Preamble-Free Frame Synchronization for Short-Packet Low-Latency Communications

Javier Villares, Meritxell Lamarca
Dept. of Signal Theory and Communications
Technical University of Catalonia (UPC)
Barcelona, Spain
javier.villares@upc.edu, meritxell.lamarca@upc.edu

***Abstract*—In this paper we show that the transmission of preambles is very inefficient in case of short-packet asynchronous communications. The study concludes that packet error rate is minimized when preambles are omitted and the receiver performs code-aided frame synchronization. In that case, it is possible to approach the theoretical limits derived in the literature for the short blocklength regime at the expense of increasing the computational cost of the receiver (multiple decodings per packet). To illustrate the potential of preamble-less communications, we focus on tail-biting convolutional codes, proving the suitability of this coding scheme for supporting asynchronous low-latency communications.**

***Index Terms*—frame synchronization, short-packet communications, tail-biting convolutional codes**

I. Introduction

EMERGING 5G (and beyond) networks are envisaged to provide support to delay-sensitive applications such as remote monitoring and control for industrial automation, remote surgery and autonomous driving, among others [1]. A fundamental ingredient of low latency communications is the reduction of the transmitted blocklength. If the packet length is greatly reduced, the gap to the Shannon's channel capacity increases and new performance bounds [2] [3] as well as efficient error-correcting codes need to be used [4] [5] [6].

The study of the theoretical limits of short blocklength communications [2] [3] is extended in [7] [8] [9] to *asynchronous* discrete memoryless channels (DMC). The main conclusion is that there is no loss in terms of error probability, neither rate, if the receiver is not initially synchronized and packet synchronization has to be performed while decoding the received message. This important result holds even if the temporal uncertainty on the packets time of arrival increases exponentially with the packet length. Moreover, Polyanskiy states in [8] that the use of known training sequences to facilitate packet synchronization is not efficient; contrarily, frame synchronization should be carried out using exclusively the redundancy introduced by the channel code (code-aided synchronization [10]). Although no proof is available in the literature, the conclusions in [8] for DMC are expected to apply also to other channels of interest. To illustrate the inefficiency of preambles, we have simulated the physical layer of a low-latency *asynchronous* binary-input additive white Gaussian noise (BI-AWGN) channel with packet lengths on the order of 50 to 100 symbols and state-of-the-art short blocklength channel codes [4] [5] [6] concluding that, as already suggested in the aforementioned theoretic studies, the minimum probability of error is attained without preambles.

Although the referred theoretic works date back to the early 2010s [7] [8], most of the papers dealing with low latency communications are still based on inefficient training sequences. In [11], for example, Bana et al. tackle the problem of joint frame synchronization and decoding and determine the optimum signaling overhead of time-multiplexed and superimposed training sequences assuming that information is encoded using a theoretic code attaining the finite-blocklength normal approximation for the AWGN channel [2, Th. 54]. Using the random channel union bound [2, Th. 16], similar results are derived in [12] [13], and then tested numerically, using a practical short blocklength Polar code. Regarding the related problem of joint channel estimation and decoding in fading channels, [14] [15] optimize the number and power of pilot symbols that minimize the required signal-to-noise ratio (SNR) for a given probability of error, transmission rate and (short) packet length. Contrarily, to our best knowledge, only a few papers have addressed so far the design and analysis of preamble-free *short-packet* communications in the context of emerging low-latency applications. Three distinguished examples are [16] (frame synchronization), [17] (channel estimation) and [18] (frame detection). This paper aims to contribute to this research topic by studying the feasibility of code-aided frame synchronization for short-blocklength communications. To exemplify the proposed receiver architecture and assess its performance, we adopt tail-biting convolutional codes (TB-CC) as a relevant example of an efficient short-blocklength error-correcting code [4] [5] [6] [19].

Notation: vectors are denoted by boldfaced lower case letters ($\mathbf{v}$). The k-th element of vector $\mathbf{v}$ is denoted by $[\mathbf{v}]_k$. Transpose and conjugate transpose operators are denoted as $(\ )^T$ and $(\ )^H$, respectively. $\Re\{x\}$ and $\Im\{x\}$ extract the real and imaginary part of the complex number x, respectively. $\|\mathbf{v}\|_p$ stands for the p-norm of vector $\mathbf{v}$. $\Pr(A|B)$ is the probability of event A given the occurrence of event B, $\mathrm{p}(\mathbf{a}|\mathbf{b})$

Work supported by projects MAYTE (PID2022-136512OB-C21) funded by MICIU/AEI/10.13039/501100011033 and ERDF/EU, and grant 2021 SGR 01033 by the Catalan Government (AGAUR)

the conditional distribution of random vector $\mathbf{a}$ given $\mathbf{b}$ and $\mathbb{E}_{\mathbf{a}|\mathbf{b}}\{\cdot\}$ the expectation with respect to $\mathrm{p}(\mathbf{a}|\mathbf{b})$.

II. Problem Statement

Adopting the setup in [11], our objective is to minimize the packet error rate (PER) for a *fixed* packet length of n symbols when the packet time of arrival τ_0 is unknown (Fig. 1) and therefore the receiver has to determine when the initial symbol of every packet is received (frame synchronization). The blocklength n is fixed to guarantee that the maximum tolerated latency is not exceeded. Focusing on low-latency services, we study the case of packet lengths (n) not larger than 100 symbols.

