A Sliding-Window LMMSE Turbo Equalization Scheme for OTFS

Franz Lampel, *Student Member, IEEE*, Alex Alvarado, *Senior Member, IEEE*, and Frans M. J. Willems, *Life Fellow, IEEE*

***Abstract*— This letter discusses a recently proposed linear minimum mean square error (LMMSE) turbo equalization scheme for OTFS. The complexity of the LMMSE-based equalizer, measured by the number of complex multiply-and-addition operations, scales with the cube of the number of transmitted symbols in an OTFS frame. To lower the complexity, we propose to limit the number of symbols in the equalization process by introducing a sliding window and exploiting the periodicity properties of the delay-Doppler domain. We demonstrate that for a practical channel model, which introduces fractional delay and Doppler shifts, even a narrow sliding window provides a bit error rate (BER) performance within $0.17\,\mathrm{dB}$ compared to the unconstrained LMMSE-based turbo equalizer. At the same time, the complexity of the sliding window LMMSE turbo equalizer is reduced by a factor of 10. The BER gap becomes even smaller with increasing window size, allowing for a complexity/performance trade-off.**

***Index Terms*— LMMSE equalization, OTFS, turbo equalization.**

I. Introduction

ORTHOGONAL Time Frequency Space (OTFS) modulation is a candidate waveform for sixth-generation communication systems. In OTFS, information-carrying signals are described by their delay-Doppler domain representation, which is advantageous for reliable communication in high-mobility scenarios [1]. The interaction of an OTFS waveform with a wireless channel in high-mobility scenarios results in inter-symbol interference (ISI) in the delay-Doppler domain, which is countered by equalizers.

Various equalizer structures are studied in the literature, which can be classified into iterative and non-iterative equalizers. See [2] for an overview. Examples of iterative equalizers are message-passing algorithms, such as in [3] or [4]. The performance of message-passing equalizers depends on the underlying factor graph and its properties, such as girth, sparsity, and interference strength [5, Sec. II]. Non-iterative equalizers, on the other hand, are mainly based on linear minimum mean square error (LMMSE) equalizers, which do not rely on factor graphs, making them an attractive alternative.

Manuscript received 30 August 2023; revised 25 September 2023; accepted 12 October 2023. Date of publication 20 October 2023; date of current version 12 December 2023. This research is supported by the Dutch Technology Foundation TTW, which is part of the Netherlands Organisation for Scientific Research (NWO) and is partly funded by the Ministry of Economic Affairs under the project Integrated Cooperative Automated Vehicles (i-CAVE). The associate editor coordinating the review of this letter and approving it for publication was W. Yuan. *(Corresponding author: Franz Lampel.)*

The authors are with the Information and Communication Theory Laboratory, Signal Processing Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (e-mail: f.lampel@tue.nl).

Digital Object Identifier 10.1109/LCOMM.2023.3326168

The main drawback of LMMSE equalizers is their computational complexity. In [6], an LMMSE equalizer that operates in the time domain is presented. The time domain formulation allows to reduce the complexity by exploiting the banded structure of the time domain channel. Similarly, a so-called cross-domain equalizer is presented in [7] and its low-complexity variant in [8]. The cross-domain equalizer iteratively passes soft information between the delay-Doppler and the time domain. A successive interference cancellation-based OTFS equalizer is discussed in [9], and an LMMSE-based turbo equalizer is presented in [10]. Both techniques employ reduced complexity algorithms to solve a linear system of equations in an approximate manner.

This letter proposes a novel yet simple LMMSE-based equalizer for OTFS based on a sliding window in the delay-Doppler domain. The sliding window effectively limits the number of received symbols in the equalization process, thereby reducing complexity in the first place. The sliding window is based on the limited spread in the Doppler domain, especially in low-latency applications, as discussed in [3, Sec. VI]. Operation in the delay-Doppler domain compared to the time domain allows the periodicity properties of the delay-Doppler domain to be exploited, further reducing the complexity of the LMMSE equalizer. We demonstrate that a sliding window that covers the effective ISI in the delay-Doppler domain provides performance close to that of an unconstrained LMMSE equalizer that considers all received symbols. Computational complexity is paramount when considering equalization schemes in which equalization is carried out iteratively, e.g., turbo or cross-domain equalization. For this purpose, we consider an LMMSE-based turbo equalizer in this letter.

Unlike the algorithmic approach in [10], our sliding-window equalizer offers a straightforward implementation and analysis. In addition, our equalizer has a fixed computational complexity, regardless of the number of reflectors in the delay-Doppler domain. In our work, we also do not make any restrictions on the fractional delay and Doppler shift introduced by the multipath propagation, which is a common assumption made in the literature, which is indeed the case, e.g., in [3] and [10].

