Super Permutation Frequency-Shift-Keyed Underwater Acoustic Communication

Viktor Lidström

***Abstract*—A method that defines the superset of Mary permutation frequency-shift keying (SPFSK) alphabets as the symbol space is investigated as a possible robust data link for the underwater acoustic (UWA) channel. The alphabet is equivalent to ON–OFF keying for $M = 1$. Polar coding and time/frequency guards are used for redundancy. The bit error probability of an SPFSK symbol is evaluated to construct the Polar code; it is found to be uniform across bit positions in the label by using natural labeling due to a symmetric fractal behavior in the pairwise symbol error probability, and the heuristic code construction method is directly applicable. A Ricean noncoherent data model is used, and a reduced complexity algorithm for calculating binary log-likelihoods is provided. The method is simulated in an additive white Gaussian noise channel to find a good Polar code construction and evaluate different alphabet parameterizations; the results show that the receiver statistics improve with the alphabet size. SPFSK is further evaluated in the public Watermark benchmark, using both genie- and pilot-assisted parameter tracking, as well as perfect and nonperfect synchronization. Results from the Watermark replay channels NOF1, NCS1, and BCH1 show that $\lesssim 1\%$ frame error rate is attained at $16 \leq E_b/N_0 \leq 23$ dB depending on the channel, using ~ 0.2 bit/s/Hz spectral efficiency, indicating its potential as a robust data link. Performance in the UWA channel increases substantially by using many tones, and the statistical model must be sufficiently accurate to leverage the processing gain of an SPFSK alphabet with higher dimensionality.**

***Index Terms*—Noncoherent, permutation frequency-shift keying, Polar code, robust data link, Watermark.**

I. Introduction

While the availability of terrestrial and space communication has grown substantially in recent times, the same is not true for underwater communication. Tethers are frequently used, which are heavy and expensive to deploy, and absorption by the water prohibits the use of standard wireless signaling, like electromagnetic and optical waves, for distances > 100 m. The underwater acoustic (UWA) channel permits communication at a very long range; however, it suffers from increasing absorption with frequency, limiting the available bandwidth. In addition, the propagation speed is low, ~ 1500 m/s, and varies with depth, creating in situ unpredictable propagation paths due to refraction in the medium, which can produce shadow zones. The channel is characterized by time-varying multipath due to reflections from surface waves, the sea bottom, and other bathymetric features like islands and introduces Doppler shifts even if the modems are stationary [1]. The ocean background noise is often modeled as colored Gaussian, whose power is affected by the sea state and weather. Site-specific noise sources, e.g., harbors, ships, construction, ice cracking, and fauna, can be colored, non-Gaussian, and impulsive. In addition, reverberation can be regarded as self-noise, sharing the same spectral characteristics as the transmitted signal, and a platform may have several acoustic instruments that share the available bandwidth [2, Ch. 4].

The UWA channel is typically modeled as a time-varying linear filter $c(\tau;t)$, which is the response at time t due to an impulse at $t-\tau$. A transmitted signal, $s(t)$, is received as [3, Ch. 1]

$$r(t) = c(\tau;t) * s(t) + w(t) \tag{1}$$

where $w(t)$ is here assumed to be a zero mean Gaussian process, and $c(\tau;t)$ includes the transmitter and receiver impulse response. One often class methods as either *coherent*, where information is mapped to phase shifts in $s(t)$, requiring the tracking of phase shifts in $c(\tau;t)$, or *noncoherent*, where only the envelope of $c(\tau;t)$ is of consequence, a reason for which it is generally referred to as a robust class of methods. This is a tradeoff with the method's rate, $R = R_0 B$, for a bandwidth B and spectral efficiency R_0, and receiver gain in an ideal additive white Gaussian noise (AWGN) channel, which are both higher for coherent methods [3, Chs. 3, 4].

Early UWA link research focused on noncoherent links; however, since the 1990s, coherent methods have been successfully demonstrated [4] (see, e.g., [5] and [6]). Recent advances have mainly been made in coherent methods, in particular by using variants of orthogonal frequency-division multiplexing (OFDM), decision feedback equalization (DFE) combined with iterative decoding, and spatial diversity [7].

As detailed in [8], many noncoherent methods were using variations of multicarrier (MC) frequency-shift keying (MFSK) with $M = \{2, 4\}$ tones by the late 1990s (see, e.g., [9], [10], and [11]) because of $R_0 = \log_2(M)/M = 0.5$ bit/s/Hz during signal transmission, which is the highest possible for MFSK. Common strategies for improving the signal-to-noise ratio (SNR) include placing a guard time T_g after each tone and a guard band B_g between each tone band, thereby reducing the intersymbol and intercarrier interference. A similar effect is achieved by choosing a spectral null placement $C_{\varnothing} > 1$ for

Manuscript received 28 September 2022; revised 29 May 2023 and 29 August 2023; accepted 12 September 2023. This work was supported by the Swedish Foundation for Strategic Research.

Associate Editor: M. Stojanovic.

The author is with the KTH Royal Institute of Technology, 114 28 Stockholm, Sweden, and also with the FOI Swedish Defence Research Agency, 164 90 Stockholm, Sweden (e-mail: viklid@kth.se).

Digital Object Identifier 10.1109/JOE.2023.3316706

the tone frequencies (see, e.g., [12]). More recent contributions have aimed at improving the interoperability and reliability in a multiuser scenario by using frequency hopping, like in the Janus standard [13], and achieving error-free communication in a non-line-of-sight scenario [14].

While noncoherent methods are generally considered to be robust, the $R_0 = 0.5$ bit/s/Hz limit on spectral efficiency is a drawback of MFSK, compared to coherent methods, which can increase R_0 by using a larger phase-shift alphabet. A natural question of interest is: Can the bitrate of robust methods be increased by another member in the noncoherent class of methods? By concatenating two 2FSK alphabets and letting all permutations form a new alphabet, one obtains $R_0 = \log_2\{\binom{4}{2}\}/4 = 0.6462$ bit/s/Hz, larger than that of MFSK. Forming an alphabet by taking all permutations of a sequence was first described in [15], and defining the sequence as the transmission of N out of M tones has been termed permutation frequency-shift keying (PFSK) [16], which is here denoted as $\binom{M}{N}$PFSK. PFSK has seen some recent use in the UWA channel [17], [18]; however, to the author's knowledge, no example of defining the alphabet as the superset of *all* PFSK alphabets has been investigated in the literature. Concatenating the alphabets of $\binom{M}{N}$PFSK, for $0 \leq N \leq M$, the new alphabet is named a *super* permutation alphabet, and its use in FSK as super permutation frequency-shift keying (SPFSK). It has several interesting properties, in particular a maximum spectral efficiency of $R_0 = 1$ bit/s/Hz. The feasibility of using SPFSK as a *noncoherent* UWA data link is the *primary focus* of this work, where the higher bitrate is complemented by strong channel coding, longer frame lengths, and the inherent robustness of noncoherent signaling to the UWA channel.

Due to the low propagation speed of acoustic signals, a short package time length is typically required to avoid collisions in an UWA network. Meanwhile, the low bandwidth of the UWA channel limits the available bitrate, necessitating the use of *short* coded frames for error correction. Polar codes have been shown to have an AWGN channel performance that is superior to contenders like low-density parity check (LDPC) and Turbo codes for short frames [19] [20] and have been selected for the control channel of the 5G standard [21]. For a longer 1024-bit frame, the Polar code can be parameterized to have as good performance as an LDPC code, at a lower complexity [20]; these attractive properties motivate choosing Polar codes for channel coding. Polar codes constitute a recent group of codes that recursively combine a frame of N bits prior to transmission, which then *polarize* to unity- or zero-capacity *bit channels*, when decoded at the receiver. Information bits are mapped to the K best bit channels, and the remaining $(N - K)$ bits are populated with *frozen* (known) bits. The reliability is identified by evaluating the error probability of each decoded bit channel, for a suitable channel model, which is termed the *Polar Code Construction Problem*. Constructing a good Polar code requires the consideration of the underlying signaling method; the super permutation alphabet becomes large even for modest values of M, making evaluation of the bit channels nontrivial. The application of Polar coding to SPFSK is, therefore, the *secondary focus* of this work. The code requires a soft input, which necessitates calculating binary likelihoods from the potentially large SPFSK symbol space; therefore, a reduced-complexity algorithm specific to the super permutation alphabet is also provided.

A challenge in methodology is defining a model that accurately describes all instances of the UWA channel that one might encounter, an exhibition of which can be found in [22], in terms of realistic performance; therefore, verification through field testing remains a de facto standard [23]. Since these are resource intensive, the number of measurements might be limited, and if postprocessing is used, the control over SNR is often poor. The resulting sparse "random" channel selection, together with the wide range of channels one might encounter, makes it difficult to compare between methods in the literature.

In replay simulation, a method is simulated by convolving with multiple channel estimates recorded in situ. While having the drawback of estimates being noisy, and that overspread channels are impossible to estimate without errors[1] [25], the channel recordings can be made available as a benchmark, making results *comparable* between publications. Since field testing is resource intensive, replay simulation is an effective way to generate promising UWA link candidates. Watermark [26] is a publicly available benchmark that enables replay of five recorded UWA channels and sets guidelines on publishing results for ease of comparison and is chosen here as the main method of performance validation.

The primary interest is in the feasibility of Polar-coded SPFSK. To that end, the effect of different alphabet parameterizations on the soft decoder input is first studied in a simple AWGN channel. Continuing to the Watermark UWA replay channels, the impact of imperfect knowledge of the likelihood model and the impact of *Doppler spreading* on the more energy-dense signaling spectrum, compared to MFSK, are in particular focus. For the convenience of conciseness, the methods are studied without a fixed time–Doppler shift, which are minor problems relative to the aforementioned issues of channel time variability; the detection and correction of linear time–Doppler shifts have been previously studied for MFSK, and the approach presented in [27] is directly applicable here. The three Watermark channels, i.e., NOF1, NCS1, and BCH1, have impulse responses with a stable energetic main tap, enabling the start time to be set manually; therefore, these are used in the evaluation.

The rest of this article is organized as follows. Section II introduces notation and defines the super permutation alphabet. Section III provides an overview of the SPFSK method. Section IV describes the receiver data model. Section V describes how parameters are obtained in the genie-aided and pilot estimation cases. Section VI introduces the Polar code and evaluates the bit error probability of SPFSK, relevant to code construction. Section VII presents and discusses results from AWGN and Watermark simulations. Finally, Section VIII concludes this article.

[1] Careful design of experiments can improve reusability of data, for comparison between multiple methods postmeasurement, and incorporating the estimation error into the noise model to improve playback simulation has been suggested in [24].

TABLE I
SPFSK Alphabet With $M = 2$ and Average Symbol Power $P_s = 3/4$.

$\mathcal{S}$	0	1	0	0.7071
	0	0	1	0.7071
$\boldsymbol{q}$	0	1	2	3

II. Super Permutation Alphabet

Notation: A bit is a binary variable, e.g., $b, c, u \in \mathbb{F}_2$, and a symbol $\boldsymbol{s}$ or $\boldsymbol{S}_q$, in boldface, a column vector with elements $\in \mathbb{R}$, where $[\cdot]$, or subscript s_i, indexes a discrete variable, starting from 0; $(\cdot)$ is a discrete-to-discrete or continuous-to-continuous mapping, and $\{\cdot\}$ refers to a set; $[\cdot,\cdot]$ and $(\cdot,\cdot)$ signify closed- and open intervals, respectively. In the context of concatenating vectors, $[\boldsymbol{s}, \boldsymbol{s}]$ is a matrix and $[\boldsymbol{s}^T, \boldsymbol{s}^T]^T$ is an extended column vector.

This section defines the symbol alphabet, $\mathcal{S}$, used for SPFSK. Although one can use the PFSK alphabet definition to define $\mathcal{S}$ as the superset, a simpler definition is available: let $\boldsymbol{u}(q) = \{u_m(q) \in \mathbb{F}_2 : m \in [0, M-1]\}$ be the *label* of an index q, written as a column vector; $\boldsymbol{u}(q)$ is a binary representation of the integer value q, where $q \in [0, Q-1]$, and $Q = 2^M$. Since $\boldsymbol{u}$ can be any binary sequence of length M,

$$\boldsymbol{S}_q = \begin{cases} \mathbf{0}, & q = 0 \\ \sqrt{\frac{QP_s}{(Q-1)}} \cdot \boldsymbol{u}(q)/||\boldsymbol{u}(q)||, & 1 \le q < Q \end{cases} \tag{2}$$

generates SPFSK alphabet $\mathcal{S} = \{\boldsymbol{S}_q : 0 \le q < Q\}$, where $||\boldsymbol{u}(q)||$ is a vector norm that normalizes the symbol by the number of nonzero elements. Each symbol carries M bits of information, and if one selects symbols at random, with $\mathrm{P}(\text{choose } \boldsymbol{S}_q) = 1/Q$, the scalar $(Q/(Q-1))^{1/2}$ ensures that the average power of the sequence is P_s. Enforcing the same symbol power across the alphabet is important for balancing the symbol decision regions. The presence of $\mathbf{0}$ in the alphabet has a power benefit for nonzero symbols, which is greatest for $M = 1$ and vanishingly small as M increases. Define a *labeling* to be a row vector of indices, where $\boldsymbol{q} = [0, 1, \ldots, Q-1]$ is termed *natural labeling*; let $\mathcal{S}(q) = \boldsymbol{S}_q$ denote the qth symbol in the alphabet, and $\mathcal{S}(\boldsymbol{q}) \triangleq \mathcal{S}$ the symbol alphabet indexed in the order defined in (2). The SPFSK alphabet is essentially the binary representation of natural labeling, with a power scaling, and an example is given in Table I. Define $\pi(\cdot)$ as a permutation operator that reorders indices according to some specification. For example, if π_G is defined as Gray labeling [28], where neighboring labels only differ in one bit, then $\mathcal{G} = \mathcal{S}(\pi_G(\boldsymbol{q}))$ is the reordered alphabet; when discussing labeling, the reordering of $\mathcal{S}$ by some definition of $\pi(\cdot)$ is what is referred to.

