information

Article

On Channel Codes for Short Underwater Messages

Melanie Falk [1,*], Gerhard Bauch [1] and Ivor Nissen [2]

[1] Institute of Communications, Hamburg University of Technology, 21073 Hamburg, Germany; bauch@tuhh.de

[2] German Technical Center for Ships and Naval Weapons, Naval Technology and Research (WTD 71), Underwater Communication Branch, 24340 Eckernförde, Germany; IvorNissen@Bundeswehr.org

[*] Correspondence: melanie.falk@tuhh.de

Received: 11 December 2019; Accepted: 20 January 2020; Published: 23 January 2020

Abstract: Acoustic underwater communication is a challenging task. For a reliable transmission, not only good channel estimation and equalization, but also strong error correcting codes are needed. In this paper, we present the results of the coding competition "Wanted: Best channel codes for short underwater messages" as well as our own findings on the influence of the modulation alphabet size in the example of non-binary polar codes. Furthermore, the proposals of the competition are compared to other commonly used channel codes.

Keywords: channel coding; underwater acoustic sensor network; forward error correction; phase shift keying; polar code

1. Introduction

Wireless communication is a challenging task in an environment where the channel impulse response varies rapidly over time and space, e.g., in car2car, outer space or underwater communication scenarios. The results of the in-situ channel estimation have to be analyzed in real time in order to determine optimal parameter settings. This is a difficult task, however, without this step the communication is not only inefficient, but also unreliable.

Looking to the underwater world, whales, dolphins and some fishes are successfully using acoustic click signals in this noise-superimposed multi-path and Doppler environment by carrying the information in different sequence patterns. Inspired by this, burst communication—the transmission of short time signals—is proposed in [1] as a solution to avoid inter-symbol-interference. However, the transmission of, in total, one multi-carrier symbol has the disadvantage of only a few bytes of transfer volume.

In recent years, many research groups have investigated underwater acoustic sensor networks (UWASNs). Important fields of study are network deployment, routing protocols and localization. Due to the harsh environment, the use of coding techniques is required to achieve a reliable communication. Approaches on the network layer can be found, e.g., in [2,3]. In [4,5], a combination of channel coding on the physical layer and network coding is considered. Also several channel codes have been investigated in an underwater scenario, e.g., convolutional codes in [6,7], Reed–Solomon codes in [7], low density parity check (LDPC) codes in [8,9], and (irregular) repeat accumulate ((I)RA) codes in [10,11], respectively. Due to the different message lengths and channels, the results of these works cannot be compared.

In this paper, we define a common framework in order to compare various channel codes for short underwater messages which are used in burst communication. While a broad community is working on efficient channel codes for large messages and streams of binary data, the study of channel codes for short messages is a less public topic. Therefore, we organized an international competition to support research in this field: The task was to find the best mapping–coding combination to transmit 128 information bits within 256 phase shift keying (PSK) symbols under the constraint of given restricted computational power. The winner received the Heinrich Hecht award.

In Table 1, possible combinations of modulation alphabet sizes and code rates are given. In general, higher order modulations result in lower Euclidean distances d_{eu} and thus, are more prone to transmission errors. On the other hand, they allow a lower code rate R_C of the channel code in the scenario under investigation. A lower code rate often results in a stronger protection such that this might overcompensate for the lower Euclidean distance. In the last column, the quotient d_{eu}/R_C is given as an indicator for a good mapping–coding combination. It suggests that QPSK (4-PSK) is the best modulation for the given problem, followed by the rather uncommon schemes 3-PSK and 5-PSK.

Table 1. Modulation and coding schemes for the transmission of 128 information bits in 256 μ-phase shift keying (PSK) symbols.

μ	Name	Distance d_{eu}	code rate R_C	d_{eu}/R_C
2	BPSK	2	0.5	4
3	3-PSK	1.732	0.316	5.47
4	QPSK	1.414	0.25	5.66
5	5-PSK	1.176	0.219	5.37
6	6-PSK	1	0.195	5.12
7	7-PSK	0.868	0.180	4.83
8	8-PSK	0.765	0.168	4.56

In the competition, polar codes, LDPC codes and a Reed–Solomon code have been submitted. Most of the solutions propose indeed a $R_C = 1/4$ channel code with QPSK. To complete the picture, we additionally investigate the performance of a convolutional code and a turbo code so that all well-known classes of error-correcting codes are evaluated for the given scenario. Furthermore, we address the question whether unconventional coding schemes like 3-PSK, 5-PSK and 7-PSK are beneficial for our application. We combine each of these modulation schemes with a non-binary polar code which works on Galois field GF(3), GF(5) or GF(7), respectively. To the best of our knowledge, the decoding process and simulation results for non-binary polar codes have only been presented for codes on GF(2^m), e.g., in [12,13]. Therefore, we provide a detailed description of the decoding of the binary polar code and the necessary changes which have to be made for codes on GF(q) where q is a prime number larger than 2.

The rest of the paper is structured as follows: In the next section, the underwater sensor network scenario is described. Section 3 is about the competition. First, the task is described, then the results are shown and evaluated. The binary polar code is presented in Section 4 as some readers might be not familiar with this relatively new class of code. In Section 5, the novel non-binary polar code is introduced and evaluated in combination with unconventional PSK sizes. The evaluation of channel codes for the given underwater communication scenario is completed in Section 6 by showing results for convolutional and turbo codes. Finally, the paper is concluded.

2. Scenario: Underwater Sensor Network

In this paper, we investigate channel codes and modulation for the application in an underwater acoustic sensor network (UWASN) as described in [14]. It consists of bottom-mounted and moving nodes acting as a mobile ad-hoc network (MANET). If a node has a message, it will be transmitted using a short burst. The message is forwarded by neighboring sensor nodes until it arrives at the destination. In the network, all nodes work as both sensors and relays. The main sensor is an acoustic sensor, hence, communication and sensing share the same medium. Due to the burst communication, the interference of these two services is kept at a low level.

As in [14], each message consists of 128 information bits. To reduce the probability of erroneous messages at the network layer, each message is encoded with an error-correcting channel code. Coding strategies at the network layer are out of the scope of this paper. As modulation, μ-ary phase shift keying (PSK) is used because the phase distortions caused by the channel can be more reliably corrected than amplitude distortions. For each message, the fixed number of 256 μ-PSK symbols is used which results from the assumed burst communication. This opens a design trade-off as higher order PSK is more noise-sensitive, but allows for a lower code rate and thus, a stronger error correction capability of the channel code.

For the simulations, a rather simple channel model has been chosen. It consists of additive white Gaussian noise (AWGN) and a small multiplicative amplitude distortion which models that amplitude distortions are corrected less accurately than phase distortions. This is an oversimplification of a real acoustic underwater channel. However, there is no well-established stochastic channel model as an underwater counterpart to, e.g., the WINNER models for cellular radio communication due to the large variety of underwater acoustic channels [15]. Although channels based on measurement campaigns can be found, e.g., in the WATERMARK benchmark [16], we decided not to use them because the proposals in the competition should not be optimized to the channel data of one specific measurement campaign.

