

A Spectral Approach to Inter-Carrier Interference Mitigation in OFDM Systems

Avi Septimus, Yosi Keller, and Itsik Bergel

Abstract—In this paper, we propose a new method for inter-carrier interference (ICI) mitigation in orthogonal frequency-division multiplexing (OFDM) systems. The proposed approach views the signal reconstruction problem at the receiver end as an integer least squares (ILS) problem, and uses a recently developed spectral approach called sequential probabilistic ILS (SPILS) to solve it. The proposed approach outperforms other state-of-the-art approaches while having the same computational complexity. In addition, we present a novel extension to the SPILS scheme that allows the generation of soft decisions (for communication systems which use soft-decision decoding). The use of soft-decision decoding (naturally) brings significant improvement in the detection reliability, and we show that the proposed method again outperforms other state-of-the-art approaches. To better address the tradeoff between performance and complexity, we first suggest a novel method to reduce the number of matrix inversions required and hence, to reduce the implementation complexity without any degradation in performance. We also introduce a novel low complexity scheme termed Quick SPILS (QSPILS) in which we lose a little in detection reliability, but significantly reduce the implementation complexity.

Index Terms—Digital communications, communication systems, decoding, fading channels.

I. INTRODUCTION

ORTHOGONAL FREQUENCY-DIVISION MULTIPLEXING (OFDM) has become popular in wideband digital communication systems. OFDM systems typically work over frequency-selective channels where the use of a cyclic prefix (CP) of adequate length maintains the orthogonality between the subcarriers and completely eliminates the inter-symbol interference (ISI). On the other hand, any change in the channel within the duration of an OFDM symbol will cause inter-carrier interference (ICI). Fast moving terminals are particularly prone to large ICI, as they suffer from a strong Doppler spread. The desire to support communications to fast moving terminals has led to an increasing interest in methods which mitigate ICI.

Various approaches to mitigating ICI are proposed in the literature, and a good review of these is given by Cai and Giannakis [1]. In particular, many approaches have tackled the problem in the frequency domain. These approaches include

the matched-filter, the zero-forcing (ZF) technique, and the linear minimum mean-squared error (MMSE) estimator. The ZF approach was simplified by taking advantage of the banded nature of the channel matrix in the frequency domain [2]. A significant improvement to the MMSE approach was the addition of successive detection (MMSE-SIC) [3]. An improvement to this method was then proposed where low-complexity recursive algorithms for determining the estimator coefficients for the MMSE-SIC estimator were derived [1]. Additionally, several works took advantage of the banded nature of the channel matrix in the frequency domain to reduce the complexity of the MMSE estimator [4]–[6].

This banded nature can be seen as an extension of the time-invariant case, where the frequency domain channel matrix is diagonal. The assumption of banded channel matrix is reasonable in cases of low Doppler spread.

However, at high speeds, the large Doppler spread causes ICI between all of the subcarriers, and there is a need for an efficient method that does not depend on the banded assumption.¹

In this paper, we propose a new method for ICI mitigation. The proposed approach views the signal reconstruction problem at the receiver end as an integer least squares (ILS) problem, and uses a recently developed spectral approach called sequential probabilistic ILS (SPILS) to solve it. We also extend SPILS to provide soft decisions for communication systems which use soft-decision coding. The proposed approach outperforms other state-of-the-art approaches while having the same computational complexity. We also introduce Quick SPILS (QSPILS) which modifies SPILS so as to give us a trade-off between complexity and accuracy.

The rest of this paper is organized as follows. In Section II, we present the OFDM system model. In Section III, we review the SPILS framework [9], which we then use to solve the OFDM decoding problem in Section IV. In Section V, we first propose a change to the SPILS algorithm which lowers its computational complexity. We then offer a further modification which allows for a tradeoff between accuracy and complexity. A detailed description of the computational complexity of the algorithms in question is presented as well. Section VI presents numerical results that demonstrate how SPILS outperforms other state-of-the-art methods. A summary and concluding remarks are given in Section VII.