If $\mathcal{S}$ is used as an FSK alphabet, the spectral efficiency is clearly $R_0 = \log_2(2^M)/M = 1$ bit/s/Hz, regardless of the choice of M. It is also clear that this is the *highest achievable* for an FSK-type method with a fixed symbol power, i.e., without amplitude modulation, since the alphabet contains all permutations of M elements.

Note that an SPFSK alphabet with $M = 1$ yields the on–off keying (OOK) (sometimes called OFDM-OOK) alphabet. Hence, an FSK modulation scheme with M carriers can either use an SPFSK alphabet with dimensionality M or map M OOK symbols to the carriers (MC-OOK) (see, e.g., [29]). These two approaches have an equivalent spectral efficiency, and if all symbols are transmitted with equal probability, they both produce the same average number of active carriers per transmission. The question of interest is whether the receiver statistics benefit from a symbol alphabet with higher dimensionality. Specifically, if the statistics improve by considering the M amplitudes jointly, i.e., MSPFSK, or independently, as in OOK on M carriers. Considering the highest rate contender in traditional FSK, i.e., 4FSK, one notes that the 4FSK alphabet is contained within a 4SPFSK alphabet. Since the 4SPFSK symbol space is more densely populated, the intersymbol distance will be smaller and noise sensitivity higher; it is, therefore, of interest to investigate the robustness penalty of doubling the information rate. The same reasoning applies to MSPFSK for larger M as several 4FSK symbols can be placed on parallel carriers to match M. Hence, both OOK and 4FSK are used for comparison in Section VII, where the implementation of 4FSK is similar to [30]. On a final note, the elements of $\mathcal{S}$ can be regarded as complex envelopes of zero-phase complex amplitudes in the baseband signal mapping described in the next section. Therefore, $\mathcal{S}$ could be extended by symbols with nonzero phases, increasing spectral efficiency by making the method semicoherent and more similar to OFDM. This would make the method more sensitive to the phase shifts of the channel, requiring a more complicated receiver, and is left for future work.

III. Super Permutation Frequency-Shift Keying

A diagram of the communication chain is shown in Fig. 1: information bits $\boldsymbol{b} = [b_0, b_1, .., b_{K-1}]^T$ enter the transmitter and are mapped to a Polar codeword, $\boldsymbol{c} = [c_0, c_1, .., c_{N-1}]^T$, which defines the code rate, $R_c = K/N$. The information input is assumed to be source encoded to maximum entropy, and $\boldsymbol{b}$ is modeled as the realization of K independent $\mathrm{Bernouilli}(0.5)$ variables. Elements of $\boldsymbol{c}$ are interleaved, which are denoted by $\mathcal{I}(\cdot)$, and $\mathcal{I}(\boldsymbol{c})$ is divided into $N_{\boldsymbol{s}} = \lceil N/\log_2 Q \rceil$ subvectors or *bit sets*. Each bit set represents an integer q, which is mapped to the symbol $\boldsymbol{s} = \boldsymbol{S}_q$. This produces a sequence of vector-valued symbols, $(\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{N_{\boldsymbol{s}}-1})$, where $\boldsymbol{s} \in \mathcal{S} = \{\boldsymbol{S}_q : 0 \le q < Q\}$; $\mathcal{S}$ is defined such that picking the qth symbol randomly from the alphabet, $\|\boldsymbol{S}_q\|^2 = P_s$, where P_s is the symbol power. Note that the uppercase $\boldsymbol{S}_q$ denotes a symbol in the alphabet, and the lowercase $\boldsymbol{s}$ a symbol chosen for transmission across the channel.

Symbols are grouped into *super* symbols $\mathfrak{s}$, each of which contain Y concatenated symbols, i.e., $\mathfrak{s} = [\boldsymbol{s}_0^T, \boldsymbol{s}_1^T, \ldots, \boldsymbol{s}_{Y-1}^T]^T$ is an extended column vector of length YM. Elements of $\mathfrak{s}$ are mapped to the amplitudes of YM complex sinusoids, or *tones*, in a baseband symbol

$$s_b(t) = \sum_{m=0}^{YM-1} \mathfrak{s}[m] \exp(j2\pi f_m t) \tag{3}$$

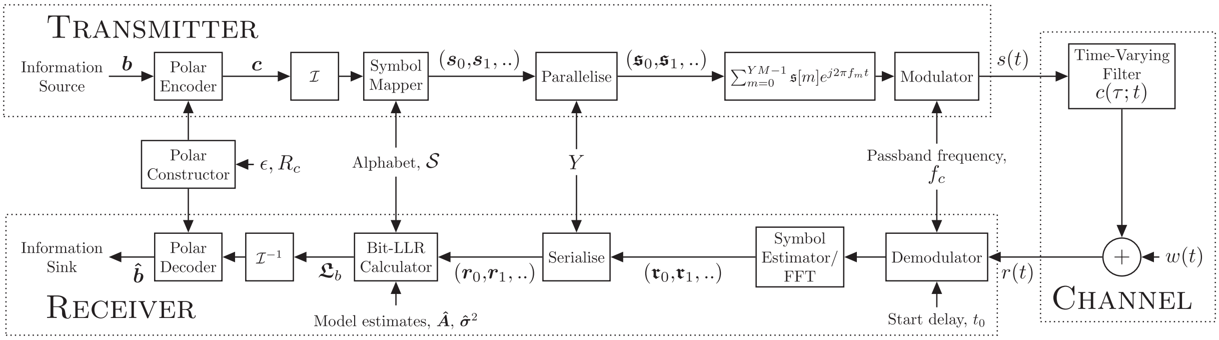

TRANSMITTER
Information Source
$\boldsymbol{b}$
Polar Encoder
$\boldsymbol{c}$
$\mathcal{I}$
Symbol Mapper
$(\boldsymbol{s}_0, \boldsymbol{s}_1, ..)$
Parallelise
$(\boldsymbol{\mathfrak{s}}_0, \boldsymbol{\mathfrak{s}}_1, ..)$
$\sum_{m=0}^{YM-1} \mathfrak{s}[m] e^{j2\pi f_m t}$
Modulator
$s(t)$
Time-Varying Filter $c(\tau; t)$
Polar Constructor
ϵ, R_c
Alphabet, $\mathcal{S}$
Y
Passband frequency, f_c
$+$
$w(t)$
Information Sink
$\hat{\boldsymbol{b}}$
Polar Decoder
$\mathcal{I}^{-1}$
$\mathfrak{L}_b$
Bit-LLR Calculator
$(\boldsymbol{r}_0, \boldsymbol{r}_1, ..)$
Serialise
$(\boldsymbol{\mathfrak{r}}_0, \boldsymbol{\mathfrak{r}}_1, ..)$
Symbol Estimator/ FFT
Demodulator
$r(t)$
RECEIVER
Model estimates, $\hat{\boldsymbol{A}}, \hat{\boldsymbol{\sigma}}^2$
Start delay, t_0
CHANNEL

Fig. 1. System diagram.

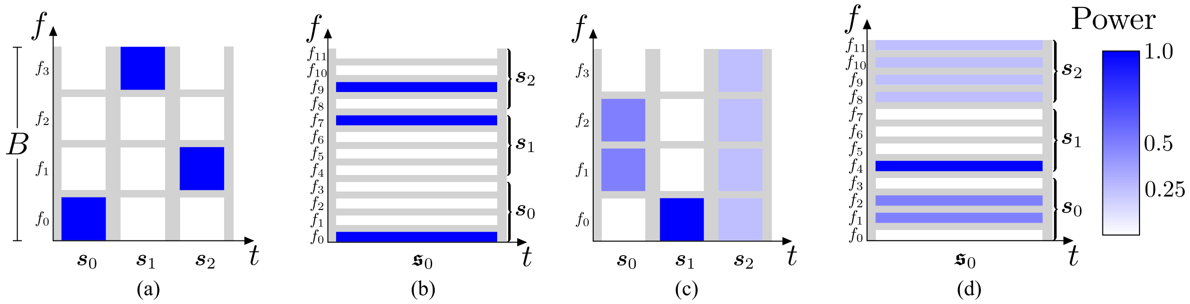


Fig. 2. Frequency parallel signaling, where three example symbols with $M = 4$ elements are shown as (a) 4FSK, (b) equivalent 3×4FSK, (c) 4SPFSK, and (d) equivalent 3×4SPFSK, for $Y = 3$; the transmission slot of an active tone is shown in color, and time/frequency padding in gray. The power scaling for SPFSK tones is due to the P_s symbol power constraint in (2). In (c), $\boldsymbol{s}_0$ can be regarded as a $\binom{4}{2}$PFSK symbol, $\boldsymbol{s}_1$ as a 4FSK symbol, and $\boldsymbol{s}_2$ as a $\binom{4}{4}$PFSK symbol; since SPFSK is the superset, all PFSK and MFSK symbols are represented in the SPFSK alphabet.

where the frequency is

$$f_m = \frac{B_t - B}{2} + m(B_t C_\varnothing + B_g) \tag{4}$$

for a total bandwidth B and tone subband B_t, given by

$$B_t = \frac{B - (YM - 1)B_g}{1 + (YM - 1)C_\varnothing}. \tag{5}$$

The guard band $B_g = F_g f_c / C_w$ Hz is given in terms of F_g throughout this article, the frequency guard in m/s, for an assumed sound speed $C_w = 1500$ m/s and passband center frequency f_c. The guard band is an additional frequency separation to the relative null placement, $C_\varnothing$, which is the number of B_t separating the tone frequencies if a square window is used; $C_\varnothing$ is useful for defining the minimum number of spectral nulls separating the tones. The use of super symbols divides B into Y subbands, which is done to separate the design of the symbol alphabet from the tone bandwidth. Throughout this article, $Y \times M$FSK denotes Y bands with an underlying Mary FSK, and $Y \times M$SPFSK likewise for SPFSK; notation of the form MFSK, MSPFSK, and OOK refers to the alphabet being used. An example of transforming $1 \times M$ to $Y \times M$ signaling is shown in Fig. 2. The information rate R_i using the signaling described by (2)–(4) and code rate R_c is

$$R_i = R_c \frac{YM}{\frac{1+(YM-1)C_\varnothing}{B-(YM-1)B_g} + T_g} \tag{6}$$

which is only defined if $B - (YM - 1)B_g > 0$.

A frame is formed from $N_{\mathfrak{s}} = \lceil N/(Y \log_2 Q) \rceil$ baseband/super symbols, where $\lceil \cdot \rceil$ is performed if padding is required in the codeword-to-symbol or symbol-to-super-symbol mapping. Each baseband symbol is multiplied with $\exp(j2\pi f_c t)$, and with a Hanning window $\{h_w(t) : h_w(t) = 0 \; \forall \; t \notin (0, T_s)\}$, to suppress tone sidelobes and improve orthogonality in the presence of Doppler spreading. The complete passband signal

$$s(t) = \sum_{i=0}^{N_{\mathfrak{s}}-1} s_{p,i}\left(t - i(T_s + T_g)\right) \tag{7}$$

is composed of $T_s = 1/B_t$ length signals, appended by a T_g time guard, where $s_{p,i}(t) = 2\,\mathrm{Re}(h_w(t) s_{b,i}(t) \exp(j2\pi f_c t))$ is the ith passband symbol.

The received signal $r(t)$, resulting from (1), enters the receiver and is brought to a baseband low-pass filter, and any linear Doppler shift v_0 is assumed to have been removed. Received super symbols, $\boldsymbol{\mathfrak{r}}$, are estimated with a fast Fourier transform (FFT) and serialized, i.e., $\boldsymbol{\mathfrak{r}}_0 \to (\boldsymbol{r}_0, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_{Y-1})$, to the Mary symbol sequence $(\boldsymbol{r}_0, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_{N_{\mathfrak{s}}-1})$. The likelihood of a received symbol, $\boldsymbol{r}$, denoted by $\boldsymbol{L}_s = \{p(\boldsymbol{r}|\boldsymbol{S}_q) : 0 \le q < Q\}$, is a Qary vector calculated using a Rice fading model. Each symbol likelihood is transformed to $\log_2(Q)$ binary log-likelihood ratios (LLRs), which are concatenated into an N-length binary LLR vector, $\boldsymbol{\mathfrak{L}}_b$. Subsequently, the binary LLRs are deinterleaved and decoded using a Polar Cyclic Redundancy Check (CRC) List Decoder.

IV. Noncoherent Data Model

Received tones widen in frequency because of the Doppler spreading caused by the UWA channel; however, due to the $B_tC_\varnothing + B_g$ frequency separation in (4), they are assumed to vary *independently*. After demodulation and sampling at rate F_s, each tone is modeled in complex baseband as

$$\tilde{r}[n] = \tilde{A}\exp\left(j2\pi f n/F_s\right) + \tilde{w}[n] \tag{8}$$

where the tilde $(\tilde{\cdot})$ denotes a complex random variable (RV), $\tilde{A} \sim \mathcal{CN}(A, \sigma_A^2)$, $\tilde{w}[n] \sim \mathcal{CN}(0, \sigma_{\tilde{w}}^2)$, and $0 \leq n < N_s$; $A = A'e^{j\phi}$ is a complex deterministic amplitude with an unknown phase shift ϕ and scaled amplitude A'; f is a known frequency since any linear Doppler shift is assumed removed. Equation (8) describes a Ricean fading model, and the estimates of $|A|$, σ_A^2, and $\sigma_{\tilde{w}}^2$ are denoted by $\hat{A}$, $\hat{\sigma}_A^2$, and $\hat{\sigma}_{\tilde{w}}^2$, respectively; their estimation is described in Section V. These estimates are viewed as deterministic constants obtained outside the context of information symbols. When a symbol with ν nonzero elements is transmitted, the amplitude of an active tone has the scaled distribution $\mathcal{CN}(A/\sqrt{\nu}, \sigma_A^2/\nu)$, since the transmitter power is split over ν elements.