II. System Model

We consider the coded OTFS system depicted in Fig. 1. As shown in Fig. 1, a data vector $\mathbf{b}$ is encoded using a channel encoder. Blocks of Q coded bits are then mapped into a symbol from an alphabet $\mathcal{A} = \{a_1, a_2, \ldots, a_M\}$ with cardinality $|\mathcal{A}| = M = 2^Q$. Thus, the coded bit vector is $\mathbf{c} = [\mathbf{c}_1^{\mathsf{T}}, \mathbf{c}_2^{\mathsf{T}}, \ldots, \mathbf{c}_N^{\mathsf{T}}]^{\mathsf{T}}$ with $\mathbf{c}_n = [c_{n,1}, c_{n,2}, \ldots, c_{n,Q}]^{\mathsf{T}}$. As discussed in [11, Sec. II], we consider alphabets for which

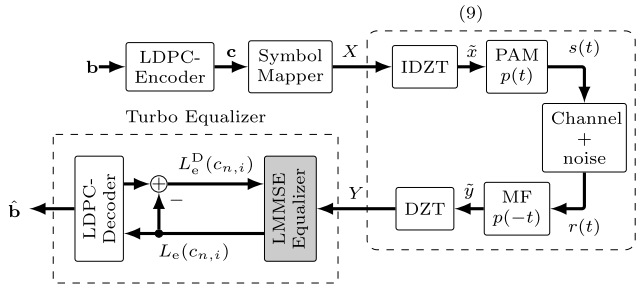

Fig. 1. System model of the considered DZT-based OTFS system. The focus of this letter is on the LMMSE equalizer.

$\sum_{i=1}^{M} a_i = 0$ and $M^{-1}\sum_{i=1}^{M}|a_i|^2 = 1$, i.e., for alphabets with zero mean and unit energy for equiprobable symbols.

The symbols are arranged on a two-dimensional grid of size $L \times K$, representing the fundamental rectangle of the discrete Zak representation or, equivalently, the delay-Doppler domain. We denote the discrete Zak domain by X and address individual symbols by subscripts, e.g., $X_{n,k}$. The indices n and k correspond to the delay and Doppler indices. Note that although we define only symbols within a rectangle of size $L \times K$, we inherently describe a two-dimensional grid in $\mathbb{C}^{\mathbb{Z}\times\mathbb{Z}}$, which follows from the periodicity properties of the discrete Zak domain, which states that $X_{n+mL,k+lK} = X_{n,k}e^{\imath 2\pi \frac{k}{K}m}$ where $\imath = \sqrt{-1}$. From the discrete Zak transform representation X, a periodic sequence $\tilde{x}$ is recovered using the inverse DZT, given as [12, eq. (2)]

$$\tilde{x}_n = \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} X_{n,k}. \tag{1}$$

The resulting sequence is periodic with period $N = KL$. One period of the sequence $\tilde{x}$ then defines the amplitude of the transmitted pulses using pulse amplitude modulation (PAM).

To preserve the cyclic properties of the discrete Zak transform when transmitting over multipath channels, a cyclic prefix is added by calculating the sequence for $n = -N_{\text{cp}}, \ldots, -1$, i.e., a single cyclic prefix for a block of $L \times K$ symbol in the delay-Doppler domain. The cyclic prefix length is chosen to satisfy $N_{\text{cp}} = \lceil \tau_{\max}/T \rceil$, where $\tau_{\max}$ is the maximum delay spread of the channel, T is the symbol period of the PAM, and $\lceil x \rceil$ denotes the smallest integer larger than or equal to x. The transmitted complex baseband signal is

$$s(t) = \sum_{n=-N_{\text{cp}}}^{N-1} \tilde{x}_n p(t - nT), \tag{2}$$

where $p(t)$ is a square-root Nyquist pulse.

We consider a linear time-variant channel consisting of P propagation paths. The received signal is a superposition of delayed and Doppler-shifted replicas of the transmitted signal, corrupted by additive noise, which may be described as

$$r(t) = \sum_{p=1}^{P} \alpha_p s(t - \tau_p) e^{\imath 2\pi \nu_p t} + w(t), \tag{3}$$

where α_p, τ_p, and ν_p are the complex attenuation factor, delay, and Doppler shift of the pth propagation path, respectively. We consider complex white Gaussian noise $w(t)$ with power spectral density N_0.