As illustrated in Fig. 1, we start considering that packets are composed of a known preamble of n_p symbols followed by a sequence of $n_c = n - n_p$ coded QPSK symbols that conveys a message $\mathbf{b}_0$ of k information bits. Later in this paper, we will show that minimum PER is achieved setting $n_p = 0$ (no preamble) and performing frame synchronization directly on the unknown codeword. Note that QPSK is adopted because it is well-suited for reliable, low data rate communications. However, the study can be easily extended to other modulations.

Coded symbols are generated using a tail-biting convolutional code (TB-CC) with large memory [20], a family of codes which has been shown to perform well for very short packet lengths such as the ones considered in this paper ($n_c < 100$). TB-CC was already adopted for the broadcast and control channels of Long-Term Evolution (LTE) [21] and it was also one of the candidates for supporting ultra-reliable low-latency services in 5G [6]. Nevertheless, we expect the results obtained in this paper to be applicable to systems employing other codes with good performance at short codeword lengths, such as Polar codes with CRC-list decoding and BCH codes using Order Statistics Decoding (OSD) [6].

The rate of the TB-CC is adjusted by means of puncturing [22] to generate $2n_c = 2(n - n_p)$ coded bits, so the code rate is $R_c = \frac{k}{2(n-n_p)}$. Note that, since k and n are fixed, the encoder rate has to be adjusted according to the preamble length n_p. As shown in Fig. 2, the encoder output is interleaved (or scrambled) and mapped into the codeword $\mathbf{x}_c$ of n_c QPSK symbols. Note that interleaving (or scrambling) is required to break the circularity of TB-CC codes and allow code-aided synchronization [23].

Finally, the transmitted packet $\mathbf{x} = [\mathbf{x}_p \; \mathbf{x}_c]$ is the stacking of the preamble and codeword vectors:

$$\mathbf{x}_p = [x_p(0), \ldots, x_p(n_p - 1)] \quad (1)$$

$$\mathbf{x}_c = [x_c(0), \ldots, x_c(n_c - 1)] = \mathrm{f}_{\mathrm{enc}}[\mathbf{b}_0] \quad (2)$$

with $\mathrm{f}_{\mathrm{enc}}[\cdot]$ denoting hereinafter the overall encoding function that includes TB-CC encoding, interleaving, and bit-to-symbol mapping (Fig. 2).

At the receiver end we assume perfect carrier and symbol (fine timing) synchronization so that the communication system in Fig. 2 observed at the symbol rate can be modeled as a memoryless additive white Gaussian noise (AWGN) channel.

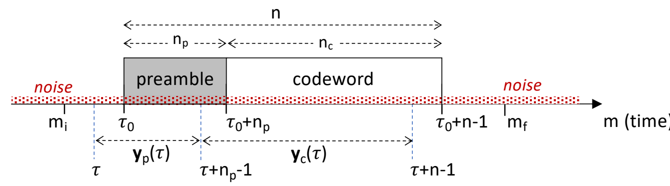

Fig. 1. Packet structure with τ_0 the time index at which the packet arrives at the receiver. Frame (coarse timing) synchronization is required to determine the value of $\tau_0 \in \mathbb{Z}$. The observed vectors $\mathbf{y}_p(\tau)$ and $\mathbf{y}_c(\tau)$ for a tentative value $\tau \in \mathbb{Z}$ are also indicated.

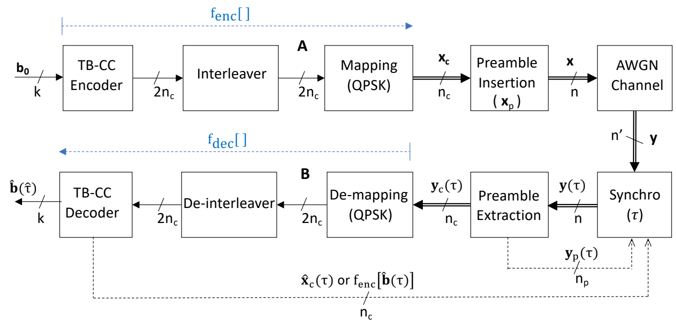

Fig. 2. Block diagram with A and B indicating the input and output of the equivalent BI-AWGN channel.

Accordingly, we will only address the problem of coarse timing synchronization, also referred to in the literature as frame synchronization [24] [25].

The demodulator output for the mth received symbol is given by

$$y(m) = \begin{cases} w(m) & m - \tau_0 < 0 \\ x_p(m - \tau_0) + w(m) & 0 \leq m - \tau_0 < n_p \\ x_c(m - \tau_0 - n_p) + w(m) & n_p \leq m - \tau_0 < n \\ w(m) & m - \tau_0 \geq n \end{cases} \quad (3)$$

where $w(m)$ is zero-mean white Gaussian noise of variance σ_w^2 and τ_0 is the unknown *integer* delay. In order to perform frame synchronization and decoding, the receiver processes the following vector of samples:

$$\mathbf{y} = [y(m_i), \cdots, y(m_f)]^T, \quad (4)$$

which contains the n (noisy) symbols of the received packet surrounded by noise. We assume that, whatever the value of τ_0, the entire packet is contained in $\mathbf{y}$, which means that $m_i \leq \tau_0$ and $m_f \geq n + \tau_0 - 1$ for any possible value of τ_0. Also, we assume that the time between consecutive packets is long enough (moderate traffic load) so that, within the receiver observation window, the signal consists of a single packet with leading/trailing noise samples.