This data model motivates the following stochastic model: for an RV defined as

$$R = |\alpha + N_0 + jN_1|, \qquad \begin{array}{l} N_0, N_1 \sim \mathcal{N}(0, \sigma^2) \\ \alpha \geq 0 \text{ is a constant} \end{array} \tag{9}$$

the probability density function (PDF) can be derived as

$$p_R\left(r; \alpha, \sigma^2\right) = \frac{r}{\sigma^2}\exp\left(-\frac{r^2+\alpha^2}{2\sigma^2}\right) I_0\left(\frac{\alpha r}{\sigma^2}\right) \tag{10}$$

where $I_0(\cdot)$ is the zeroth-order Bessel function of the first kind, and R is said to be Rayleigh-distributed if $\alpha = 0$, and Rice-distributed otherwise. The received symbol, which is the absolute value of an FFT output, is modeled as

$$R[m] = \left|\hat{A}[m]\frac{S_q[m]}{\sqrt{P_s}} + W_0[m] + jW_1[m]\right| \tag{11}$$

when a symbol $\boldsymbol{S}_q$ with ν nonzero elements is transmitted, where $0 \leq m < M$; this constitutes a set of independent RVs, which follows from the independence assumption on the tones. Here, $W_0, W_1 \sim \mathcal{CN}(0, \hat{\sigma}^2[m]/2)$, and

$$\hat{\sigma}^2[m] = \hat{\sigma}_A^2[m]\frac{(S_q[m])^2}{P_s} + \frac{\hat{\sigma}_{\tilde{w}}^2}{N}. \tag{12}$$

Note that $\nu \cdot (S_q[m])^2 = P_s$ if $S_q[m] \neq 0$, so the estimates of $\hat{A}$ and $\hat{\sigma}_A^2$ are scaled by $1/\sqrt{\nu}$ and $1/\nu$, respectively; if $S_q[m] = 0$, (11) describes the absolute value of a Complex Gaussian RV with variance $\hat{\sigma}_{\tilde{w}}^2/N$. Comparing (9) and (11), $R[m]$ is a Ricean RV, and the likelihood of a received symbol $\boldsymbol{r}$ is

$$p\left(\boldsymbol{r}; \boldsymbol{S}_q\right) = \prod_{m=0}^{M-1} p_{R[m]}\left(r[m]; \hat{A}[m]\frac{S_q[m]}{\sqrt{P_s}}, \frac{\hat{\sigma}^2[m]}{2}\right) \tag{13}$$

with $p_{R[m]}(\cdot)$ defined by (10). The $1/2$ variance scaling, compared to the standard Rice model, is motivated by viewing the complex noise in (11) as a bivariate Gaussian RV with joint variance $\hat{\sigma}^2[m]$. Calculating (13) for all symbols in the alphabet yields a Qary symbol likelihood $\boldsymbol{L}_s = \{L_s[q] : 0 \leq q < Q\}$, where $L_s[q] = p(\boldsymbol{r}; \boldsymbol{S}_q)$, which can be transformed into $\log_2(Q) = M$ binary likelihoods

$$p(\boldsymbol{r}; u_m = 0) = \sum_{q=0}^{Q-1} L_s[q]P_q \cdot \mathbf{1}_b\left(u_m(q) = 0\right)$$

$$p(\boldsymbol{r}; u_m = 1) = \sum_{q=0}^{Q-1} L_s[q]P_q \cdot \mathbf{1}_b\left(u_m(q) = 1\right) \tag{14}$$

where $0 \leq m < M$, $P_q = 1/Q$ since symbols are a priori equiprobable, and $\mathbf{1}_b(u_m(q) = b)$ is an indicator function that masks out symbols whose label has b in the mth binary digit. The symbol to binary likelihood transformation described by (14) has a similar form to the *a posteriori* LLR used in turbo equalization [31]. The binary LLR of $\boldsymbol{r}$ is an Mary vector

$$\boldsymbol{\mathfrak{L}}_b(\boldsymbol{r}) = \left\{\log\left(\frac{p(\boldsymbol{r}; u_m = 0)}{p(\boldsymbol{r}; u_m = 1)}\right) : 0 \leq m < M\right\} \tag{15}$$

and by concatenating the LLR of each received symbol, the resulting Nary $\boldsymbol{\mathfrak{L}}_b = [\boldsymbol{\mathfrak{L}}_b(\boldsymbol{r}_0)^T, \boldsymbol{\mathfrak{L}}_b(\boldsymbol{r}_1)^T, \ldots, \boldsymbol{\mathfrak{L}}_b(\boldsymbol{r}_{N_s-1})^T]^T$ is used as *soft* input to the Polar Decoder. The rest of this section defines uncoded decisions, required for constructing the Polar code in Section VI, and makes practical improvements to numerical stability and complexity.

Discovering the value of q reveals the transmitted information, and (13) can be used to make a maximum likelihood (ML) symbol decision on $\boldsymbol{r}$

$$\hat{q} = \underset{q}{\operatorname{argmax}}\, p\left(\boldsymbol{r}; \boldsymbol{S}_q\right). \tag{16}$$

Similarly, (14) can be used for M partial decisions

$$\hat{q} = (\hat{\mathfrak{q}}_0, \hat{\mathfrak{q}}_1, \ldots, \hat{\mathfrak{q}}_{M-1}), \quad \text{where}$$

$$\hat{\mathfrak{q}}_m = \begin{cases} 0, & \text{if } p(\boldsymbol{r}; u_m = 0) > p(\boldsymbol{r}; u_m = 1) \\ 1, & \text{if } p(\boldsymbol{r}; u_m = 0) \leq p(\boldsymbol{r}; u_m = 1) \end{cases} \tag{17}$$

and $(\hat{\mathfrak{q}}_0, \hat{\mathfrak{q}}_1\ \ldots,\ \hat{\mathfrak{q}}_{M-1})$ is the binary representation of a decimal value.

Note that (13) is prone to numerical instability, so transforming $\boldsymbol{\mathfrak{L}}_s = \log(\boldsymbol{L}_s)$, and moving $\log(\cdot)$ into the density in (10), is an improvement. This requires replacing $L_s[q]$ by $\exp(\mathfrak{L}_s[q])$ in (14), which introduces another instability; however, (14) and (15) is a log-sum-exp problem, which has an accurate and stable approximation (see [32]).

Direct evaluation of (13) requires $M2^M$ density calculations; however, this can be reduced to $M(M+1)$ by observing that the symbol element $S_q[m] \in \{0, \sqrt{P_s/1}, \sqrt{P_s/2}, ..\sqrt{P_s/M}\}$ for all m, depending on how many tones are active in the

Algorithm 1: Binary LLR Lookup Implementation: $\overset{+}{=}$ signifies recursive summation, $\#$ a comment, and $>^b$ is a boolean operator that evaluates $(x >^b y)$ to 1 if $x > y$, and to 0 otherwise; LOOKUP INDEX$(\cdot)$ is called once, and BINARY LOG-LIKELIHOOD$(\cdot)$ once for each received symbol.

```
LOOKUP INDEX ($\mathcal{S} = \{\boldsymbol{S}_q : 0 \leq q < Q\}$)
  for $q = 0 : Q - 1$
    $\nu = \text{SUM}\ (\boldsymbol{S}_q >^b 0)$ # Number of active tones
    for $m = 0 : M - 1$
      if $S_q[m] > 0$
        $\mathfrak{I}[m, q] = \nu$
      else
        $\mathfrak{I}[m, q] = 0$
  return $\boldsymbol{\mathfrak{I}}$
RICE ($r, A, \sigma^2$)
  return
   $\log(r/\sigma^2) - (r^2 + A^2)/2\sigma^2 + \log(I_0(Ar/\sigma^2))$
BINARY LOG-LIKELIHOOD ($\boldsymbol{r}, \hat{\boldsymbol{A}}, \hat{\boldsymbol{\sigma}}_A^2, \boldsymbol{\mathfrak{I}}, \hat{\sigma}_{\tilde{w}}^2$)
  for $\nu = 0 : M$
    for $m = 0 : M - 1$
      if $\nu = 0$ # No tones are active
        $\hat{\sigma}^2 = \hat{\sigma}_{\tilde{w}}^2/N$
        $\Lambda[m, \nu] = \text{RICE}(r[m]; 0, \hat{\sigma}^2/2)$
      else # $\nu$ tones are active
        $\hat{\sigma}^2 = \hat{\sigma}_A^2[m]/\nu + \hat{\sigma}_{\tilde{w}}^2/N$
        $\Lambda[m, \nu] = \text{RICE}(r[m]; \hat{A}[m]/\sqrt{\nu}, \hat{\sigma}^2/2)$
  for $m = 0 : M - 1$
    $L_b(\boldsymbol{r}; u_m = 0) = 0$
    $L_b(\boldsymbol{r}; u_m = 1) = 0$
    for $q = 0 : Q - 1$
      $\mathfrak{L}_s(\boldsymbol{r}; \boldsymbol{S}_q) = \sum_{\mathfrak{m}=0}^{M-1} \Lambda[\mathfrak{m}, \mathfrak{I}[\mathfrak{m}, q]]$
      if $u_m(q) = 0$
        $L_b(\boldsymbol{r}; u_m = 0) \overset{+}{=} \exp(\mathfrak{L}_s(\boldsymbol{r}; \boldsymbol{S}_q))$
      else
        $L_b(\boldsymbol{r}; u_m = 1) \overset{+}{=} \exp(\mathfrak{L}_s(\boldsymbol{r}; \boldsymbol{S}_q))$
    $\mathfrak{L}_b[m] = \log(L_b(\boldsymbol{r}; u_m = 0)) - \log(L_b(\boldsymbol{r}; u_m = 1))$
  return $\boldsymbol{\mathfrak{L}}_b$ # $M$ary binary LLR
```

qth symbol. Only $M(M+1)$ unique density evaluations are needed to calculate $p(\boldsymbol{r}; \boldsymbol{S}_q)$, so an $M \times Q$ matrix $\boldsymbol{\mathfrak{I}}$, with $\mathfrak{I}[\cdot, \cdot] \in [0, M]$, can be used to index an $M \times (M+1)$ matrix $\boldsymbol{\Lambda}$, which contains the unique symbol log-likelihoods of $\boldsymbol{r}$. In addition, $\log I_0(\cdot)$ is preferably implemented as a lookup table. A reduced-complexity algorithm, using $\{\boldsymbol{\mathfrak{I}}, \boldsymbol{\Lambda}\}$ to process a single symbol $\boldsymbol{r}$, is detailed in Algorithm 1.

V. Parameter Estimation

A description of how the receiver obtains the auxiliary parameters required for the soft decoder, for the cases when genie-provided and when estimated from pilot signals, is given in the following.

As discussed in Section I, the synchronization problem is explicitly avoided in the Watermark results presented in Section VII-B2. The first symbol arrives at $t = t_0$, where t_0 is here set manually to the most energetic arrival in $c(\tau; t)$, which is shown for the three Watermark channels in Fig. 3. Watermark nominally reinstates the linear Doppler shift of the original channel recording, which has been disabled here.

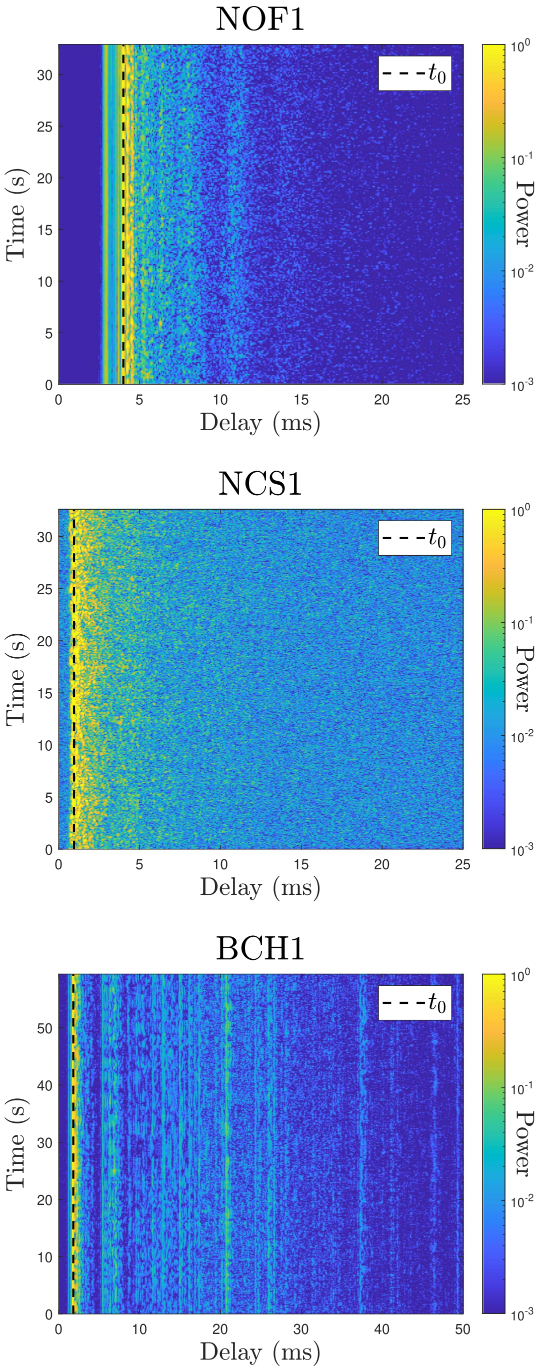


Fig. 3. Square of the impulse response for the Watermark channels NOF1, NCS1, and BCH1, normalized to the most energetic tap (shown in color). Sounding #3 is shown for all three channels. Note that the axes are different for BCH1. The start delay, t_0, is 4 ms for NOF1, 0.95 ms for NCS1, and 1.81 ms for BCH1. Both NOF1 and BCH1 exhibit little Doppler spreading, less than 1 Hz at -20 dB power; NCS1 is spread by approximately ±3 Hz at -10 dB and ±13 Hz at -20 dB, while being slightly skewed toward negative frequencies [33].