Manuscript received July 22, 2013; revised December 9, 2013 and February 25, 2014; accepted April 22, 2014. Date of publication May 1, 2014; date of current version August 20, 2014. The associate editor coordinating the review of this paper and approving it for publication was H. Steendam.

The authors are with the Faculty of Engineering, Bar Ilan University, 5290002 Ramat-Gan, Israel (e-mail: avi.septimus@gmail.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCOMM.2014.2321379

¹In some cases, and in particular when the number of multipath components is small, the average Doppler shift can be reduced by resampling the received data at a corrected rate [7], [8]. In such cases, the method in this work will only need to handle the residual Doppler spread.

II. SIGNAL MODEL

A single OFDM symbol is constructed of the $N + L$ samples

$$X_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{j2\pi kn/N}, \quad -L \leq n \leq N-1 \quad (1)$$

where N is the IFFT length, and $x_{k=0\dots N-1}$ are the data symbols. The samples X_{-1}, \dots, X_{-L} are termed the cyclic prefix (CP), which satisfy

$$X_{-i} = X_{N-i} \quad 1 \leq i \leq L. \quad (2)$$

The OFDM symbol is transmitted through a frequency selective channel, and the received signal is:

$$Y_n = \sum_{\ell=0}^{L-1} h_{n,\ell} X_{n-\ell} + W_n \quad (3)$$

where $h_{n,\ell}$ is the channel gain at a delay of ℓ samples at time n and W_n is additive white Gaussian noise (AWGN) with variance σ_w^2 . For ease of representation, we represent the input and output signals by the $N \times 1$ vectors $\mathbf{X} = [X_0, \dots, X_{N-1}]^T$ and $\mathbf{Y} = [Y_0, \dots, Y_{N-1}]^T$ respectively, and we have the relation:

$$\mathbf{Y} = \tilde{\mathbf{h}}\mathbf{X} + \mathbf{W} \quad (4)$$

where the n, ℓ element of the matrix $\tilde{\mathbf{h}}$ is defined by:

$$\tilde{h}_{n,\ell} = \begin{cases} h_{n,n-\ell} & n-L+1 \leq \ell \leq n \\ h_{n,N+n-\ell} & N+n-L+1 \leq \ell \leq N-1. \end{cases} \quad (5)$$

Note that the upper right-hand section of the matrix $\tilde{\mathbf{h}}$ takes into account the CP.

To detect the transmitted symbols, the receiver performs an FFT on the received vector. Defining $\mathbf{y} = \mathbf{F}\mathbf{Y}$, $\mathbf{x} = \mathbf{F}\mathbf{X}$ and $\mathbf{w} = \mathbf{F}\mathbf{W}$, where \mathbf{F} is the $N \times N$ FFT matrix with $\mathbf{f}_{n,m} \triangleq (1/\sqrt{N})[\exp(-j2\pi mn/N)]_{n,m=0,\dots,N-1}$, we can write:

$$\mathbf{y} = \mathbf{F}\tilde{\mathbf{h}}\mathbf{X} + \mathbf{F}\mathbf{W} = \mathbf{H}\mathbf{x} + \mathbf{w} \quad (6)$$

where $\mathbf{H} = \mathbf{F}\tilde{\mathbf{h}}\mathbf{F}^H$ is termed the frequency domain channel matrix and \mathbf{w} is also complex Gaussian with zero mean and a covariance matrix $\sigma_w^2 \mathbf{I}$. Due to the CP, if the channel is time invariant, then the time domain channel (5), is a circulant matrix. Therefore, the frequency domain channel matrix, \mathbf{H} is diagonal, implying that no ICI is experienced. The transmitted signal can thus be reconstructed by simply multiplying each value in \mathbf{y} by the inverse of its corresponding diagonal value in \mathbf{H} , explaining the motivation for the CP. When the channel is time-variant, however, the matrix $\tilde{\mathbf{h}}$ is no longer circulant, because channel impulse response coefficients appearing in one row are not necessarily equal to the coefficients in another row. This implies that \mathbf{H} is no longer diagonal, and its off-diagonal entries correspond to ICI experienced in the system. The amount of ICI experienced is directly related to the velocity of either/both the transmitter and receiver, as movement in the system causes Doppler spread. For a low Doppler spread, the system experiences low ICI, and the frequency-domain channel matrix \mathbf{H} is approximately banded. As the Doppler spread

increases, due to increasing velocity in the system, the banded assumption no longer holds.