For a given tentative delay estimate τ, we denote as $\mathbf{y}_p(\tau)$ and $\mathbf{y}_c(\tau)$ the observed samples that would correspond to the preamble and the codeword, respectively, under the assumption that the packet arrives at time τ (Fig. 1):

$$\mathbf{y}_p(\tau) = [y(\tau), \ldots, y(\tau + n_p - 1)]^T \quad (5)$$

$$\mathbf{y}_c(\tau) = [y(\tau + n_p), \ldots, y(\tau + n - 1)]^T \quad (6)$$

Assuming that the packet delay is τ, the overall decoder mapping is denoted hereinafter as $\hat{\mathbf{b}}(\tau) = \mathrm{f}_{\mathrm{dec}}[\mathbf{y}_c(\tau)]$ and includes de-mapping, de-interleaving and TB-CC decoding (see Fig. 2).

III. Frame synchronization

A. Classic non-code-aided (NCA) frame synchronization

In this section, we briefly summarize the maximum-likelihood (ML) frame synchronizer when the code structure is not exploited and the data is assumed to be composed by a known preamble followed by n_c unknown QPSK symbols. Although this paper focuses on code-aided synchronization, this low-complexity synchronizer will be employed to initialize the code-aided algorithms presented next.

Since $\mathbf{x}_c$ is QPSK-modulated, the ML frame synchronizer admits the following expression [24]:

$$\hat{\tau}_{nca} = \arg\max_{\tau} F(\tau) \tag{7}$$

with

$$\begin{aligned} F(\tau) \doteq\ & \frac{2}{\sigma_w^2}\Re\left\{\mathbf{y}_p^H(\tau)\mathbf{x}_p\right\} \\ & + \sum_{m=0}^{n_c-1} \ln\cosh\left(\frac{2\Re\{[\mathbf{y}_c(\tau)]_m\}}{\sigma_w^2}\right) \\ & + \sum_{m=0}^{n_c-1} \ln\cosh\left(\frac{2\Im\{[\mathbf{y}_c(\tau)]_m\}}{\sigma_w^2}\right) \end{aligned} \tag{8}$$

which is proportional at high SNR ($\sigma_w^2 \to \infty$) to

$$\Re\left\{\mathbf{y}_p^H(\tau)\mathbf{x}_p\right\} + \|\Re\{\mathbf{y}_c(\tau)\}\|_1 + \|\Im\{\mathbf{y}_c(\tau)\}\|_1 \tag{9}$$

Note that the detector in [24] is slightly modified here to deal with a single packet surrounded by noise instead of a continuous bit stream with inserted periodic sync words, as considered originally in the paper. Moreover, following the approach in [11], the search of the maximum in (7) is initiated when a threshold λ is surpassed: $F(\tau) > \lambda$. However, in contrast to [11], we assume that packets are not lost during the recovery of a recent false alarm. The threshold λ is chosen such that the miss-detection probability $\Pr(F(\tau_0) < \lambda)$ becomes negligible compared to the target PER.

B. Code-aided (CA) frame synchronization

If the receiver is aware of the channel code employed at the transmitter, it can exploit the redundancy in the code to improve the frame synchronizer performance. ML estimation of the delay τ_0 can be obtained as follows:

$$\hat{\tau}_{ca_1} = \arg\max_{\tau} \mathrm{p}(\mathbf{y}|\tau) = \arg\max_{\tau} \mathbb{E}_{\mathbf{b}|\tau}\left\{\mathrm{p}(\mathbf{y}|\mathbf{b},\tau)\right\} \tag{10}$$

where $\mathbf{y}$ is the vector of observed samples (4) and the expectation runs over all possible transmitted messages $\mathbf{b}$. The authors in [25] proposed to obtain this estimate employing the Expectation-Maximization (EM) algorithm which, under mild conditions [26], provides ML estimates on the unknown parameter τ_0. Using the EM principle, we can obtain $\hat{\tau}_{ca_1}$ in (10) as follows:

$$\hat{\tau}_{ca_1} = \arg\max_{\tau} \mathbb{E}_{\mathbf{b}|\mathbf{y},\tau}\left\{\ln \mathrm{p}(\mathbf{y}|\mathbf{b},\tau)\right\} \tag{11}$$

that, after some simple manipulations [25], can be expressed in case of QPSK symbols as

$$\hat{\tau}_{ca_1} = \arg\max_{\tau} \left[\Re\left\{\mathbf{y}_p^H(\tau)\mathbf{x}_p\right\} + \Re\left\{\mathbf{y}_c^H(\tau)\hat{\mathbf{x}}_c(\tau)\right\}\right] \tag{12}$$

where $\hat{\mathbf{x}}_c(\tau)$ is the vector of symbol-by-symbol MAP soft-decisions obtained under a tentative alignment τ. In particular, the soft-decision on the coded symbol $x_c(m) = [\mathbf{x}_c]_m$ is given by

$$[\hat{\mathbf{x}}_c(\tau)]_m = \sum_{i=1}^{4} s_i \Pr\left(x_c(m) = s_i|\mathbf{y}_c(\tau)\right) \tag{13}$$

where $s_i \in \{\pm 1 \pm j\}$ is one of the four possible symbols of the QPSK constellation and $\Pr\left(x_c(m)|\mathbf{y}_c(\tau)\right)$ is the a posteriori probability (APP) of $x_c(m)$ provided by the maximum a posteriori (MAP) decoder, which can be implemented with the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [27].