The transmission chain which is used to evaluate different channel code-modulation size proposals is shown in Figure 1. The information word $\mathbf{u}$ consists of 128 bits. It is encoded and mapped onto 256 symbols $x_i \in \{0, 1, \dots, \mu - 1\}$, where μ denotes the PSK cardinality. These are mapped onto complex μ-PSK symbols $s_i = e^{j\frac{2\pi x_i}{\mu}}$ and transmitted over the channel. Each transmit symbol is multiplied by a random real-valued channel factor h_i drawn from a uniform distribution $\mathcal{U}(0.9, 1.1)$ and complex-valued white Gaussian noise n_i is added. The noisy received symbols $y_i = h_i s_i + n_i$ are decoded to find an estimated information word $\hat{\mathbf{u}}$.

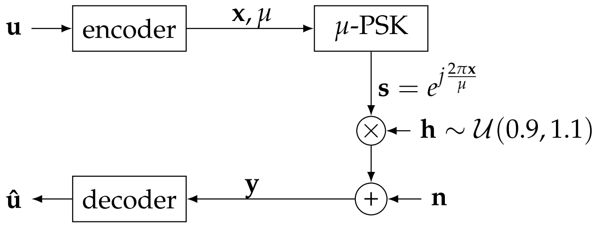

Figure 1. Transmission chain.

As 128 information bits are encoded and mapped to 256 μ-PSK symbols, the overall rate R is always 0.5 bit/channel use. For BPSK ($\mu = 2$), the code rate of the channel code is $R_C = 0.5$ while for 4-PSK (QPSK), it is $R_C = 0.25$.

3. Competition

We organized an international coding competition to find the best channel code and PSK cardinality μ for the underwater sensor network described before and to support research on channel codes for short messages. The call [17] was published in July 2018 on *researchgate.org*, the submission deadline was in January 2019.

3.1. Task Description

For the competition, a test environment written in C++ was given. It simulates message transmissions to estimate the bit and block error rate for various signal-to-noise power ratios as well as the encoding and decoding time. The encoder and demodulator–decoder functions had to be implemented and submitted by the participants.

Encoding and mapping were allowed to be performed in separate steps or to be combined into some coded modulation approach. In the competition, any PSK cardinality μ between 2 and 8 was allowed, including integers that are not a power of 2.

The main evaluation criterion for the submitted codes was the block error rate, furthermore, encoding and decoding time as well as memory consumption had to be within the following limits:

- Both the average encoding and decoding time should be below a reference time which is based on fast Fourier transform (FFT) calculations for all signal-to-noise power ratios.
- The memory requirements of the encoder and decoder should be below 1 MB.

3.2. Evaluation of the Submitted Proposals

Researchers from Japan, Germany and the United States have participated in the competition. In total, five different types of codes were submitted:

- Polar code with successive cancellation (SC) decoder and 4-PSK.
- Cyclic redundancy check (CRC) aided polar code with successive cancellation list (SCL) decoder and 4-PSK.
- LDPC code with 4-PSK.
- Non-binary LDPC code on GF(256) with 4-PSK.
- Reed–Solomon code with BPSK.

Reed–Solomon codes have been introduced in [18] in 1960, LDPC codes in [19] in 1962 and non-binary LPDC codes in [20] in 1998. The LDPC codes proposed in the competition are based on [21,22], respectively. Polar codes were first presented in [23] in 2008, CRC aided polar codes with SCL decoding in [24]. The construction of polar code with SCL decoder has been proposed in [25]. In all proposals of the competition, channel coding and mapping are performed in separate steps. For 4-PSK, Gray mapping is used.

All participants except one group have allowed the publication of the results based on their implementations. Therefore, the results for the polar code with SC decoding shown in this work are based on the work of the authors of the paper instead of the submitted code. The shown en- and de-coding times are higher while the block error rate is lower than those of the proposal.

In the following, it is checked first whether the complexity constraints are fulfilled. Then, the performance in terms of block error rate is evaluated. All submitted codes require significant less than 1 MB of RAM. For low to moderate signal-to-noise power ratios, the mean decoding time of the non-binary LDPC code is far beyond the time limit. Therefore, the code is excluded. The other four codes fulfill the time constraint.

In Figure 2, the average encoding and decoding time, respectively, normalized by the reference time are displayed over the signal-to-noise power ratio $\bar{E}_b/N_0$ while the block error rate is shown in Figure 3. It can be seen that the encoding time is negligible for all proposals. The decoding times do not depend on the signal-to-noise power ratio except for the decoding of the LPDC code. The LDPC code is decoded with a belief propagation decoder which needs on average very few iterations at sufficiently high $\bar{E}_b/N_0$ to find a valid codeword whereas in very noisy conditions, the algorithm stops when a given maximum number of iterations is reached. Thus, the worst case decoding time can be influenced by adjusting the maximum number of iterations and a trade-off between worst case complexity and error correction performance has to be found.

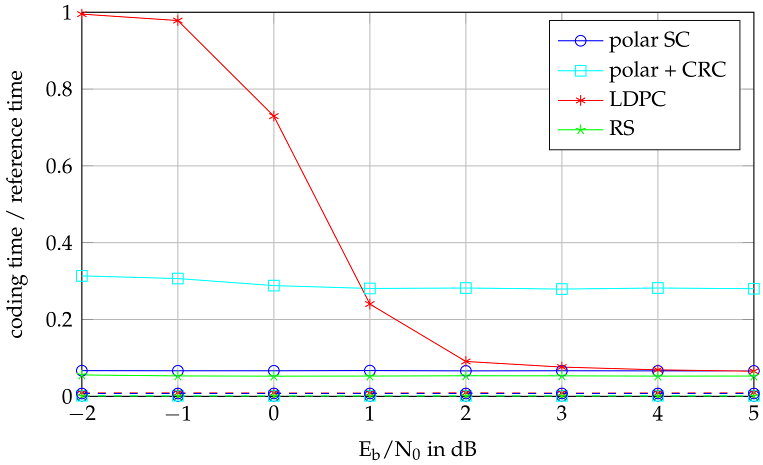

Figure 2. Average encoding time (dashed line) and decoding time (solid line) normalized to the reference time.

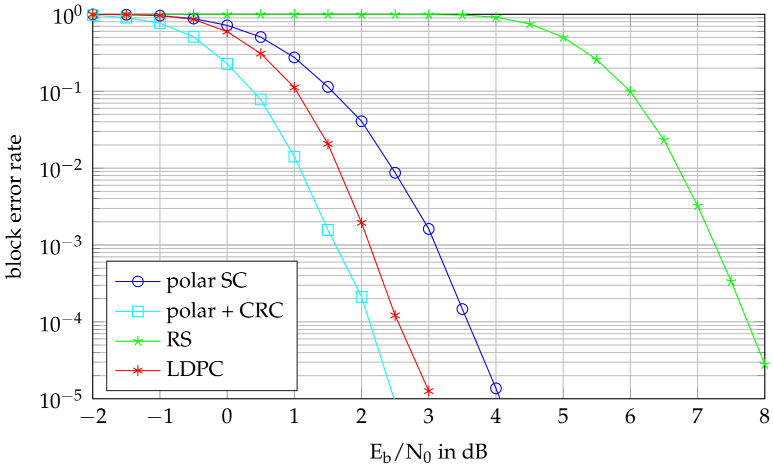

Figure 3. Block error rate.