III. SEQUENTIAL PROBABILISTIC INTEGER LEAST SQUARES

Integer Least Squares (ILS) is a well studied mathematical problem, with applications to a wide variety of mathematical and engineering problems. ILS refers to the case where the signal's values come from a set of integer values. This problem is known to be NP-hard [10], and it is therefore of great value to research sub-optimal algorithms to solve it. Keller [9] developed a probabilistic method for solving the ILS problem in the multiple-input-multiple-output (MIMO) channel scenario. In the following subsection, we briefly describe the PILS method and its extension termed SPILS. In the next section we discuss the application of PILS and SPILS to OFDM, including some extensions and low complexity methods.

A. Probabilistic Integer Least Squares

Consider the received vector:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \quad (7)$$

where $\mathbf{x} = [x_0, \dots, x_{N-1}]^T$, $\mathbf{H} \in \mathbb{C}^{N \times N}$, and $\mathbf{w} = [w_0, \dots, w_{N-1}]^T$ is white additive Gaussian noise with variance σ_w^2 . The variables x_k are assumed to be drawn from an alphabet $\mathcal{A} = \{a_1, \dots, a_M\}$ which is a set of M integer values. \mathbf{H} and \mathbf{y} are assumed to be known, and we aim to recover \mathbf{x} . The maximum likelihood (ML) solution to (7) is given as the LS solution which is known to be NP-hard. Following Goldberger and Leshem [11] and Keller [9], the distribution of \mathbf{x} is approximated by its pairwise posteriors $P(x_k, x_r | \mathbf{y})$. We then use the pairwise posteriors estimate to compute the marginal posteriors $P(x_k | \mathbf{y})$, and solve the ILS problem by choosing the value $a_m \in \mathcal{A}$ with the highest posterior for each $x_k \in \mathbf{x}$.

1) *Estimating the Pairwise Posteriors:* Let $\mathbf{h}_0, \dots, \mathbf{h}_{N-1}$ be the columns of \mathbf{H} and let $\mathbf{B}_{k,r}$, $0 \leq k < r \leq N-1, k \neq r$ be the matrix obtained from \mathbf{H} by removing the k and r columns. The projection matrix of:

$$\mathbf{P}_{k,r} = \mathbf{I} - \mathbf{B}_{k,r} (\mathbf{B}_{k,r}^T \mathbf{B}_{k,r})^{-1} \mathbf{B}_{k,r}^T \quad (8)$$

is the orthogonal projection of space spanned by $\{\mathbf{h}_i\}$ onto the complement of the subspace spanned by $\{\mathbf{h}_i | i \neq k, r\}$. Hence:

$$\mathbf{P}_{k,r} \mathbf{H}\mathbf{x} = \mathbf{P}_{k,r} \mathbf{h}_k x_k + \mathbf{P}_{k,r} \mathbf{h}_r x_r. \quad (9)$$

Applying $\mathbf{P}_{k,r}$ to both sides of (7), we obtain:

$$\mathbf{P}_{k,r} \mathbf{y} = \mathbf{P}_{k,r} \mathbf{h}_k x_k + \mathbf{P}_{k,r} \mathbf{h}_r x_r + \tilde{\mathbf{w}}_{k,r} \quad (10)$$

where:

$$\tilde{\mathbf{w}}_{k,r} \triangleq \mathbf{P}_{k,r} \mathbf{w} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{P}_{k,r} \mathbf{P}_{k,r}^T) = \mathcal{N}(0, \sigma_w^2 \mathbf{P}_{k,r}). \quad (11)$$