Once the receiver is synchronized, ML decoding is performed as follows:

$$\hat{\mathbf{b}} = \hat{\mathbf{b}}(\hat{\tau}_{ca_1}) = \mathrm{f}_{\mathrm{dec}}[\mathbf{y}_c(\hat{\tau}_{ca_1})] \tag{14}$$

As the BCJR decoder is already used for synchronization in (13), it is reasonable to also use it in (14) to decide the transmitted message $\mathbf{b}_0$. In that case, because the BCJR decoder takes MAP decisions at the bit level, i.e., $\max_{b(k)} \mathrm{p}(b(k)|\mathbf{y}_c(\hat{\tau}_{ca_1}))$, the resulting receiver minimizes the Bit Error Rate (BER) instead of the PER. Although not optimal in terms of PER, this will be the decoder adopted in Sec. V to evaluate the code-aided synchronizer presented in this section.

C. Joint code-aided (CA) frame synchronization and data detection

As the frame synchronization stage described above requires multiple decoding attempts for different values of the delay τ, it is worth comparing the complexity and performance of the algorithm presented in the previous section with the minimum-PER receiver that performs joint MAP synchronization and decoding:

$$(\hat{\mathbf{b}}, \hat{\tau}_{ca_2}) = \arg\max_{\mathbf{b},\tau} \ln \mathrm{p}(\mathbf{b},\tau|\mathbf{y}) \tag{15}$$

Assuming that all the sequences $\mathbf{b}$ and time alignments τ are equally likely, MAP is equivalent to ML and (15) can be written as

$$\begin{aligned} \max_{\mathbf{b},\tau} \ln \mathrm{p}(\mathbf{y}|\mathbf{b},\tau) = \max_{\mathbf{b},\tau} \ln \mathrm{p}(\mathbf{y}_p(\tau), \mathbf{y}_c(\tau)|\mathbf{b}) = \\ \max_{\tau}\left[\ln \mathrm{p}(\mathbf{y}_p(\tau)) + \max_{\mathbf{b}} \ln \mathrm{p}(\mathbf{y}_c(\tau)|\mathbf{b})\right] = \\ \max_{\tau}\left[\Re\{\mathbf{y}_p^H(\tau)\mathbf{x}_p\} + \max_{\mathbf{b}} \Re\{\mathbf{y}_c^H(\tau)\mathrm{f}_{\mathrm{enc}}(\mathbf{b})\}\right] \end{aligned} \tag{16}$$

assuming QPSK-modulated symbols. Using that $\hat{\mathbf{b}}(\tau) = \mathrm{f}_{\mathrm{dec}}[\mathbf{y}_c(\tau)]$, we can estimate τ_0 from (16) as follows:

$$\hat{\tau}_{ca_2} = \arg\max_{\tau} \left[\Re\{\mathbf{y}_p^H(\tau)\mathbf{x}_p\} + \Re\{\mathbf{y}_c^H(\tau)\mathrm{f}_{\mathrm{enc}}[\hat{\mathbf{b}}(\tau)]\} \right] \quad (17)$$

and, using this estimate of τ_0, to retrieve the information message using the ML decoder:

$$\hat{\mathbf{b}} = \hat{\mathbf{b}}(\hat{\tau}_{ca_2}) = \mathrm{f}_{\mathrm{dec}}[\mathbf{y}_c(\hat{\tau}_{ca_2})] \quad (18)$$

In case of the studied TB-CC, the ML decoder in (14) can be implemented approximately by means of the Wrap-Around Viterbi Algorithm (WAVA) decoder [28] or, alternatively, by taking hard-decisions on the Max-Log-MAP [29] soft output.

Similar to the synchronizer in (12) (Sec. III-B), the frame synchronizer proposed in (17) requires to decode the received packet for every possible alignment τ. In order to reduce the computational burden, we have limited the search in (17) to a reduced interval $\mathcal{W} = [\hat{\tau}_{nca} - N/2, \ldots, \hat{\tau}_{nca} + N/2]$ around an initial estimate $\hat{\tau}_{nca}$, which is calculated employing the non-CA estimator in Sec. III-A. In that way, the reception of each packet requires only $N+1$ decoding attempts with N adjusted so that $\Pr(\tau_0 \notin \mathcal{W}) = \Pr(|\hat{\tau}_{nca} - \tau_0| > N/2)$ is negligible compared to the target PER.