A. Genie-Aided Estimates

Genie estimates are obtained by probing noiseless Watermark channels with pilot signals in the same time locations as the received super symbols, i.e., at $t_i = t_0 + i(T_s + T_g)$ for $\mathfrak{r}_i$. The pilot signals are generated by replacing $\mathfrak{s}[m]$ with $\mathfrak{p}[m]$ in (3), where $\mathfrak{p} = (P_s/M)^{1/2} \cdot [1, 1, \ldots, 1]^T$ is an YMary vector with equal power to the information super symbols. Estimates of the pilot symbols are serialized in the same way as the information super symbols so that each $\boldsymbol{r}$ has a matching amplitude and noise variance estimate, $\{\hat{\boldsymbol{A}}, \hat{\boldsymbol{\sigma}}^2\}$, where the amplitudes are estimated with an FFT. For the noise variance of the mth tone, $\hat{\sigma}^2 = \hat{\sigma}_A^2[m] + \hat{\sigma}_{\tilde{w}}^2/N$, the background noise variance $\hat{\sigma}_{\tilde{w}}^2$ is assumed estimated outside the frame and set to $\sigma_{\tilde{w}}^2$ used in the simulation; the amplitude $\hat{A}[m]$ and its variance $\hat{\sigma}_A^2[m]$ are calculated from pairs of time-consecutive pilot symbols, as described in [34]. Note that this approach provides near-perfect time tracking (once per pair of symbols) of the parameters but is not implementable in reality.

B. Pilot Estimates

In a realistic case, the amplitude and amplitude variances need to be estimated from dedicated pilot signals repeated at regular intervals in the frame. Excessive transmission of pilots reduces the overall spectral efficiency, and as discussed in [34], at least two pilot estimates per repetition are required to calculate $\hat{\sigma}_A^2$ for each tone. However, the results of [34] indicate that an MFSK receiver can employ sparse *single* pilots by limiting the SNR of the likelihoods. A benefit to the approach is increased robustness to time variation in the true amplitude between the pilot estimates. Moreover, when $\sigma_{\tilde{w}}^2$ is small, the likelihood SNR limitation models the remaining variance due to a nonzero σ_A^2. Since MFSK and SPFSK only differ in alphabet definition, the approach is employed here. A single pilot symbol is added to the start and end of the information super symbol sequence, resulting in a sequence of the form $(\mathfrak{p}, \mathfrak{s}, .., \mathfrak{s}, \mathfrak{p})$; the amplitude used in the likelihood calculation is subsequently obtained from the two pilot estimates through linear interpolation. The likelihood input SNR, or Ricean shape factor, is defined as $(\hat{A}[m]/\nu)^2/(2\hat{\sigma}^2)$ for the mth tone in Algorithm 1, where one sets $\hat{\sigma}_A^2 = 0$ in $\hat{\sigma}^2$. If the likelihood SNR is larger than a maximum value, η, then $\hat{\sigma}^2$ is scaled up accordingly. This rule is applied prior to calculating the Ricean likelihood in Algorithm 1.

VI. Polar Coding

The information bits input to the code are mapped to a set of indices $n \in \mathcal{A}$, and binary 1s are mapped to indices $n \in \mathcal{A}^C$, where $\{\mathcal{A} \cup \mathcal{A}^C\} \sim [0, N-1]$; $\mathcal{A}$ is a set of unity-capacity channels and $\mathcal{A}^C$ is a set of zero-capacity channels in the asymptotic sense of large N. Encoding and decoding are $\mathcal{O}(N \log_2 N)$ in complexity, and both are single-pass and explicit. Successive cancellation decoding (SCD) is done in bit-reversed label order,[2] and each bit decision made by the decoder is used in the evaluation of the remaining bit channels; SCD has been shown to attain asymptotic channel capacity for any binary-symmetric channels [35].

A nonsystematic $\mathcal{O}(LN \log_2 N)$ generalization named successive cancellation list decoding is presented in [36], where decoder bit decisions are postponed until a *list* of $2L$ decoder states is available, after which the L most likely are kept. The number of frozen bits is $N - K - N_{\text{CRC}}$, and at the end of decoding, N_{CRC} bits are used for CRC on the bits in each list entry; the one that fulfills the CRC and or is the most likely is chosen as the decoded frame. The list decoder is shown in [36] to have a performance close to ML decoding in an AWGN channel, for modest values of L. A logarithmic adaptation presented in [37], which only uses numerically stable quantities, is the decoder used here.

To construct the code for a continuous channel output alphabet, which is the case for the statistical model in (11), with an exact bit error probability of the decoded bit channels, requires convolving the PDF in each recursive step in the code definition, which is intractable for any applicable frame length N. An overview of construction algorithms, and an adaption to solve the problem of convolving PDFs, is given in [38]. However, applying this approach to the binary LLRs of an SPFSK symbol is outside the scope of this article. A simple recursive construction algorithm for the binary erasure channel (BEC) was provided in the original publication [35], which has been suggested as an *heuristic* method since it yields good results also for non-BECs [39]. The heuristic method works well since the recursive code definition introduces a *partial order* in the reliability of the bit channels, although the performance is expected to be worse than for a code constructed by evaluation of the bit channels actually seen by the bits [40]. In fact, the partial ordering has motivated the "universal" reliability ordering proposed for constructing the Polar codes in the 5G standard; this ordering was found through extensive Monte Carlo simulation of codes with $N = 1024$ [41], which is an approach to construction that was also suggested in the original publication of Polar codes [35]. Meanwhile, the heuristic method is a deterministic algorithm only requiring the frame length, N, and an uncoded bit error probability, ϵ, to evaluate the bit channel reliabilities, where the value of ϵ, or the *Design SNR*, is an open choice. Since the UWA channel is generally unknown a priori, the generality of the heuristic method motivates its use here.

It is, therefore, of interest to know the bit error probability, ϵ_m, of the M different label positions m, when a symbol $\boldsymbol{S}_q$ is chosen randomly from the alphabet and distorted by noise; a nonuniform error probability needs to be accounted for in the code construction if it varies across the label. Since SPFSK has 2^M symbols with varying pairwise Euclidean distance, the a priori assumption is that this is the case. However, if one can define a labeling that balances $\epsilon_m = \epsilon$ for all bit positions in the label $[u_0, u_1, \ldots, u_{M-1}]^T$, the heuristic construction method is directly applicable. The rest of this section evaluates the bit error probability, and the construction equations are stated in Section VI-B for completeness. The decoder performance using SPFSK symbols in a simple AWGN channel, presented in Section VII-A, verifies that this is a reasonable approach, where some comments are also made on good values of ϵ.

[2] Bit reversing the labels yields the decoding order, e.g., $\{00, 01, 10, 11\} \rightarrow \{00, 10, 01, 11\}$, for $N = 4$.

A. Bit Error Probability

1) AWGN Simulation: The model described by (11) is similar to a static AWGN model producing a Ricean RV, so it is reasonable to evaluate the internal properties of SPFSK using

$$R[m] = \big|X[m] + W_0[m] + jW_1[m]\big| \tag{18}$$

where $W_0[m], W_1[m] \sim \mathcal{CN}(0, \sigma^2), m \in [0, M-1]$. Statistics for an uncoded symbol are obtained by repeated generation of random, equiprobable, symbol realizations $\boldsymbol{X} = \boldsymbol{x} \in \mathcal{S}$, and Gaussian noise $\boldsymbol{W}_0 = \boldsymbol{w}_0$, $\boldsymbol{W}_1 = \boldsymbol{w}_1$. Applying Algorithm 1 to the resulting

$$\boldsymbol{r} = \left\{ \Big| x[m] + w_1[m] + jw_2[m] \Big| : 0 \le m < M \right\} \tag{19}$$

with $\hat{\boldsymbol{A}} = \mathbf{1}\sqrt{P_s}$ and $\hat{\boldsymbol{\sigma}}^2 = \mathbf{1}\sigma^2$, bit decisions are obtained by using (17) on the output. For a *chosen* $C_{\text{SNR}/b} = E_b/N_0$, $\mathcal{S}$ is constructed with $P_s = E_b R_0$, and the noise variance is $\sigma^2 = E_b/C_{\text{SNR}/b}$, where $R_0 = 1$, and the bit energy E_b is arbitrary. The uncoded bit error rate, resulting from several iterations using *binary* LLR decisions, is denoted $\boldsymbol{\epsilon}(\hat{b})$.

2) Integration of Symbol Decision Regions: Although the symbol AWGN simulation is the most straightforward approach, it yields little insight into the internal structure of $\mathcal{S}$, and an alternative method that evaluates the probability space is provided. An ML *symbol* decision receiver chooses $\boldsymbol{S}_q$ if $\boldsymbol{r}$ falls into its decision region

$$\begin{aligned} D_q = \{\boldsymbol{r} \in \mathbb{R}^M_{\ge \mathbf{0}} : p(\boldsymbol{r}|\boldsymbol{S}_q) > p(\boldsymbol{r}|\boldsymbol{S}_{q'}) \\ \forall\, q' \ne q \in [0, Q-1]\} \end{aligned} \tag{20}$$

where q' indicates a *different* symbol. This is equivalent to the symbol ML decision described by (16). The probability of a symbol error event, $e_{\hat{s}}$, is [3, Ch. 4]

$$\begin{aligned} \mathrm{P}(e_{\hat{s}}) &= \sum_{q=0}^{Q-1} P_q\, \mathrm{P}(\boldsymbol{r} \notin D_q | \boldsymbol{S}_q) &(21)\\ &= \sum_{q=0}^{Q-1} P_q \sum_{\substack{q'=0\\ q' \ne q}}^{Q-1} \underbrace{\int_{D_{q'}} p(\boldsymbol{r}|\boldsymbol{S}_q) d\boldsymbol{r}}_{\triangleq P_{q'|q}} &(22) \end{aligned}$$

where $P_{q'|q}$ is defined as the *symbol mix-up probability*; this error occurs at the mth label digit if the decided symbol, with index q', has $u_m(q') = b$ in its label, whereas the transmitted symbol, with index q, has $u_m(q) = 1 - b$. The bit error probability, from a *symbol decision*, is

$$\epsilon_m(\hat{s}) = \sum_{q=0}^{Q-1} P_q \sum_{q'=0}^{Q-1} P_{q'|q} \mathbf{1}_b \left(u_m(q) \oplus u_m(q') = 1\right) \tag{23}$$

where $\oplus$ denotes modulo-2 addition, and $\mathbf{1}_b$ selects the $P_{q'|q}$ terms that cause binary error events on the mth digit.

What remains is evaluating the Q^2 integrals $P_{q'|q}$. The density $p(\boldsymbol{r}|\boldsymbol{S}_q)$ consists of M independent[3] Rice distributions, parameterized by $\boldsymbol{S}_q$ and the *chosen* σ^2, and the integration domain is $D_{q'}$. Note that the PDF simplifies to the Rayleigh distribution along axes with $S_q[\cdot] = 0$. Direct integration is intractable, and numerical integration is required. Because of the high dimensionality, the Monte Carlo integration technique *importance sampling*, first introduced in [42], is used: λ samples $\{\boldsymbol{x}_0, \boldsymbol{x}_1, .., \boldsymbol{x}_{\lambda-1} : \boldsymbol{x}_\ell \in D_{q'}\}$ are drawn from a distribution $g(\boldsymbol{x})$ to approximate $P_{q'|q}$ as the sample mean

$$\begin{aligned} \int_{D_{q'}} p(\boldsymbol{r}|\boldsymbol{S}_q) d\boldsymbol{r} &= \int \frac{\tilde{p}(\boldsymbol{x}|\boldsymbol{S}_q)}{g(\boldsymbol{x})} g(\boldsymbol{x}) d\boldsymbol{x} &(24)\\ &\approx \frac{1}{\lambda} \sum_{\ell=0}^{\lambda-1} \frac{\tilde{p}(\boldsymbol{x}_\ell|\boldsymbol{S}_q)}{g(\boldsymbol{x}_\ell)} &(25) \end{aligned}$$

where $g(\boldsymbol{x})$ is a sample generator and

$$\tilde{p}(\boldsymbol{x}|\boldsymbol{S}_q) = \begin{cases} p(\boldsymbol{r}|\boldsymbol{S}_q) & \forall\, \boldsymbol{x} \in D_{q'} \\ 0 & \forall\, \boldsymbol{x} \notin D_{q'} \end{cases}. \tag{26}$$

A good generator is $g(\boldsymbol{x}) = \{p(\boldsymbol{x}|\boldsymbol{S}_{q'}) : \boldsymbol{x} \in \mathbb{R}^M_{\ge \mathbf{0}}\}$, since its mass defines $D_{q'}$, and it can be implemented using (18).

The complexity of the above operation is problematic for large alphabets. Hence, an approach for reducing the complexity is proposed in the following. To calculate $\tilde{p}(\boldsymbol{x}_\ell|\boldsymbol{S}_q)$, one must test if $\boldsymbol{x}_\ell \in D_{q'}$, for $0 \le q' \le 2^M - 1$, where 2^M can be a large number. However, it is reasonable to think that this region, defined by $\boldsymbol{S}_{q'}$, only shares a decision boundary with *some* of the other symbols in $\mathcal{S}$. Define the open neighborhood $\mathfrak{N}(\boldsymbol{S}_{q'}) = \{\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{\varphi-1}\}$ as the set of symbols that share a decision boundary with $\boldsymbol{S}_{q'}$. For a generated $\boldsymbol{x}_\ell$, we have $\tilde{p}(\boldsymbol{x}_\ell|\boldsymbol{S}_q) = 0$ if $p(\boldsymbol{x}_\ell|\boldsymbol{S}_{q'}) < p(\boldsymbol{x}_\ell|\boldsymbol{v})$ for *any* $\boldsymbol{v} \in \mathfrak{N}(\boldsymbol{S}_{q'})$, since then $\boldsymbol{x}_\ell \notin D_{q'}$, which requires at most φ comparisons. The neighborhood of $\boldsymbol{S}_{q'}$ can be found by projecting the alphabet onto the hyperplane that $\boldsymbol{S}_{q'}$ is normal to

$$\begin{aligned} \mathcal{P}_{\mu q'} = \{\boldsymbol{S}_\mu \perp \boldsymbol{S}_{q'} : 0 \le \mu \le Q - 1, \\ \mu \ne q',\ q' \ne 0\} \end{aligned} \tag{27}$$

$$\boldsymbol{S}_\mu \perp \boldsymbol{S}_{q'} = \boldsymbol{S}_\mu - \frac{\langle \boldsymbol{S}_{q'}, \boldsymbol{S}_\mu \rangle}{||\boldsymbol{S}_{q'}||^2} \boldsymbol{S}_{q'} \tag{28}$$

where $\langle \cdot \rangle$ is a scalar product, and $\mathfrak{N}(\boldsymbol{S}_{q'})$ is the set $\{\boldsymbol{S}_\mu\}$ that generate linearly independent vectors, with *minimal* norm, in $\mathcal{P}_{\mu q'}$. It is not possible to project onto $\mathbf{0}$; however, it is a neighbor to all symbols, and vice versa. If several $\boldsymbol{S}_\mu$ generate parallel vectors in $\mathcal{P}_{\mu q'}$, only the one with the smallest norm is in $\mathfrak{N}(\boldsymbol{S}_{q'})$, since its decision region "screens out" the other symbols with overlapping projections. Calculation of the Q neighborhoods is done once, before the importance sampling, and the error probability across the label, defined by (23), is written $\boldsymbol{\epsilon}(\hat{s})$.