Finding the true pairwise posterior probabilities $P(x_k, x_r | \mathbf{y})$ is an NP-hard problem as well. Instead, we compute an estimate

of the pairwise posteriors from the relevant projection of the received signal, $\mathbf{P}_{k,r}\mathbf{y}$. Using (10) and (11) we have:

$$P(\mathbf{y}|x_k, x_r) \approx P(\mathbf{P}_{k,r}\mathbf{y}|x_k, x_r) \\ = c e^{-\frac{1}{2\sigma_w^2} \|\mathbf{P}_{k,r}\mathbf{y} - \mathbf{P}_{k,r}\mathbf{h}_k x_k - \mathbf{P}_{k,r}\mathbf{h}_r x_r\|^2} \quad (12)$$

where c is a constant. By Bayes' formula, the pairwise posterior with respect to the alphabet values (a_m, a_ℓ) is then:

$$P(x_k = a_m, x_r = a_\ell|\mathbf{y}) \\ = \frac{P(\mathbf{y}|x_k = a_m, x_r = a_\ell)P(x_k = a_m, x_r = a_\ell)}{P(\mathbf{y})}. \quad (13)$$

Assuming that the prior probabilities are statistically independent, (13) becomes:

$$P(x_k = a_m, x_r = a_\ell|\mathbf{y}) \\ = \frac{P(\mathbf{y}|x_k = a_m, x_r = a_\ell)P(x_k = a_m)P(x_r = a_\ell)}{P(\mathbf{y})}. \quad (14)$$

Note that (14) incorporates the prior information $P(x_k = a_m)\forall a_m \in \mathcal{A}$.

2) *From Pairwise Posteriors to Marginal Posteriors:* We aim to compute the marginal posteriors:

$$P(x_k = a_m|\mathbf{y}) \quad \forall k = 0 \dots N-1, a_m \in \mathcal{A}$$

given the pairwise posteriors:

$$P(x_k = a_m, x_r = a_\ell|\mathbf{y})$$

computed in the previous section. For that we consider the set of pairwise posteriors as a two-dimensional Markov-Random-Field (2D-MRF) and aim to infer its unary probabilities. Utilizing a spectral approach to the 2D-MRF inference problem, we define $\mathbf{p} \in [0, 1]^{NM}$ to be the vector of marginal posteriors such that:

$$\mathbf{p} = [\mathbf{p}_0^T \dots \mathbf{p}_{N-1}^T]^T \quad (15)$$

where:

$$\mathbf{p}_k = [P(x_k = a_1|\mathbf{y}) \dots P(x_k = a_M|\mathbf{y})]^T. \quad (16)$$

Let $\mathbf{A} \in [0, 1]^{NM \times NM}$ be the matrix obtained via:

$$\mathbf{A} = \mathbf{p}\mathbf{p}^T. \quad (17)$$

Our goal will be to first calculate the matrix \mathbf{A} and then use it to determine the unary probabilities in \mathbf{p} . \mathbf{A} consists of blocks $\mathbf{C}_{k,r}$ such that:

$$\mathbf{A} = \begin{bmatrix} \mathbf{C}_{0,0} & \dots & \mathbf{C}_{0,N-1} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{N-1,0} & \dots & \mathbf{C}_{N-1,N-1} \end{bmatrix} \quad (18)$$

where $\mathbf{C}_{k,r}\forall k, r$ are defined as:

$$\mathbf{C}_{k,r} = \begin{bmatrix} \mathbf{p}_{k,1}\mathbf{p}_{r,1} & \dots & \mathbf{p}_{k,1}\mathbf{p}_{r,M} \\ \vdots & \ddots & \vdots \\ \mathbf{p}_{k,M}\mathbf{p}_{r,1} & \dots & \mathbf{p}_{k,M}\mathbf{p}_{r,M} \end{bmatrix} \quad (19)$$

and $\mathbf{p}_{k,m}$ refers to the m -th element of the vector \mathbf{p}_k . To fill in the entries of \mathbf{A} , the diagonal blocks $\mathbf{C}_{k,k}$ will require separate treatment from the off-diagonal blocks $\mathbf{C}_{k,r}\forall k \neq r$. Off-diagonal blocks each involve two different sub-symbols. To approximate the product on these off-diagonal blocks, we first use the chain rule to get:

$$P(x_k = a_m, x_r = a_\ell|\mathbf{y}) \\ = P(x_k = a_m|\mathbf{y})P(x_r = a_\ell|x_k = a_m, \mathbf{y}). \quad (20)$$