For a given block length N_C, code rate R_C and decoding algorithm, there is no further parameter to influence the complexity of the polar code with SC decoder and the Reed–Solomon code. In case of the polar code with SCL decoder, there is an almost linear dependence of the decoding time on the selected list size. A larger list size allows for better error correction. While doubling the list size results in significant improvements for the regime of very low list sizes, the gains quickly decrease.

The polar code with successive cancellation decoding and 4-PSK (polar SC) and the Reed–Solomon code with BPSK (RS) both have a very low decoding time. However, the polar code is around 4 dB better than the Reed–Solomon code in terms of block error rate. The lowest block error rate is achieved by the CRC-aided polar code with successive cancellation list decoding and 4-PSK (polar + CRC). It is around 0.5 dB better than the proposed LDPC code with 4-PSK (LDPC) and approximately 1.5 dB better than the polar code with the simpler decoder. Since the CRC-aided polar code has the best error rate performance and a complexity within the given constraints, it is the winner proposal of the competition. It is described in detail in the next section.

The winner proposal has been submitted by Tobias Prinz, Fabian Steiner, Thomas Wiegart and Peihong Yuan from Technical University of Munich, Germany. They have received the Heinrich Hecht award which is endowed with 5000 Euros. The award is named after Karl Heinrich Hecht (1880–1961), a German pioneer of acoustic underwater communication.

4. Polar Code

Polar codes have been introduced by Arikan in 2008 [23,26]. The basic building block of a polar code is shown in Figure 4. It is represented mathematically as

$$\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix}. \tag{1}$$

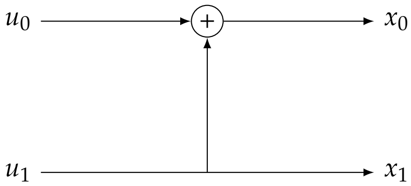

Figure 4. Building block of polar codes.

The structure of a polar code with length $N = 2^n$ is obtained by concatenating basic building blocks. It can be described by the matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^{\otimes n} \mathbf{B} \tag{2}$$

where $\mathbf{F}^{\otimes n}$ denotes the nth Kronecker power of $\mathbf{F}$ and $\mathbf{B}$ is the bit reversal permutation matrix. In Figure 5 an example for $n = 2$ is shown.

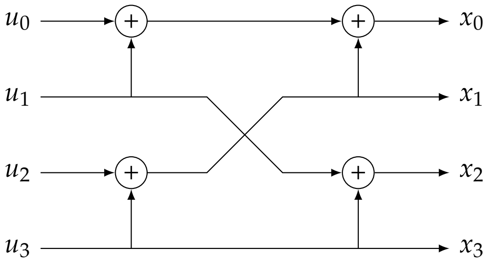

Figure 5. Polar encoder of size $N = 4$.

The input vector $\mathbf{u}$ consists of K information bits and $N - K$ so called frozen bits which have a fixed value, are usually set to zero. The positions of the frozen bits are determined with a construction algorithm. Popular construction algorithms are the Gaussian approximation [27] and the Tal and Vardy method [28]. Recently, construction methods using the information bottleneck method [29] or a genetic algorithm [30] have been proposed.

4.1. Successive Cancellation Decoding

Polar codes have been introduced with a decoding scheme called successive cancellation (SC) decoding [26]. The combination of the encoder, channel and SC decoder forms a virtual channel or bit channel for each input bit u_i. Most of these N virtual channels are either very good (capacity close to one) or very bad (capacity close to zero). This polarization effect is the stronger the larger the code size N is. Most of the aforementioned construction algorithms aim at evaluating the quality of the virtual channels and assigning the frozen bits to the $N - K$ worst virtual channels.

A successive cancellation decoder estimates the information bits one after the other, using decisions on previous bits for the estimation of the current bit. The decoding process is illustrated with a small example for $N = 4$ depicted in Figure 6. Assume that u_0 is a frozen bit known to be zero.

The log-likelihood ratios (LLR) $L_{i,0} = \log\left(\frac{Pr(x_i=0|y_i)}{Pr(x_i=1|y_i)}\right)$ have been calculated from the noisy received symbols y_i and $\boxplus$ denotes the boxplus-operation which is usually approximated by

$$L_i \boxplus L_j \approx \mathrm{sgn}(L_i) \cdot \mathrm{sgn}(L_j) \cdot \min\left(|L_i|, |L_j|\right). \tag{3}$$

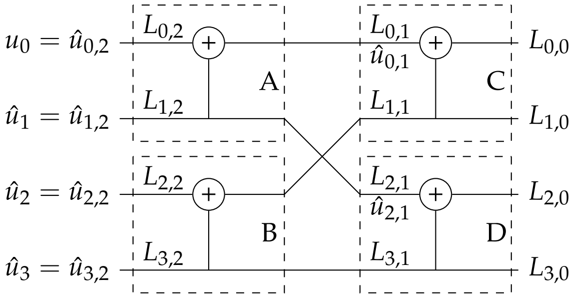

Figure 6. Polar decoder of size $N = 4$.

Then, the process is as follows:

1. Calculate the LLRs of the upper branches of block C and D as $L_{0,1} = L_{0,0} \boxplus L_{1,0}$ and $L_{2,1} = L_{2,0} \boxplus L_{3,0}$.
2. As u_0 is frozen, the estimate $\hat{u}_{0,2} = u_0 = 0$.
3. Calculate the LLR of the lower branch of block A as $L_{1,2} = (-1)^{\hat{u}_{0,2}} L_{0,1} + L_{2,1}$ and estimate u_1 as $\hat{u}_1 = \hat{u}_{1,2} = \frac{1}{2}(1 - \mathrm{sgn}(L_{1,2}))$.
4. Propagate the decision on $\hat{u}_{1,2}$ to determine $\hat{u}_{0,1} = \hat{u}_{0,2} \oplus \hat{u}_{1,2}$ and $\hat{u}_{2,1} = \hat{u}_{1,2}$.
5. Calculate the LLRs of the lower branches of block C and D as $L_{1,1} = (-1)^{\hat{u}_{0,1}} L_{0,0} + L_{1,0}$ and $L_{3,1} = (-1)^{\hat{u}_{2,1}} L_{2,0} + L_{3,0}$.
6. Calculate the LLR of the upper branch of block B as $L_{2,2} = L_{1,1} \boxplus L_{3,1}$ and estimate u_2 as $\hat{u}_2 = \hat{u}_{2,2} = \frac{1}{2}(1 - \mathrm{sgn}(L_{2,2}))$.
7. Use $\hat{u}_{2,2}$ to calculate the LLR of the lower branch of block B as $L_{3,2} = (-1)^{\hat{u}_{2,2}} L_{1,1} + L_{3,1}$ and estimate u_3 as $\hat{u}_3 = \hat{u}_{3,2} = \frac{1}{2}(1 - \mathrm{sgn}(L_{3,2}))$.

In general, the LLRs are calculated in the following manner:

$$L_{i,j} = \begin{cases} L_{k,j-1} \boxplus L_{k',j-1} & i \text{ even} \\ (-1)^{\hat{u}_{i-1,j}} L_{k,j-1} + L_{k',j-1} & i \text{ odd} \end{cases} \tag{4}$$

where $k = \lfloor \frac{i}{2} \rfloor + \lfloor \frac{i}{2^j} \rfloor 2^{j-1}$ and $k' = k + 2^{j-1}$.

