Figure 16.** Constellations of the 16 blocks of data with S256 after 0, 1 and 2 iterations for Yellow Sea 1 in Figure 17. (a1) Rate-1/4, 0 iteration; (a2) rate-1/4, 1 iteration; (a3) rate-1/4, 2 iterations; (b1) rate-1/8, 0 iteration; (b2) rate-1/8, 1 iteration; (b3) rate-1/8, 2 iterations; (c1) rate-1/16, 0 iteration; (c2) rate-1/16, 1 iteration; (c3) rate-1/16, 2 iterations.



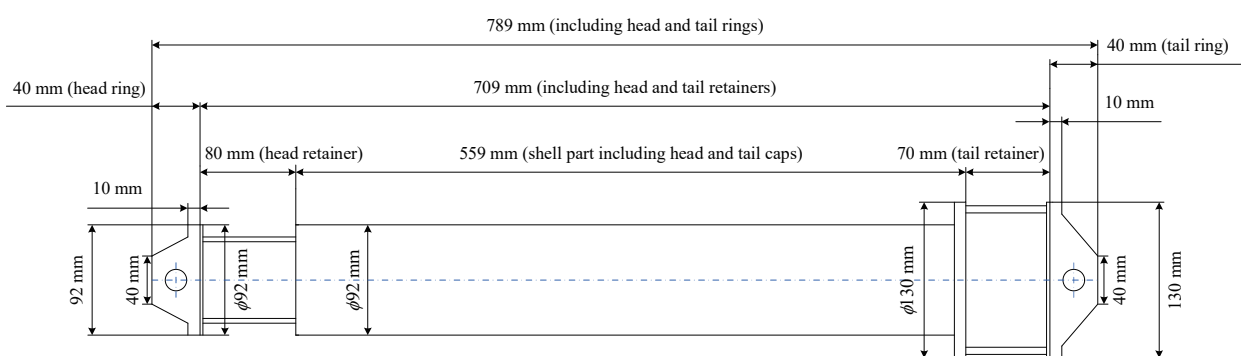
**Figure 17.** BER performance of the GL turbo system with S256 for Yellow Sea 1.

### 5.2. Adaptive Underwater Acoustic Communications with SNR = 13 dB

The experimental deployment in the Yellow Sea is shown in Figure 11b. An underwater acoustic communication machine (Seatrix Modem) was used, whose illustration and dimensions are shown in Figures 18 and 19, respectively. The introduction of the machine is listed in Table 4. An SD card was plugged in Seatrix Modem to collect data at the receiver, and the collected data were analyzed by using a computer. The transmitting ship floated away from the receiving ship at a speed of approximately 0.5 m/s. For Yellow Sea 2, rate-1/2 and rate-1/4 convolutional codes were used. S256 was used, and the CP was set to 16 symbols. The transmission rates were 3765 bits/s and 1882 bits/s, respectively, and the corresponding bandwidth efficiencies were 0.94 bps/Hz (rate-1/2) and 0.47 bps/Hz (rate-1/4), respectively. The deployment depths of the transducer and the hydrophone were 4 m and 5 m, respectively. The main goal of the experiment is to demonstrate a successful implementation on modem hardware for the proposed algorithms. As the receiver in Section 5.2 has a higher SNR than the receiver in Section 5.1, higher code rates can be used in Section 5.2. As the communication environments in Sections 5.1 and 5.2 are similar, their channels are comparable.



**Figure 18.** Illustration of underwater acoustic communication machine (Seatrix Modem). (a) Single machine; (b) multiple machines.



