

Article

Demodulation of Chaos Phase Modulation Spread Spectrum Signals Using Machine Learning Methods and Its Evaluation for Underwater Acoustic Communication

Chao Li ^{1,2,*} , Franck Marzani ³  and Fan Yang ³

¹ State Key Laboratory of Acoustics, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China

² University of Chinese Academy of Sciences, Beijing 100190, China

³ LE2I EA7508, Université Bourgogne Franche-Comté, 21078 Dijon, France; Franck.Marzani@u-bourgogne.fr (F.M.); fanyang@u-bourgogne.fr (F.Y.)

* Correspondence: chao.li.1986@ieee.org

Received: 25 September 2018; Accepted: 28 November 2018; Published: 1 December 2018



Abstract: The chaos phase modulation sequences consist of complex sequences with a constant envelope, which has recently been used for direct-sequence spread spectrum underwater acoustic communication. It is considered an ideal spreading code for its benefits in terms of large code resource quantity, nice correlation characteristics and high security. However, demodulating this underwater communication signal is a challenging job due to complex underwater environments. This paper addresses this problem as a target classification task and conceives a machine learning-based demodulation scheme. The proposed solution is implemented and optimized on a multi-core center processing unit (CPU) platform, then evaluated with replay simulation datasets. In the experiments, time variation, multi-path effect, propagation loss and random noise were considered as distortions. According to the results, compared to the reference algorithms, our method has greater reliability with better temporal efficiency performance.

Keywords: underwater acoustic communication; direct sequence spread spectrum; chaos phase modulation sequence; partial least square regression; machine learning

1. Introduction

The underwater acoustic communication has always been a crucial research topic [1–6]. In some special applications, high-performance communication is required, resulting that the system must satisfy the constraints of reliability, security and efficiency simultaneously. The direct-sequence spread spectrum (DSSS) communication is one of the most effective potential solutions to the problem of confidential underwater acoustic communication. It spreads the frequency spectrum of the carrier wave with a spreading code sequence, so the modulated signal is hard to be detected by a third party within the underwater noise, possessing greater concealment [7,8].

Conventional DSSS technique is used to modulating the carrier wave with pseudo-noise (PN) sequences, such as m-, Gold and Kasami sequences [9–11]. However, these PN sequences can only provide finite keyings and limited code resources, (e.g., binary phase-shift modulation keying, quadrature amplitude modulation, frequency shift keying), so the transmitted DSSS signals usually possess binary-value and periodic characters. Estimating the modulation parameters from the signals of this type is not very hard even with the blind methods [12–15]. For example, Figure 1a plots

the power spectrum density reprocessing S_{rep} of the m-sequence spread spectrum signal computed as follows:

$$S_{rep}(g) = |\mathcal{F}[S(f)]|^2 = \left| \int_{-\infty}^{+\infty} S(f) e^{-i2\pi gf} df \right|^2 \quad (1)$$

where S is the power spectrum of the received signal, and \mathcal{F} refers to the Fourier Transform. Due to the periodicity of m-sequence, regular spikes occur at the integer multiple of the sequence period along the time axis, the period of the transmitted signals can be, therefore, accurately estimated by measuring the interval width between two spikes [15].

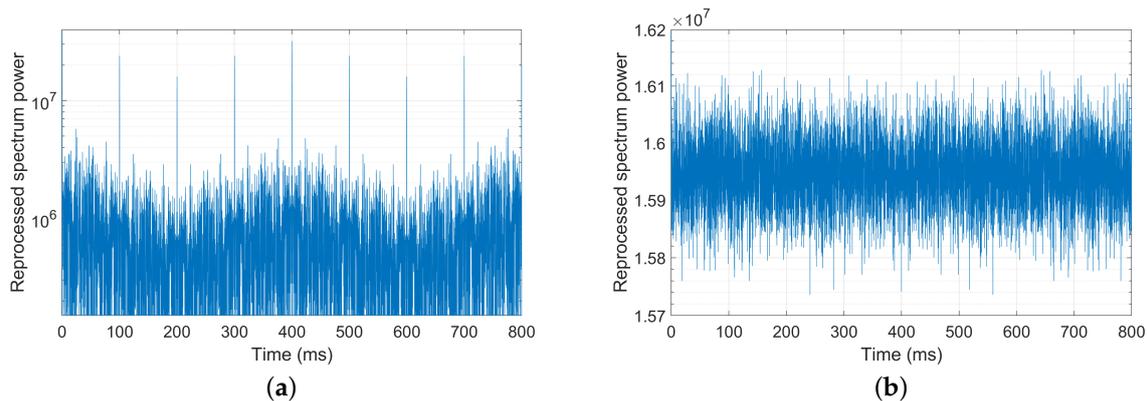


Figure 1. Period estimations of m- and chaos sequence spread spectrum signals with power spectrum density (PSD) reprocessing [15]. (a) m-sequence; (b) Chaos sequence.

Over the recent twenty years, chaos theory has been incorporated into the conceptions of PN sequences for the benefits of broad band, irregular complexity and orthogonality [16–18]. Usually, chaos (deterministic chaos) refers to irregular, unpredictable behavior in deterministic, dissipative, and nonlinear dynamical systems [19]. Within DSSS, it can be considered as a kind of disorder sensitive to initial conditions, allowing to generate a large number of orthogonal chaos sequences by setting different initial values. This method is helpful to overcome the binary-value and periodic problems of old code sequences. As the simulation results shown in Figure 1b, adopting chaos sequence significantly increases the difficulties of the parameter retrieving. Up till now, a series of ingenious chaos maps have been proposed (e.g., logistic [20], tent [21], cubic [22], Chebyshev [23], Bernoulli [24], multi-segment piecewise linear maps [25]), considerably improving communication security.

Despite of multiple advantages, adopting chaos spread spectrum into real-life underwater acoustic communication is far from easy. The de-spreading and demodulation of chaos spread spectrum signals at the receiver is challenged by the signal distortions caused by time and space variations of underwater environments, reverberation, multi-path effect, low signal-to-noise ratio, etc. [26]. Matched filter is one of the most widely used demodulators in underwater acoustic communication [27]:

$$s_{out}(t) = \int_{-\infty}^{+\infty} \mathbf{H}(f) S(f) e^{j2\pi ft} df \quad (2)$$

where s_{out} is the instantaneous output of the filter, \mathbf{H} is its transfer function and S is the spectrum density function of the given signal s . When the channel impairment is only the white gaussian noise, according to the Cauchy–Bunyakovsky–Schwarz inequality, the optimal matched filter is, therefore, $\mathbf{H}(f) = \alpha S^*(f) e^{-2\pi f t_0}$, where α and t_0 are, respectively, a constant value and the time delay, implying that its impulse response is actually the replica of s . The matched filters obviously do not have the ability to suppress the non-Gaussian-distributing noise and channel multi-path effects. In underwater communications, it is, therefore, usually combined with the equalization techniques or passive-phase conjugation method to mitigate the distortions from the channels of this type [7,28–30]. Stojanovic et al. successfully realized the deep-water 204-km 660-bps, shallow-water

89-km 1000-bps, and shallow-water 3.7-km 10-kbps underwater communications with equalization optimizations, respectively [28]. It is claimed that a passive-phase conjugation-based communication method achieved an average bit error ratio (BER) of 10^{-6} at $SNR = -5$ dB with simulations [7], and a successful communication with the distance of 900 km in a deep-water experiment [31].

Yet, unlike common acoustic signal detections which distinguish the interested signal from the ambient noise, chaos spread spectrum signal demodulation is a recognition task that is much more complicated. It has to find out the right class of the received symbol from a large candidate set, necessitating the target classification capacity via feature analysis. In our opinions, today's underwater sound communication techniques can be further improved from the following two aspects:

- High-performance symbol classification method. Though the conventional matched filters are able to enhance the symbol energy over noise by providing a processing gain with maximal SNRs, all of the components of the symbol sequences or vectors are unintentionally considered to contribute equally to the model precision without feature analysis, resulting in values of the key components that are diluted by the others. Hence, there are still some opportunities to further improve the accuracy performance of the demodulators through feature analysis.
- Noise analysis. Submarine noise is one of the main interferences with the quality of underwater communication. They come from a variety of sources, such as marine life, sea quakes, rain, artificial constructions, etc. The ambient noise of transducers are highly random, and the analysis of their statistical distribution feature helps potentially to improve the reliability of underwater communication modalities.

For the issues mentioned above, we focus our work on the investigation of machine learning-based chaos spread spectrum signal demodulation techniques. It is motivated by the fact that machine learning methods handle the classification problem by making data-driven predictions or decisions through building a statistical model from sample inputs. It provides a nice solution to statistically analyze the data features from the given sample sets. Because this procedure is automatic, it is especially appropriate to establish the physical models which are hard to be described using a precise analytic function. Within underwater communications, the received signals are distorted from the transmitted signals via the underwater sound channels with random noise interferences, so it can be considered that they contain all of the necessary factors of the propagation model. Learning directly from the original data with a proper approach may help to build a precise model to benefit underwater communication techniques from signal and environment feature analysis.

This paper is devoted to the underwater acoustic communication scheme. A new proposed chaos phase modulation method for underwater communication [32] is used to modulate the communication symbols. The chaos phase modulation method first arose in the 1990s [33,34] and was then used in radar systems [35,36] and high-performance multi-channel secure underwater communications [32]. With chaos phase modulation techniques, a synchronized copy of the chaotic symbol signal is generated at the receiver side and the demodulation is performed by exploiting this replica in different methods to recover the transmitted data, which is known as a coherent communication system [37]. Compared to the nonherent systems, such as differential chaos shift keying, chaos phase modulation does not have the problem of low data transmission rate, in which half of bit energy is used to transmit the reference signal [38,39]. Because it modulates the phases of the carrier wave with chaos sequence, which can provide a huge number of modulation keyings and code resources, it can also overcome the problem of periodicity and binary values caused by conventional PN sequences. Furthermore, the signals of this type have nice orthogonality, providing high distinguishability between symbols, which can potentially facilitate the demodulation task at the receiver end.