As shown in Fig. 1, the received signal $r(t)$ is passed through a matched filter and sampled with period T. After discarding the cyclic prefix (not shown in Fig. 1), the discrete sequence is

$$\tilde{y}_n = \sum_{p=0}^{P-1}\sum_{m=0}^{N-1} \alpha_p \tilde{x}_m e^{\imath 2\pi \frac{k_p}{N} m} \tilde{h}^p_{n-m} + w_n, \tag{4}$$

where $\tilde{h}^p_n = \sum_{m-\infty}^{\infty} h((n-n_p)T - mNT)$ are the samples of the fractional delayed and periodically extended Nyquist pulse (i.e., the convolution of transmit and receive filters), and $n_p = \tau_p/T$ is the normalized delay. The spacing between two adjacent points in the Doppler domain is $1/(NT)$, see [12, eq. (42)]. Hence, we express the Doppler shift in (4) as a multiple of the Doppler resolution, i.e., $k_p = \nu_p/(NT)$. The noise samples w_n are identically and independently distributed complex zero-mean Gaussian random variables with variance $\sigma^2 = N_0$. The sequence $\tilde{y}_n$ is subsequently mapped to the delay-Doppler domain using the discrete Zak transform, given as [12, eq. (1)]

$$Y_{n,k} = \frac{1}{\sqrt{K}} \sum_{l=0}^{K-1} y_{n+lL} e^{-\imath 2\pi \frac{k}{K} l}, \tag{5}$$

for $n = 0, 1, \ldots, L-1$ and $k = 0, 1, \ldots, K-1$.

Due to the linearity of the channel and the DZT, the received signal in the delay-Doppler domain is [12, eq. (46)]

$$Y_{n,k} = \sum_{p=0}^{P-1} \alpha_p \sum_{m=0}^{L-1} \left(\sum_{l=0}^{K-1} X_{m,l} U^p_{m,k-l} \right) H^p_{n-m,k} + W_{n,k}. \tag{6}$$

Here, $U^p_{n,k}$ and $H^p_{n,k}$ are the Doppler and delay spread functions and are given on the fundamental rectangle as

$$U^p_{n,k} = \frac{1}{\sqrt{K}} e^{\imath 2\pi \frac{k_p}{KL} n} e^{-\imath\pi \frac{K-1}{K}(k-k_p)} \frac{\sin\left(\pi(k-k_p)\right)}{\sin\left(\frac{\pi}{K}(k-k_p)\right)}, \tag{7}$$

and $H^p_{n,k}$ is the delay spread. If L is chosen such that LT covers the essential support of $h(t)$, i.e., $h(t)$ is essentially zero for $|t| > LT/2$, and if $\tau_p < LT$, then $H^p_{n,k}$ is

$$H^p_{n,k} = \frac{1}{\sqrt{K}} \begin{cases} e^{-\imath 2\pi \frac{k}{K}} \tilde{h}_{n+L}, & \text{if } n - n_p \le -L/2, \\ \tilde{h}_n, & \text{if } |n - n_p| < L/2, \\ e^{\imath 2\pi \frac{k}{K}} \tilde{h}_{L-n}, & \text{if } n - n_p \ge L/2. \end{cases} \tag{8}$$

The transformed noise $W_{n,k}$ in (6) has the same statistics as w_n because the DZT is an orthonormal transform. The spread in the delay-Doppler For more details on the derivations and examples of OTFS, we refer the reader to [12]. To illustrate the delay-Doppler spread, we show in Fig. 2 the spread of a symbol at position $n = 0$ and $k = 17$ for a grid of size $L = 27$ and $K = 36$ and for the channel discussed in Sec. IV. We note that the interference pattern is approximately the same for all symbols in the delay-Doppler domain. The received symbols are then passed on to a turbo equalizer to provide an estimate of the transmitted bits, denoted by $\hat{\mathbf{b}}$.

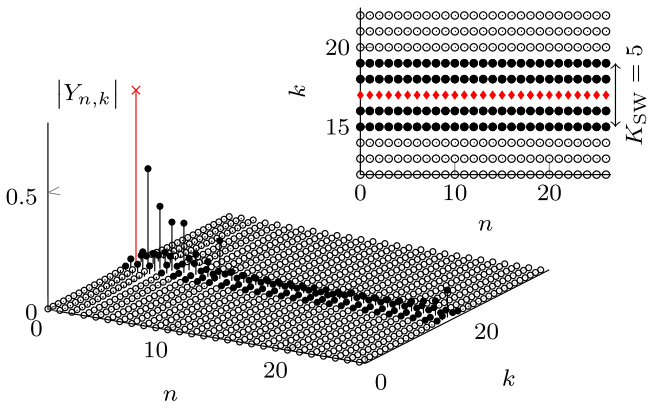

Fig. 2. Example of the spread of a symbol in the delay-Doppler domain (linear scale). The effective spread in the delay-Doppler domain is contained to a narrow stripe. The inlay shows the received symbols considered in the sliding window (filled markers). The sliding window can be used to estimate the symbols in the center of the sliding window (diamond shape).