IV. Performance analysis

The performance of the proposed CA synchronization schemes can be analyzed in terms of Frame Synchronization Error Rate $\mathrm{FSER} = \Pr(\hat{\tau} \neq \tau_0)$ and the Packet Error Rate $\mathrm{PER} = \Pr(\hat{\mathbf{b}} \neq \mathbf{b}_0)$. Taking into account that the probability of successful decoding is vanishingly small when the frame synchronizer fails, PER can be expressed accurately as the sum of two terms:

$$\mathrm{PER} \approx \mathrm{FSER} + (1 - \mathrm{FSER})\mathrm{PER}_{\mathrm{sync}} \quad (19)$$

where $\mathrm{PER}_{\mathrm{sync}} = \Pr(\hat{\mathbf{b}}(\hat{\tau}) \neq \mathbf{b}_0 | \hat{\tau} = \tau_0)$ stands for the probability of decoding error for those packets that are correctly synchronized. If we define $\mathrm{PER}_0 = \Pr(\hat{\mathbf{b}}(\tau_0) \neq \mathbf{b}_0)$ as the probability of wrong decoding in the synchronous case (known τ_0), it follows that $\mathrm{PER}_{\mathrm{sync}}$ is always upper-bounded by PER_0. The reason is that packets that are correctly aligned by the CA synchronizer are also more likely to be successfully decoded.

While FSER decreases with the preamble length n_p, both $\mathrm{PER}_{\mathrm{sync}}$ and PER_0 are increasing functions of n_p because the codeword length $n_c = n - n_p$ decreases with n_p since the packet length is fixed to n. Advances in the study of performance limits of channel codes in the finite-length regime [4] [5] [6] have shown that the value of PER_0 for optimal short-blocklength codes is approximately lower-bounded by the following normal approximation (NA) [3]:

$$\mathrm{PER}_{\mathrm{NA}} = Q\left(\frac{C - \frac{k}{2n_c} + \frac{\ln_2(2n_c)}{4n_c}}{\ln_2(e)\sqrt{V/(2n_c)}} \right) \quad (20)$$

where $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-t^2/2}dt$ is the complementary Gaussian cumulative distribution function and, C and V the capacity and dispersion, respectively, of the considered BI-AWGN channel [3, Fig. 6] (Fig. 2). These two constants are increasing functions of the signal-to-noise ratio per bit $\frac{E_b}{N_0}$, and can be obtained numerically using saddle point approximations [3] [30]. A salient feature of (20) is that it is an accurate approximation even if the codeword length n_c is as short as 50 symbols, becoming in that way a useful benchmark for the evaluation of short packet communications.

Since closed-form expressions for FSER, PER, and PER_0 are not available, they are evaluated numerically in the next section through Monte Carlo simulations for the two CA synchronization schemes considered in this paper, and compared with the normal approximation in (20).

V. Simulations

As described in Sec. II, short messages of k bits are encoded using a binary TB-CC. A mother code of rate $1/2$ is punctured to reduce the encoder output length from $2k$ to $2n_c$ bits, with n_c the length of the transmitted codeword in QPSK symbols. Taking into consideration [22], regular (periodic) puncturing is applied. The TB-CC of memory 11 bits in [20] (octal polynomials 5537 and 6131) is adopted as the baseline because it represents a good trade-off between performance and decoding complexity. To make the results independent of the selected interleaver, the interleaver is changed randomly at every transmitted packet. Simulations have evidenced that the election of the interleaver is not critical and all randomly-generated interleavers have basically the same performance.

Regarding the TB-CC decoder, two different algorithms are considered: BCJR [31] and WAVA [28]. While BCJR is the decoder adopted in Sec. III-B for ML synchronization, WAVA is the decoding scheme employed in Sec. III-C for optimal joint synchronization and decoding. We implement the sub-optimal BCJR decoder in [31], where the initial forward and backward metrics are trained through an additional recursion over the TB-CC circular trellis, yielding a total of four one-way recursions. Note that the complexity of the optimal BCJR [32] is not affordable in case of the adopted large-memory TB-CC because it would require to find the best circular path in the trellis for each possible initial state. For the WAVA algorithm, we perform up to four iterations, each corresponding to one one-way trellis recursion.

As a baseline, we have considered messages of $k = 72$ bits and a blocklength of $n = 72$ QPSK symbols. Different preamble lengths n_p are simulated varying the codeword length $n_c = n - n_p$ from 48 symbols ($n_p = 24$) up to 72 symbols ($n_p = 0$). Replicating [11], the preamble is a complex-valued Zadoff-Chu sequence of maximal aperiodic autocorrelation [33].

The received E_b/N_0 is set to 3 dB and packets are synchronized following the procedure in sections Sec. III-B and III-C. The search interval $\mathcal{W} = [\hat{\tau}_{nca} - N/2, \ldots, \hat{\tau}_{nca} + N/2]$ in (12) and (17) is adjusted to guarantee that the initial NCA synchronization has minor impact on the achieved PER. In particular, N is set to the lowest value satisfying $\Pr(\tau_{nca} \notin \mathcal{W}) < \mathrm{PER}/10$, and therefore it changes with n and the code rate R_c.