**Figure 19.** Dimension of underwater acoustic communication machine (Seatrix Modem).

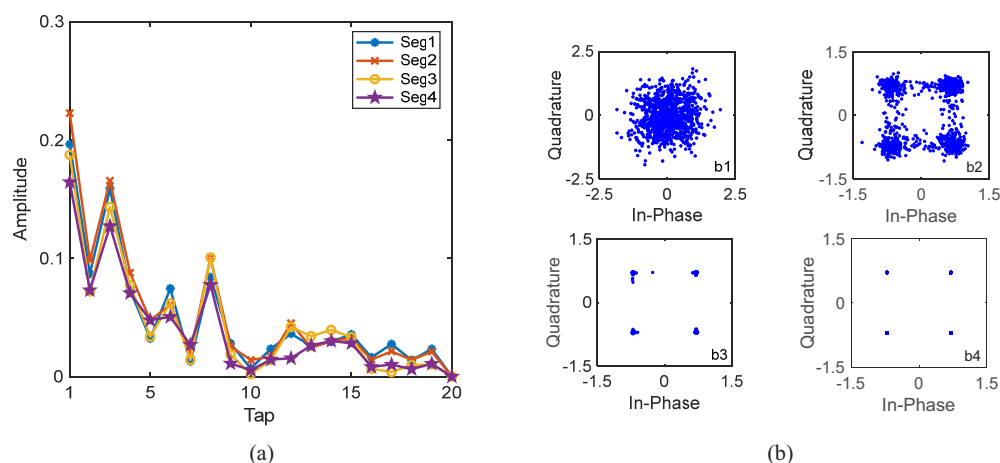
**Table 4.** Introduction of underwater acoustic communication machine (Seatrix Modem).

Communication frequency	9 kHz to 14 kHz (10 kHz to 14 kHz was used)
Communication distance	6000 m with high SNR
Work depth	2000 m
Electrical parameters	Received power consumption 1 w; Transmission power consumption 10 w to 60 w; Built-in 400 wh rechargeable battery; RS-232 interface.
Mechanical parameters	Length 790 mm × Width (Diameter) 130 mm; Weight 15 kg (including the battery)



We used rate-1/2 and rate-1/4 convolutional codes and S256. The BER performance based on the GL algorithm is shown in Table 5. It can be seen that S256 was very effective for underwater acoustic communications with moving transceivers. After only one iteration, all information bits were correctly decoded. Taking the fourth block with rate-1/2 convolutional code in Table 5 as an example, there were four consecutive segments for the fourth data block, and their channels were different due to floating transceivers, i.e.,  $\alpha_p \neq 1$ . When turbo equalization was not used, i.e., with 0 iterations, the corresponding channel equalization results are shown in Figure 20(b1), and the constellation points after LMMSE equalization were very scattered. Then, the automatically determined  $\alpha_p$  was updated. When turbo equalization was performed once, i.e., after one iteration, as shown in Figure 20(b2), the constellation points after LMMSE equalization were still scattered. After three iterations, as shown in Figure 20(b4), the constellation points after LMMSE equalization were ideally condensed together. The corresponding estimated channels of the four segments in Figure 20a were significantly different, where  $\alpha_p = 0.09$  after three iterations, demonstrating the time-variation of the channel and the effectiveness of the ST scheme and the GL algorithm. Comparing Figure 15a in Yellow Sea 1 and Figure 20a in Yellow Sea 2, it can be seen that their channel lengths are almost the same. This is because the communication environments are basically the same. We did not show BERs for S512 and W1024, as they basically do not work.