3) Conclusion on Code Construction and Labeling: A comparison of $\epsilon_m(\hat{b})$ and $\epsilon_m(\hat{s})$ over the label is shown in Fig. 4(a) for natural labeling. There is a small difference between them, caused by bit- and symbol-level decisions. However, they are consistent, indicating that both are valid descriptions in the context of construction since the design SNR is an open choice. An overall decrease in error probability can be observed for larger M. A bit error event is the union of an error given that

[3] The independence assumption continues from the assumption of independent tones in Section IV.

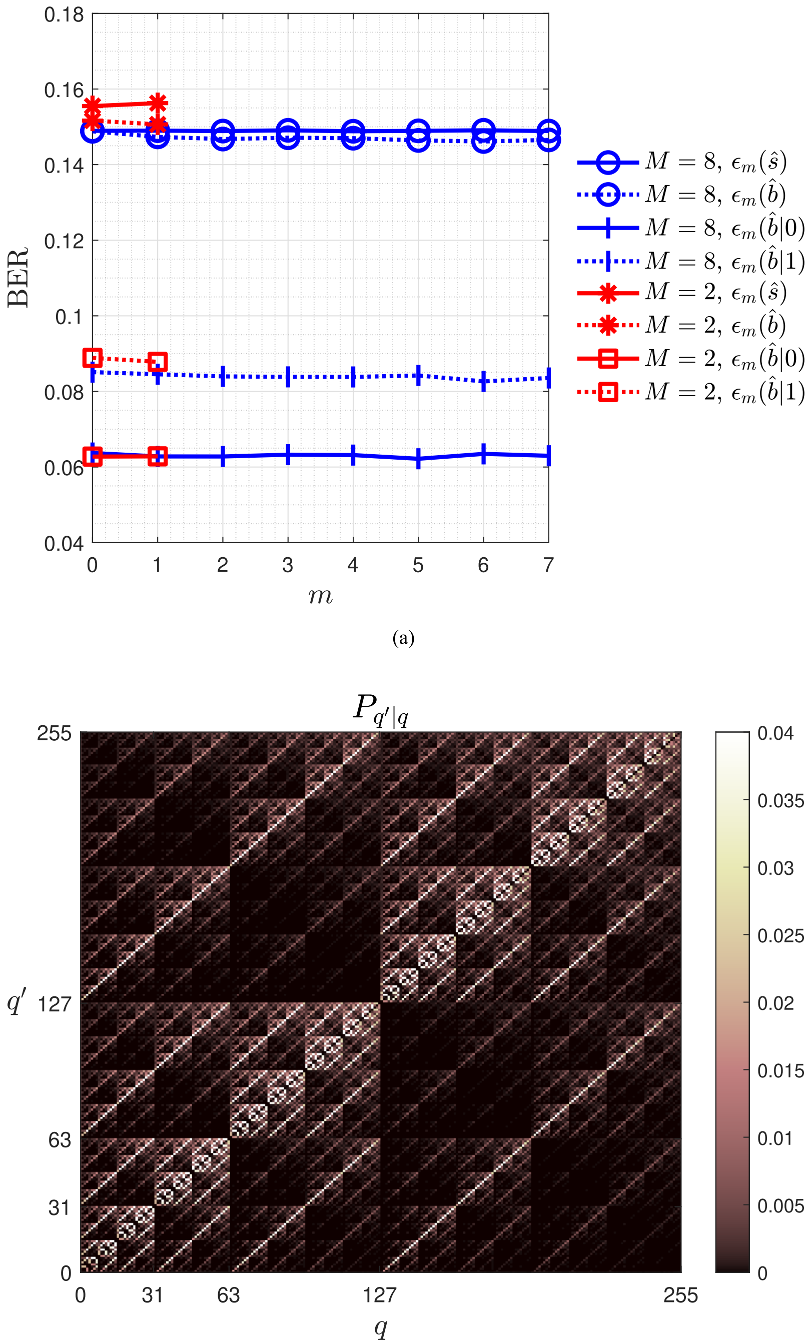
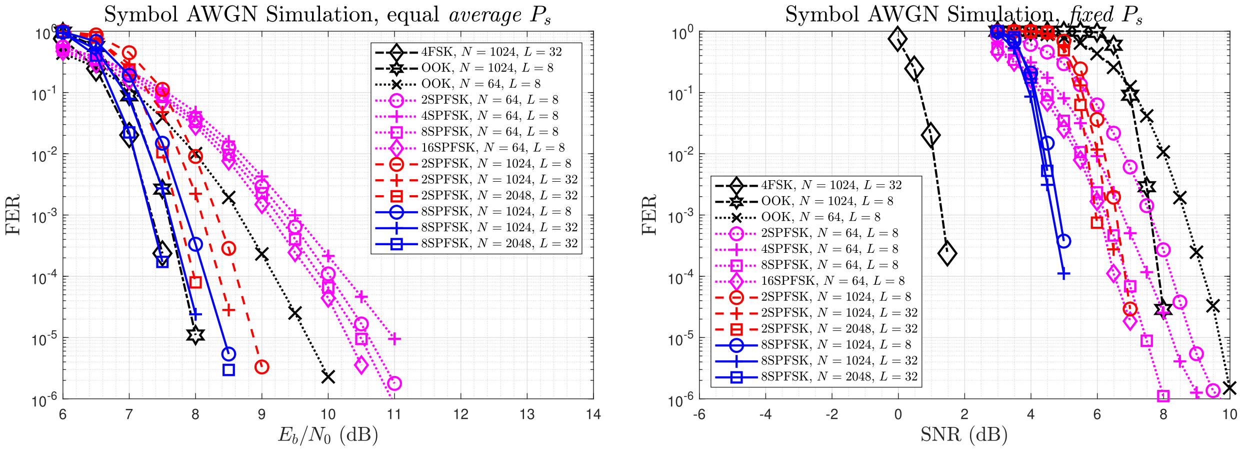


Fig. 4. (a) Bit error probability for a single SPFSK symbol, depending on the label position; $\epsilon_m(\hat{b})$ is generated at 3-dB design SNR with an AWGN simulation of 1.6×10^5 iterations, and $\epsilon_m(\hat{s})$ by using importance sampling of the likelihood function with $\lambda = 5 \times 10^3$. (b) Probability that $\boldsymbol{S}_{q'}$ is erroneously chosen when $\boldsymbol{S}_q$ is transmitted, for a $Q = 2^8$ alphabet $\mathcal{S}$ with natural labeling ($P_{q'|q}$ is set to 0 along the symmetry line $q' = q$ in the plot).

0 or 1 was transmitted, so $\epsilon_m(\hat{b}) = \epsilon_m(\hat{b}|0) + \epsilon_m(\hat{b}|1)$. The conditionals are observed to differ due to bit-zeros being mapped to zeros in the symbols, which are Rayleigh distributed at the receiver, whereas bit-ones map to nonzero elements, which are Ricean distributed. This is unusual in a communication method; however, the information source is assumed to be maximum entropy, so the main practical implication is that a random source with uniform bit probability should be used in the performance evaluation. Natural labeling produces an approximately uniform error probability over the label elements, particularly for $M = 8$. No other labeling with this property, or with a lower average bit probability, is known to the author; one resorts to evaluating specific schemes, e.g., Gray labeling, since there are $2^M!$ possible choices for labeling. Fig. 4(b) shows $P_{q'|q}$,



(a) (b)

Fig. 5. FER in a static AWGN channel for Polar-coded SPFSK, 4FSK, and OOK, using several frame lengths N, decoder list lengths L, and the number of elements in each symbol, M. Observe that the OOK alphabet is produced by the alphabet definition in (2) for $M = 1$ and is, therefore, equivalent to 1SPFSK. All cases have code rate $R_c = 0.5$. Codes are constructed with $\{N = 64, \epsilon = 0.01\}$, $\{N = 1024, \epsilon = 0.1\}$, and $\{N = 2048, \epsilon = 0.2\}$; the number of CRC bits are $N_{\text{CRC}} = 6$ for $N = 64$, and $N_{\text{CRC}} = 16$ for $N = \{1024, 2048\}$. (a) Equal P_s on average. The noise is scaled down by the probability of $\mathbf{0}$ symbols being transmitted. (b) Fixed P_s (power constrained), where $\text{SNR} = P_s/(BN_0)$, $P_s = E_b R_0 R_c$, and N_0 is the noise spectral density.

for $M = 8$ using natural labeling: it appears symmetric about the index line $q' = 2^M - 1 - q$, which explains the uniformity of $\boldsymbol{\epsilon}(\hat{s})$. $P_{q'|q}$ is also antisymmetric about $q' = q$ due to the number of Rayleigh/Rice-distributed elements in the symbol, which causes the difference in the conditionals. A fractal pattern can be observed in regions of low error probability, similar to a Sierpiński triangle. This special structure produces a uniform error probability across the label, regardless of the dimensionality of the alphabet. It is concluded that natural labeling enables the use of Polar code construction that assumes approximately uniform $\epsilon_m = \epsilon$, given that the information source has maximum entropy.

B. Construction

The heuristic construction algorithm, originally given for the BEC in [35], is described in the following for completeness. The codeword $\boldsymbol{c} = \{c_n : 1 \leq n \leq N\}$ is transmitted across the channel; define $Z(c_n)$ as the unreliability of the bit c_n, when the nth bit-channel is decoded. Then

$$Z_{\mathfrak{n}}(c_{2n-1}) = 2Z_{\mathfrak{n}/2}(c_n) - Z_{\mathfrak{n}/2}(c_n)^2 \tag{29}$$

$$Z_{\mathfrak{n}}(c_{2n}) = Z_{\mathfrak{n}/2}(c_n)^2 \tag{30}$$

computes $\{Z_{\mathfrak{n}}(c_n) : n \in [1, \mathfrak{n}]\}$ from a code of size $\mathfrak{n}/2$. Note that indexing starts from 1. The recursion is started with $Z_1(c_1) = \epsilon$ and is repeated until $\mathfrak{n} = N$. One then defines $\mathcal{A}$ as the set of indices, n, with the K smallest $Z_N(c_n)$, to which information and CRC bits are mapped, and $\mathcal{A}^C$ as the set of largest $Z_N(c_n)$, which are populated with 1:s.[4]

VII. Result and Discussion

A. Symbol Simulation in an AWGN Channel

To find a good value on the construction parameter ϵ, the decoding performance was verified by performing several MSPFSK symbol simulations in the static AWGN channel described by (18), and Fig. 5 shows the resulting frame error rate (FER) as a function of the SNR per bit, E_b/N_0, for varying frame length N, alphabet size 2^M, and number of list elements in the decoder, L. The code rate is $R_c = 0.5$ in all cases. Polar-coded 4FSK- and OOK symbol simulations are shown as reference, where OOK is equivalent SPFSK with $M = 1$, or 1SPFSK. The ϵ yielding the best results were found to increase with N and are stated in the caption of Fig. 5. All symbols in the alphabet are assumed equiprobable, and comparisons in the following part are made at FER$= 10^{-3}$.

1) Equal Average Symbol Power: Fig. 5(a) shows MSPFSK, MFSK, and OOK (1SPFSK) compared with equal *average* symbol power. MFSK conveys a symbol with P_s power in every use of the symbol AWGN channel; meanwhile, the SPFSK alphabets contain a symbol requiring no power, i.e., a zero symbol, which is transmitted with probability $1/Q$. For $N = 64$, the $M = 2$ SPFSK alphabet performs better than one with $M = 4$, while OOK ($M = 1$) outperforms all other alphabet sizes. This is due to the $Q/(Q-1)$ power scaling to balance the presence of the zero symbol, which is most prominent for OOK. However, the performance is seen to increase with M, and the difference between $M = 4$ and $M = 16$ is ~ 0.4 dB. Increasing the frame length, the performance difference between $N = 1024$ and $N = 2048$ is ~ 0.3 dB, and between $L = \{8, 32\}$, the difference is ~ 0.2 dB for the same N. The slope does not seem to change for $N \geq 1024$, and since the frame transmission time is an important parameter for avoiding network collisions, $N = 1024$ is of interest for a UWA system. Also, the performance is

[4] Although this produces more 1:s in the final codeword, it is inconsequential because of the noise definition used in Watermark.