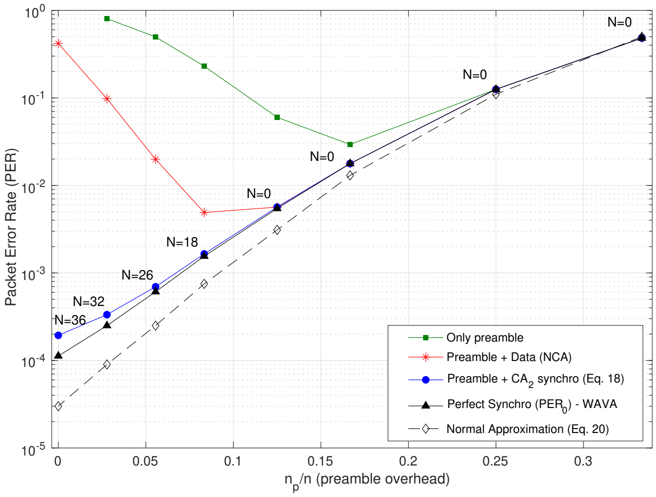

Fig. 3. PER as a function of the preamble overhead (n_p/n) for $E_b/N_0 =$ 3dB and a TB-CC of memory 11 bits. The message and packet length is set to $k = 72$ bits and $n = 72$ QPSK symbols, respectively. The number of decoding attempts $(N+1)$ is indicated for each simulated packet configuration.

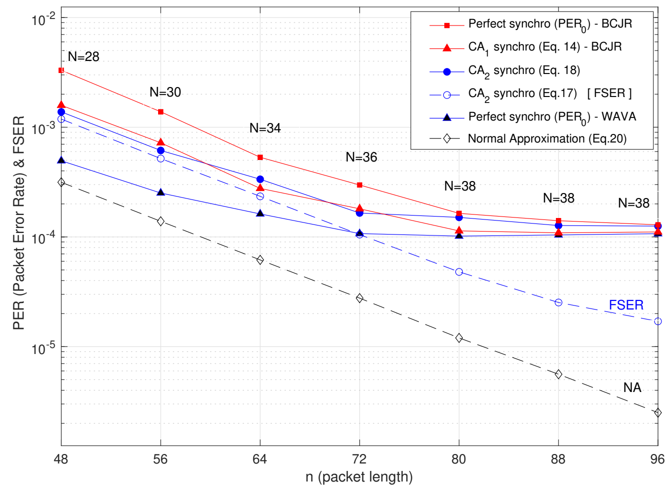

Fig. 4. PER and FSER versus packet length n for $E_b/N_0 = $ 3dB and a TB-CC of memory 11 bits. The code rate is fixed to $R_c = 1/2$ (no puncturing) and no preamble is transmitted ($n_p = 0$). The number of decoding attempts $(N+1)$ is indicated for each simulated packet configuration.

In Fig. 3, PER is represented as a function of the preamble overhead n_p/n. As concluded in [11], if synchronization is based exclusively on the preamble (green square markers), there is an optimum preamble length that minimizes the receiver PER. This optimum value arises as a trade-off between the performance of packet synchronization (FSER), which calls for a longer preamble, and message decoding (PER_0), which asks for a longer codeword of lower code rate. Simulations indicate that the preamble can be shortened for a better PER by processing the whole packet with the non-code-aided synchronizer in (7) (red asterisk markers). On the other hand, if code-aided (CA) synchronization is carried out according to (17)-(18), the minimum PER is achieved when the preamble length n_p goes to zero (blue circle markers), as already suggested in [8] [9]. As shown in Fig. 3, the loss with respect to perfect synchronization is only appreciable if the preamble length is very short or null. In the figure, the number of required decoding attempts per packet $(N+1)$ is indicated, showing that the receiver's computational cost increases considerably if the preamble is shortened or eliminated. Finally, to assess the efficiency of the adopted TB-CC code, the normal approximation in (20), which predicts the PER_0 of theoretically-optimum short blocklength codes, is also depicted in the figure (black diamond markers).

In Fig. 4, the receiver PER is evaluated as a function of the packet length n, assuming preamble-less transmissions ($n_p = 0$) and fixed code rate $R_c = 1/2$. Three main conclusions can be drawn from this figure. First, the studied TB-CC code of memory 11 bits is only efficient for packet lenghs ($n \leq 72$). For longer codewords, the gap to the normal approximation in (20) increases severely, as already noticed in [20]. Second, the achievable PER is limited by the synchronizer performance (FSER) for very short packets ($n \leq 72$), as shown in Fig. 4 for the CA synchronization scheme presented in Sec. III-C (CA_2). Third, the advantage of CA_2 with respect to CA_1 (Sec. III-B) is significant for $n \leq 72$ but it vanishes for longer packet lengths. The reason is that CA_1 is based on a suboptimal implementation of the BCJR algorithm [31] that performs two initial recursions on the trellis to estimate the initial value of the forward and backward metrics. This estimation improves if the trellis length n_c increases. Therefore, for packets longer than 80 symbols ($n_c \geq 80$), both synchronization algorithms show similar performance in terms of PER.