4.2. Successive Cancellation List Decoder

In the successive cancellation decoder, error propagation can easily occur because each decision depends on previous decisions. As the algorithm is not iterative, errors once made cannot be corrected. The successive cancellation list (SCL) decoder has been proposed in [24] to mitigate these problems. The SCL decoder can be viewed as up to L SC decoders working in parallel, where L denotes the list size. Whenever a decision on an information bit has to be made, the SC decoder is duplicated including all LLRs $L_{i,j}$ and hard values $\hat{u}_{i,j}$ computed so far. One SC decoder continues with estimate $\hat{u}_i = 0$, the other with $\hat{u}_i = 1$. As soon as the number of decoding paths exceeds L, only the L most promising paths are further pursued, all other are deleted. The path metric $M_i^{(l)}$ of path l at stage i is recursively calculated as

$$M_i^{(l)} = M_{i-1}^{(l)} - \ln(1 + e^{-(1-2\hat{u}_{i,n}^{(l)})L_{i,n}^{(l)}}) \approx \begin{cases} M_{i-1}^{(l)} & if\ \mathrm{sgn}(L_{i,n}^{(l)}) = 1 - 2\hat{u}_{i,n}^{(l)} \\ M_{i-1}^{(l)} + |L_{i,n}^{(l)}| & otherwise \end{cases}. \tag{5}$$

The path metric is also computed when the input bit u_i at stage i is frozen. An efficient software implementation of the SCL decoder is described in [24] which is extended to LLR-based computations in [31].

It has been observed that frequently, the correct codeword is in the list, but it has not the lowest path metric. Therefore, polar codes are often combined with an outer code for error detection, e.g., a cyclic redundancy check (CRC) code [24]. The number of frozen bits is reduced by the number of parity bits of the error detecting code. The weakening of the polar code is usually overcompensated by the gain of excluding invalid codewords from the candidate list.

In the winner proposal, the polar code is combined with a CRC length of 7 for $\bar{E}_b/N_0 \leq 1.75\,dB$, while for larger signal-to-noise power ratios, a CRC size of 11 is used. The list size L is set to 32.

5. Non-Binary Polar Codes and Influence of PSK Size

In most of the proposals, a PSK cardinality of 4 is used. In Figure 7, the capacity (Gaussian input) and the mutual information for various PSK modulations are shown for the AWGN channel which is very similar to the channel model used in the competition. In the given scenario, the overall rate is 0.5 bit/channel use. Considering the mutual information in Figure 7, all complex-valued PSK modulations are equally suitable while the real-valued BPSK needs an almost 1 dB better signal-to-noise power ratio.

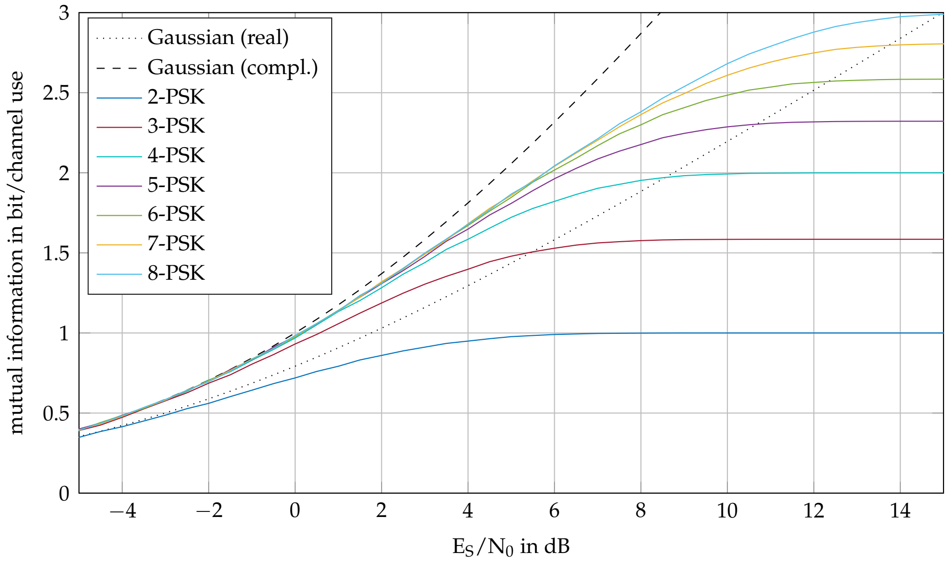

Figure 7. Mutual information for various PSK cardinalities.

To estimate whether 4-PSK is the best modulation in the given scenario, the winner's proposal is compared to our non-binary polar codes on Galois field GF(q) with q-PSK where q is a prime number.

5.1. Polar Codes on GF(q)

In [32], it is shown that polarization is achieved for non-binary input alphabets $\mathcal{A} = \{0, \ldots, q-1\}$ with q being a prime number using a similar construction as in the binary case, i.e., the basic building block and the structure of the code are the same, but the addition modulo 2 is replaced by the addition modulo q. However, following research has been focused on non-binary polar codes on GF(2^m) which need a modified building block to achieve polarization [32], but can easily be used on binary data and be combined with common modulation schemes such as BPSK, QPSK or 16-QAM. In contrast, polar codes with q being a prime number are suitable for unconventional modulation schemes like 3-PSK, 5-PSK and 7-PSK which shall be evaluated in this section.

In the following, we show how the polar decoder has to be modified for q being a prime number larger than 2. Due to the larger alphabet, the decoder works on probability vectors $\mathbf{p}$ instead of log-likelihood ratios. Considering the equations of the basic building block

$$x_0 = (u_0 + u_1)_{mod\, q} \tag{6}$$

$$x_1 = u_1 \tag{7}$$

the probability of $u_0 = m$ given the received vector $\mathbf{y} = [y_0, y_1]^T$ is

$$Pr\{u_0 = m|\mathbf{y}\} = Pr\{(x_0 - x_1)_{mod\, q} = m|\mathbf{y}\} = \sum_{n=0}^{q-1} Pr\{x_0 = (m+n)_{mod\, q}|\mathbf{y}\} \cdot Pr\{x_1 = n|\mathbf{y}\}. \tag{8}$$

Considering this, the operation of the upper branch becomes $\mathbf{p}_{u_0} = \text{funUpperBranch}(\mathbf{p}_{x_0}, \mathbf{p}_{x_1})$ described in Algorithm 1.

Algorithm 1: funUpperBranch($\mathbf{p}_{x_0}$,$\mathbf{p}_{x_1}$)

```
for $m = 0$ to $q - 1$ do
    $\mathbf{p}_{u_0}(m) = \sum_{n=0}^{q-1} \mathbf{p}_{x_0}((m+n)_{mod\, q}) \cdot \mathbf{p}_{x_1}(n)$;
end
return $\mathbf{p}_{u_0}$;
```

In contrast to u_0, u_1 is included in both Equations (6) and (7). In the decoding process, the estimation of u_1 depends on the decision $\hat{u}_0$. Therefore,

$$Pr\{u_1 = m|\mathbf{y}, \hat{u}_0\} = Pr\{(x_0 - \hat{u}_0)_{mod\, q} = m|\mathbf{y}\} \cdot Pr\{x_1 = m|\mathbf{y}\} \cdot \beta \tag{9}$$

where the first factor results from Equation (6), the second factor results from Equation (7) and the third factor β is needed for normalization so that $\sum_{m=0}^{q-1} Pr\{u_1 = m|\mathbf{y}, \hat{u}_0\} = 1$. Thus, the operation of the lower branch becomes $\mathbf{p}_{u_1} = \text{funLowerBranch}(\mathbf{p}_{x_0}, \mathbf{p}_{x_1}, \hat{u}_0)$ described in Algorithm 2.