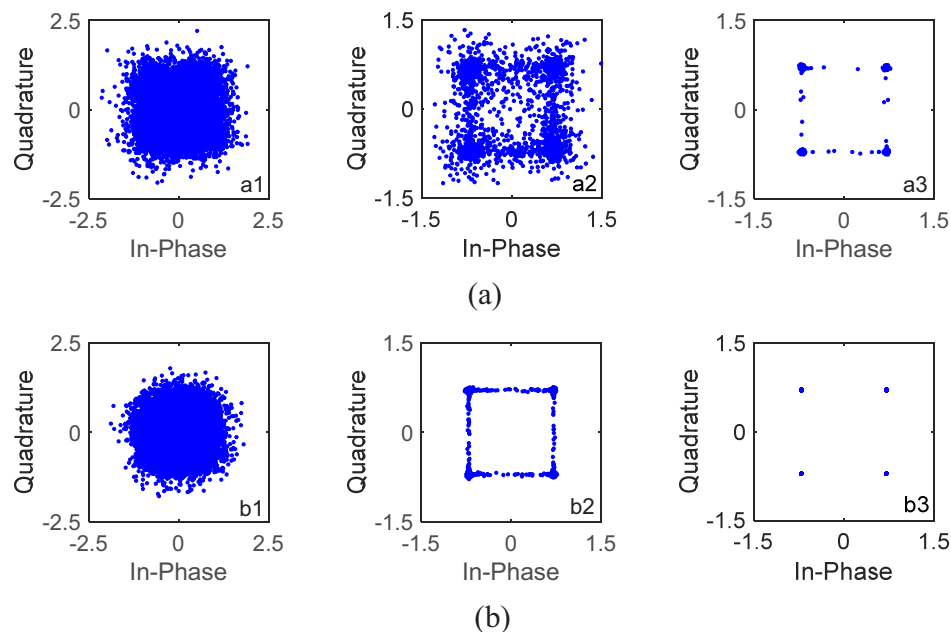
Then, we tested the effectiveness of direct adaptive communications with real underwater acoustic communication machines. The adaptive threshold setting is shown in Table 2. We still used the mean aggregation degree after one iteration to compare the threshold, where the inner boundary is set to  $\xi_{in} = 0.03$  and the outer boundary is set to  $\xi_{ex} = 0.2$ . When the mean aggregation degree is  $\xi_m < 0.03$ , the encoding (transmission) rate will be improved automatically; when the mean aggregation degree is  $\xi_m > 0.2$ , the encoding (transmission) rate will be reduced automatically; when the mean aggregation degree is  $0.03 \leq \xi_m \leq 0.2$ , the encoding (transmission) rate will be kept the same.



**Figure 20.** Estimated channels and equalization results of the fourth block with S256 and 1/2 code rate for Yellow Sea 2 in Figure 14. (a) The estimated channels of the consecutive four segments after 3 iterations; (b1) constellations after 0 iteration; (b2) constellations after 1 iteration; (b3) constellations after 2 iterations; (b4) constellations after 3 iterations.

The calculation of the mean aggregation degree  $\xi_m$  is shown in Table 3. For Yellow Sea 2, assuming that rate-1/4 convolutional code was used first, after one iteration, the mean aggregation degree was  $\xi_m = 0.007$ , which was less than 0.03. Therefore, the encoding rate was improved automatically, i.e., the channel code was adjusted from rate-1/4 convolutional code to rate-1/2 convolutional code automatically. After one iteration with rate-1/2 convolutional code, the mean aggregation degree was  $\xi_m = 0.041$ , which belonged to  $[0.03, 0.2]$ . Therefore, the encoding rate was kept the same. Assuming that rate-1/2 convolutional

code was used first, after one iteration, the mean aggregation degree was  $\xi_m = 0.041$ , which belonged to  $[0.03, 0.2]$ ; therefore, the encoding rate was kept. The aggregation performance of the 16 blocks of information bits with S256 for Yellow Sea 2 is shown in Figure 21. From Figure 21a, after one iteration with rate-1/2 convolutional code, the constellation points of the 16 blocks of information bits became obviously clustered.



**Figure 21.** Constellations of the 16 blocks of data with S256 after 0, 1 and 2 iterations for Yellow Sea 2 in Table 5. (a1) Rate-1/2, 0 iteration; (a2) rate-1/2, 1 iteration; (a3) rate-1/2, 2 iterations; (b1) rate-1/4, 0 iteration; (b2) rate-1/4, 1 iteration; (b3) rate-1/4, 2 iterations.

The BER performance based on the GL algorithm with S256 is shown in Table 5. After one iteration, the decoding performance with rate-1/2 convolutional code was sufficient in meeting the needs for underwater acoustic communications. Therefore, the rate-1/2 convolutional code was kept, which was in keeping with the result from the mean aggregation degree. The experiment demonstrated the effectiveness and practicability of the proposed algorithms in real underwater acoustic communication machines.