The two-dimensional convolution of the delay-Doppler spreads in (6) can equivalently be expressed as a matrix-vector product

$$\mathbf{y} = \mathbf{H}\,\mathbf{x} + \mathbf{w}, \tag{9}$$

where $\mathbf{y}$, $\mathbf{x}$, and $\mathbf{w}$ are the received symbols, transmitted symbols, and noise samples in the delay-Doppler domain represented as vectors. The matrix $\mathbf{H}$ models the ISI in the delay-Doppler domain. Each element of $\mathbf{x}$ corresponds to an element of the fundamental rectangle of X. A common choice of arranging the elements of X into a vector is by stacking the columns of the fundamental rectangle X into a vector, i.e., the symbol at $X_{n,k}$ is the $(k+nK)$th element of $\mathbf{x}$. In that particular case, the elements of the channel matrix are given by

$$\mathbf{H}_{i,j} = \sum_{p=0}^{P-1} \alpha_p U^p_{\lfloor j \rfloor_K,(j)_K} \gamma(i,j,K) H^p_{(\lfloor i \rfloor_K - \lfloor j \rfloor_K)_L,(i)_K}, \tag{10}$$

where $\gamma(i,j,K) = e^{-\imath 2\pi \frac{(i)_K}{K} \mathbb{1}(\lfloor i \rfloor_K < \lfloor j \rfloor_K)}$, $(\cdot)_L$ is the modulo-L operation, $\lfloor i \rfloor_K$ is the largest integer smaller than or equal to i/K, and $\mathbb{1}(\cdot)$ is the indicator function.

III. Sliding Window LMMSE-Based Turbo Equalizer

A. LMMSE Equalizer

For the linear model in (9), the LMMSE estimate for x_n is given as [13, Th. 12.1 and eq. (15.64)]

$$\hat{x}_n = \mathrm{E}(x_n) + \mathrm{Var}(x_n)\mathbf{f}_n^{\mathrm{H}}(\mathbf{y} - \mathbf{H}\,\mathrm{E}(\mathbf{x})), \tag{11}$$

where

$$\mathrm{E}(x_n) = \sum_{i=1}^{M} a_i \Pr(x_n = a_i), \tag{12}$$

$$\mathrm{Var}(x_n) = \sum_{i=1}^{M} |a_i|^2 \Pr(x_n = a_i) - |\,\mathrm{E}(x_n)|^2, \tag{13}$$

and

$$\mathbf{f}_n = \mathrm{Cov}(\mathbf{y},\mathbf{y})^{-1}\mathbf{h}_n, \tag{14}$$

with $\mathbf{h}_n$ being the nth column of $\mathbf{H}$ and covariance matrix[1] $\mathrm{Cov}(\mathbf{y},\mathbf{y}) = \sigma^2\mathbf{I}_N + \mathbf{H}\,\mathrm{Cov}(\mathbf{x},\mathbf{x})\mathbf{H}^{\mathrm{H}}$ with $\mathbf{I}_N$ being the identity matrix of size N. We assume that the symbols x_n are independent. Thus, the covariance matrix $\mathrm{Cov}(\mathbf{x},\mathbf{x})$ is a diagonal matrix.

As discussed in [11, p. 677], the computational complexity of the LMMSE estimate in (11) is dominated by the inversion of the covariance matrix $\mathrm{Cov}(\mathbf{y},\mathbf{y})$ in (14). This letter considers one operation to consist of one complex multiplication and addition. Matrix inversion using the Gauss-Jordan algorithm requires N^3 such operations.

B. Sliding Window LMMSE Equalizer

From Fig. 2, it can be observed effective spread in the Doppler domain is concentrated in a narrow stripe. Based on this observation and the fact that all symbols exhibit the same interference pattern, we propose estimating the transmitted symbols from a subset of the received symbols defined by a window. The window size is chosen to cover the entire delay domain while covering only a narrow stripe of K_{SW} rows in the Doppler domain. For symmetry reasons, we choose K_{SW} to be odd. In the inlay of Fig. 2, received symbols within the window with width $K_{\mathrm{SW}} = 5$ are indicated by the filled markers.