This paper innovatively processes the demodulation of chaos phase modulation signals as a target recognition problem, so it can be solved using either matched filter or machine learning methods. For the purpose of low bit error rate, a partial least square (PLS) regression-based demodulation framework is conceived. PLS algorithm is initially a standard tool for processing a wide spectrum of chemical data problems. Its success in chemometrics resulted in a lot of applications in other

scientific areas including bioinformatics [40], food research [41], medicine [42], pharmacology [43], social sciences [44], etc. Recently, this regression method has been successfully used in hyperspectral image recognition applications in the biometric field [45,46], which effectively improve the test accuracy by modeling the relations between training and prediction matrices. The use of PLS regression in chaos phase modulation communications allows us to weigh the received discrete signal through statistical analyses, further raising the accuracy of the system.

The proposed demodulation scheme is implemented with a multi-core center processing unit (CPU). The evaluation experiments of this paper are conducted using simulation datasets, within which the time variation, multi-path effect, propagation loss and random noise are considered as distortions. In order to obtain unbiased results, two reference algorithms are implemented with different optimization forms. The results demonstrate that the proposed method achieves the best accuracy performance with high temporal efficiency within the experimental protocol of this paper.

The remainder of this paper is organized as follows: Section 2 describes the chaos phase modulation scheme; Sections 3 and 4 present the proposed demodulation method and its parallel implementation, respectively; Section 5 analyzes the evaluation experiment results; finally, a conclusion is given in Section 6.

2. Message Signal Modulation

This paper modulates the phases of the communication symbols directly on the complex exponential function [32]:

$$p_l = \exp\{j2\pi k_l\} \quad (3)$$

where $k_l \in [-0.5, 0.5]$ is the l -th element of chaos sequence. We can see that chaos phase modulation sequences are complex sequences with randomly-distributed phases, and their envelopes are constant.

We use the multi-segment piecewise linear mapping [25,47] to generate the desired chaos sequence k . It is an iteration procedure that can be programmatically expressed as follows:

- (1) Initialize the first element of the sequence k_0 with a random value, known as "seed" in computer science;
- (2) If $k_l > 1/2$, repeat the computation $k_l := k_l - |2\zeta k_l|$ with $\zeta \in [0, 1]$ until the constraint is satisfied, otherwise, go to the next step;
- (3) If $k_l < -1/2$, repeat the computation $k_l := k_l + |2\zeta k_l|$ until the constraint is satisfied, otherwise, go to the next step;
- (4) Compute the next element k_{l+1} :

$$k_{l+1} = \begin{cases} 2(N - \zeta)k_l + (N - 0.5) & -1/2 \leq k_l < (1 - N)/(2N) \\ 2(N - \zeta)k_l + (N - 1.5) & (1 - N)/(2N) \leq k_l < (2 - N)/(2N) \\ \vdots & \\ 2(N - \zeta)k_l + (-N + 0.5) & (N - 1)/(2N) \leq k_l < 1/2 \end{cases} \quad (4)$$

- (5) If $l = L - 1$, task completed, otherwise go back to the second step.

Steps (2)–(4) above correct iteratively k_l to keep it in the given interval with the help of a step coefficient $\zeta \in [0, 1]$. L is the length of sequence k . Step (4) segments the interval $[-0.5, 0.5]$ into $2N$ pieces ($N \in \mathbb{N}^+$). Now we can create the chaos spreading spectrum symbol $u(t)$:

$$u(t) = \sum_{l=0}^{L-1} d(t) \times g(t, l, p_l) \quad (5)$$

with

$$\mathbf{d}(t) = A \cdot \frac{e^{j2\pi f_0 t} + e^{-j2\pi f_0 t}}{2} \quad (6)$$

$$g(t, l, \mathbf{p}_l) = \begin{cases} \mathbf{p}_l & \text{if } t \in (\frac{T \cdot l}{L}, \frac{T \cdot (l+1)}{L}] \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where $\mathbf{d}(t)$ is the carrier wave, A is amplitude, f_0 is the frequency of carrier wave, g is a shaping function, $T = T_c \times L$ is the time length of symbol and T_c is the time length of chip. In this work, cosine signal is used as the carrier wave.

We can write the fourier transform of $\mathbf{u}(t)$ as follows:

$$\mathcal{F}[\mathbf{u}(t)] = \sum_{l=0}^{L-1} \int_{-\infty}^{+\infty} \mathbf{d}(t) \times g(t, l, \mathbf{p}_l) \times e^{-i2\pi f t} dt \quad (8)$$

$$= \sum_{l=0}^{L-1} \int_{\frac{T \cdot l}{L}}^{\frac{T \cdot (l+1)}{L}} \mathbf{d}(t) \times e^{i2\pi(k_l - f)t} dt \quad (9)$$

The equation above implies that chaos phase modulation associates the frequency spectrum of $\mathbf{u}(t)$ with the chaos sequence \mathbf{k} . According to the chaos theory, chaos sequences are sensitive to the initial conditions, even if they are approximately equal. Consequently, correlators will have a low output unless they are highly synchronized with the inputs. That allows to create a large number of highly orthogonal symbol candidates by changing the seeds.

Figure 2a plots the normalized autocorrelation of a chaos phase modulation signal measured by using a classical matched filter (see Equation (2)). The correlation values of main and side lobes are marked. We can see clearly that their ratio is around $4.1 \times$, whereas, as shown in Figure 2b, it is hard to distinguish the main and side lobes on the cross-correlation curves, and the maximum cross-correlation coefficient is only 0.217, demonstrating that chaos phase modulation signals have nice orthogonality.

For the purpose of textual communication, we map the 8-bit ASCII table to chaos phase modulation signals. Each ASCII code is represented by using a symbol with certain time length, performing a symbol candidate set for message edition at the transmitter and signal demodulation at the receiver.

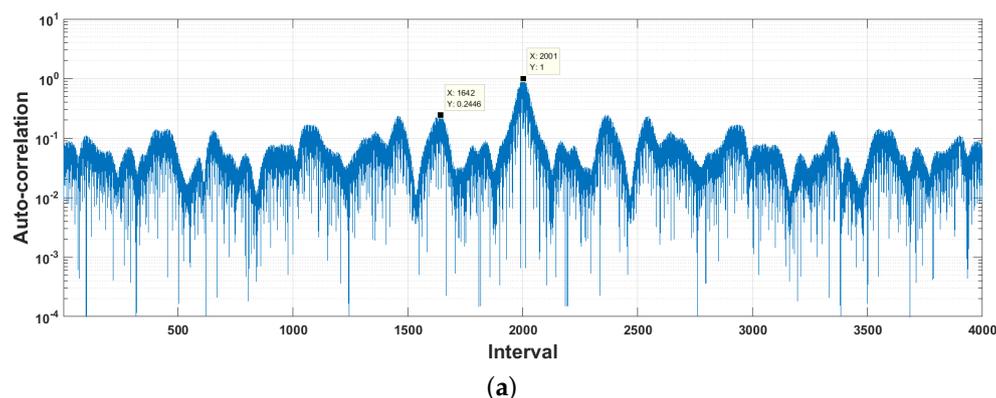


Figure 2. Cont.

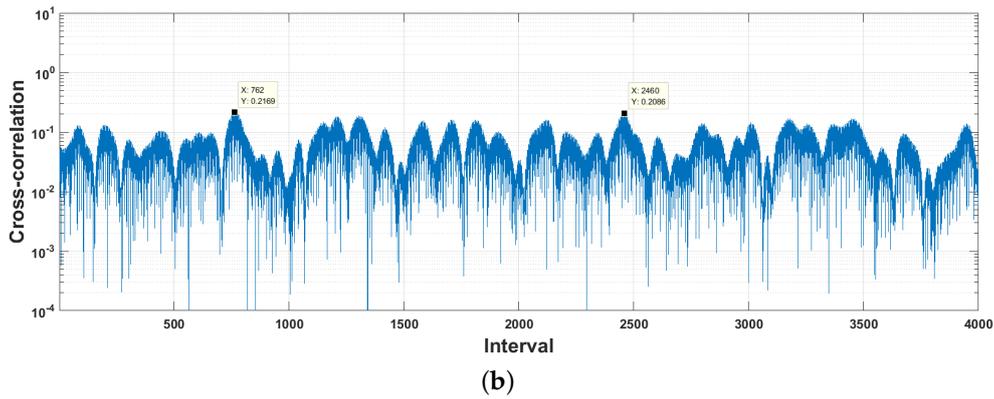


Figure 2. Correlation of chaos phase modulation signals. (a) Autocorrelation; (b) Cross-correlation.

3. Signal Demodulation

We consider the demodulation problem as a symbol recognition one. This section describes the proposed demodulation scheme at the receiver.

3.1. Architecture of Demodulator

Figure 3 shows the demodulation flowchart of this paper. The input signal is first resampled and pre-processed via a band pass filter, then reshaped into the symbol matrix X_{in} . The rows of X_{in} correspond to the received symbols u'_m , and the columns to the sampling points. Thirdly, u'_m are associated to the units of a detector array, respectively. The units of detector array are essentially binary classifiers, which score the similarity degree between u'_m and the symbol candidate u_n . The output of the detector array is a score matrix denoted using Y_{out} , whose rows correspond to the symbol candidates and the columns to the received symbols. The element of the score matrix a_{nm} represents the similarity between u_n and u'_m . Finally, the transmitted message is recovered through a decision function and ASCII code mapping. This paper supposes that a higher score means better similarity; the decision function, therefore, returns the indices of the maxima of the columns of Y_{out} :

$$M_m = \max\{a_m^T\} \tag{10}$$

where $a_m^T = [a_{1m}, a_{2m}, \dots, a_{nm}, \dots]$.