**Table 5.** BER performance of the GL turbo system with S256 at SNR = 13 dB for Yellow Sea 2.

Block Number	Rate-1/2			Rate-1/4		
	Iteration Number			Iteration Number		
	0	1	2	0	1	2
1	8.8%	0	0	0	0	0
2	2.3%	0	0	0	0	0
3	0.7%	0	0	0	0	0
4	3.1%	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	2.0%	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
Mean	0.9%	0	0	0.1%	0	0

From the above simulations and experimental results, we can conclude that the best segment length is  $2^n$ , and should be close to and longer than the channel length, and shorter than a period of the training sequence, where  $n$  is an integer. Considering the transmission rate and time variation of the channel, S256 is better than S128, S512 and W1024. The two separate experimental results show that, with SNR = 9 dB, the 1/8 code rate is effective; with SNR = 13 dB, the 1/2 code rate is effective. Even if  $\alpha_p = 0.07$  and  $\alpha_p = 0.09$  ( $\alpha_p$  can be obtained by calculating the correlation coefficient of the consecutive segments), i.e., the channels of the four segments are weakly correlated, the proposed system is still effective.

## 6. Conclusions

The GL algorithm and the CA algorithm have been proposed to achieve a global accurate channel estimation of each segment and automatic encoding rate adjustment. To improve the estimation and tracking capability of time-varying channels, the ST scheme has been used. For channel estimate fusion of the segments, S256 is the best for practical moving underwater acoustic communications. Even if the channel correlation coefficients of the segments are as low as 0.7 and 0.9, the proposed GL turbo system is still effective. The experimental results demonstrate that a 1/8 code rate is effective at SNR = 9 dB, and a 1/2 code rate is effective at SNR = 13 dB. In the process of iteration, direct adaptive communications based on constellation aggregation have been realized. The experimental results illustrate that the encoding rate can be adjusted automatically among the 1/2 code rate, 1/4 code rate, 1/8 code rate and 1/16 code rate by using the mean aggregation degree decision. Simulations and experimental results have verified the effectiveness of the proposed system.

**Author Contributions:** Conceptualization, L.W. and G.Y.; methodology, L.W. and P.Q.; validation, P.Q. and L.W.; formal analysis, L.W. and J.L.; investigation, G.Y. and L.W.; resources, L.W.; data curation, L.W. and G.Y.; writing—original draft preparation, P.Q., J.L. and G.Y.; writing—review and editing, T.C., L.W., G.Y., J.L. and P.Q.; visualization, P.Q. and J.L.; supervision, L.W.; project administration, L.W. and G.Y.; funding acquisition, L.W., X.W. and G.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the China Scholarship Council, in part by the General Program of National Natural Science Foundation of China under Grant 61771271, in part by the General Project of Natural Science Foundation of Shandong Province under Grant ZR2020MF010 and Grant ZR2020MF001 and in part by the Qingdao Source Innovation Program under Grant 19-6-2-4-cg.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

Assuming that the circulant matrix of the impulse response for the channel is  $\mathbf{H}_n$ , the transmitted signals are  $\mathbf{s}_n$  and the white Gaussian noise is  $\mathbf{w}$ . Then, the received signal can be represented as (A1).

$$\begin{aligned}
 \mathbf{y}_n &= \mathbf{H}_n \mathbf{s}_n + \mathbf{w} = \mathbf{H}_n (r \mathbf{t}_{L_s} + \mathbf{f}_n) + \mathbf{w} \\
 &= \begin{pmatrix} h_1 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & h_2 \\ h_2 & h_1 & 0 & \ddots & & \ddots & & \vdots \\ \vdots & h_2 & h_1 & & \ddots & & \ddots & h_{L_c} \\ h_{L_c} & \vdots & h_2 & \ddots & & \ddots & & 0 \\ 0 & h_{L_c} & \vdots & & \ddots & & \ddots & 0 \\ 0 & 0 & h_{L_c} & \ddots & & \ddots & & \vdots \\ \vdots & \vdots & 0 & & \ddots & & \ddots & 0 \\ 0 & 0 & \vdots & \cdots & \cdots & \cdots & \cdots & h_1 \end{pmatrix}_{(L_s+L_c-1) \times (L_s+L_c-1)} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{L_s} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{(L_s+L_c-1) \times 1} + \mathbf{w} \\
 &= \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_{L_c} \end{pmatrix}_{L_c \times 1} * \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{L_s} \end{pmatrix}_{L_s \times 1} + \mathbf{w} \\
 &= \mathbf{h}_n * \mathbf{s}_n + \mathbf{w}
 \end{aligned} \tag{A1}$$