We note that the window can be used to estimate all symbols with an identical Doppler index k, i.e., for all elements of $\mathbf{x}$ whose indices are given by $n = k+n'K$ with $0 \le n' \le L-1$. Thus, we denote the received symbols within the window by $\mathbf{y}_k$. An example for the received symbols within the sliding window for $k = 17$ and $K_{\mathrm{SW}} = 5$ is provided in the inlay of Fig. 2. The vector $\mathbf{y}_k$ contains the N_{SW} elements of $\mathbf{y}$ whose index are from the set $\{n = k'+n'K : 0 \le n' \le L-1, k - \lfloor K_{\mathrm{SW}}/2 \rfloor \le k' \le k + \lfloor K_{\mathrm{SW}}/2 \rfloor\}$. From (9), we have the following linear relation between the transmitted and received symbols of the sliding window

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x} + \mathbf{w}_k. \tag{15}$$

The matrix $\mathbf{H}_k$ is given by the N_{SW} rows of $\mathbf{H}$ that correspond to the elements of $\mathbf{y}_k$, and $\mathbf{w}_k$ are the corresponding noise samples. The estimate $\hat{x}_n$ based on the window is obtained by replacing $\mathbf{y}$ and $\mathbf{H}$ in (11) with $\mathbf{y}_k$ and $\mathbf{H}_k$. This reduces the computational complexity of the matrix inversion by a factor of $(N/N_{\mathrm{SW}})^3$.

Because the sliding window can be reused for all L symbols with the same Doppler index, we reduce the computational complexity for the matrix inversion by

$$\frac{N^3}{KN_{\mathrm{SW}}^3} = \frac{K^3L^3}{K(K_{\mathrm{SW}}^3L^3)} = \frac{K^2}{K_{\mathrm{SW}}^3}. \tag{16}$$

Using the sliding window, we now have K covariance matrices $\mathrm{Cov}(\mathbf{y}_k,\mathbf{y}_k)$ rather than a single covariance matrix $\mathrm{Cov}(\mathbf{y},\mathbf{y})$, which can be inverted independently, providing an additional degree of parallelization.

[1]The covariance matrix between two random vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as $\mathrm{Cov}(\mathbf{x},\mathbf{y}) = \mathrm{E}((\mathbf{x} - \mathrm{E}(\mathbf{x}))(\mathbf{y} - \mathrm{E}(\mathbf{y}))^{\mathrm{H}})$.

C. SISO LMMSE Equalizer

In this part, we briefly discuss the SISO LMMSE and refer to [11] for more details and derivations of the presented equations. The bitwise a-priori information for the equalizer is provided, e.g., by a channel decoder, in the form of log-likelihood ratios (LLR), defined as

$$L(c_{n,i}) = \ln \frac{\Pr(c_{n,i}=0)}{\Pr(c_{n,i}=1)}. \tag{17}$$

The LMMSE equalizer (11) considers a-priori information via the mean and variance of the symbol x_n, given in (12) and (13), and which can be found using the symbol probability derived from the LLRs as [11, p. 676]

$$\Pr(x_n = a_j) = \prod_{i=1}^{Q} \frac{1}{2}(1 + (1-2a_{j,i})\tanh(L(c_{n,i})/2), \tag{18}$$

where $[a_{j,1}, a_{j,2}, \ldots, a_{j,Q}]^T$ is the bit pattern associated with the symbol a_j.

A SISO LMMSE equalizer, in turn, computes extrinsic LLRs, i.e., information about the bit $c_{n,i}$ that is contained in $\mathbf{y}$ via $\hat{x}_n$, and that excludes the a-priori LLR $L(c_{n,i})$, which later become a priori LLRs for the decoder. The extrinsic LLR is defined as [11, eq. (2)]

$$L_{\mathrm{e}}(c_{n,i}) = \ln \frac{\sum_{\forall:\mathbf{c}_n:c_{n,i}=0} p(\hat{x}_n|\mathbf{c}_n) \prod_{\forall i':i'\neq i} \Pr(c_{n,i'})}{\sum_{\forall:\mathbf{c}_n:c_{n,i}=1} p(\hat{x}_n|\mathbf{c}_n) \prod_{\forall i':i'\neq i} \Pr(c_{n,i'})}, \tag{19}$$