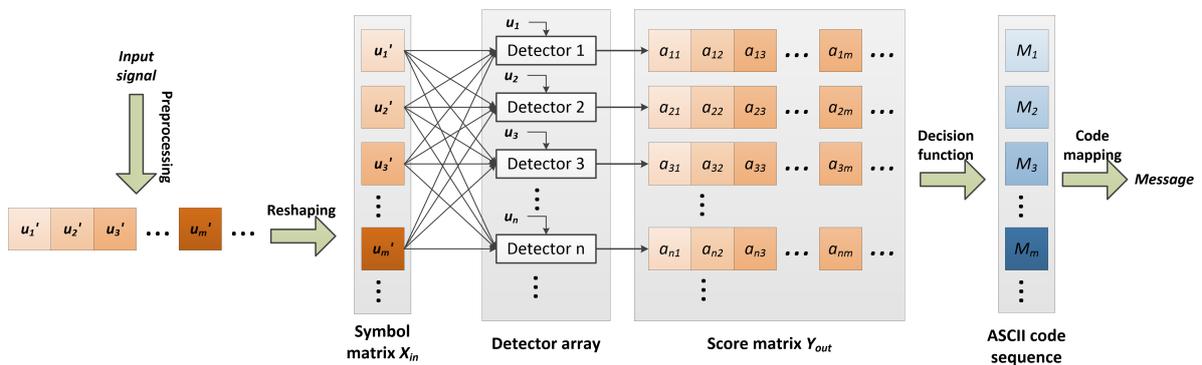


Figure 3. Overall flowchart of demodulation.

3.2. PLS-Based Detection Method

Signal detection plays an important role in the low-BER acoustic communications. One of the most frequently used acoustic signal detection methods is the matched filter (see Equation (2)). The underwater sound channels and ambient noise usually distort the message signals seriously,

resulting in marked degradation in signal detection performance. Figure 4a plots the autocorrelation between the original chaos phase modulation signal and its observation at the receiver simulated by using the sound channel response estimated from the trail data. We can see that the main-to-side lobe ratio goes from $4.1\times$ (see Figure 2) down to lower than $1.94\times$. Whereas, the maximum cross-correlation increases from 0.217 up to 0.42, see Figure 4b.

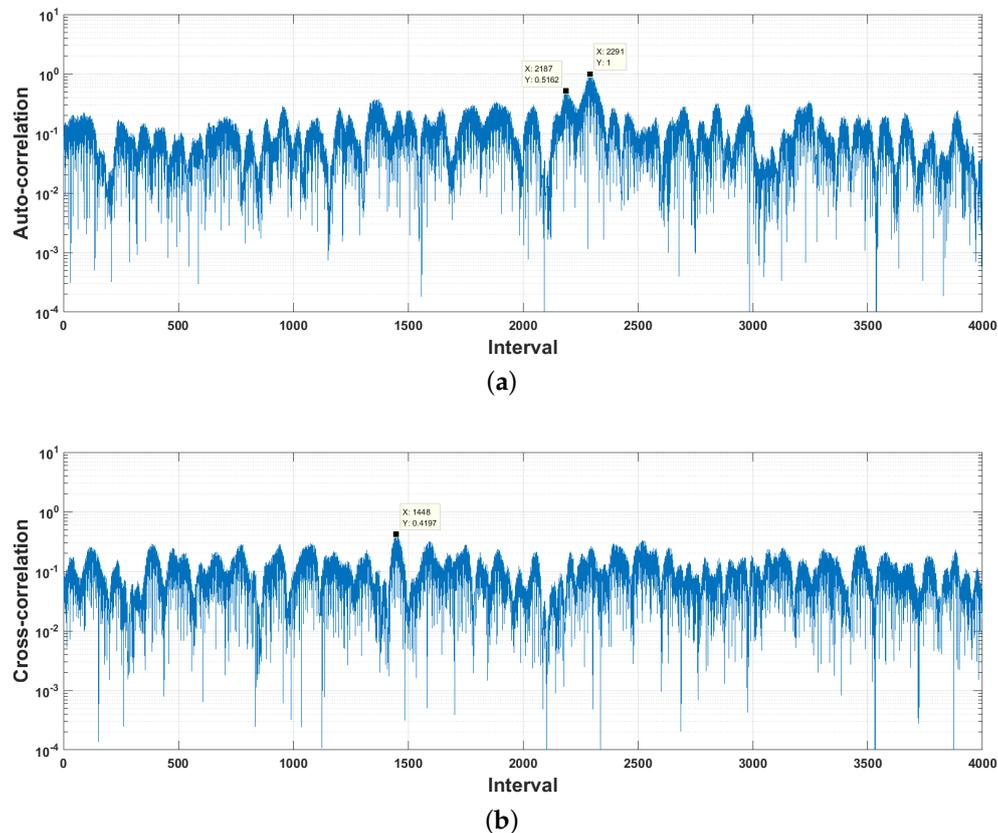


Figure 4. Correlation between the original chaos phase modulation signal and its underwater sound channel output at $SNR = -10$ dB. (a) Autocorrelation; (b) Cross-correlation.

The simulations above demonstrate that the distinguishability of the received chaos signals is reduced significantly during the underwater propagation. For the purpose of high accuracy performance, we use a partial-least square (PLS) regression-based classifier as the detector array in Figure 3.

PLS is a wide class of methods for modeling relations between sets of observed variables by means of latent variables. It comprises regression and classification tasks, as well as dimension reduction techniques and modeling tools. Projection of the observed data to its latent structure by means of PLS was developed by Herman Wold and coworkers [48–50]. The underlying assumption of all PLS methods is that the observed data is generated by a system or process which is driven by a small number of latent (not directly observed or measured) variables. Its goal is to maximize the covariance between the two parts of a paired dataset even though those two parts are in different spaces. This method is essentially a machine learning one, which necessitates the training and testing processes, simultaneously.

3.2.1. Training Process of PLS Regression

Let x be a random example for training and y its response. Either x or y is zero-mean column vectors. In order to assess their relation with covariance, we project them onto two separate directions

specified by unit vectors w_x and w_y in order to obtain two random variables $w_x^T x$ and $w_y^T y$. Now we present two matrices X and Y whose i -th rows are the feature vectors of corresponding examples x_i and y_i . According to the nonlinear iterative PLS algorithm, PLS searches for the directions w_x and w_y such that [45,51]

$$\max_{w_x, w_y: \|w_x\|=\|w_y\|=1} \mathbb{C}(w_x, w_y) = w_x^T C_{xy} w_y = \frac{1}{m} w_x^T X^T Y w_y \quad (11)$$

where $C_{xy} = \frac{1}{m} X^T Y$ is the covariance matrix of X and Y , and m is the example number. The direction that solve the maximal covariance optimization are the first singular vectors $w_x = U_1$ and $w_y = V_1$ of the singular value decomposition of C_{xy} :

$$C_{xy} = U \Sigma V^T \quad (12)$$

where the value of the covariance is given by the corresponding singular value σ_1 .

In this paper, more than one projection direction are wanted. To do this, the same strategy of data projecting is applied by deflation. This creates the new data matrix

$$X' = X(I - w_x w_x^T) \quad (13)$$

Let P_j be $\frac{X_j^T X_j U_j}{X_j^T U_j^T U_j X_j}$ with $j = 1, 2, \dots, k$, where k is the projection direction number, the deflation of $X = X_1$ is obtained as follows:

$$X_2 = X_1(I - U_1 P_1^T), \text{ with } X_1 = X \quad (14)$$

Considering a test point with the feature vector $\varphi(x)$, we define $\tilde{\varphi}(x)$ as the feature vector needed for the regression, whose columns are $\varphi_j(x)^T U_j$. The new feature vector can be expressed as follows:

$$\tilde{\varphi}(x)^T = \varphi(x)^T \tilde{U} (P^T \tilde{U})^{-1} \quad (15)$$

where $\tilde{U} = [U_1, U_2, U_3, \dots, U_j]$.

Now, we can start to compute the regression coefficients vector W , which performs the regression of the variables Y in terms of $X \tilde{U}$. We seek a coefficient matrix B that solves the optimization

$$\min_B \|X \tilde{U} B - Y\|^2 = \min_B \langle X \tilde{U} B - Y, X \tilde{U} B - Y \rangle \quad (16)$$

The final regression coefficients W are given by $\tilde{U} B$. We seek the minimum by computing the gradient with respect to B and setting it to zero. The overall regression coefficients can be computed as follows:

$$W = \tilde{U} (P^T \tilde{U})^{-1} C^T \quad (17)$$

where C is the matrix with columns $c_j = \frac{Y^T X_j U_j}{U_j^T X_j^T U_j}$.

The PLS regression algorithm is shown in Algorithm 1. The repeat loop computes the first singular value by the iterative method. We can train the PLS model by assigning the training matrix X_{train} and its responses Y_{train}^T to X and Y , respectively. The element $Y_{train}(n, m)$ of Y_{train} is 1 if the m -th received symbol u'_m is matched with the n -th symbol candidate u_n and 0 otherwise. For a training matrix having 4 symbol candidates and two received symbols per candidate, Y_{train} is as follows:

$$Y_{train} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \quad (18)$$

Algorithm 1 Pseudocode of PLS Regression Algorithm.**Input:** training matrix X , response variables Y , projection direction number k **Output:** regression coefficients W

```

1: centering the data
2: for  $j = 1, 2, \dots, k$  do
3:    $U_j \leftarrow$  first column of  $X_j^T Y$ 
4:    $u_j \leftarrow U_j / \|U_j\|$ 
5:   repeat
6:      $U_j \leftarrow X_j^T Y Y^T X_j U_j$ 
7:      $u_j \leftarrow U_j / \|U_j\|$ 
8:   until convergence
9:    $P_j \leftarrow X_j^T X_j U_j / (U_j^T X_j^T X_j U_j)$ 
10:   $c_j \leftarrow Y^T X_j U_j / (U_j^T X_j^T X_j U_j)$ 
11:   $X_{j+1} \leftarrow X_j (I - U_j P_j^T)$ 
12: end for
13:  $W \leftarrow \tilde{U} (P^T \tilde{U})^{-1} C^T$ 

```

3.2.2. Demodulation with PLS Regression

The symbol matrix to be demodulated X_{in} is formed of the normalized received symbol \bar{u}'_m :

$$\bar{u}'_m = \frac{u'_m - \mu}{\sigma} \quad (19)$$

where \bar{u}'_m is the normalized received symbol sequence and is stored as the rows of X_{in} , μ and σ are the mean and standard deviation of the training matrix X_{train} . We associate X_{in} to the well-trained PLS model to compute the score matrix Y_{out} :

$$Y_{out}^T = X_{in} \times W \quad (20)$$

The similarity score $Y_{out}(n, m)$ of u'_m related to the n -th symbol candidate u_n is expected to be maximal if matched, otherwise 0. The index of the maximum of the score vector of u'_m (stored as the m -th column of Y_{out}) is considered as the estimated class of this received symbol.