Take elements of  $\mathbf{h}_n, \mathbf{s}_n, \mathbf{t}_{L_s}, \mathbf{f}_n$  and  $\mathbf{w}$  to build the channel estimator based on the LS algorithm. An element of the impulse response of the channel is denoted by  $h_n$ ; an element of the symbol sequence is denoted by  $f_n$ ; an element of the periodic training sequence is denoted by  $t_n$  with a period of  $T$ ; and the superimposed symbol sequence is denoted by  $s_n = f_n + r \times t_n$ . As  $r$  is a constant number, the superimposed symbol sequence can be simplified as  $s_n = f_n + t_n$ . Then, an element of the received signal sequence can be expressed as

$$\begin{aligned}
 y_n &= h_n * s_n + w_n \\
 &= \sum_{i=1}^{L_c} f_{(n+1-i)} h_i + \sum_{i=1}^{L_c} t_{(n+1-i)} h_i + w_n,
 \end{aligned} \tag{A2}$$

where  $w_n$  and  $L_c$  is the white Gaussian noise and the channel length, respectively. Assuming that the mean value of the symbol sequence is 0, and  $T \geq L_c$ , then we can obtain

$$E[y_n] = \sum_{i=1}^{L_c} t_{(n+1-i)} h_i. \tag{A3}$$

Let  $L_s = pT$ . We divide  $E[\mathbf{y}_n]$  into  $p$  subsegments with a length of  $T$ , i.e.,

$$E[\mathbf{y}_n] = E \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{L_s} \end{pmatrix}_{L_s \times 1} = E \begin{pmatrix} \mathbf{y}_{1T} \\ \mathbf{y}_{2T} \\ \vdots \\ \mathbf{y}_{pT} \end{pmatrix}, \tag{A4}$$

where

$$E[\mathbf{y}_{iT}] = \begin{bmatrix} t_{iT+1} & t_{iT} & \cdots & t_{iT-L_c+2} \\ t_{iT+2} & t_{iT+1} & \cdots & t_{iT-L_c+3} \\ \vdots & \vdots & \ddots & \vdots \\ t_{iT+T} & t_{iT+T-1} & \cdots & t_{iT+T-L_c+1} \end{bmatrix}_{T \times L_c} \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_{L_c} \end{pmatrix}_{L_c \times 1} \triangleq \mathbf{A} \mathbf{h}_n. \tag{A5}$$

As we use a periodic training sequence with a period of  $T$ , we can acquire

$$\mathbf{A} = \begin{pmatrix} t_1 & t_T & \cdots & t_{T-L_c+2} \\ t_2 & t_1 & \cdots & t_{T-L_c+3} \\ \vdots & \vdots & \ddots & \vdots \\ t_T & t_{T-1} & \cdots & t_{T-L_c+1} \end{pmatrix}. \quad (\text{A6})$$

It can be seen from (A5) that  $\mathbf{A}$  is a Toeplitz matrix, and its mean is not related to  $i$ . A cumulative sum calculation is performed for  $E[\mathbf{y}_n]$ . When  $p$  is a large value, we can obtain

$$\frac{1}{p} \sum_{i=1}^p E[\mathbf{y}_{iT}] = \frac{1}{p} \sum_{i=1}^p \mathbf{y}_{iT} = \mathbf{A}\mathbf{h}_n. \quad (\text{A7})$$

When  $T \geq L_c$ , based on the LS algorithm, we can obtain (3).