Algorithm 2: funLowerBranch($\mathbf{p}_{x_0}$,$\mathbf{p}_{x_1}$,$\hat{u}_0$)

```
$\mathbf{p}_{\hat{u}_0}$ = all zero vector of size $q$;
$\mathbf{p}_{\hat{u}_0}(\hat{u}_0) = 1$; //hard-decision
$\mathbf{p}_{xp}$ = funUpperBranch($\mathbf{p}_{x_0}$,$\mathbf{p}_{\hat{u}_0}$); //permutation of $\mathbf{p}_{x_0}$
$beta = 0$;
for $m = 0$ to $q - 1$ do
    $\mathbf{p}_{u_1}(m) = \mathbf{p}_{xp}(m) \cdot \mathbf{p}_{x_1}(m)$;
    $beta = beta + \mathbf{p}_{u_1}(m)$;
end
$\mathbf{p}_{u_1} = \mathbf{p}_{u_1}/beta$; //normalization
return $\mathbf{p}_{u_1}$;
```

Similar to the log-likelihood ratio calculation in Equation (4)

$$\mathbf{p}_{i,j} = \begin{cases} \text{funUpperBranch}(\mathbf{p}_{k,j-1}, \mathbf{p}_{k',j-1}) & i \text{ even} \\ \text{funLowerBranch}(\mathbf{p}_{k,j-1}, \mathbf{p}_{k',j-1}, \hat{u}_{i-1,j}) & i \text{ odd} \end{cases} \tag{10}$$

where $k = \lfloor \frac{i}{2} \rfloor + \lfloor \frac{i}{2^j} \rfloor 2^{j-1}$ and $k' = k + 2^{j-1}$.
The path metric is computed using log-likelihoods:

$$M_i^{(l)} = M_{i-1}^{(l)} - \ln(\mathbf{p}_{i,n}^{(l)}(\hat{u}_{i,n}^{(l)})) \tag{11}$$

5.2. Results

We have implemented non-binary polar codes with 3-PSK, 5-PSK and 7-PSK. In the following, their block error rates are compared to the block error rate of the winner proposal. In a first step, the binary source is replaced by a q-ary source so that a conversion between bits and q-ary symbols is not needed. Therefore, the performance cannot be diminished by conversion losses. The number of information symbols K' is chosen so that again approximately 0.5 bit/channel use are transmitted. All polar codes are combined with a CRC on GF(q) and decoded using a successive cancellation list decoder. The number of information symbols K' as well as the CRC polynomial are given in Table 2.

Table 2. Parameters of the non-binary polar codes.

q	K'	**CRC Polynomial**
3	80	$x^6 + x^5 + 2x^2 + 1$
5	55	$x^5 + x^4 + x^3 + 3x^2 + 4x + 2$
7	46	$x^4 + x^2 + 3x + 5$

In Figure 8, the block error rate curves are shown for the list sizes $L = 32$, $L = 100$ and $L = 625$. It can be seen that the polar code with 4-PSK is the best for $L = 32$ and $L = 100$ while for $L = 625$, the non-binary polar code on GF(3) is slightly better for $E_b/N_0 > 0.5\,dB$. The block error rate of the polar code on GF(5) is considerably worse than the error rate of the binary and ternary polar codes. However, the gain of an increasing list size is larger for the polar code on GF(5) so there might be a very large list size for which all three codes have a similar error rate.

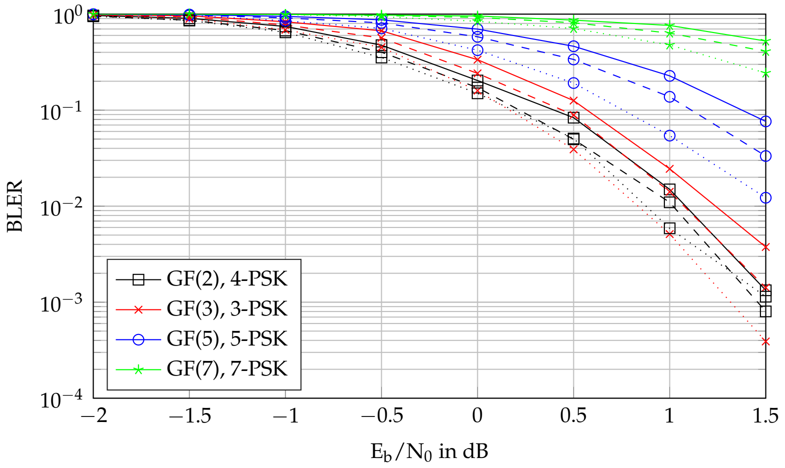

Figure 8. Block error rate for polar codes on GF(q) for list size $L = 32$ (solid line), $L = 100$ (dashed line) and $L = 625$ (dotted line).

For the given application, this is irrelevant as the decoding is far too complex for large list sizes. Furthermore, the decoding complexity increases with the alphabet size because the probability vectors become larger. Therefore, only 3-PSK should be further considered as an alternative to 4-PSK.

So far, the source emitted q-ary symbols. In the following, a binary source is considered. The transmission chain for the ternary polar code is shown in Figure 9. First, a binary check sum is appended to the information word $\mathbf{u}$ which is calculated using the CRC polynomial $x^6 + x^2 + x + 1$. Then, the sequence of 134 binary values is converted to a sequence of 85 symbols that take values from $\{0, 1, 2\}$. The resulting ternary sequence is encoded by the polar code on GF(3) and mapped to 256 3-PSK symbols. The combination of a binary CRC and the ternary polar code has been chosen because the conversion from 134 bits to 85 ternary symbols introduces less loss than the conversion from 128 bits to 81 ternary symbols. At the receiver, first successive cancellation list decoding is performed on GF(3), then the candidate sequences, sorted according to the path metric, are converted to binary sequences until a sequence that fulfills the CRC check is found or all candidate sequences are converted and checked. The block error rate of this code is shown in Figure 10. For comparison, the error rates of the winner proposal (4-PSK, L = 32) and the polar code on GF(3) with ternary source (ref 3-PSK, L = 32) are given.