4. Implementation and Optimization

The pseudocode of the demodulation algorithm of this paper is shown in Algorithm 2. In order to facilitate the description, the decision function in Figure 3 is inlined into the top function. After program initialization, the input signal is first filtered via an FIR filter for denoising, then segmented and reshaped into the form of the symbol matrix X_{in} through the loop in Line 3. Thirdly, the PLS regression function $PlsModel()$ is invoked. It returns the score matrix Y_{out} . Fourthly, $findmax()$ function searches for the column maxima Y_{out} , and return their indices, performing an index vector ind . Finally, ind is assigned to the message generation function $MesGen()$. The desired message string is generated by mapping the indices to the characters in the ASCII table.

Algorithm 2 Pseudocode of the Proposed Demodulation Algorithm.

Input: input signal s , frequency band f_b , normalization coefficients μ and σ , regression coefficients W , sampling rate f_s , symbol time length T , ASCII table $ascii$

Output: message str

```

1: initialization
2:  $s' \leftarrow fir(s, f_b)$ 
3: for  $i = 1, 2, \dots, m, \dots$  do
4:    $\bar{u}'_i \leftarrow [s'(((i-1) \times f_s \times T + 1) : (i \times f_s \times T)) - \mu] / \sigma$ 
5:    $X_{in}(i, :) \leftarrow \bar{u}'_i$ 
6: end for
7:  $Y_{out} \leftarrow PlsModel(X_{in}, W)$ 
8: for the  $i$ -th column of  $Y_{out}$ ,  $i = 1, 2, \dots, m, \dots$  do
9:    $ind(i) \leftarrow findmax(Y_{out}(i))$ 
10:   $str(i) \leftarrow MesGen(ind(i), ascii)$ 
11: end for

```

According to Algorithm 2, the proposed demodulation algorithm is lightweight and possesses high parallelism. From the view point of programming, the loop in Line 3 reshapes the input signal sequence s' without any computational operations. $PlsModel()$ function is essentially a matrix multiplier. The loop of Line 8 is an ideal independent loop, whose iterations can be pipelined or parallelized completely without initiation interval.

Hence, we optimized the original code for a multi-core CPU platform. Figure 5 shows the overall architecture of the optimized demodulation implementation. The CPU cores are interconnected with a shared bus. The first core of CPU masters the preprocessing, resampling, segmentation, reshaping tasks and PLS model. When these computations are finished, the score vectors stored in Y_{out} are distributed to the other cores via a shared bus for ASCII mapping. The mapping task consists of $findmax()$ and $MesGen()$ functions. We parallelize them with the single instruction multiple data (SIMD) approach, which enables us to apply the same description code (thread function) to process different data in parallel. The thread functions formed of $findmax()$ and $MesGen()$ functions are distributed to the different CPU cores which share the memories and data bus. Since the data or operations are independent, all of the threads are able to execute simultaneously so that the running speed is accelerated.

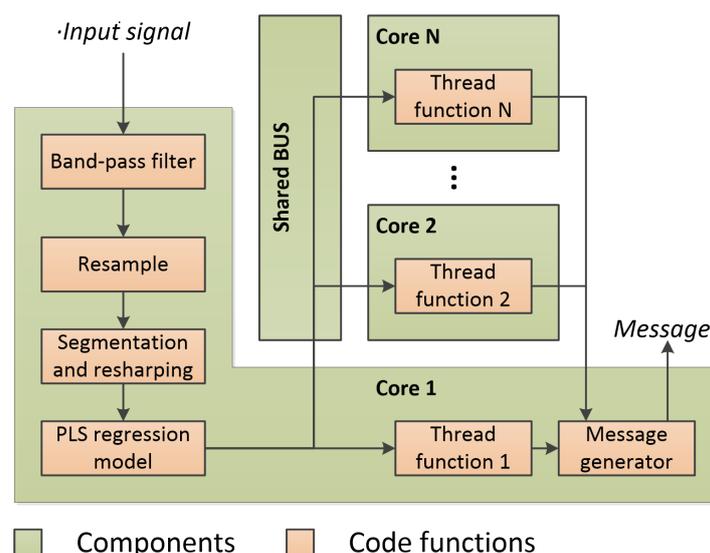


Figure 5. Architecture of the optimized demodulation implementation.

5. Experiments and Evaluations

The proposed method is evaluated with the simulation experiments. In order to obtain an unbiased result, all the used datasets are simulated with the data measured and recorded in the sea trail. The user-controlled parameters of the algorithm are optimized experimentally. The accurate and computational performance of the proposed method are evaluated by comparison with the two reference algorithms.

5.1. Dataset

The experiments of this paper are conducted by using simulation data, which are generated via the replay simulation method presented in [52]:

$$s(t) = \int_{\tau=0}^{T_h} h(t, \tau) \cdot s_o(t - \tau) d\tau + s_n(t) \quad (21)$$

where $s_o(t)$ is the transmitted signal, s is the received signal, h is the measured underwater sound channel response, T_h is the tap number of the channel response and s_n is the ambient noise.

Equation (21) allows to simulate time variation, multi-path effect, propagation loss and random noise occurred during the underwater acoustic propagations by specifying the channel response h and ambient noise s_n . The real-life underwater noise is used as the ambient noise in this work. The sound channel response is estimated from a set of experimental data recorded by the Institute of Acoustics of Chinese Academy of Sciences in the South China Sea. Figure 6 plots the smoothed sound velocity profiles at the sending and receiving sites. The acoustic signal was transmitted from a fixed source to a fixed receiver array. Table 1 displays the parameters of the experimental environment and channel estimation. The time coherence of the experimental environment is evaluated by measuring the normalized correlation of two 10-s chaotic signals with a time interval of 5 min. The result value of 0.82 demonstrates that the underwater sound channels are highly correlated during the measurement time.

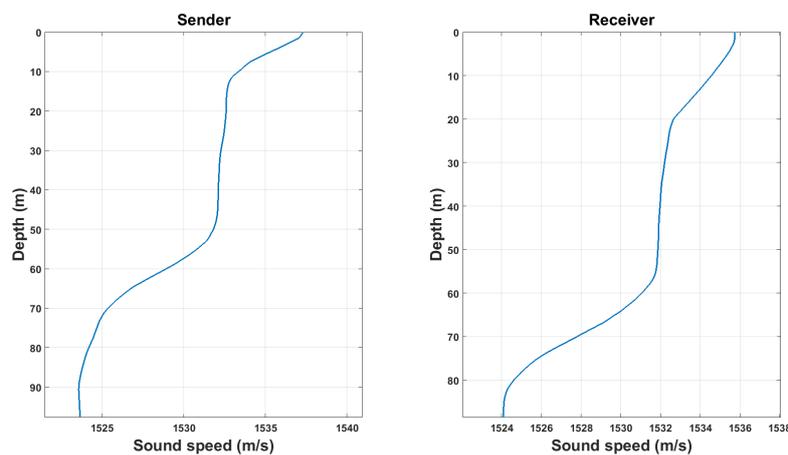


Figure 6. Smoothed sound velocity profiles at the sender (left) and receiver (right).

Table 1. Experimental parameters of underwater sound channel estimation.

Parameters	Values	Descriptions
Sea depth	90–100 m	Minimum and maximum sea depths during the propagation.
Distance	16.5 km	Both of the sender and receiver are stationary during the transmission.
Source level	200 dB	-
Impulse period	2 s	-
Impulse form	Chaotic	-
Sampling rate f_s	4000 Hz	-
Channel tap number W_h	400	$W_h = f_s \times 0.1$ s.
Channel estimation interval	10 ms	-

In order to obtain an accurate ocean channel, three frequently used channel estimation methods are evaluated for selection, including least mean square (LMS) [53], recursive least squares (RLS) [54] and least squares matching pursuit (LSMP) [55] algorithms. Figure 7 plots the mean square error (MSE) between the original channel response and the ones measured with the three methods over the signal-to-noise ratio. It is obtained with the simulated sparse channel response and spreading frequency signal. Finally, we selected LSMP for its high reliability and accuracy.

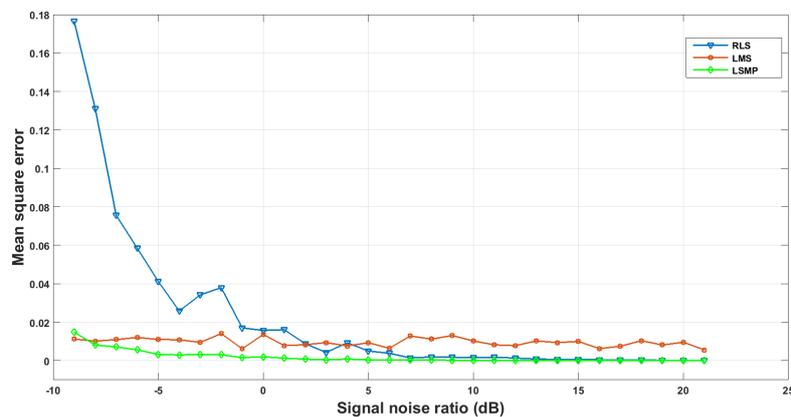
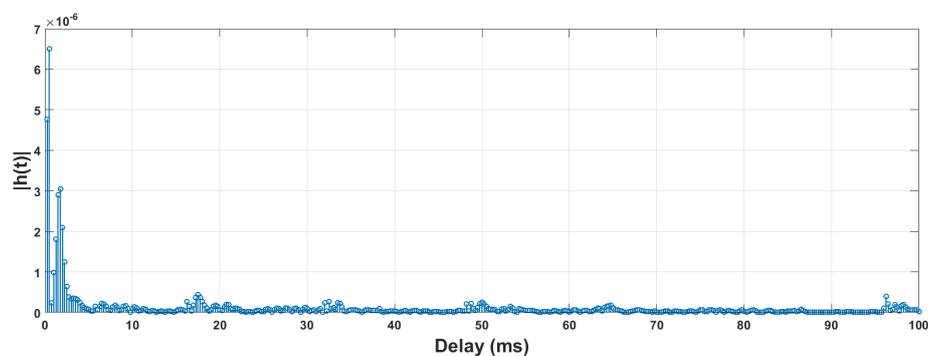
**Figure 7.** Accuracy comparison of channel estimation methods.