## References

1. Qarabaqi, P.; Milica, S. Statistical characterization and computationally efficient modeling of a class of underwater acoustic communication channels. *IEEE J. Ocean. Eng.* **2013**, *38*, 701–717. [[CrossRef](#)]
2. Song, A.; Stojanovic, M.; Chitre, M. Editorial underwater acoustic communications: Where we stand and what is next? *IEEE J. Ocean. Eng.* **2019**, *44*, 1–6. [[CrossRef](#)]
3. Yang, T.C. Properties of underwater acoustic communication channels in shallow water. *J. Acoust. Soc. Am.* **2012**, *131*, 129–145. [[CrossRef](#)] [[PubMed](#)]
4. Benson, A.; Proakis, J.; Stojanovic, M. Towards robust adaptive acoustic communications. In Proceedings of the OCEANS 2000 MTS/IEEE Conference and Exhibition. Conference Proceedings (Cat. No. 00CH37158), Providence, RI, USA, 11–14 September 2000; Volume 2, pp. 1243–1249.
5. Radosevic, A.; Ahmed, R.; Duman, T.M.; Proakis, J.G.; Stojanovic, M. Adaptive OFDM modulation for underwater acoustic communications: Design considerations and experimental results. *IEEE J. Ocean. Eng.* **2014**, *39*, 357–370. [[CrossRef](#)]
6. Wan, L.; Zhou, H.; Xu, X.; Huang, Y.; Zhou, S.; Shi, Z.; Cui, J.H. Adaptive modulation and coding for underwater acoustic OFDM. *IEEE J. Ocean. Eng.* **2015**, *40*, 327–336. [[CrossRef](#)]
7. Barua, S.; Rong, Y.; Nordholm, S.; Chen, P. Adaptive modulation for underwater acoustic OFDM communication. In Proceedings of the OCEANS 2019-Marseille, Marseille, France, 17–20 June 2019; pp. 1–5.
8. Chao, L.; Franck, M.; Fan, Y. Demodulation of chaos phase modulation spread spectrum signals using machine learning methods and its evaluation for underwater acoustic communication. *Sensors* **2018**, *18*, 4217.
9. Yang, W.; Yang, T.C. High-frequency channel characterization for M-ary frequency-shift-keying underwater acoustic communications. *J. Acoust. Soc. Am.* **2006**, *120*, 2615–2626. [[CrossRef](#)]
10. Yang, T.C. Spatially multiplexed CDMA multiuser underwater acoustic communications. *IEEE J. Ocean. Eng.* **2016**, *41*, 217–231. [[CrossRef](#)]
11. Yin, J.; Yang, G.; Huang, D.; Jin, L.; Guo, Q. Blind adaptive multi-user detection for under-ice acoustic communications with mobile interfering users. *J. Acoust. Soc. Am.* **2017**, *141*, 70–75. [[CrossRef](#)]
12. Ma, L.; Zhou, S.; Qiao, G.; Liu, S.; Zhou, F. Superposition coding for downlink underwater acoustic OFDM. *IEEE J. Ocean. Eng.* **2017**, *42*, 175–187. [[CrossRef](#)]
13. Tao, J. DFT-precoded MIMO OFDM underwater acoustic communications. *IEEE J. Ocean. Eng.* **2018**, *43*, 805–819. [[CrossRef](#)]
14. Qiao, G.; Song, Q.; Ma, L.; Wan, L. A low-complexity orthogonal matching pursuit based channel estimation method for time-varying underwater acoustic OFDM systems. *Appl. Acoust.* **2019**, *148*, 246–250. [[CrossRef](#)]
15. Roudsari, H.; Bousquet, J. A time-varying filter for doppler compensation applied to underwater acoustic OFDM. *Sensors* **2018**, *19*, 105. [[CrossRef](#)]
16. Bai, Y.; Bouvet, P.J. Orthogonal chirp division multiplexing for underwater acoustic communication. *Sensors* **2018**, *18*, 3815. [[CrossRef](#)] [[PubMed](#)]
17. Xi, J.; Yan, S.; Xu, L. Direct-adaptation based bidirectional turbo equalization for underwater acoustic communications: Algorithm and undersea experimental results. *J. Acoust. Soc. Am.* **2018**, *143*, 2715–2728. [[CrossRef](#)] [[PubMed](#)]
18. Yin, J.; Ge, W.; Han, X.; Guo, L. Frequency-domain equalization with interference rejection combining for single carrier multiple-input multiple-output underwater acoustic communications. *J. Acoust. Soc. Am.* **2020**, *147*, EL138–EL143. [[CrossRef](#)]
19. Qin, X.; Qu, F.; Zheng, Y.R. Bayesian iterative channel estimation and turbo equalization for multiple-input-multiple-output underwater acoustic communications. *IEEE J. Ocean. Eng.* **2021**, *46*, 326–337. [[CrossRef](#)]
20. Wang, S.; Liu, M.; Li, D. Bayesian learning-based clustered-sparse channel estimation for time-varying underwater acoustic OFDM communication. *Sensors* **2021**, *21*, 4889. [[CrossRef](#)] [[PubMed](#)]
21. Sun, L.; Wang, M.; Zhang, G.; Li, H.; Huang, L. Filtered multitone modulation underwater acoustic communications using low-complexity channel-estimation-based MMSE turbo equalization. *Sensors* **2019**, *19*, 2714. [[CrossRef](#)] [[PubMed](#)]