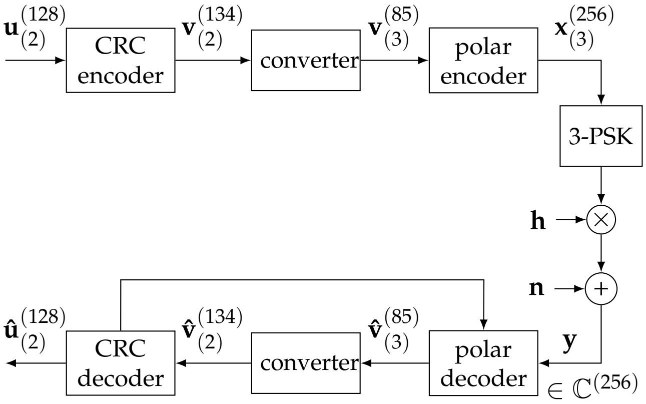
$\mathbf{u}_{(2)}^{(128)}$
CRC encoder
$\mathbf{v}_{(2)}^{(134)}$
converter
$\mathbf{v}_{(3)}^{(85)}$
polar encoder
$\mathbf{x}_{(3)}^{(256)}$
3-PSK
$\mathbf{h}$
$\mathbf{n}$
$\mathbf{y}$
$\in \mathbb{C}^{(256)}$
polar decoder
$\hat{\mathbf{v}}_{(3)}^{(85)}$
converter
$\hat{\mathbf{v}}_{(2)}^{(134)}$
CRC decoder
$\hat{\mathbf{u}}_{(2)}^{(128)}$

Figure 9. Transmission chain with binary source, cyclic redundancy check (CRC)-aided polar code and 3-PSK. Subscripts indicate the alphabet size, superscripts indicate the length of the vectors.

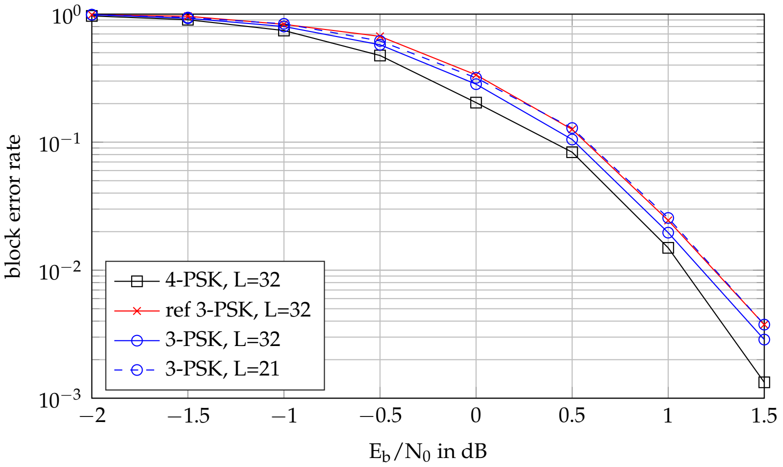

Figure 10. Block error rate for binary and ternary polar codes + CRC.

Interestingly, the ternary polar code with the binary CRC and the converter performs slightly better than the ternary polar code with the ternary CRC. The reason might be the following: The ternary sequence of information and CRC symbols has a length of 85 for the case with converter, while it consists of 80 information symbols and 6 CRC symbols in the all-ternary case. Thus, the polar code which is combined with the binary CRC is slightly stronger.

Compared to the binary polar code with 4-PSK, the ternary polar code with the same list size is 0.1–0.2 dB worse. For the competition, a time limit was given as stated in Section 3.1. The decoding time of the ternary polar code is around 1.5 times the reference time for $L = 32$. To fulfill the time limit, the list size has to be reduced to $L = 21$ which leads to an additional loss of 0.1 dB in the block error rate performance.

6. Comparison to Other Codes

The competition has shown that CRC-aided polar codes outperform LDPC codes and Reed–Solomon codes in the given scenario. With convolutional codes and turbo codes, two well-known classes of error-correcting codes have not been evaluated yet. In this section, we briefly introduce these codes and evaluate their performance to finally answer the question which channel code is the best for short underwater messages in the scenario described in Section 2.

6.1. Tail-Biting Convolutional Code

Convolutional codes are a well-known class of channel codes. In contrast to polar codes, they are very flexible in codeword length as they work stream-based instead of block-wise. The encoder of a classic convolutional code can be viewed as a finite-state shift register circuit with M memory elements where the state depends on the previous input bits. The output depends on the state and the current input bit. To provide the same error protection to all input bits, M extra padding bits are appended to the input stream such that the final state equals the initial state. Due to the padding bits, the code rate is reduced which is negligible for long messages. For short messages, tail-biting convolutional codes are used to completely avoid the rate loss. For tail-biting convolutional codes, the shift register of the encoder is not always initialized with zeros, but in such way that the initial state and final state are equal. For non-recursive codes, this is achieved by initializing the shift register with the last M bits of the input sequence. The procedure to determine the initial state for recursive codes is described in [33].

For the terminated convolutional code, the most likely information word can efficiently be found by the Viterbi algorithm. For the tail-biting convolutional code, 2^M Viterbi decoders would be necessary, one for each possible initial state. However, a close-to-optimum solution can be found by the wrap-around Viterbi algorithm (WAVA) [34]. The WAVA is an iterative decoding scheme based on the Viterbi decoder. In the first iteration, it is similar to the conventional Viterbi decoder, unless it starts not only from the all-zero state, but from all possible states. After each iteration, it is checked whether the maximum likelihood path is a tail-biting path. If this is true, the algorithm stops, else a Viterbi decoder is run with the initial state metrics set to the final state metrics of the previous iteration. A very small number (≈ 3) of allowed iterations is sufficient to achieve close-to-optimum performance.

In the following comparison, a non-recursive tail-biting convolutional code with memory $M = 8$ is decoded using the WAVA algorithm with at most two iterations as hardly any improvement has been observed for three allowed iterations. In the choice of the memory size it has been considered that a longer memory results in a stronger code, but increasing the memory M by one doubles the number of possible states and therefore doubles the complexity. A memory of $M > 8$ would not fulfill the time requirements.

6.2. Turbo Code

Turbo codes consist of two parallel concatenated codes which are often identical recursive convolutional codes with low memory. The order of the information bits is changed by an interleaver Π before they are encoded by the second constituent code so that the constituent encoder outputs differ. The structure of the encoder and decoder is depicted in Figure 11.

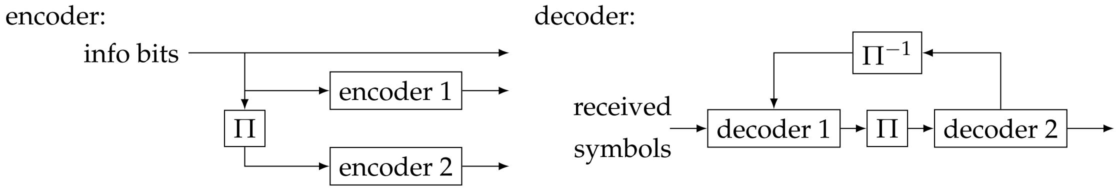

Figure 11. Encoder and decoder of a turbo code.

In the decoder, the two constituent decoders work alternately and exchange so called extrinsic information. Therefore, they have to calculate the a-posteriori probability for each information bit so that the BCJR algorithm instead of the Viterbi algorithm has to be used. Similar to the tail-biting convolutional code, the decoding complexity depends linearly on the number of iterations and exponentially on the memory size.