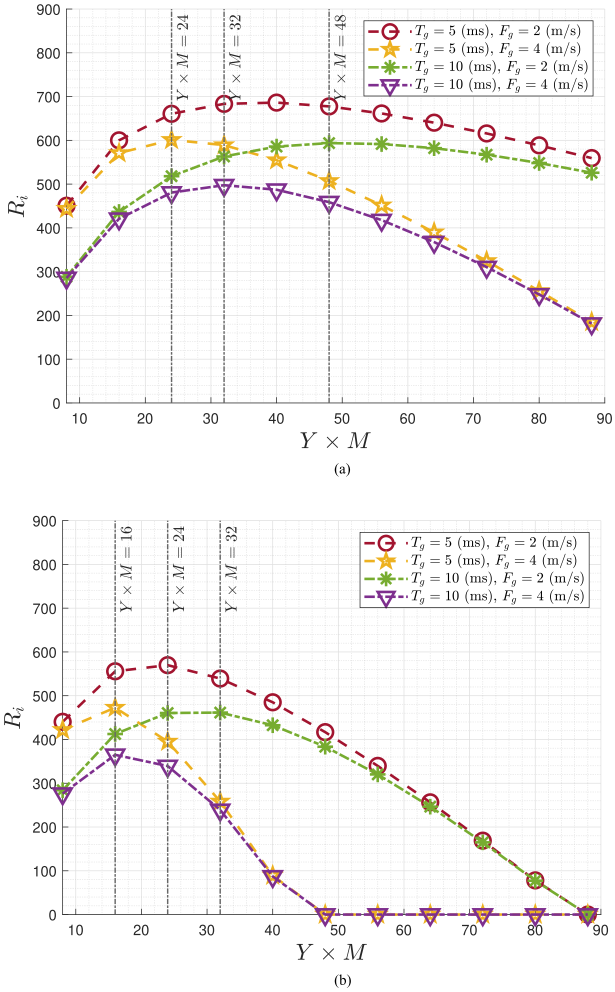
acceptable for both values of L, so the choice depends on the receiver hardware. It is interesting to note that 8SPFSK has ~ 0.3 dB worse performance than 4FSK with equal N and L, whereas for 8SPFSK with $N = 2048$, their performance in a static AWGN channel is similar in terms of robustness and energy efficiency. The reason for this is the different rates, and the increased performance in SPFSK from larger M, and likewise in the decoder from the larger N. Here, the coded rate of 4FSK is $R_0 \cdot R_c = 0.25$ bit/s/Hz, whereas it is $1 \cdot 0.5$ bit/s/Hz for SPFSK. Hence, twice the information is conveyed at an equivalent E_b/N_0 performance, indicating that longer frames are necessary for 8SPFSK to attain the same performance to bit energy efficiency as 4FSK. Meanwhile, OOK with $L = 8$ has equivalent performance to 8SPFSK with $L = 32$. This indicates that conveying information by transmitting the zero symbol as often as possible is more energy efficient in an AWGN symbol channel. However, the performance gain from larger M closes the energy efficiency gap, which is only ~ 0.6 dB between OOK and 16SPFSK for the $N = 64$ frame.

2) Fixed Symbol Power: Considering a scenario where the transmitter is power limited, it is relevant to also compare the alphabet for a fixed P_s. Note that the noise level is calculated from the signal time samples in Watermark, so this symbol model might be more applicable than the average-P_s model. The same configurations as previously are shown in Fig. 5(b) versus the SNR, which is defined in the caption. The robustness of SPFSK increases monotonically with M, albeit with decreasing returns, which is evident by comparing the difference between $M = \{8, 16\}$ and $M = \{4, 8\}$ at $N = 64$ frame length. Comparing 4FSK and 8SPFSK at $N = 1024$, the robustness penalty for doubling the information rate is ~ 3.25 dB. Comparing SPFSK over M, a processing gain of ~ 2.5–3 dB over OOK ($M = 1$) is achieved with large M. Note that SPFSK has the same spectral efficiency for all M.

It is unclear precisely why the performance increases with the alphabet size, 2^M, and what limit exists on the processing gain. A possibility is that the mass of the symbol likelihood becomes more localized in the high-dimensional space. However, the complexity becomes problematic at a certain point. For $M = 16$, the alphabet size is $Q = 2^{16}$; hence, $M \leq 8$ are mainly of interest for the UWA channel, particularly $M = 8$ as each symbol conveys a standard byte, simplifying the implementation. From an AWGN standpoint, the symbol simulations show that OOK and SPFSK, with large M, have similar energy efficiency in terms of performance, given the average energy expenditure per bit. For symbols transmitted with fixed power, the symbol simulations also show the performance of SPFSK increasing monotonically with M. Hence, the receiver statistics improve by jointly considering more symbol elements, i.e., a larger quantity of information in the likelihood calculation.

B. Watermark

1) Setup: The Watermark version is 1.0, and the stated information rates in the simulation results are obtained from the framework. The bandwidth is $B = 4$ kHz for all channels, and the number of information bits in each simulated frame is given by $K = R_c N$. The number of transmitted frames (packets) processed in each iteration is 780–2280 for NOF1, 1500–2520 for NCS1, and 144–212 for BCH1, and the performance is averaged over all soundings in the framework. The default AWGN noise definition is used for the passband signal, which sets the noise variance based on an E_b/N_0 measure; E_b is calculated from the convolver output samples as $\sum r_p[\cdot]^2/K$, where $r_p[\cdot]$ are the received passband samples, which scale the noise based on the total energy in the channel output. No noise is added to the genie estimates, and the frame start time, t_0, is set manually in Sections VII-B2 and VII-B3, where the linear Doppler normally reinstated by Watermark has also been disabled; see Fig. 3 for a description of the channels. Randomized time–Doppler shifts are used in the results of Section VII-B4.

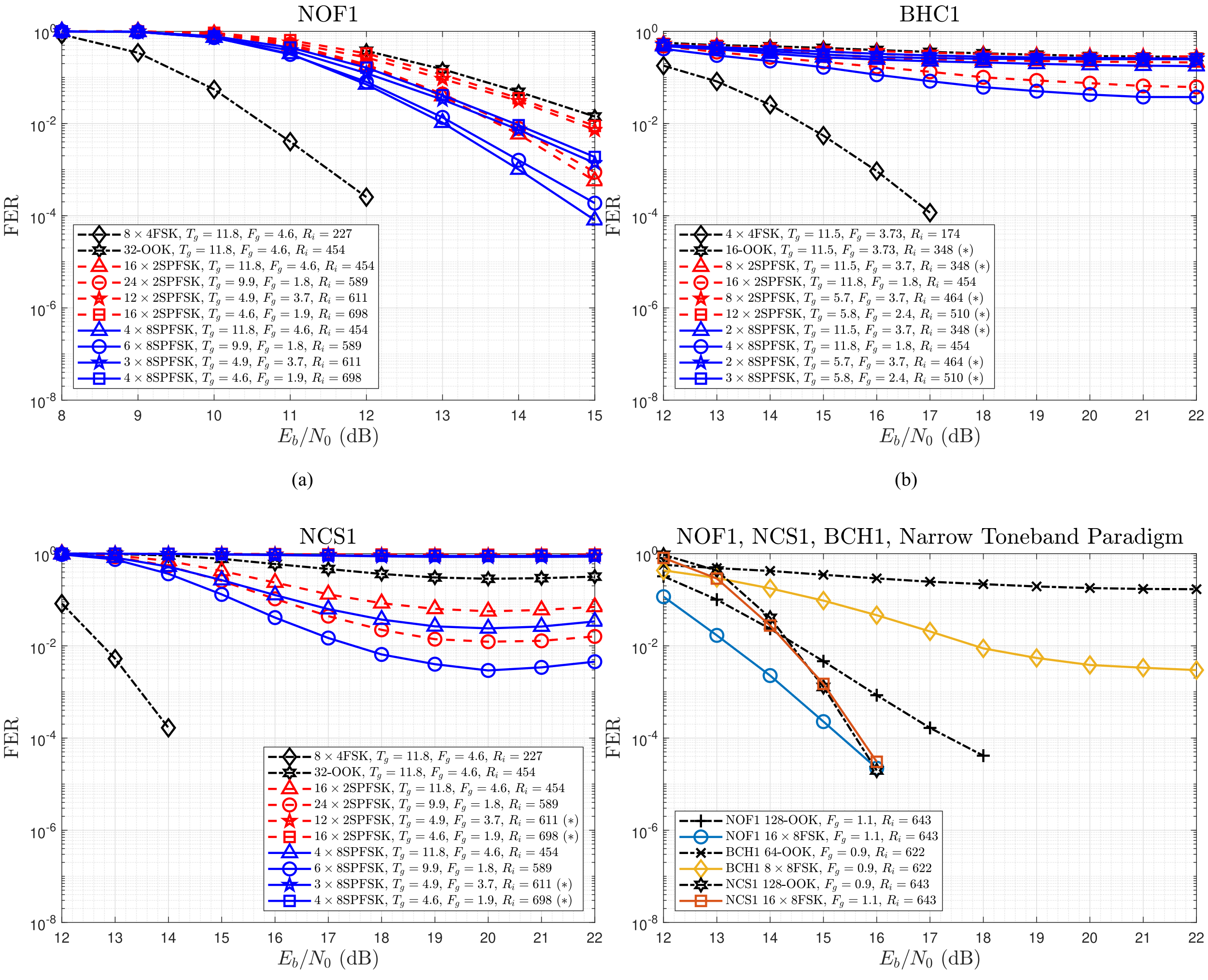


(a)

(b)

Fig. 6. Information rate of $Y \times M$SPFSK, calculated with (6) for $B = 4$ kHz, $R_c = 0.5$, and $C_\emptyset = 2$, where maxima are highlighted by vertical lines, and R_i is set to zero for Y where the total guard bandwidth exceeds B in (5). (a) $f_c = 14$ kHz: NOF1, NCS1. (b) $f_c = 35$ kHz: BCH1.

2) Simulation in the Channels NOF1, NCS1, and BCH1, Using Genie-Aided Estimates: First, the impact of guard parameters and choice of alphabet is evaluated: $M = \{1, 2, 8\}$, $T_g \sim \{5, 10\}$ ms, and $F_g \sim \{2, 4\}$ m/s (these units are used throughout), and the results are shown in Fig. 7(a)–(c). The total number of tones has been chosen to maximize R_i for the different guard configurations, as shown in Fig. 6. A configuration is considered *usable* at noise powers for which the FER is $\lesssim 1\%$.



(a)

(b)

(c)

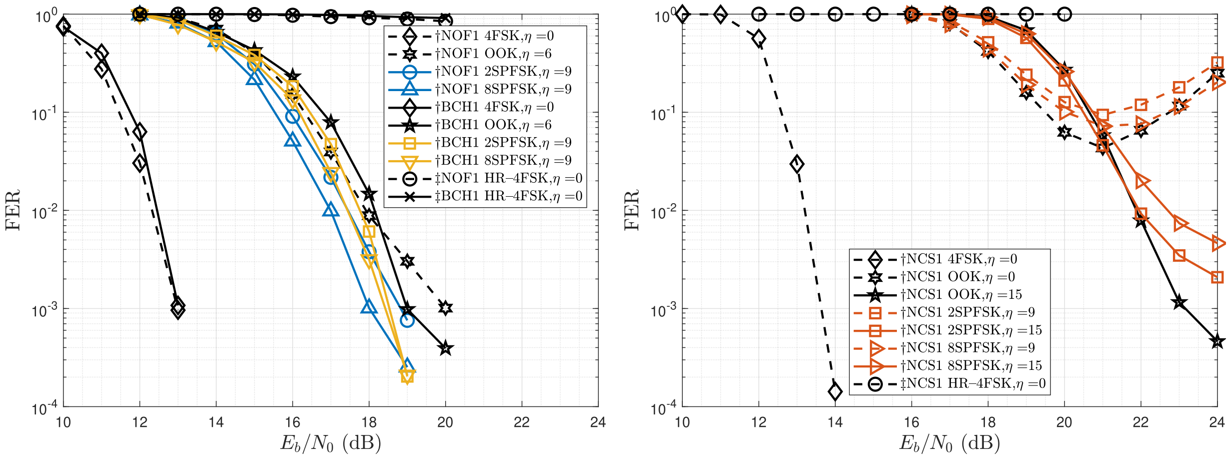
(d)

Fig. 7. Watermark results; genie-aided parameter estimation. (a)–(c) show the impact of guard parameters, $T_g \sim \{5, 10\}$ (ms), $F_g \sim \{2, 4\}$, and choice of alphabet, $M = \{1, 2, 8\}$, for NOF1, BCH1, and NCS1, respectively. Note that $(*)$ indicates poor performance, with several configurations' overlapping, and the horizontal scale being different in (a). All configurations use $N = 1024$, $R_c = 0.5$, $N_{\text{CRC}} = 16$, $L = 32$, and a Polar code constructed with $\epsilon = 0.1$. In (d), a more narrowband approach is shown, which aims to improve performance in BCH1 and NCS1 by increasing the number of tones and reducing the time–frequency guards; $T_g = 4$ (ms) is used in (d).

All SPFSK configurations are usable at 15 dB in NOF1; meanwhile, only two are usable in NCS1, $M = 2$ and $M = 8$, and none in BCH1. MFSK is usable in all channels, at half the rate of SPFSK. It is clear that NOF1 is a benign channel compared to the others; most of the energy contribution is from a stable path with little Doppler spreading, whereas NCS1 has no stable paths, and BCH1 has a mixture of stable and fluctuating and trailing arrivals [26]. BCH1 seems the most challenging, where sounding #3 (of 4) is the most time dispersive (see Fig. 3).

Interestingly, the performance *decreases* at high E_b/N_0 in NCS1, which implies that the narrow likelihood becomes overly sensitive to amplitude fluctuations between the genie measurements, or, that the Ricean function does not properly describe the statistical behavior of the envelope. Furthermore, the best SPFSK configurations in NCS1 and BCH1 use smaller guard bands and a larger number of tones, with $M = 8$ producing the best performance in all three channels. Therefore, a more narrow toneband design paradigm, by increasing the number of tones and reducing the time–frequency guards, is shown in Fig. 7(d) for $M = \{1, 8\}$. The performance is much improved in all three channels, indicating that the number of tones have a greater impact than the choice of time–frequency guards; however, an error floor remains in BCH1, likely due to the long reverberation in sounding #3. Comparing OOK (1SPFSK) with 8SPFSK, the latter performs significantly better in the channels with less Doppler spreading, NOF1 and BCH1, whereas in NCS1, in which the Doppler spread is large, their performances are similar.