The estimate $\hat{x}_n$, however, depends on the mean and variance of x_n, and thus, on $\Pr(x_n = a_j)$ and consequently on $L(c_{n,i})$, see (18). The dependency of $\hat{x}_n$ on $L(c_{n,i})$ can be removed by considering $\mathrm{E}(x_n) = 0$ and $\mathrm{Var}(x_n) = 1$ for all n, see [11, Sec. III A]. Thus, the mean and variance of x_n are zero and one, which is applied in (11). Alternatively, the estimate that does not depend on $L(c_{n,i})$ can equivalently be obtained through [11, eq. (6)]

$$\hat{x}_n = K_n \mathbf{f}_n^{\mathrm{H}}(\mathbf{y}_k - \mathbf{H}_n \,\mathrm{E}(\mathbf{x}) + \mathrm{E}(x_n)\mathbf{h}_n) \tag{20}$$

where $K_n = (1 + (1 - \mathrm{Var}(x_n))\mathbf{f}_n^{\mathrm{H}}\mathbf{h}_n)^{-1}$.

The exact computation of the distribution of $p(\hat{x}_n|\mathbf{c}_n = \mathbf{a}_j) = p(\hat{x}_n|x_n = a_j)$ comes with a high computational burden. However, $p(\hat{x}_n|x_n = a_j)$ is well approximated by a Gaussian distribution [11, eq. (7)]

$$p(\hat{x}_n|x_n = a_j) = \frac{1}{\pi\sigma_{n,j}^2} \exp\left(-\frac{|\hat{x}_n - \mu_{n,j}|^2}{\sigma_{n,j}^2}\right). \tag{21}$$

The mean $\mu_{n,j}$ is obtained by computing the conditional mean $\mathrm{E}(\hat{x}_n^{\mathrm{e}}|x_n = a_j)$ which is

$$\mu_{n,j} = K_n a_j \mathbf{f}_n^{\mathrm{H}}\mathbf{h}_n. \tag{22}$$

Likewise, the variance $\sigma_{n,j}^2$ is the conditional variance $\mathrm{Var}(\hat{x}_n^{\mathrm{e}}|x_n = a_j)$ given as

$$\sigma_{n,j}^2 = K_n^2(\mathbf{f}_n\mathbf{h}_n - \mathrm{Var}(x_n)\mathbf{f}_n^{\mathrm{H}}\mathbf{h}_n\mathbf{h}_n^{\mathrm{H}}\mathbf{f}_n). \tag{23}$$

For quadrature phase-shift keying, applied in Sec. IV, with mapping $[0,0] \to (1+\imath)/\sqrt{2}$, $[0,1] \to (1-\imath)/\sqrt{2}$, $[1,0] \to (-1+\imath)/\sqrt{2}$, and $[1,1] \to (-1-\imath)/\sqrt{2}$, the extrinsic a-posterior LLR in (19) becomes [11, Table III]

$$L_{\mathrm{e}}(c_{n,1}) = K' \cdot \Re(\mathbf{f}_n^{\mathrm{H}}(\mathbf{y}_k - \mathrm{E}(\mathbf{y}_k) + E(x_n)\mathbf{h}_n)$$
$$L_{\mathrm{e}}(c_{n,2}) = K' \cdot \Im(\mathbf{f}_n^{\mathrm{H}}(\mathbf{y}_k - \mathrm{E}(\mathbf{y}_k) + E(x_n)\mathbf{h}_n)$$

where $K' = \sqrt{8}/(1 - \mathrm{Cov}(x_n, x_n)\mathbf{h}_n\mathbf{f}_n)$. $\Re(\cdot)$ and $\Im(\cdot)$ denote the real and imaginary parts, respectively.

In a turbo equalizer, the extrinsic LLR is passed to a decoder. The decoder then provides the extrinsic LLR $L_{\mathrm{e}}^{\mathrm{D}}(c_{n,i})$ after decoding, which is the a-priori information for the equalizer in the next iteration step.

IV. Numerical Results

We consider an OTFS modulation with $L = 27$, $K = 36$, and QPSK modulation. We choose $p(t) = \sin(\pi t/T)/(\sqrt{T}(\pi t/T)$ and symbol period $T = 100\,\mathrm{ns}$. The corresponding bandwidth is $10\,\mathrm{MHz}$. We consider an LDPC code with a code rate of 1/2 [14]. We consider the Extended Vehicular A (EVA) channel model. The channel consists of nine multipath components. The individual delay and relative powers pairs (τ_p, α_p) (in ns and dB) of the multipath components are in ascending order with respect to p: (0,0), (30,-1.5), (150,-1.4), (310,-3.6),(370,-0.6), (710,-9.1), (1090,-7), (1730,-12), (2510,16.9). See [15, Annex B2.1] for details. We assigned a single Doppler shift to each multipath component, which is modeled as $\nu_p = \nu_{\max}\cos(\psi_p)$, where ψ_p are independent random variables that are uniformly distributed in the interval $[0, 2\pi)$. The maximum Doppler shift is $\nu_{\max} = 6.222\,\mathrm{kHz}$, which corresponds to a velocity of $240\,\mathrm{km\,h^{-1}}$ at a carrier frequency of $28\,\mathrm{GHz}$. Thus, the maximum shift in the Doppler domain of $k_p \leq \nu_{\max} NT = 0.6048$.