In this work, the turbo code consists of two identical non-systematic recursive convolutional codes with memory $M = 3$. The shift registers are initialized with zeros, the codes are not terminated. The interleaver is designed such that the last information bits are shifted to positions where they are well-protected by the second convolutional code. The decoding process consists of up to eight iterations. If both constituent codes estimate the same information word, the decoding is stopped earlier. Increasing the number of iterations to more than eight yields only very small improvement of the block error rate.

6.3. Results

The block error rate and the normalized decoding time of the tail-biting convolutional code (TBCC), the turbo code and three of the proposals from the competition are shown in Figures 12 and 13, respectively. All channel codes are combined with 4-PSK modulation.

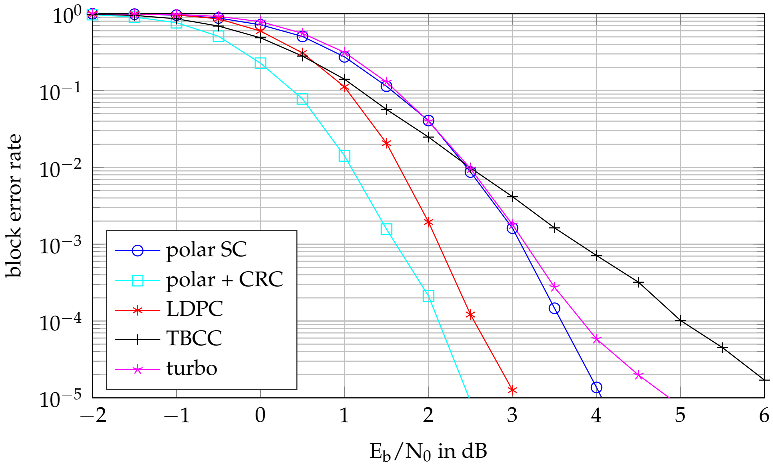

Figure 12. Block error rate.

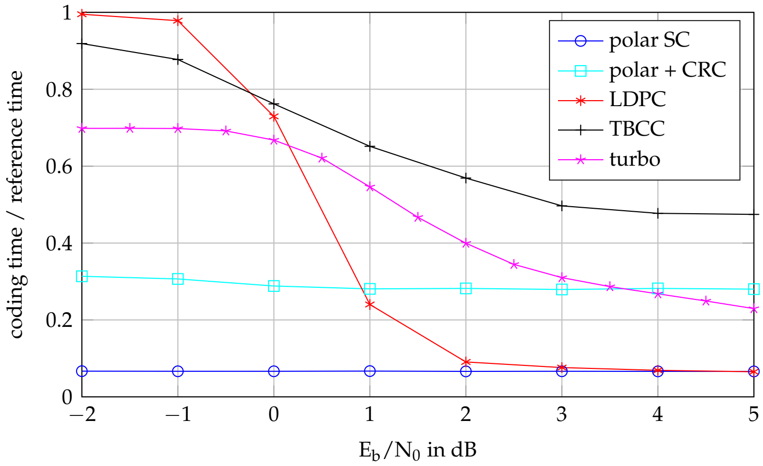

Figure 13. Average decoding time normalized to the reference time.

The tail-biting convolutional code is worse than the CRC-aided polar code in both error rate and decoding time. The slope of its error rate curve is less steep compared to those of the other codes. Hence, it can compete with the LDPC code at very low signal-to-noise power ratios while it is even worse than the simple polar code with successive cancellation decoder for very good channel conditions. The decoding time of the tail-biting convolutional code is always within the given limit. As for the LDPC code, the iterative nature of the decoder is visible: The average decoding time decreases with the E_b/N_0 until it converges to the time for one iteration because the lower the noise, the more often a valid code word is found in the first iteration.

The turbo code is outperformed by the polar code with CRC and the LDPC code in terms of block error rate. The gap to the CRC-aided polar code is around 1.2 dB to 1.5 dB in the waterfall region. Beginning from a signal to noise power ratio of $E_b/N_0 = 4$ dB, the error rate curve flattens out. This can be mitigated by a more severe stopping criterion. The decoding in the low SNR regime is significantly faster for the turbo code than for the other iteratively decoded codes. However, in the more interesting waterfall region, the turbo code cannot compete with the LDPC code neither in decoding time nor in error rate.

7. Conclusions

In this paper, the results of the coding competition "Wanted: Best channel code for short underwater messages" [17] are presented. Besides the candidate codes submitted in the competition, the performance of tail-biting convolutional codes and turbo codes has been evaluated. Furthermore, non-binary polar codes on GF(q), q prime, have been presented and simulation results for polar codes on GF(3), GF(5) and GF(7) with 3-PSK, 5-PSK or 7-PSK, respectively, have been discussed. A CRC-aided polar code with successive cancellation list decoding and 4-PSK modulation has been identified as the best channel code-modulation pair for the given underwater acoustic sensor network scenario. It shows the lowest block error rate of all tested channel codes and has a moderate decoding time which does not depend on the transmission quality. Furthermore, it has been shown that both 4-PSK and 3-PSK are suitable modulations for the transmission of 0.5 bit/channel use in a channel with AWGN and small multiplicative amplitude distortions. However, 4-PSK can be easily combined with binary channel codes which are usually less computationally complex than codes on GF(3) and is therefore preferable in applications.

Author Contributions: Conceptualization, I.N.; software and investigation, M.F.; writing, M.F. and I.N.; review, supervision and project administration, G.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the German Technical Center for Ships and Naval Weapons, Naval Technology and Research (WTD 71), grant number E/E71S/H0531/CF081, project "UnderwaterTRANSEC". The publishing fees were supported by the funding programme *Open Access Publishing* of Hamburg University of Technology (TUHH).

Acknowledgments: The authors would like to thank all participants of the coding competition.

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

References

1. Nissen, I. Burst communication—A solution for the underwater information management. *Hydroacoustics* **2015**, *18*, 113–126.
2. Kebkal, V.; Kebkal, O.; Kebkal, K. Network coding for underwater acoustic sensor networks. In Proceedings of the 2013 MTS/IEEE OCEANS, Bergen, Norway, 10–14 June 2013; pp. 1–5. doi:10.1109/OCEANS-Bergen.2013.6607957.
3. Sprea, N.; Bashir, M.; Truhachev, D.; Srinivas, K.V.; Schlegel, C.; Sacchi, C. BATS Coding for Underwater Acoustic Communication Networks. In Proceedings of the OCEANS 2019, Marseille, France, 17–20 June 2019; pp. 1–10. doi:10.1109/OCEANSE.2019.8867299.
4. Wang, H.; Wang, S.; Zhang, E.; Zou, J. A network coding based hybrid ARQ protocol for underwater acoustic sensor networks. *Sensors* **2016**, *16*, 1444.
5. Barreto, G.; Simão, D.; Pellenz, M.; Souza, R.; Jamhour, E.; Penna, M.; Brante, G.; Chang, B. Energy-Efficient Channel Coding Strategy for Underwater Acoustic Networks. *Sensors* **2017**, *17*, 728.
6. Behgam, M.; Zheng, Y.R.; Liu, Z. Coding for Short Messages in Multipath Underwater Acoustic Communication Channels. In Proceedings of the OCEANS 2018 MTS/IEEE Charleston, Charleston, SC, USA, 22–25 October 2018; pp. 1–5. doi:10.1109/OCEANS.2018.8604711.