Figure 8 plots a channel response measured using the LSMP algorithm. It can be seen that the histogram possesses multiple peak points, representing the different multi-path arrivals [55]. We do not normalize it in order to keep the attenuation information. The normalized correlation coefficient between the original probe signal and the signals simulated with Equation (21) is 0.79.

**Figure 8.** An example of channel response measured using LSMP algorithm.

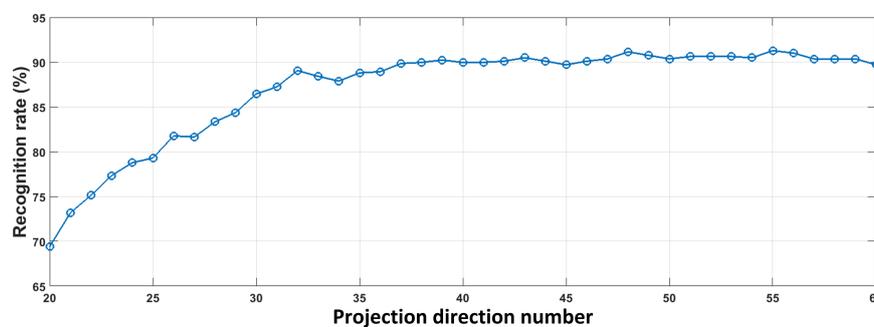
Now, we can start to generate the desired dataset with Equation (21). Given an 8-bit ASCII code, there are 256 symbol candidates u ; for each one, we generate six sessions of simulated signals u' . The dataset, therefore, contains $256 \text{ candidates} \times 6 \text{ sessions} = 1536$ simulated symbols.

5.2. Parameter Configuration

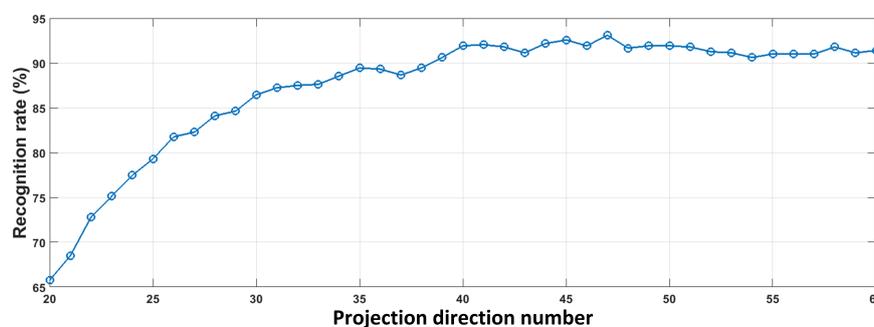
The subject of this experiment is to determine the following two parameters of the proposed algorithm: the time length of the symbol T and the projection direction number of PLS regression k . To do this, three versions of datasets are generated with different symbol time length: $T = 0.3$, 0.6 and 1 s. Next, the proposed method is evaluated with different projection direction numbers. 3 randomly-selected sessions are used for training and the rests for test. The final experiment results are computed by averaging the results of three independent executions.

We seek the optimal value of k by maximizing the recognition rate of the systems experimentally. The recognition rate is defined as the ratio of the correct classification counts and the number of input test symbols. This criterion has been widely used to evaluate the classification performance in pattern recognition field. Figure 9 plots the recognition rate curves of the three dataset versions over k . As expected, the recognition rate raises with the increasing of the k linearly at the beginning then slows down. It can be seen that the curves of $T = 0.3$ and $T = 0.6$ trend towards to stability when $k = 40$, whereas $T = 1$ when $k = 65$. Meanwhile, the optimal recognition rates are obtained when k is 42, 55 and 70 for the three dataset versions, respectively. That is, the optimal recognition rates are 90.36%, 92.58% and 99.74% when T is equal to 0.3, 0.6 and 1 s with $SNR = -8$ dB.

Additionally, it is found also that the dataset with larger symbol time length leads to better accuracy. This is because the larger T is, the more time gains can be achieved in the slow-time-varying environment. This experiment demonstrates that the symbols with the period of 1 s lead to an accuracy of 99.74%, to nearly 100%, in a low signal-to-noise ratio environment ($SNR = -8$ dB).



(a)



(b)

Figure 9. Cont.

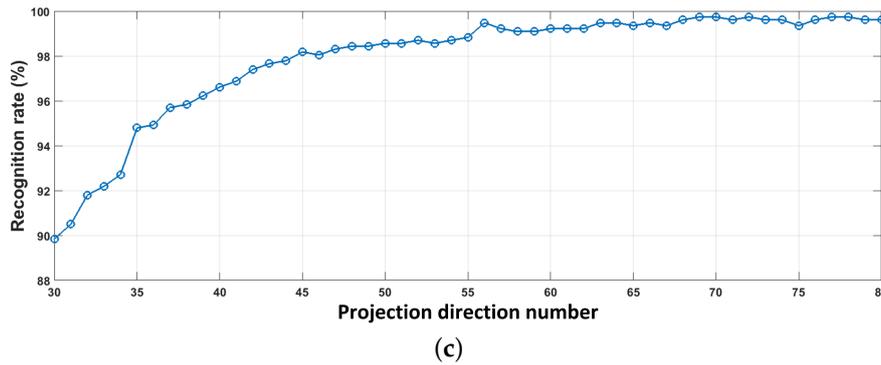


Figure 9. Recognition rate over the number of projection directions: $SNR = -8$ dB and $f_s = 4000$ Hz. (a) $T = 0.3$ s; (b) $T = 0.6$ s; (c) $T = 1$ s

5.3. Demodulation Accuracy Evaluation

The experiment of this subsection evaluates the accuracy performance of the proposed method by comparing it with two reference algorithms, including traditional matched filter [7] and a three-layer back propagation (BP) neural network [56]. Bit error rate is used as the criterion of accuracy performance. Both of the reference algorithms can be directly inserted into the scheme described in Section 3.1.

We compare the three demodulators with all of the three versions of the datasets: three of the six sessions for training, and the rests for testing. That is, for each data version, either the training or testing dataset has 256 symbol candidates \times three sessions = 768 simulated symbols. For the purpose of unbiased evaluation, the simulated symbols are randomly divided into two groups. The average bit error rate of three repetitions with different training and testing data is used as the evaluation criterion of the demodulation accuracy performance.

Figure 10 compares the accuracy performance between the proposed method and the reference algorithms with the three versions of datasets. First of all, we can see that PLS regression always possesses the lowest bit error rates. For the three dataset versions, when SNR is around -2 , -6 and -6.5 dB, the BERs of PLS reach 1.0%, whereas the matched filter and BP network are 22.24%, 36.05%, 6.44%, and 16.79%, 40.33%, 39.87%, respectively. Additionally, PLS achieves an error-free demodulation results with the three datasets at $SNR = 0$, -5 and -5 dB, out of the available range of the y-axis with logarithmic scale. This demonstrates that the proposed method has the best accuracy performance with the experiment protocol of this paper.

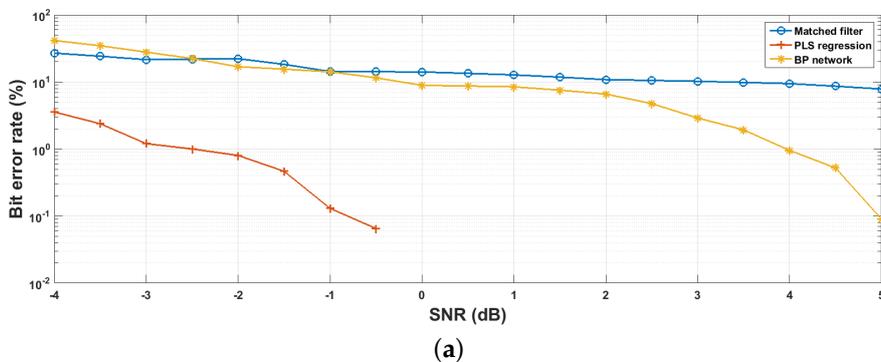


Figure 10. Cont.

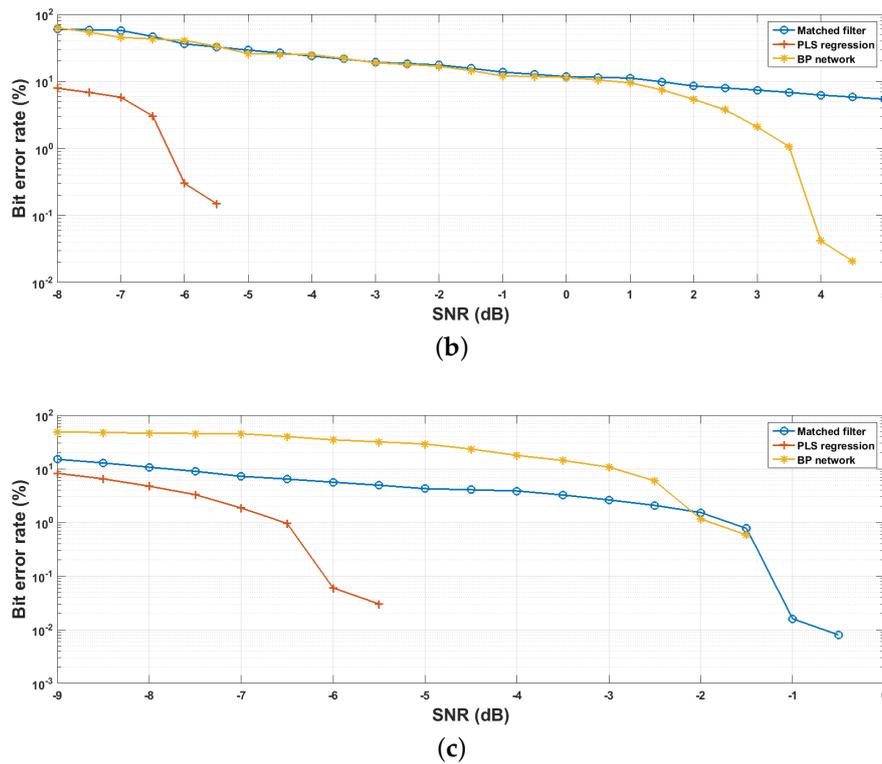


Figure 10. Accuracy comparison results with bit error ratios: $f_s = 4000$ Hz. (a) $T = 0.3$ s, $k = 40$; (b) $T = 0.6$ s, $k = 55$; (c) $T = 1$ s, $k = 70$.