3) Simulation in the Channels NOF1, NCS1, and BCH1, Using Pilot Estimated Parameters: SPFSK is further evaluated by replacing the genie estimates with estimates obtained from pilot signals at the start and end of the frame, as defined in Section V-B, and the results are shown in Fig. 8. To offset the reduced spectral efficiency from inserting pilot signals, the frame length is extended to $N = 2048$, as parameterized in Fig. 5. Attaining usable performance was found to require further narrowband tones, which had a greater impact than the frequency



(a)

(b)

Fig. 8. Watermark results; genie-aided synchronization, amplitude estimation from pilots, 256 tones. The SNR of each likelihood (or Ricean shape parameter), $(\hat{A}/\nu)^2/(2N_s\hat{\sigma}^2_{\tilde{w}})$, is limited to η (dB), as defined in [34]. Note that the horizontal and vertical scales differ from Fig. 7. All configurations marked with † employ 256 tones with $C_{\varnothing} = 2$ null separation, and a Polar code with $N = 2048$, $\epsilon = 0.2$, $R_c = 0.5$, $N_{\text{CRC}} = 16$, and $L = 32$. Hence, the number of frequency-parallel symbols, Y, is 64, 256, 128, and 32 for 4FSK, OOK (1SPFSK), 2SPFSK, and 8SPFSK, respectively; the spectral efficiency is 0.108 for 4FSK and 0.194 for SPFSK (bit/s/Hz). ‡ marks 512-tone configurations with $C_{\varnothing} = 1$ and $Y = 128$, which are equal to † in other regards; these are an HR implementation of 4FSK, with equal spectral efficiency to SPFSK, showcasing that the higher rate is unobtainable using 4FSK.

guards. Hence, $Y \times M = 256$, and $\{T_g = 4 \text{ ms}, F_g = 0 \text{ (m/s)}\}$ is used in configurations marked by †, where 4FSK and SPFSK have 0.108 and 0.194-bit/s/Hz spectral efficiency, respectively; the tones are placed at $C_{\varnothing} = 2$ with no additional frequency padding. A high-rate (HR) version of 4FSK, with the same spectral efficiency as SPFSK, is shown for reference; these employ $Y \times M = 512$ and $C_{\varnothing} = 1$ separation and are marked by ‡.

A wide range of limitations on the likelihood SNR, η, were simulated in steps of 3 dB, and the results with the lowest FER are shown in Fig. 8. As observed in [34], MFSK benefits from a small η, with $\eta = 0$ dB producing the lowest FER in all three channels for 4FSK. A small η causes the variance used in calculating the likelihood to be dominated by the amplitude, enforcing "wide" likelihoods at higher E_b/N_0. More tones and a smaller frequency separation are required for 4FSK to achieve the same spectral efficiency as SPFSK, which produces no useful performance due to insufficient protection from Doppler spreading. Therefore, the following discussion focuses on the case with $C_{\varnothing} = 2$ (†).

In contrast to 4FSK, the best performance in the more benign channels NOF1 and BCH1, shown in Fig. 8(a), is obtained with $\eta = 9$ dB for 2SPFSK and 8SPFSK, and $\eta = 6$ dB for OOK (1SPFSK); however, the performance of both OOK and MFSK is similar over $0 \leq \eta \leq 6$ dB. There is a degree of amplitude variation between nonzero SPFSK symbols for alphabets with $M > 2$, which is why SPFSK necessitates "narrower" likelihoods. Like MFSK, OOK only has one amplitude level per symbol element, resulting in a similar behavior in terms of η. However, this behavior changes in NCS1, the channel with higher Doppler spreading; η with best performance at low and high E_b/N_0 is shown for SPFSK and OOK in Fig. 8(b). Clearly, the best choice of η changes for different E_b/N_0, where a small η is more efficient at low E_b/N_0, and a large η at high E_b/N_0. This indicates that the performance for both SPFSK and OOK would improve with a more adaptive likelihood SNR limitation rule. In contrast to NOF1 and BCH1, OOK behaves less like 4FSK in NCS1. This is likely due to the Doppler spreading, since more frequency-adjacent tones are active simultaneously in OOK, compared to 4FSK. While the performance in the pilot case is improved by using 256 tones, it is circa 3.7–7.5 dB worse than the genie-aided case when comparing NOF1 and NCS1 between Figs. 7(d) and 8 at 10^{-2} FER. Also noted is that the performance increases with M for SPFSK in the genie-aided case, whereas in the pilot case, the performance using large M is slightly better in the benign channels and slightly worse in NCS1, where OOK (1SPFSK) has a lower error floor. Hence, the sensitivity to inaccuracies in the likelihood parameters due to sparse pilots increases with M for SPFSK.

4) *Feasibility of SPFSK in a Scenario With Nonperfect Time–Doppler Synchronization:* The results of the previous section are strengthened by validating that the performance is achievable also when perfect knowledge of the start time and linear Doppler shift is not assumed. The following argument on the inherently deterministic property of the problem also suggests avenues of approach to synchronization. Assume a signal containing a frame arrives at time t_0, and the entire frame is Doppler shifted proportional to the relative velocity v_0. The true meaning of the quantity $\{t_0, v_0\}$ is ambiguous due to the multipath and is often defined as the maximum over some energy measure. However, another interpretation is that $\{t_0, v_0\}$ is the quantity that *allows* the frame to be decoded after its effects on the signal are inverted. Rather, the frame is decodable at $\{t_0, v_0\}$ provided that the underlying link method is sufficiently robust to extract the information from the realization of the noisy signal. This viewpoint suggests that the limitations imposed by synchronization are defined by processing complexity rather than the communication channel since the receiver can attempt decoding arbitrarily close to $\{t_0, v_0\}$ by exhaustive search with the proper

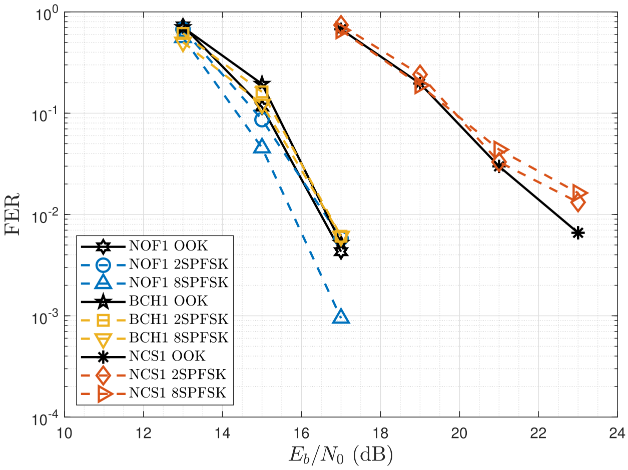


Fig. 9. Watermark results; exhaustive-search synchronization, amplitude estimation from pilots, 256 tones. The alphabets OOK (1SPFSK), 2SPFSK, and 8SPFSK, were simulated with a randomized time of arrival $t_0 \in [1.5, 2]$ s and linear Doppler shift $v_0 \in [-1, 1]$ m/s. The receiver performed a limited exhaustive search of $\{t_0, v_0\}$ by attempting to decode with certain parameterizations and rejecting results with an incorrect CRC (see Section VII-B4). OOK employed the likelihood SNR limitations $\eta = \{0, 15\}$ dB; both 2SPFSK and 8SPFSK employed $\eta = \{9, 15\}$ dB. The implementations were otherwise equal to † in Fig. 8.

time–Doppler resolution. The above argument is exemplified by the Watermark results in Fig. 9, where the simulations of OOK, 2SPFSK, and 8SPFSK, shown in Fig. 8, have been repeated with randomized $t_0 \in [1.5, 2]$ s and $v_0 \in [-1, 1]$ m/s. The receiver performs a limited exhaustive search by inverting the effects of several $\{t_0, v_0\}$ and attempting to decode the frame, rejecting the result if the CRC is incorrect; the low-complexity test statistic and associated detector presented in [27] are used to suggest values for t_0, with $T_s/10$–13 ms time resolution, and the Doppler shifts are evaluated with ~ 0.13 m/s resolution over the range ± 1.1 m/s. Furthermore, the parameter search includes the values of η identified in Fig. 8.

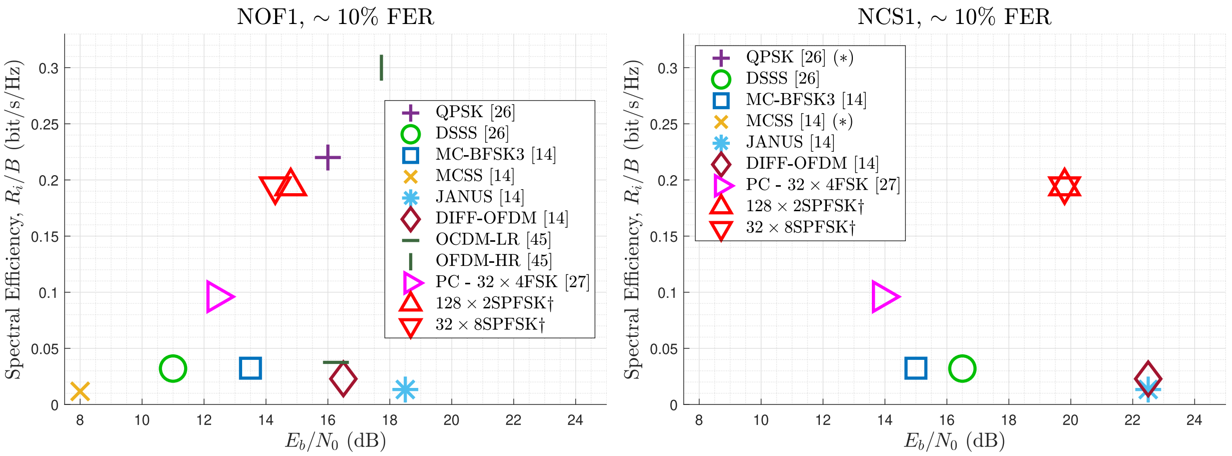
Compared with Fig. 8, the performance is improved in NOF1 and BCH1, and at $\sim 10\%$ FER in NCS1; meanwhile, the FER is slightly worse at the highest E_b/N_0 in NCS1. The reason for improved performance is that the receiver attempts decoding at several time shifts close to t_0; meanwhile, using several η improves the performance at both high and low E_b/N_0. Hence, there is no performance loss provided that the receiver inverts the correct $\{t_0, v_0\}$, and synchronization can be approached by *narrowing* the number of time–Doppler values the receiver should attempt to decode. This idea is similar to [14], where decoded pilot tones are used to assess a time–Doppler hypothesis before decoding. Furthermore, the range and variation of possible velocities is defined by the relative movement of the transmitter and the receiver and, therefore, mainly constitutes a design tradeoff between platform mobility and the receiver processing complexity. While not limited to noncoherent methods, the practicality of such an approach is enabled by the reduced complexity in methods that avoid channel estimation.

5) Literature Comparison: The performance shown in Fig. 9 is compared with the FER results available in the literature for the benchmark channels used here. All methods are compared at 10% FER, in terms of E_b/N_0, since most publications employ a linear scale on the FER. Results for a modified Janus implementation are reported for NOF1 and NCS1 in [14], where 10% FER is attained at approximately 18.5 and 22.5 dB for the respective channels; Zetterberg et al. [14] present an MC-2FSK with Turbo coding with this FER performance at ~ 13.5 and 15 dB in the two channels. Also shown in [14] is a Turbo-coded differential OFDM based on [43], and a multicarrier spread spectrum (MCSS) with DFE, based on [44]; the OFDM method has a lower performance than MC-2FSK and accomplishes 10% FER at 16.5 and 22.5 dB, whereas MCSS has 10% FER at ~ 8 dB in NOF1 and no useful performance in NCS1. Example implementations of quadrature phase-shift keying (QPSK) and direct sequence spread spectrum (DSSS) are presented in the original Watermark publication [26]: for QPSK, the FER passes 10% at ~ 16 dB in NOF1, with no useful performance in NCS1; for DSSS, the FER is below 10% at ~ 11 and 16.5 dB, for the respective channels. Results from NOF1 are presented for several implementations of OFDM and orthogonal chirp division multiplexing in [45], where the best performing high- and low-rate methods reach 10% FER at 2 and 17 dB SNR, which have been converted to E_b/N_0 using (31) in [14]. Recent results for Polar-coded 4FSK (PC-4FSK) using a hypothesis approach to synchronization attain 10% FER at 12.4 and 13.9 dB [27]. Except for [27], no methods that report FER for BCH1 could be found in the literature.

A graphical illustration of the literature results is shown in Fig. 10, together with the SPFSK results from Fig. 9, which are marked with †. Each method's spectral efficiency, R_i/B, is shown versus the E_b/N_0 required for 10% FER. Note that R_i/B relates to the throughput, compared to the upper limit R_0 used in Section I. One observes in Fig. 10 that robust methods are generally more energy efficient in the UWA channel, given a target FER; 4FSK requires a relatively low E_b/N_0 in both channels, indicating that it is interesting for energy-constrained platforms. Note that some methods are tailored to specifications other than spectral or energy efficiency; JANUS was developed for interoperability and low-complexity receivers, whereas DSSS has applications in covert communication. Meanwhile, a transmitter using SPFSK can achieve a significantly increased spectral efficiency in both channels at a higher energy cost per bit. SPFSK fills a performance gap between the noncoherent and coherent methods in the benign channel NOF1, whereas in NCS1, it is the sole method to achieve higher spectral efficiency since the channel is prohibitively challenging for other HR methods.

C. OOK Versus SPFSK

Since the MSPFSK alphabets have equal spectral efficiency for different M, a relevant question is whether high-dimensionality alphabets should be pursued for the UWA channel, as opposed to OOK ($M = 1$). SPFSK with large M performs better in the symbol AWGN simulations in Fig. 5(b), and in the more benign Watermark channels NOF1 and BCH1, shown in Fig. 7(d), when genie-aided estimates are used; in the



(a)

(b)

Fig. 10. Literature comparison. The spectral efficiency of each method is compared with the E_b/N_0 required for a 10% FER. The approximate values from the literature are obtained visually from the published figures. Only the best performing low- and high-rate methods are included from [45], denoted by "-LR" and "-HR," where no results are stated for the NCS1 channel. The results for 2SPFSK and 8SPFSK in Fig. 9 are shown, which were simulated with randomized time–Doppler shift. Note that the 10% performance of 256-OOK is similar to or worse than 128 × 2SPFSK and 32 × 8SPFSK, and that QPSK and MCSS do not attain 10% FER in NCS1, which is indicated by (∗). (a) Results for the Watermark channel NOF1. (b) Results for the Watermark channel NCS1.

more difficult channel NCS1, the performance is similar. The Ricean likelihood employed here models the statistical behavior of the amplitudes; while the receiver has near-perfect knowledge of the likelihood parameterization in both cases, the true model is only known in the symbol AWGN case. The amplitude distribution might deviate from a Ricean likelihood, particularly in a channel like NCS1. When the likelihood parameters are estimated from pilots, as shown in Fig. 8, the accuracy of the likelihood model decreases further, leading to a performance varying within ~ 1 dB at 10^{-2} FER for SPFSK with different M. These results indicate that higher accuracy is needed in the statistical model to leverage the benefits of a high-dimensional alphabet; by improving the model and the parameter tracking, the symbol simulations indicate that a performance gain of 2.5–3 dB over OOK may be achieved. Meanwhile, 2SPFSK finds a middle ground between the performance of higher dimensional SPFSK and OOK for the approach to likelihood modeling taken here, a reason for which it may also be of interest for future applications.