Although theoretical studies based on random coding suggest the existence of channel codes that allow perfect frame synchronization without degrading the PER [8], we have found in Fig. 4 that the synchronization of TB-CC can introduce a small degradation on the PER in case of the largest constraint lengths if the packet length is not sufficiently large ($n < 88$). The reason is that the probability of wrong synchronization FSER increases slightly as we enlarge the memory of the code, becoming comparable to the code PER with perfect synchronization (PER_0) for the longer memory lengths. This conclusion is evidenced in Table I (next page) where we have evaluated the CA synchronizer in Sec. III-C using TB-CCs of increasing memory. In this table, the initial non-CA synchronization is assumed to be perfect ($\hat{\tau}_{nca} = \tau_0$) to make the results independent of N and focus on the inherent limitations of TB-CC when used for code-aided frame synchronization.

VI. Conclusions

In this work, we have shown that the use of preambles is very inefficient in case of asynchronous short blocklength communications. Without preambles, the receiver has to determine the start of incoming packets by exploiting the redundancy of the error-correcting code with the aid of the decoder (code-aided frame synchronization). By adopting code-aided

Code Memory	FSER	PER_0	PER
4	1.8×10^{-5}	2.6×10^{-2}	2.6×10^{-2}
8	6.8×10^{-5}	1.0×10^{-3}	1.0×10^{-3}
11	8.0×10^{-5}	1.2×10^{-4}	1.8×10^{-4}
14	8.4×10^{-5}	5.0×10^{-5}	1.2×10^{-4}

TABLE I
PROBABILITY OF INCORRECT SYNCHRONIZATION FOR DIFFERENT MEMORY LENGTHS AND THE CA SYNCHRONIZER OF SEC. III-C. SIMULATIONS ARE CARRIED OUT FOR $n_p = 0$ (NO PREAMBLE), $n = k = 72$ AND $E_b/N_0 = 3dB$.

synchronization, it is shown that near-perfect synchronization of short packets can be achieved without preambles. This result is demonstrated in the relevant case of large-memory tail-biting convolutional codes (TB-CC). This family of codes is chosen because it performs close to the theoretical limits when the blocklength is really short ($n \leq 72$) and is a distinguished candidate for ultra-reliable low-latency communications. For these very short bursts, we have evaluated the synchronization limits of large-memory TB-CC and determined the computational growth (number of decodings per packet) that is required to attain the referred performance limits. Future research on efficient joint decoding and synchronization strategies will become of paramount importance to keep within bounds the computational load of short blocklength receivers when they are asked to operate without preambles.

REFERENCES

[1] M. Bennis, M. Debbah and H. V. Poor, “Ultrareliable and Low-Latency Wireless Communication: Tail, Risk, and Scale,” in Proc. of the IEEE, vol. 106, no. 10, pp. 1834-1853, Oct. 2018.

[2] Y. Polyanskiy, H. V. Poor and S. Verdu, “Channel Coding Rate in the Finite Blocklength Regime,” in IEEE Trans. on Information Theory, vol. 56, no. 5, pp. 2307-2359, May 2010.

[3] T. Erseghe, “Coding in the Finite-Blocklength Regime: Bounds Based on Laplace Integrals and Their Asymptotic Approximations,” in IEEE Trans. on Information Theory, vol. 62, no. 12, pp. 6854-6883, Dec. 2016.

[4] G. Liva, L. Gaudio and T. Ninacs, “Code design for short blocks: A survey”, 2016 European Conf. on Networks and Communications, Athens (Greece), June 2016.

[5] O. Iscan, D. Lentner and W. Xu, “A Comparison of Channel Coding Schemes for 5G Short Message Transmission,” 2016 IEEE Globecom Workshops, Washington, DC, 2016.

[6] M. Shirvanimoghaddam et al., “Short Block-Length Codes for Ultra-Reliable Low Latency Communications,” in IEEE Communications Magazine, vol. 57, no. 2, pp. 130-137, Feb. 2019.

[7] A. Tchamkerten, V. Chandar, and G. W. Wornell, “Communication under strong asynchronism,” in IEEE Trans. on Information Theory, vol. 55, no. 10, pp. 4508–4528, Oct. 2009.

[8] Y. Polyanskiy, “Asynchronous Communication: Exact Synchronization, Universality, and Dispersion,” in IEEE Trans. on Information Theory, vol. 59, no. 3, pp. 1256-1270, March 2013.

[9] L. Li and A. Tchamkerten, “Second-Order Asymptotics for Communication Under Strong Asynchronism,” in IEEE Trans. on Information Theory, vol. 65, no. 5, pp. 2838-2849, May 2019.

[10] H. Wymeersch, H. Steendam, H. Bruneel and M. Moeneclaey, “Code-aided frame synchronization and phase ambiguity resolution,” in IEEE Trans- on Signal Processing, vol. 54, no. 7, pp. 2747-2757, July 2006.

[12] A. T. P. Nguyen, R. Le Bidan and F. Guilloud, “Trade-Off Between Frame Synchronization and Channel Decoding for Short Packets,” in IEEE Communications Letters, vol. 23, no. 6, pp. 979-982, June 2019.

[11] A. Bana, K. F. Trillingsgaard, P. Popovski and E. de Carvalho, “Short Packet Structure for Ultra-Reliable Machine-Type Communication: Tradeoff between Detection and Decoding,” 2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, April 2018.