In our case, the communication symbols are modulated with multi-segment piecewise linear mapping, which are essentially a set of pseudo-random sequences, implying that the predictors (elements of the symbol vector) are somehow correlated, resulting in a multicollinearity problem. The multicollinearity problem is a phenomenon where one predictor variable in a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy, resulting in model distortions [57]. Mathematically, a set of variables is perfectly multicollinear if one or more exact linear relationships exist among some of the variables:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_i x_i = 0 \quad (22)$$

where β_i are constants and x_i is the i -th element of the symbol vector u'_n . We quantify the multicollinearity of the simulated dataset by using condition indices (CIs) [58]:

$$CI_i = \sqrt{\frac{\lambda_{max}}{\lambda_i}} \quad (23)$$

where λ_{max} is the maximum eigenvalue of the symbol set, and λ_i is its i -th eigenvalue. Belsey et al. [59] suggest that when the value of CI is higher than 10, data dependencies are starting to affect the regression estimates. Figure 11 plots the condition indices of a 0.3-s symbol set, in which 30.42% of the elements are collinear with the others.

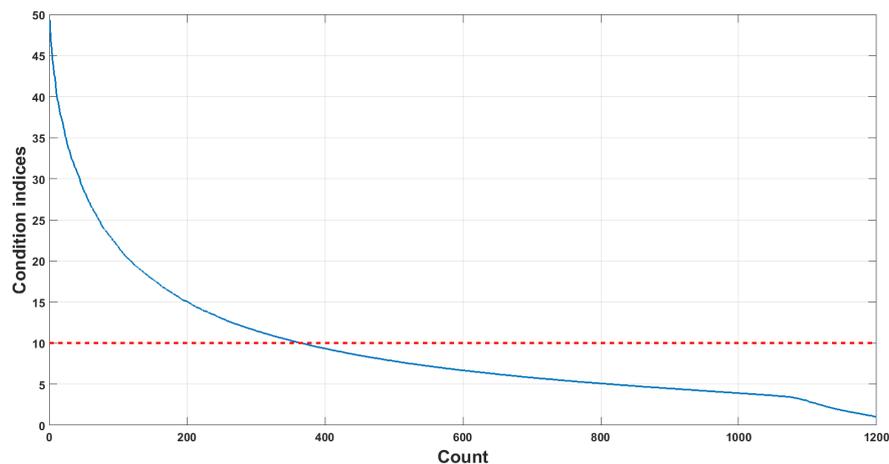


Figure 11. Condition indices of the 0.3-s symbol set.

Unlike the reference algorithms, PLS models the fundamental relations between the training and response matrices by modeling their covariance structure. More precisely, a PLS model will try to find the multidimensional direction in the space of training matrix that explains the maximum multidimensional variance direction in the space of response matrix. The PLS family of methods is, therefore, particularly suited for multicollinearity problem, providing better accuracy performance than the others.

As for the BP networks, the matched filters achieve better accuracy than when the SNR is low. In our opinion, this is caused by the fact that the power of the noise components is higher than the transmitted signals within low SNR environment ($SNR < 0$ dB), whereas the given training examples are too sparse, resulting that the BP networks are drowned by the strong ambient noise and can hardly capture the useful feature information even if they are well converged. Yet, if the signal-to-noise ratio exceeds certain thresholds (in the case of this evaluation they are -1 , 0 and -2 dB for the three datasets), the networks will provide better accuracy than matched filter. Additionally, the PLS models are trained by using the same datasets with the BP networks; it demonstrates, therefore, that incorporating variable relationship analysis into the regression problem can improve the capacity of the regressors in terms of handling the sparse training datasets.

Finally, the accuracy performance of the matched filters significantly rises with the increasing of the symbol time length T . For example, the bit error rates of the matched filters are around 14%, 11% and 0% at $SNR = 0$ dB. As mentioned in Section 5.2, the distinguishability of the symbols varies in function of its length due to the time gain. According to our further test, with the same experimental protocol, both matched filter and PLS regression achieve the 0% BER at $SNR = -5$ dB when $T = 1.5$ s. However, it should be noted that the communication rate is inversely proportional to T , so the method of this paper is more appropriate when requiring a high communication rate.

5.4. Practicability Analysis

Collecting high-density quality underwater acoustic data is technically difficult for today's acoustic engineering. In this subsection, the environment and sample quality compatibility is evaluated to investigate whether the training datasets could be sparse to a certain extent. The evaluations are performed by testing the well-trained PLS models using the datasets with strange underwater sound channels and different SNRs. To do this, we have 50% of the testing symbols simulated via strange channels (not included in the training channels). The time interval between the two channel measurements is around 5 minutes, and their normalized correlation is around 0.8. The evaluation result is shown in Table 2, in which the rows correspond to the training datasets and the columns to the testing datasets.

Table 2. BER (%) between the training and testing datasets with different underwater sound channel response and SNRs (dB): $T = 0.6$ s, $k = 55$ and $f_s = 4000$ Hz, the rows correspond to the training datasets and the columns to the testing datasets.

BER \ SNR	SNR	−10	−9	−8	−7	−6	−5	−4	−3	−2	−1	0
−10	−10	44.61	33.57	22.77	21.21	13.78	8.98	6.91	4.52	4.52	3.98	2.76
−9	−9	30.78	25.09	10.32	9.87	6.31	5.92	3.33	2.51	0.91	0.69	0.54
−8	−8	27.14	14.12	9.75	5.01	3.82	2.01	1.21	0.82	0.51	0.42	0.21
−7	−7	15.34	7.82	4.98	4.57	1.27	0.98	0.39	0.26	0.13	0.00	0.00
−6	−6	10.52	4.21	2.61	1.55	0.26	0.13	0.00	0.00	0.00	0.00	0.00
−5	−5	6.45	3.13	1.27	0.52	0.26	0.13	0.00	0.00	0.00	0.00	0.00
−4	−4	4.33	1.01	0.13	0.13	0.13	0.00	0.00	0.00	0.00	0.00	0.00
−3	−3	3.78	0.26	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
−2	−2	2.21	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
−1	−1	1.88	0.31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	0	0.91	0.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Comparing the diagonal of Table 2 with the simulation results of Figure 10b which trains and tests with the same sound channel, we can see that a strange channel somehow reduces the accuracy performance of the demodulator, but this impact is tiny, and the demodulator reaches the BER of 0% at $SNR = -4$ dB, which is still much better than either BP networks or matched filters. This demonstrates that the proposed method has the ability to resist the channel variant. Meanwhile, both of the SNRs of the training and testing datasets positively affect the accuracy of the demodulator. Poor training data do not necessarily result in high BERs, and vice versa. Hence, the performance of our demodulator is not seriously affected by the gap between the training and testing datasets, implying that the training data can be collected with few constraints.

5.5. Temporal Efficiency Evaluation

This subsection evaluates the temporal efficiency performance of the proposed demodulator by comparing it with the matched filter and BP network. This paper theoretically analyzes the efficiency performance of all the algorithms with asymptotic time complexity (also known as time complexity) [60]:

$$\mathcal{T}(n) = O(f(n)) = \eta f(n) \quad (24)$$

where n is the scale of the problem to be solved and $f(n)$ is a function having the same order of magnitude with $\mathcal{T}(n)$, which makes their ratio η a non-zero constant.

The pseudocode of the demodulation scheme shown in Figure 3 is given in Algorithm 2, in which Line 7 is replaced by using the matched filter or BP network to perform the comparisons. The time complexity of this demodulation scheme \mathcal{T}_{total} can be expressed as:

$$\mathcal{T}_{total} = \mathcal{T}_{FIR} + \mathcal{T}_{norm} + \mathcal{T}_* + \mathcal{T}_{Dec} \quad (25)$$

with

$$\mathcal{T}_{FIR} = m \times T \times f_s \times n_{ord} \quad (26)$$

$$\mathcal{T}_{norm} = m \times T \times f_s \quad (27)$$

$$\mathcal{T}_{Dec} = \frac{3}{2} \times m \times n \quad (28)$$

where \mathcal{T}_{FIR} , \mathcal{T}_{norm} and \mathcal{T}_{Dec} are the time complexities of FIR filter, normalization operation and decision function, \mathcal{T}_* with $*$ = "PLS", "MF" or "BP" is the time complexity of the selected demodulator, n_{ord} is the order of the FIR filter, m is the symbol number of the input message, $n = 256$ is the number of symbol candidates, and $f_s = 4000$ Hz is the sampling rate.

For each symbol, according to the algorithms presented in Section 3, the PLS model runs in polynomial time (An algorithm is said to be of polynomial time if its running time is upper-bound by a polynomial expression in the size of the input for the algorithm, i.e., $\mathcal{T}(\approx \sim \sim \sim n) = O(\approx \sim \sim n^\kappa)$ for some positive constant κ [60,61].): $\mathcal{T}_{PLS} = n \times T \times f_s$. Similarly, the time complexity of the matched filter is $\mathcal{T}_{MF} = n \times T^2 \times f_s^2$. As for the forward propagation of a BP network, when the size of a hidden layer is $n - 1$, its time complexity is $\mathcal{T}_{BP} = (n - 1) \times T \times f_s + n^2$. With the experiment protocol of this paper, PLS model clearly has the lowest time complexity, whereas matched filter the highest.

Table 3 lists the implementations of the proposed and the two reference algorithms. They are implemented by using the multi-core CPU platform presented in Section 4. The indices “_ori” and “_opt” refer to the original and optimized implementations, respectively. The detection step of the matched filter (see Equation (2)) and the message generation steps of all the three algorithms are parallelized. The running speed measurements cover all the accelerated steps, including detection and decision. The CPU used is Intel(R) Core(TM) i7-6700 CPU 3.4 GHz. The ratio of the running time, known as acceleration ratio, is used as the efficiency evaluation criterion, and the running time of *mf_ori* with $T = 0.3$ s (72.8 ms per symbol) is set as the measuring basis.