The number of symbols in the delay domain L is chosen according to two criteria. First, the size of the delay dimension TL should be larger than $\tau_{\max}$ to avoid fading in the delay-Doppler domain [1, Sec. IV]. Second, we aim to minimize the computational complexity by choosing L as small as possible. The smallest L that satisfies both criteria is $L = 27$. We choose sliding window sizes of three, five, and seven, which achieve complexity reductions of 48, 10.36, and 3.77, respectively.

We create in total 100 channel realizations and compute the bit error rate (BER) for the different sliding window sizes for each channel realization. The BER is presented as a function of the signal-to-noise ratio (SNR) per bit, defined as [11, Sec. V]

$$\frac{E_b}{N_0} \triangleq \frac{E_s}{N_0 RQ} = \frac{\mathrm{E}(|y_n|^2)}{N_0 RQ} = \frac{N^{-1}\sum_{i=1}^{N} G_{i,i}}{\sigma^2 RQ} \tag{24}$$

where $G_{i,i}$ are the main diagonal elements of $\mathbf{G} = \mathbf{H}^{\mathrm{H}}\mathbf{H}$ and from which we derive the noise variance. In our simulation, we normalize the channel such that the numerator in (24) is one. In the equalization process, we assume perfect channel knowledge, which is acquired in a separate OTFS frame, and set the elements of $\mathbf{H}$ for which the absolute value is smaller than 0.01 to zero. The resulting average BER curves are shown in Fig. 3. Additionally, we plot the BER for the unconstrained LMMSE equalizer and the additive white

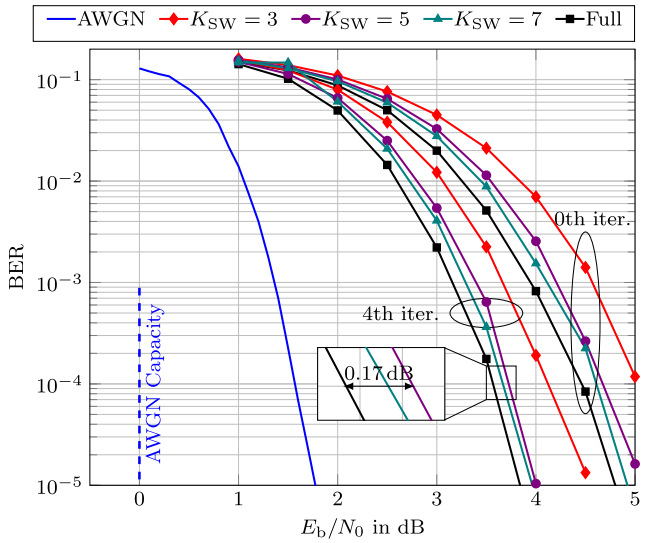

Fig. 3. BER curves for the sliding window for different window sizes after the 0th and 4th iteration.

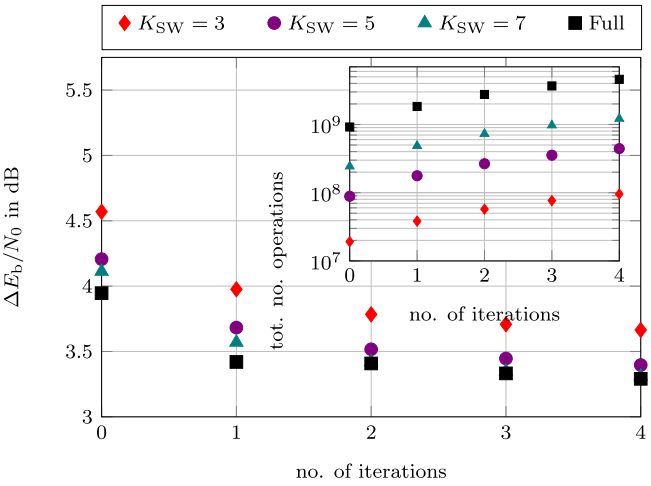

Fig. 4. E_{b}/N_0 gap to AWGN capacity (0 dB) at a BER of 10^{-3} at various iteration steps. **Inset:** The total number of operation operations for matrix inversion as a function of the number of iterations.