7. Goalic, A.; Trubuil, J.; Beuzelin, N. Channel Coding for Underwater Acoustic Communication System. In Proceedings of the OCEANS 2006, Boston, MA, USA, 18–21 September 2006; pp. 1–4. doi:10.1109/OCEANS.2006.307093.
8. Xu, X.; Chen, Y.; Zhang, L.; Feng, W. Comparison of the performance of LDPC codes over different underwater acoustic channels. In Proceedings of the 2010 IEEE 12th International Conference on Communication Technology, Nanjing, China, 11–14 November 2010; pp. 155–158. doi:10.1109/ICCT.2010.5689306.
9. Wu, Y.; Zhu, M.; Zhu, W.; Xing, Z.; Xu, L.; Bo, Y.. Nonbinary LDPC code for noncoherent underwater acoustic communication and its experiment results. In Proceedings of the 2013 OCEANS, San Diego, CA, USA, 23–27 September 2013; pp. 1–5. doi:10.23919/OCEANS.2013.6741109.
10. Zhang, L.; Xu, X.; Sun, H.; Chen, Y. Performance Analysis of IRA Codes for Underwater Acoustic OFDM Communication System. In Proceedings of the 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing, Beijing, China, 24–26 September 2009; pp. 1–4. doi:10.1109/WICOM.2009.5302465.
11. Liu, L.; Wang, Y.; Li, L.; Zhang, X.; Wang, J. Design and implementation of channel coding for underwater acoustic system. In Proceedings of the 2009 IEEE 8th International Conference on ASIC, Changsha, China, 20–23 October 2009; pp. 497–500. doi:10.1109/ASICON.2009.5351627.
12. Chen, P.; Bai, B.; Ma, X. Two-stage polarization-based nonbinary polar codes for 5G URLLC. *arXiv* **2019**, arXiv:1801.08059v2 .
13. Yuan, P.; Steiner, F. Construction and Decoding Algorithms for Polar Codes based on 2 × 2 Non-Binary Kernels. In Proceedings of the 2018 IEEE 10th International Symposium on Turbo Codes Iterative Information Processing (ISTC), Hong Kong, China, 3–7 December 2018; pp. 1–5. doi:10.1109/ISTC.2018.8625284.
14. Goetz, M.; Nissen, I. GUWMANET—Multicast Routing in Underwater Acoustic Networks. In Proceedings of the Communications and Information Systems Conference (MCC), 2012 Military, Gdansk, Poland, 8–9 October 2012, IEEE: Piscataway, NJ, USA , 2012; pp. 1–8.
15. Van Walree, P.A. Propagation and Scattering Effects in Underwater Acoustic Communication Channels. *IEEE J. Ocean. Eng.* **2013**, *38*, 614–631. doi:10.1109/joe.2013.2278913.
16. Van Walree, P.A.; Socheleau, F.; Otnes, R.; Jenserud, T. The Watermark Benchmark for Underwater Acoustic Modulation Schemes. *IEEE J. Ocean. Eng.* **2017**, *42*, 1007–1018. doi:10.1109/JOE.2017.2699078.
17. Falk, M.; Nissen, I.; Bauch, G. Wanted: Best Channel Codes for Short Underwater Messages. 2018. Available online: researchgate.net (accessed on 15 April 2019) .
18. Reed, I.S.; Solomon, G. Polynomial codes over certain finite fields. *J. Soc. Ind. Appl. Math.* **1960**, *8*, 300–304.
19. Gallager, R. Low-density parity-check codes. *IRE Trans. Inf. Theory* **1962**, *8*, 21–28. doi:10.1109/TIT.1962.1057683.
20. Davey, M.C.; MacKay, D.J.C. Low density parity check codes over GF(q). In *1998 Information Theory Workshop (Cat. No.98EX131)*; IEEE: Piscataway, NJ, USA; 1998; pp. 70–71. doi:10.1109/ITW.1998.706440.
21. 3GPP TS 38.212 V15.0.0: Multiplexing and Channel Coding. 2017. Available online: 3gpp.org (accessed on 25 November 2019).
22. Kasai, K.; Declercq, D.; Poulliat, C.; Sakaniwa, K. Multiplicatively Repeated Nonbinary LDPC Codes. *IEEE Trans. Inf. Theory* **2011**, *57*, 6788–6795. doi:10.1109/TIT.2011.2162259.
23. Arikan, E. A performance comparison of polar codes and Reed-Muller codes. *IEEE Commun. Lett.* **2008**, *12*, 447–449. doi:10.1109/LCOMM.2008.080017.
24. Tal, I.; Vardy, A. List Decoding of Polar Codes. *IEEE Trans. Inf. Theory* **2015**, *61*, 2213–2226. doi:10.1109/TIT.2015.2410251.
25. Yuan, P.; Prinz, T.; Boecherer, G.; Iscan, O.; Boehnke, R.; Xu, W. Polar Code Construction for List Decoding. In Proceedings of the SCC 2019; 12th International ITG Conference on Systems, Communications and Coding, Rostock, Germany, 11–14 February 2019; pp. 1–6. doi:10.30420/454862022.
26. Arikan, E. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Trans. Inf. Theory* **2009**, *55*, 3051–3073. doi:10.1109/TIT.2009.2021379.
27. Trifonov, P. Efficient Design and Decoding of Polar Codes. *IEEE Trans. Commun.* **2012**, *60*, 3221–3227. doi:10.1109/TCOMM.2012.081512.110872.

28. Tal, I.; Vardy, A. How to Construct Polar Codes. *IEEE Trans. Inf. Theory* **2013**, *59*, 6562–6582. doi:10.1109/TIT.2013.2272694.
29. Stark, M.; Shah, A.; Bauch, G. Polar code construction using the information bottleneck method. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Barcelona, Spain, 15–18 April 2018; pp. 7–12. doi:10.1109/WCNCW.2018.8368978.
30. Elkelesh, A.; Ebada, M.; Cammerer, S.; ten Brink, S. Genetic Algorithm-based Polar Code Construction for the AWGN Channel. In Proceedings of the SCC 2019; 12th International ITG Conference on Systems, Communications and Coding, Rostock, Germany, 11–14 February 2019; pp. 1–6. doi:10.30420/454862007.
31. Balatsoukas-Stimming, A.; Parizi, M.B.; Burg, A. LLR-Based Successive Cancellation List Decoding of Polar Codes. *IEEE Trans. Signal Process.* **2015**, *63*, 5165–5179. doi:10.1109/TSP.2015.2439211.
32. Sasoglu, E.; Telatar, E.; Arikan, E. Polarization for arbitrary discrete memoryless channels. In Proceedings of the 2009 IEEE Information Theory Workshop, Taormina, Italy, 11–16 October 2009; pp. 144–148. doi:10.1109/ITW.2009.5351487.
33. Weiss, C.; Bettstetter, C.; Riedel, S. Code construction and decoding of parallel concatenated tail-biting codes. *IEEE Trans. Inf. Theory* **2001**, *47*, 366–386. doi:10.1109/18.904537.
34. Shao, R.; Lin, S.; Fossorier, M. Two decoding algorithms for tailbiting codes. *IEEE Trans. Commun.* **2003**, *51*, 1658–1665. doi:10.1109/tcomm.2003.818084.