Table 3. Description of multi-core CPU implementations.

Names	Algorithms	Optimizations
<i>mf_ori</i>	Matched filter	None
<i>mf_opt</i>	Matched filter	Multi-thread parallelization
<i>bp_ori</i>	BP network	None
<i>bp_opt</i>	BP network	Multi-thread parallelization
<i>pls_ori</i>	PLS regression	None
<i>pls_opt</i>	PLS regression	Multi-thread parallelization

Figure 12 displays the acceleration ratios of all the implementations. Firstly, we can see that the implementations based on machine learning methods are much more efficient than the matched filters. Their running speed gaps are hundreds of times. With matched filters, the received symbol is correlated with the replicas of the original symbols point by point along the time axis, which is quite time-costly. By contrast, the PLS model and the forward propagation process of BP networks are essentially matrix/vector multiplication operations with low computation intensity. So, even if the running speed has been multiplied through multi-thread optimization, their temporal efficiency performance are still not on the same order of magnitude.

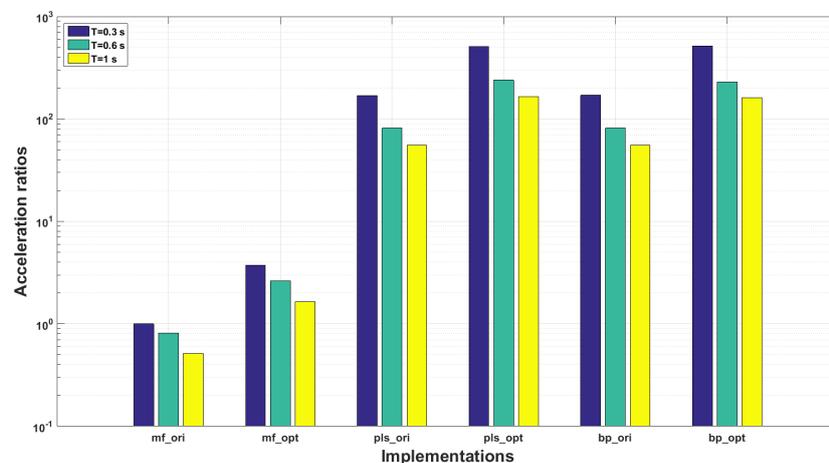


Figure 12. Running speed comparison results: $f_s = 4000$ Hz.

Secondly, comparing with the original implementations, the optimized implementations are accelerated by around three times. In this experiment, a four-core CPU is used, so the speed-up should have been four times higher than measured values (in theory). This is caused by the following two reasons:

- a the implementations are partly optimized instead of completely parallelized. The multi-thread optimizations of *pls_cpu* and *bp_cpu* cover only the message generations without detection steps;
- b the shared memory architecture is constrained by the von Nuemann bottle neck, the memory access conflicts may be occurred when multiple thread functions are invoked simultaneously.

Depending on the efficiency evaluation result, the time costs of all the implementations are lower than T , satisfying the constraints of real-time processing. Therefore, the communication rate \mathcal{R} is only constrained by T . Given that a symbol represents a bit stream of 8-bits with ASCII code, \mathcal{R} can be estimated as $(8 \times m)/(m \times T + d/c_s)$ in theory, where m is the received symbol number or the size of the transmitted message, d is the distance between the sender and receiver and c_s is the underwater sound speed.

6. Conclusions

This paper presents a novel machine learning available demodulation scheme for underwater communication applications based on the chaos phase modulation spread spectrum techniques. Thanks to the characteristic of a large number of phase-shift keying, it is more appropriate for confidential communication applications. Compared to the reference algorithms, experiments demonstrate that the proposed approach possesses higher accuracy performance with low SNR and high communication rate. The main originalities and contributions of this paper include:

- The proposed demodulation scheme handles the signal demodulation as a target recognition problem, allowing us to incorporate advanced classification methods into it, benefiting from the recent progresses of pattern recognition techniques. Because machine learning methods are able to statistically analyze the symbol features and noise characters, it can provide better demodulation accuracy performance than matched filters with shorter symbol sequences. That is, if the same accuracy performance is desired, the conventional matched filters need long symbol sequences in order to get enough time gains, so adopting machine learning methods, especially PLS regression, can considerably improve the underwater communication applications in terms of communication rates;
- It is found that the chaos-spreading spectrum signals have the characteristic of multicollinearity which may potentially impact the performance of certain classifiers negatively. Fortunately, the algorithm family of PLS can somehow overcome this disadvantage by modeling the relationships between the predictor and response variables, even with a sparse training dataset;
- The running cost of the proposed approach is very low. The detection steps of the proposed scheme costs only several milliseconds with the experiment protocol of this paper, which even can be ignored during the overall demodulation procedure, so it possesses high temporal efficiency performance;
- The proposed demodulation algorithm can be easily transplanted to other hardware platforms for different purposes. The detection process of the proposed algorithm is actually an operation of matrix production, which can be easily implemented and optimized by using any currently available computing platforms.

Meanwhile, some issues exist still. Firstly, the spatial variations and frequency-shift characteristic of underwater sound propagations are not considered in the simulation; some more evaluations with real-life experiment data are strongly required in future studies. Secondly, the experiment of this paper demonstrates that PLS regression has better accuracy performance than BP networks. However, it cannot be concluded that PLS regression is always the best classifier for all the three mentioned

methods, because the training set of this paper is sparse, far away from the requirement of deep learning. In the future work, we aim to attempt to further improve the evaluation experiment with the standard of deep learning. Thirdly, the PLS-based demodulator is able to resist some variation of communication channel, but it should be noted that the underwater sound channels sometimes rapidly vary in real-life applications. The training frequency of the PLS model will, therefore, be studied and a solution with real-time learning will be investigated. Finally, the parallel implementation of the training process of the proposed algorithm is not discussed in this paper. Some optimization properties within the instruction and data levels with other refined development tools, i.e., MPI for multi-core CPUs or CUDA for GPUs, still exist and are worth being investigated further.

Author Contributions: C.L. and F.Y. conceived the algorithm and analyzed the data. C.L. implemented the algorithm, designed the experiments, and wrote the paper; F.M. and F.Y. revised the paper. F.M. financed this work.

Funding: CAS Pioneer Hundred Talents Program provided funding for the studies.

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

References

1. Song, H.C. Acoustic communication in deep water exploiting multiple beams with a horizontal array. *J. Acoust. Soc. Am.* **2012**, *132*, EL81–EL87. [[CrossRef](#)] [[PubMed](#)]
2. Qiao, G.; Babar, Z.; Ma, L.; Liu, S.; Wu, J. MIMO-OFDM underwater acoustic communication systems—A review. *Phys. Commun.* **2017**, *23*, 56–64. [[CrossRef](#)]
3. Akyildiz, I.F.; Pompili, D.; Melodia, T. Challenges for efficient communication in underwater acoustic sensor networks. *ACM Sigbed Rev.* **2004**, *1*, 3–8. [[CrossRef](#)]
4. Diamant, R.; Lampe, L. Low Probability of Detection for Underwater Acoustic Communication: A Review. *IEEE Access* **2018**, *6*, 19099–19112. [[CrossRef](#)]
5. Ganapathy, H.; Pados, D.A.; Karystinos, G.N. New Bounds and Optimal Binary Signature Sets—Part II: Aperiodic Total Squared Correlation. *IEEE Trans. Commun.* **2011**, *59*, 1411–1420. [[CrossRef](#)]
6. Sklivanitis, G.; Demirors, E.; Batalama, S.N.; Melodia, T.; Pados, D.A. Receiver configuration and testbed development for underwater cognitive channelization. In Proceedings of the 2014 48th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 2–5 November 2014; pp. 1594–1598.
7. Yang, T.C.; Yang, W.B. Low signal-to-noise-ratio underwater acoustic communications using direct-sequence spread-spectrum signals. In Proceedings of the OCEANS 2007—Europe, Aberdeen, UK, 18–21 June 2007; pp. 1–6.
8. Yang, T.C.; Yang, W.B. Performance analysis of direct-sequence spread-spectrum underwater acoustic communications with low signal-to-noise-ratio input signals. *J. Acoust. Soc. Am.* **2008**, *123*, 842–855. [[CrossRef](#)] [[PubMed](#)]
9. Rovatti, R.; Setti, G.; Mazzini, G. Chaotic complex spreading sequences for asynchronous DS-CDMA. Part II. Some theoretical performance bounds. *IEEE Trans. Circuits Syst.* **1998**, *45*, 496–506. [[CrossRef](#)]
10. Mazzini, G.; Setti, G.; Rovatti, R. Corrections to “Chaotic complex spreading sequences for asynchronous DS-CDMA. I. System modeling and results”. *IEEE Trans. Circuits Syst.* **1998**, *44*, 515–516. [[CrossRef](#)]
11. Chen, C.C.; Yao, K.; Umeno, K.; Biglieri, E. Design of spread-spectrum sequences using chaotic dynamical systems and ergodic theory. *IEEE Trans. Circuits Syst.* **2001**, *48*, 1110–1114. [[CrossRef](#)]
12. Wang, X.Y.; Zhu, Z.F.; Fang, S.L. Noncooperative detection and parameter estimation of underwater acoustic DSSS-BPSK signal. In Proceedings of the International Conference on Mechatronics and Machine Vision in Practice, Xiamen, China, 4–6 December 2007; pp. 52–56.
13. Zhang, X. Spread Spectrum Sequence Estimation in Acoustic Non-Cooperative Communication System. In Proceedings of the 2009 International Symposium on Computer Network and Multimedia Technology, Wuhan, China, 18–20 January 2009; pp. 1–4.
14. Zhang, T.; Dai, S.; Zhang, W.; Ma, G.; Gao, X. Blind estimation of the PN sequence in lower SNR DS-SS signals with residual carrier. *Digital Signal Process.* **2012**, *22*, 106–113. [[CrossRef](#)]