Gaussian noise (AWGN) channel ($\mathbf{H} = \mathbf{I}_N$). The figure shows that a window size of $K_{\mathrm{SW}} = 5$ results in an SNR loss of about $0.17\,\mathrm{dB}$ at a BER of 10^{-4} after four iterations. Increasing the window size K_{SW} above five only marginally improves the performance.

In Fig. 4, we plot the SNR gap $\Delta E_{\mathrm{b}}/N_0$ at a BER of 10^{-3}, i.e., the difference between the required E_{b}/N_0 and the AWGN capacity, which is the lower bound on E_{b}/N_0 required for reliable communication at rate $1/2$ and QPSK, for the different turbo equalizers. It can be seen that after the second iteration, all investigated turbo equalizers only improved marginally for each subsequent iteration. The inset in Fig. 4 illustrates the total number of operations required for the matrix inversions as a function of the iterations. We can see that window size $K_{\mathrm{SW}} = 5$ after four iterations has approximately the same complexity as $K_{\mathrm{SW}} = 7$ after zero iterations, but the former is about $0.17\,\mathrm{dB}$ better than the latter, and thus, $K_{\mathrm{SW}} = 5$ and four iterations would be our recommendation.

V. Conclusion

In this letter, we proposed a simple approach to reduce the complexity of an LMMSE-based turbo equalizer for OTFS. For this purpose, we introduced a sliding window that effectively reduces the number of symbols in the equalization process. We showed that a substantial complexity reduction can be achieved by choosing the sliding window to cover the essential spread of a symbol in the delay-Doppler domain.

In our approach, we focused on reducing the complexity in the first place, i.e., by reducing the size of the required matrix inversion. Further complexity reduction could be achieved by addressing the complexity of the equalizer itself or exploiting certain matrix features, as discussed in [11].

References

[1] S. K. Mohammed, R. Hadani, A. Chockalingam, and R. Calderbank, "OTFS—Predictability in the delay-Doppler domain and its value to communication and radar sensing," 2023, *arXiv:2302.08705*.

[2] Z. Q. Zhang, H. Liu, Q. L. Wang, and P. Fan, "A survey on low complexity detectors for OTFS systems," *ZTE Commun.*, vol. 19, no. 4, pp. 3–15, Dec. 2021.

[3] P. Raviteja, K. T. Phan, Y. Hong, and E. Viterbo, "Interference cancellation and iterative detection for orthogonal time frequency space modulation," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6501–6515, Oct. 2018.

[4] L. Gaudio, M. Kobayashi, G. Caire, and G. Colavolpe, "On the effectiveness of OTFS for joint radar parameter estimation and communication," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 5951–5965, Sep. 2020.

[5] G. Colavolpe and G. Germi, "On the application of factor graphs and the sum-product algorithm to ISI channels," *IEEE Trans. Commun.*, vol. 53, no. 5, pp. 818–825, May 2005.

[6] S. Tiwari, S. S. Das, and V. Rangamgari, "Low complexity LMMSE receiver for OTFS," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2205–2209, Dec. 2019.

[7] S. Li, W. Yuan, Z. Wei, and J. Yuan, "Cross domain iterative detection for orthogonal time frequency space modulation," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2227–2242, Apr. 2022.

[8] M. Liu, S. Li, B. Bai, and G. Caire, "Reduced-complexity cross-domain iterative detection for OTFS modulation via delay-Doppler decoupling," 2023, *arXiv:2307.00926*.

[9] P. S. Sanoopkumar, S. McWade, and A. Farhang, "Truncated turbo equalizer with SIC for OTFS," 2023, *arXiv:2305.14966*.

[10] H. Li and Q. Yu, "Doubly-iterative sparsified MMSE turbo equalization for OTFS modulation," *IEEE Trans. Commun.*, vol. 71, no. 3, pp. 1336–1351, Mar. 2023.

[11] M. Tuchler, A. C. 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.

[12] F. Lampel, H. Joudeh, A. Alvarado, and F. M. J. Willems, "Orthogonal time frequency space modulation based on the discrete zak transform," *Entropy*, vol. 24, no. 12, p. 1704, Nov. 2022.

[13] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, 1st ed. London, U.K.: Pearson, 1993.

[14] *IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks–Specific Requirements—Part 11: Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2020, Feb. 2021.

[15] *Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) Radio Transmission and Reception*, document TS 36.101, Version 14.3.0, Release 14, 3GPP, Apr. 2017.