[13] A. T. P. Nguyen, R. Le Bidan and F. Guilloud, “Superimposed Frame Synchronization Optimization for Finite Blocklength Regime,” 2019 IEEE Wireless Communications and Networking Conf. Workshop (WCNCW), Marrakech, Morocco, 2019.

[14] G. Liva, G. Durisi, M. Chiani, S. S. Ullah and S. C. Liew, “Short codes with mismatched channel state information: A case study,” 2017 IEEE Int. Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Sapporo, Japan, Jul. 2017

[15] J. Östman, G. Durisi, E. G. Ström, M. C. Coşkun and G. Liva, “Short Packets Over Block-Memoryless Fading Channels: Pilot-Assisted or Noncoherent Transmission?,” in IEEE Trans. on Communications, vol. 67, no. 2, pp. 1521-1536, Feb. 2019.

[16] R. Imad, C. Poulliat and S. Houcke, “Frame Synchronization Techniques for Non-Binary LDPC Codes over GF(q),” 2010 IEEE Global Communications Conf. (Globecom), Miami, FL, USA, Dec. 2010.

[17] M. Dorpinghaus, A. Ispas and H. Meyr, ”On the Gain of Joint Processing of Pilot and Data Symbols in Stationary Rayleigh Fading Channels,” in IEEE Trans. on Information Theory, vol. 58, no. 5, pp. 2963-2982, May 2012.

[18] A. Lancho, J. Östman and G. Durisi, ”On Joint Detection and Decoding in Short-Packet Communications,” 2021 IEEE Global Communications Conf. (Globecom), Madrid, Spain, Dec. 2021

[19] C. Yue, M. Shirvanimoghaddam, B. Vucetic, and Y. Li, “Channel coding and decoding schemes for URLLC,” in *Ultra-Reliable and Low-Latency Communications (URLLC) Theory and Practice*, T. Duong, S. Khosravirad, C. She, P. Popovski, M. Bennis, and T. Quek, Eds. Hoboken, NJ, USA: Wiley, 2023.

[20] L. Gaudio, T. Ninacs, T. Jerkovits and G. Liva, “On the Performance of Short Tail-Biting Convolutional Codes for Ultra-Reliable Communications,” 2017 Int. ITG Conf. on Systems, Communications and Coding (SCC), Hamburg, Germany, Feb. 2017.

[21] 3GPP TS 36.212 version 16.3.0 Release 16, “LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding”. Sep. 2020.

[22] J. Hagenauer, “Rate-compatible punctured convolutional codes (RCPC codes) and their applications,” in IEEE Trans. on Communications, vol. 36, no. 4, pp. 389-400, April 1988.

[23] J. Sun and M. C. Valenti, “Optimum frame synchronization for preamble-less packet transmission of turbo codes,” 2004 Asilomar Conf. on Signals, Systems and Computers (Asilomar), Pacific Grove, CA, USA, Nov. 2004.

[24] J. Massey, “Optimum Frame Synchronization,” in IEEE Trans. on Communications, vol. 20, no. 2, pp. 115-119, April 1972.

[25] H. Wymeersch, M. Moeneclaey, “ML frame synchronization for turbo and LDPC codes”, Proc. of Int. Symp. DSP and Communications Systems, Coolangatta, Australia, Dec. 2003.

[26] C. Herzet, H. Wymeersch, F. Simoens, M. Moeneclaey and L. Vandendorpe, “MAP-Based Code-Aided Hypothesis Testing,” in IEEE Trans. on Wireless Communications, vol. 7, no. 8, pp. 2856-2860, Aug. 2008.

[27] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate (Corresp.),” in IEEE Transactions on Information Theory, vol. 20, no. 2, pp. 284-287, Mar. 1974.

[28] R. Y. Shao, Shu Lin and M. P. C. Fossorier, “Two decoding algorithms for tailbiting codes,” in IEEE Trans. on Communications, vol. 51, no. 10, pp. 1658-1665, Oct. 2003.

[29] P. Robertson, P. Hoeher and E. Villebrun. ”Optimal and Sub-Optimal Maximum A Posteriori Algorithms Suitable for Turbo Decoding”. European Transactions on Telecommunications. 8 (2): 119–125. 1997.

[30] J. Font-Segura, G. Vazquez-Vilar, A. Martinez, A. Guillén i Fàbregas and A. Lancho, “Saddlepoint approximations of lower and upper bounds to the error probability in channel coding,” 2018 Annual Conf. on Information Sciences and Systems (CISS), Princeton, NJ, Mar. 2018.

[31] J. B. Anderson and S. M. Hladik, “Tailbiting MAP decoders,” in IEEE Journal on Selected Areas in Communications, vol. 16, no. 2, pp. 297-302, Feb. 1998.

[32] R. Johannesson and K. Zigangirov, *Fundamentals of Convolutional Coding*, 1st Edition, Piscataway, NJ, USA: Wiley IEEE Press, 1999.

[33] D. Chu, “Polyphase codes with good periodic correlation properties (Corresp.),” in IEEE Trans. on Information Theory, vol. 18, no. 4, pp. 531-532, July 1972.