15. Zhang, T.; Zhang, W.; Dai, S.; Ma, G.; Jiang, Q. A Spectral Method for Period Detection of PN Sequence for Weak DS-SS Signals in Dynamic Environments. In Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, Wuhan, China, 25–26 April 2009; pp. 266–269.
16. Abel, A.; Schwarz, W. Chaos communications-principles, schemes, and system analysis. *Proc. IEEE* **2002**, *90*, 691–710. [[CrossRef](#)]
17. Yang, T. A Survey of Chaotic Secure Communication Systems. *Int. J. Comput. Cogn.* **2004**, *2*, 81–130.
18. Bowong, S.; Kakmeni, F.M.M.; Siewe, M.S. Secure communication via parameter modulation in a class of chaotic systems. *Commun. Nonlinear Sci. Numer. Simul.* **2007**, *12*, 397–410. [[CrossRef](#)]
19. Kilfoyle, D.B.; Baggeroer, A.B. The state of the art in underwater acoustic telemetry. *IEEE J. Ocean. Eng.* **2000**, *25*, 4–27. [[CrossRef](#)]
20. Kocarev, L.; Jakimoski, G. Logistic map as a block encryption algorithm. *Phys. Lett. A* **2001**, *289*, 199–206. [[CrossRef](#)]
21. Yoshida, T.; Mori, H.; Shigematsu, H. Analytic study of chaos of the tent map: Band structures, power spectra, and critical behaviors. *J. Stat. Phys.* **1983**, *31*, 279–308. [[CrossRef](#)]
22. Rogers, T.D.; Whitley, D.C. Chaos in the cubic mapping. *Math. Model.* **1983**, *4*, 9–25. [[CrossRef](#)]
23. Bayliss, A.; Turkel, E. Mappings and accuracy for Chebyshev pseudo-spectral approximations. *J. Comput. Phys.* **1992**, *101*, 349–359. [[CrossRef](#)]
24. Zhang, Z.; Wang, H.; Zhao, Y.; Liu, J.; Yang, L. Frequency modulated radar signal with combined chaotic sequence based on Bernoulli map. *J. Commun. Technol. Electron.* **2016**, *61*, 971–979. [[CrossRef](#)]
25. Costa, R.A.D.; Loiola, M.B.; Eisenkraft, M. Correlation and spectral properties of chaotic signals generated by a piecewise-linear map with multiple segments. *Signal Process.* **2016**, *133*, 187–191. [[CrossRef](#)]
26. Liu, L.; Zhou, S.; Cui, J. Prospects and problems of wireless communication for underwater sensor networks. *Wirel. Commun. Mob. Comput.* **2008**, *8*, 977–994.
27. Turin, G.L. An introduction to matched filters. *IRE Trans. Inf. Theory* **1960**, *6*, 311–329. [[CrossRef](#)]
28. Stojanovic, M.; Catipovic, J.A.; Proakis, J.G. Phase-coherent digital communications for underwater acoustic channels. *IEEE J. Ocean. Eng.* **1994**, *19*, 100–111. [[CrossRef](#)]
29. Laot, C.; Coince, P. Experimental results on adaptive MMSE turbo equalization in shallow underwater acoustic communication. In Proceedings of the OCEANS'10 IEEE SYDNEY, Sydney, Australia, 24–27 May 2010; pp. 1–5.
30. Cannelli, L.; Leus, G.; Dol, H.; van Walree, P. Adaptive turbo equalization for underwater acoustic communication. In Proceedings of the 2013 MTS/IEEE OCEANS, Bergen, Norway, 10–14 June 2013; pp. 1–9.
31. Shimura, T.; Ochi, H.; Watanabe, Y.; Hattori, T. Demonstration of time-reversal communication combined with spread spectrum at the range of 900 km in deep ocean. *Acoust. Sci. Technol.* **2012**, *33*, 113–116. [[CrossRef](#)]
32. Shu, X.; Wang, H.; Wang, J.; Yang, X. A method of multichannel chaotic phase modulation spread spectrum and its application in underwater acoustic communication. *Chin. J. Acoust.* **2017**, *2*, 159–168.
33. Ushio, T.; Innami, T.; Kodama, S. Chaos Shift Keying Based on In-Phase and Anti-Phase Chaotic Synchronization. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **1996**, *79*, 1689–1693.
34. Yang, J.; Qiu, Z.K.; Li, X.; Zhuang, Z.W. Uncertain chaotic behaviours of chaotic-based frequency- and phase-modulated signals. *IET Signal Proc.* **2012**, *5*, 748–756. [[CrossRef](#)]
35. Wang, R.; Guo, J.; Leung, H. Orthogonal Circulant Structure and Chaotic Phase Modulation Based Analog to Information Conversion. *Signal Proc.* **2017**, *144*, 104–117. [[CrossRef](#)]
36. Yang, Q.L.; Yan-Fei, L.I.; Zhang, X.Y. The Design of Low Sidelobe and Orthogonal Waveform for MIMO Radar Based on Chaotic Phase-Spectrum Modulation. *Electron. Inf. Warfare Technol.* **2017**. [[CrossRef](#)]
37. Kaddoum, G. Wireless Chaos-Based Communication Systems: A Comprehensive Survey. *IEEE Access* **2017**, *4*, 2621–2648. [[CrossRef](#)]
38. Bai, C.; Ren, H.P.; Grebogi, C.; Baptista, M.S. Chaos-Based Underwater Communication With Arbitrary Transducers and Bandwidth. *Appl. Sci.* **2018**, *8*, 162. [[CrossRef](#)]
39. Van Nguyen, B.; Jung, H.; Kim, K. On the Anti-Jamming Performance of the NR-DCSK System. *Secur. Commun. Netw.* **2018**, *2018*, 7963641. [[CrossRef](#)]
40. Nguyen, D.V.; Rocke, D.M. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* **2002**, *18*, 39. [[CrossRef](#)]

41. Martens, M.; Martens, H. Partial Least Squares Regression. In *Statistical Procedures in Food Research*; Elsevier Applied Science: England, UK, 1986; pp. 293–359.
42. Worsley, K. An overview and some new developments in the statistical analysis of PET and fMRI data. *Hum. Brain Map* **1997**, *5*, 254–258. [[CrossRef](#)]
43. Nilsson, J.; Jong, S.D.; Smilde, A.K. Multiway calibration in 3D QSAR. *J. Chemom.* **1997**, *11*, 511–524. [[CrossRef](#)]
44. Hulland, J. Use of partial least squares (PLS) in strategic management research: A review of four recent studies. *Strat. Manag. J.* **1999**, *20*, 195–204. [[CrossRef](#)]
45. Uzair, M.; Mahmood, A.; Mian, A. Hyperspectral Face Recognition With Spatiospectral Information Fusion and PLS Regression. *IEEE Trans. Image Process.* **2015**, *24*, 1127–1137. [[CrossRef](#)]
46. Li, C.; Benezeth, Y.; Nakamura, K.; Gomez, R.; Yang, F. A robust multispectral palmprint matching algorithm and its evaluation for FPGA applications. *J. Syst. Archit.* **2018**, *88*, 43–53. [[CrossRef](#)]
47. Li, S.; Mou, X.; Cai, Y. Pseudo-random Bit Generator Based on Couple Chaotic Systems and Its Applications in Stream-Cipher Cryptography. In Proceedings of the International Conference on Cryptology in India, Chennai, India, 16–20 December 2001; pp. 316–329.
48. Wold, H. Soft Modelling: The Basic Design and Some Extensions. *Syst. Indirect Obs.* **1982**, 36–37. Available online: <https://ci.nii.ac.jp/naid/10006132197/en/> (accessed on 28 November 2018).
49. Wold, H. Partial least squares. In *Encyclopedia of Statistical Sciences*; Kotz, S., Johnson, N., Eds.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2004.
50. Wold, S.; Ruhe, A.; Wold, H.; Dunn, W.J., III. The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverse. *SIAM J. Sci. Stat. Comput.* **1984**, *5*, 745–743. [[CrossRef](#)]
51. Shawe-Taylor, J.; Cristianini, N. *Kernel Methods for Pattern Analysis*; Cambridge University Press: New York, NY, USA, 2004.
52. Walree, P.A.V.; Socheleau, F.X.; Otnes, R.; Jenserud, T. The Watermark Benchmark for Underwater Acoustic Modulation Schemes. *IEEE J. Ocean. Eng.* **2017**, *42*, 1007–1018. [[CrossRef](#)]
53. Widrow, B.; Mccool, J.M.; Larimore, M.G.; Johnson, C.R. *Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter*; Springer: Cham, Switzerland, 1977; pp. 355–393.
54. Lu, L.; Zhao, H.; Chen, B. Improved-Variable-Forgetting-Factor Recursive Algorithm Based on the Logarithmic Cost for Volterra System Identification. *IEEE Trans. Circuits Syst. Express Briefs* **2016**, *63*, 588–592. [[CrossRef](#)]
55. Li, W.; Preisig, J.C. Estimation of Rapidly Time-Varying Sparse Channels. *IEEE J. Ocean. Eng.* **2008**, *32*, 927–939. [[CrossRef](#)]
56. Jin, W.; Li, Z.J.; Wei, L.S.; Zhen, H. The improvements of BP neural network learning algorithm. In Proceedings of the 2000 5th International Conference on Signal Processing Proceedings, Beijing, China, 21–25 August 2000; pp. 1647–1649.
57. Barons, M.J.; Parsons, N.; Griffiths, F.; Thorogood, M. A comparison of artificial neural network, latent class analysis and logistic regression for determining which patients benefit from a cognitive behavioural approach to treatment for non-specific low back pain. In Proceedings of the 2013 IEEE Symposium on Computational Intelligence in Healthcare and e-health (CICARE), Singapore, 16–19 April 2013; pp. 7–12.
58. Fildes, R. Conditioning Diagnostics: Collinearity and Weak Data in Regression. *Technometrics* **1993**, *35*, 85–86.
59. David, A.B.; Edwin, K.; Welsch, R.E. *Conditioning Diagnostics: Collinearity and Weak Data in Regression*; Wiley: Hoboken, NJ, USA, 2005.
60. Sipser, M. *Introduction to the Theory of Computation*, 2nd ed.; Thomson Course Technology: Boston, MA, USA, 2005.
61. Johnson, D.S.; Papadimitriou, C.H. Computational complexity. In *Local Search in Combinatorial Optimization*; John Wiley and Sons Ltd.: Chichester, UK, 1994; pp. 36–60.