VIII. Conclusion

A novel approach to noncoherent UWA communication is presented, called SPFSK, and its feasibility as a robust data link using Polar coding is evaluated. SPFSK is an extension of MFSK and PFSK that includes all permutations of an M-length sequence with average power P_s in its alphabet. It has 1 bit/s/Hz maximum spectral efficiency, which is twice that of 2FSK and 4FSK, enabling an increase in the information rate for FSK-type methods. It has a simple symbol alphabet definition, and its size increases with the symbol length M as 2^M, where $M = 1$ produces the OOK alphabet. The fundamental spectral efficiency of the alphabet is equal for all choices of M.

Polar coding is used for error correction, and an analysis of the bit error probability of SPFSK symbols shows that the heuristic code construction method is directly applicable if the information source is encoded to maximum entropy and if natural labeling is used on the alphabet. The error probability is approximately equal across the label due to the SPFSK pairwise symbol error probability having a symmetric fractal shape in the form of a Sierpiński triangle. A Ricean data model with genie estimates is used, and a low-complexity algorithm for calculating binary LLRs for the large alphabet is presented.

Symbol AWGN simulations with a fixed symbol power show that the performance increases with the size of the alphabet, 2^M, with a processing gain of ~ 2.5–3 dB for large M, compared to OOK ($M = 1$). Watermark simulations with genie-aided estimates show a significantly improved performance for large M in the benign channels NOF1 and BCH1, whereas in the challenging channel NCS1, the performance is similar to OOK. A more realistic evaluation is performed by replacing the genie-aid with sparse pilot estimates and an SNR limitation on the likelihoods. The performance difference is smaller in this case, with OOK having a lower error floor in NCS1. These results indicate that the statistical model must be sufficiently accurate to leverage the processing gain of an SPFSK alphabet with higher dimensionality. However, 2SPFSK finds an interesting middle ground between the performance increase from higher dimensionality and lower sensitivity to inaccuracies in the statistical model. The feasibility of the performance in a more realistic scenario is validated by the Watermark simulations without perfect time–Doppler synchronization.

Tones are placed at the second spectral null in frequency and evaluated with a fixed time–frequency guard, $\{T_g, F_g\}$, inserted between tones to decrease time/frequency interference. The number of tones, $Y \times M$, affecting how narrowband the tones are, is found to greatly affect the performance in the Watermark channels NOF1, NCS1, and BCH1, with more tones improving the performance. The absolute frequency guard band B_g, calculated from the physical speed F_g, imposes a penalty on the information rate, increasing with the passband frequency, f_c, and the number of tones. While the sensitivity to Doppler spreading increases with $Y \times M$, a relative frequency separation, in the form of spectral null placement $C_{\varnothing}$, is found to be more effective in the channels under consideration.

Any implementation should consider $C_{\varnothing} = \{2, 3\}$ as the sole frequency guard, depending on the robustness requirement, and removing the time guard, since the best performance was obtained using 256 narrowband tones for the 4 kHz bandwidth. The poor peak-to-power ratio resulting from a large number of tones could be addressed by randomizing the phase. An MSPFSK alphabet can be implemented on any FSK system by replacing the Mary FSK symbols by (2) and using Algorithm 1 to calculate the binary LLRs. As observed in [27], the pilot signals used to track the likelihood parameters can also be used for Doppler shift detection and the CRC of the Polar code for frame verification. The method would benefit from an adaptive likelihood model, which should be investigated further in future work.

Compared with FER results in the literature, SPFSK fills a performance gap between traditional noncoherent FSK methods and coherent methods. A doubling of the throughput compared to a traditional 4FSK is demonstrated, achieving a spectral efficiency of $\sim$ 0.2 bit/s/Hz with FER $< 1\%$ in the more benign Watermark channels NOF1, BCH1, and the more challenging NCS1. Particularly for NCS1, no method with equivalent spectral efficiency is available in the literature. Polar-coded SPFSK is, therefore, a candidate for a *robust* data link and is of special interest for channels that are challenging for coherent methods.

Acknowledgment

The author would like to acknowledge P. Sigray, J. Kuttenkeuler, and E. Strandell-Erstorp for a discussion that led to the idea of a super permutation alphabet. Special thanks are given to M. Nordenvaad for invaluable advice and discussion, to S. Tavildar for publishing an implementation of the Polar list decoder, to P. Zetterberg, Fredrik Lindqvist, and Bernt Nilsson for helpful discussions, and to the researchers who have worked on the release of Watermark as a public benchmark.

References

[1] J. Poncela, M. C. Aguayo, and P. Otero, “Wireless underwater communications,” *Wireless Pers. Commun.*, vol. 64, no. 3, pp. 547–560, 2012.

[2] X. Lurton, *An Introduction to Underwater Acoustics: Principles and Applications*, 2nd ed. New York, NY, USA: Springer, 2010.

[3] J. G. Proakis, *Digital Communications*, 5th ed. New York, NY, USA: McGraw-Hill Higher Educ., 2008.

[4] M. Stojanovic, “Recent advances in high-speed underwater acoustic communications,” *IEEE J. Ocean. Eng.*, vol. 21, no. 2, pp. 125–136, Apr. 1996.

[5] M. Stojanovic, J. Catipovic, and J. Proakis, “Phase-coherent digital communications for underwater acoustic channels,” *IEEE J. Ocean. Eng.*, vol. 19, no. 1, pp. 100–111, Jan. 1994.

[6] M. Chitre, S. Ong, and J. Potter, “Performance of coded OFDM in very shallow water channels and snapping shrimp noise,” in *Proc. MTS/IEEE OCEANS*, 2005, pp. 996–1001.

[7] M. Chitre, S. Shahabudeen, L. Freitag, and M. Stojanovic, “Recent advances in underwater acoustic communications & networking,” in *Proc. OCEANS*, 2008, pp. 1–10.

[8] D. Kilfoyle and A. Baggeroer, “The state of the art in underwater acoustic telemetry,” *IEEE J. Ocean. Eng.*, vol. 25, no. 1, pp. 4–27, Jan. 2000.

[9] J. Catipovic, M. Deffenbaugh, L. Freitag, and D. Frye, “An acoustic telemetry system for deep ocean mooring data acquisition and control,” in *Proc. OCEANS*, 1989, pp. 887–892.

[10] S. Coatelan and A. Glavieux, “Design and test of a coding OFDM system on the shallow water acoustic channel,” in *Proc. OCEANS*, 1995, pp. 2065–2070.

[11] K. Scussel, J. Rice, and S. Merriam, “A new MFSK acoustic modem for operation in adverse underwater channels,” in *Proc. Oceans MTS/IEEE Conf.*, 1997, pp. 247–254.

[12] S. Coatelan and A. Glavieux, “Design and test of a multicarrier transmission system on the shallow water acoustic channel,” in *Proc. OCEANS*, 1994, pp. III/472–III/477.

[13] J. Potter, J. Alves, D. Green, G. Zappa, I. Nissen, and K. McCoy, “The JANUS underwater communications standard,” in *Proc. Underwater Commun. Netw. Conf.*, 2014, pp. 1–4.

[14] P. Zetterberg, F. Lindqvist, and B. Nilsson, “Underwater acoustic communication with multicarrier binary frequency shift keying,” *IEEE J. Ocean. Eng.*, vol. 47, no. 1, pp. 255–267, Jan. 2022.

[15] D. Slepian, “Permutation modulation,” *Proc. IEEE*, vol. 53, no. 3, pp. 228–236, Mar. 1965.

[16] H. L. Schneider, “Data transmission with FSK permutation modulation,” *Bell Syst. Tech. J.*, vol. 47, no. 6, pp. 1131–1138, 1968.

[17] Y. Wu and M. Zhu, “On approaching noncoherent channel capacity using nonbinary IRA code and multi-tone FSK,” in *Proc. OCEANS*, 2017, pp. 1–6.

[18] V. Lidström, E. S. Erstorp, M. L. Nordenvaad, P. Sigray, and J. Kuttenkeuler, “Non-coherent acoustic modulation for energy constrained underwater platforms,” in *Proc. OCEANS*, 2019, pp. 1–7.

[19] L. Fanari, E. Iradier, I. Bilbao, R. Cabrera, J. Montalban, and P. Angueira, “Comparison between different channel coding techniques for IEEE 802.11be within factory automation scenarios,” *Sensors*, vol. 21, no. 21, 2021, Art. no. 7209.

[20] O. Iscan, D. Lentner, and W. Xu, “A comparison of channel coding schemes for 5G short message transmission,” in *Proc. IEEE Globecom Workshops*, 2016, pp. 1–6.

[21] “3GPP TR 21.915 V15.0.0,” 2020. Accessed: Mar. 14, 2020. [Online]. Available: https://www.3gpp.org/release-15

[22] P. A. van Walree, “Propagation and scattering effects in underwater acoustic communication channels,” *IEEE J. Ocean. Eng.*, vol. 38, no. 4, pp. 614–631, Oct. 2013.

[23] M. Stojanovic and J. Preisig, “Underwater acoustic communication channels: Propagation models and statistical characterization,” *IEEE Commun. Mag.*, vol. 47, no. 1, pp. 84–89, Jan. 2009.

[24] S. Yang, G. B. Deane, J. C. Preisig, N. C. Sevuktekin, J. W. Choi, and A. C. Singer, “On the reusability of postexperimental field data for underwater acoustic communications R&D,” *IEEE J. Ocean. Eng.*, vol. 44, no. 4, pp. 912–931, Oct. 2019.

[25] P. van Walree, “Channel sounding for acoustic communications: Techniques and shallow-water examples,” Norwegian Defence Res. Establishment, Kjeller, Norway, Tech. Rep. FFI-rapport 2011/00007, 2011.

[26] P. A. van Walree, F.-X. Socheleau, R. Otnes, and T. Jenserud, “The Watermark benchmark for underwater acoustic modulation schemes,” *IEEE J. Ocean. Eng.*, vol. 42, no. 4, pp. 1007–1018, Oct. 2017.

[27] V. Lidström, “Evaluation of polar-coded noncoherent acoustic underwater communication,” *IEEE J. Ocean. Eng.*, early access, doi: 10.1109/JOE.2022.3227233.

[28] F. Gray, “Pulse code communication,” U.S. Patent 2 632 058, 1953. [Online]. Available: http://www.google.com/patents/US2632058

[29] Y. Yao, Y. Wu, M. Zhu, D. Li, and J. Tao, “Efficient on-off keying underwater acoustic communication for seafloor observation networks,” *Appl. Sci.*, vol. 10, no. 6, 2020, Art. no. 1986.

[30] V. Lidström, “Polar coded non-coherent acoustic underwater communication,” in *Proc. 5th Underwater Commun. Netw. Conf.*, 2021, pp. 1–5.

[31] M. Tuchler, A. Singer, and R. Koetter, “Minimum mean squared error equalization using a priori information,” *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 673–683, Mar. 2002.

[32] P. Blanchard, D. J. Higham, and N. J. Higham, “Accurately computing the log-sum-exp and softmax functions,” *IMA J. Numer. Anal.*, vol. 41, no. 4, pp. 2311–2330, 2021.

[33] P. van Walree, R. Otnes, and T. Jenserud, “The Watermark manual and user’s guide—Version 1.0,” Forsvarets Forskningsinstitutt, Kjeller, Norway, Tech. Rep. 16/01378, 2016. [Online]. Available: https://www.ffi.no/en/publications-archive/the-watermark-manual-and-users-guide-version-1.0

[34] V. Lidström, “Implications of time variability on a Ricean noncoherent model in acoustic underwater communication,” in *Proc. 6th Underwater Commun. Netw. Conf.*, 2022, pp. 1–5.

[35] E. Arikan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[36] I. Tal and A. Vardy, “List decoding of polar codes,” *IEEE Trans. Inf. theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

[37] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, “LLR-based successive cancellation list decoding of polar codes,” *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5165–5179, Oct. 2015.

[38] D. Kern, S. Vorkoper, and V. Kuhn, “A new code construction for polar codes using min-sum density,” in *Proc. 8th Int. Symp. Turbo Codes Iterative Inf. Process.*, 2014, pp. 228–232.

[39] E. Arkan, “A performance comparison of polar codes and Reed-Muller codes,” *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447–449, Jun. 2008.

[40] R. Mori and T. Tanaka, “Performance of polar codes with the construction using density evolution,” *IEEE Commun. Lett.*, vol. 13, no. 7, pp. 519–521, Jul. 2009.

[41] V. Bioglio, C. Condo, and I. Land, “Design of polar codes in 5G new radio,” *IEEE Commun. Surv. Tut.*, vol. 23, no. 1, pp. 29–40, First Quarter 2021.

[42] T. Kloek and H. Dijk, “Bayesian estimates of equation system parameters: An application of integration by Monte Carlo,” *Econometrica*, vol. 46, no. 1, pp. 1–19, 1978.

[43] Y. M. Aval and M. Stojanovic, “Differentially coherent multichannel detection of acoustic OFDM signals,” *IEEE J. Ocean. Eng.*, vol. 40, no. 2, pp. 251–268, Apr. 2015.

[44] P. van Walree and G. Leus, “Robust underwater telemetry with adaptive turbo multiband equalization,” *IEEE J. Ocean. Eng.*, vol. 34, no. 4, pp. 645–655, Oct. 2009.

[45] Y. Bai and P.-J. Bouvet, “Orthogonal chirp division multiplexing for underwater acoustic communication,” *Sensors*, vol. 18, no. 11, 2018, Art. no. 3815.

Viktor Lidström received the M.Sc. degree in space engineering, with a specialization in spacecraft and instrumentation, from the Luleå University of Technology, Luleå, Sweden, in 2017. He is currently working toward the Ph.D. degree in underwater acoustic communication with a focus on robust noncoherent links at the Department of Engineering Mechanics, KTH Royal Institute of Technology, and affiliated with the Swedish Defence Research Agency FOI, Stockholm.

His research interests include link adaptation and flooding networks.