22. Cen, Y.; Liu, M.; Li, D.; Meng, K.; Xu, H. Double-scale adaptive transmission in time-varying channel for underwater acoustic sensor networks. *Sensors* **2021**, *21*, 2252. [[CrossRef](#)] [[PubMed](#)]
23. Berger, C.R.; Zhou, S.; Preisig, J.C.; Willett, P. Sparse channel estimation for multicarrier underwater acoustic communication: From subspace methods to compressed sensing. *IEEE Trans. Signal Process.* **2010**, *58*, 1708–1721. [[CrossRef](#)]
24. Huang, J.; Zhou, S.; Huang, J.; Berger, C.R.; Willett, P. Progressive inter-carrier interference equalization for OFDM transmission over time-varying underwater acoustic channels. *IEEE J. Sel. Top. Signal Process.* **2011**, *5*, 1524–1536. [[CrossRef](#)]
25. Wang, Z.; Zhou, S.; Preisig, J.C.; Pattipati, K.R.; Willett, P. Clustered adaptation for estimation of time-varying underwater acoustic channels. *IEEE Trans. Signal Process.* **2012**, *66*, 3079–3091. [[CrossRef](#)]
26. Tadayon, A.; Stojanovic, M. Iterative sparse channel estimation and spatial correlation learning for multichannel acoustic OFDM systems. *IEEE J. Ocean. Eng.* **2019**, *44*, 820–836. [[CrossRef](#)]
27. Martins, N.E.; Jesus, S.M. Blind estimation of the ocean acoustic channel by time-frequency processing. *IEEE J. Ocean. Eng.* **2006**, *31*, 646–656. [[CrossRef](#)]
28. Cai, J.; Su, W.; Zhang, S.; Chen, K.; Wang, D. A semi-blind joint channel estimation and equalization single carrier coherent underwater acoustic communication receiver. In Proceedings of the OCEANS 2016-Shanghai, Shanghai, China, 10–13 April 2016; pp. 1–6.
29. Yang, G.; Guo, Q.; Ding, H.; Yan, Q.; Huang, D. Joint message-passing-based bidirectional channel estimation and equalization with superimposed training for underwater acoustic communications. *IEEE J. Ocean. Eng.* **2021**, *46*, 1463–1476. [[CrossRef](#)]
30. Guo, Q.; Ping, L.; Huang, D. A low-complexity iterative channel estimation and detection technique for doubly selective channels. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 4340–4349.
31. Pelekanakis, K.; Chitre, M. Robust equalization of mobile underwater acoustic channels. *IEEE J. Ocean. Eng.* **2015**, *40*, 775–784. [[CrossRef](#)]
32. Orozco-Lugo, A.G.; Lara, M.M.; McLernon, D.C. Channel estimation using implicit training. *IEEE Trans. Signal Process.* **2004**, *52*, 240–254. [[CrossRef](#)]
33. Guo, Q.; Huang, D. A concise representation for the soft-in soft-out LMMSE detector. *IEEE Commun. Lett.* **2011**, *15*, 566–568.
34. Guo, Q.; Huang, D.; Nordholm, S.; Xi, J.; Yu, Y. Iterative frequency domain equalization with generalized approximate message passing. *IEEE Signal Process. Lett.* **2013**, *20*, 559–562. [[CrossRef](#)]