Obviously, the channel matrix induces a statistical dependence between x_k and x_r given \mathbf{y} . Yet, to derive a simple result, we choose to assume that the different sub-symbols are independent for all $k \neq r$, so we can use the approximation:

$$P(x_k = a_m, x_r = a_\ell|\mathbf{y}) \\ = P(x_k = a_m|\mathbf{y})P(x_r = a_\ell|\mathbf{y}) \quad \forall k \neq r. \quad (21)$$

Therefore, off-diagonal blocks can be calculated by using (14) and:

$$\mathbf{C}_{k,r}[m, \ell] = P(x_k = a_m, x_r = a_\ell|\mathbf{y}) \quad \forall k \neq r. \quad (22)$$

The diagonal blocks $\mathbf{C}_{k,k}$ each involve probabilities of only the k -th sub-symbol, and therefore we do not use the independence assumption of (21). Thus, we are left with the terms $P(x_k = a_m|\mathbf{y})P(x_k = a_\ell|\mathbf{y})$ which involve the very unary probabilities which we wish to solve for. Therefore, we should use some estimate of those unary probabilities for these terms. We choose to use the prior probabilities implying that:

$$\mathbf{C}_{k,k}[m, \ell] = P(x_k = a_m)P(x_k = a_\ell). \quad (23)$$

Hence, given the matrix \mathbf{A} containing the pairwise posteriors on the off-diagonal blocks and the prior products on the diagonal blocks, one can recover the marginals \mathbf{p} as the rank-one approximation of \mathbf{A} , computed via its eigendecomposition. Due to the inaccuracy of our approximations, the matrix \mathbf{A} will not be of rank one, and the leading eigenvector of \mathbf{A} is its optimal rank-one approximation in the Frobenius norm. The matrix \mathbf{A} is symmetric with non-negative elements. Hence, following the Perron-Frobenius theorem [12], \mathbf{A} is guaranteed to have a leading eigenvalue whose corresponding eigenvector is nonnegative. To compute the leading eigenvector of \mathbf{A} , the Power Iteration [13] is used. Although the leading eigenvector vector of \mathbf{A} is a good estimation for the probability vector \mathbf{p} , we must note that (in general) it is not an appropriate probability vector. In an appropriate probability vector as defined by (15) and (16), the sum of the probabilities for each sub-symbol must be one. Hence, we generate an appropriate estimation of the probability vector \mathbf{p} by normalizing the leading eigenvector of \mathbf{A} so that each set of M values in \mathbf{p}_k satisfies:

$$\sum_{j=1}^M \mathbf{p}_{k,j} = 1 \quad \forall k. \quad (24)$$

Once \mathbf{p} has been calculated, the vector $\hat{\mathbf{x}} = [\hat{x}_0 \cdots \hat{x}_{N-1}]^T$ can be decoded by choosing the alphabet symbol having the highest posterior marginal:

$$\hat{x}_k = \arg \max_{a_m \in \mathcal{A}} P(x_k = a_m | \mathbf{y}). \quad (25)$$

Keller [9] referred to this method for solving the ILS problem as Probabilistic ILS (PILS). The PILS algorithm is summarized in Algorithm 1.

Algorithm 1 Probabilistic ILS

- 1: Compute the matrix \mathbf{A} .
 - 2: Compute \mathbf{p} , the leading eigenvector of \mathbf{A}
 - 3: **for** $k = 0$ to $N - 1$ **do**
 - 4: $\hat{x}_k = \arg \max_{a_m \in \mathcal{A}} P(x_k = a_m | \mathbf{y})$
 - 5: **end for**
-

B. Sequential Probabilistic Integer Least Squares

Following the MMSE-SIC paradigm, Keller [9] extended the PILS scheme by sequentially decoding one variable at a time. The premise is to utilize the highest posterior marginal computed by the PILS algorithm and reduce the original problem of N variables to one of $N - 1$. To pick the best variable to decode in iteration t , we first select the index i at which the PILS algorithm has the highest posterior marginal:

$$i = \arg \max_j \{ \max \mathbf{p}_j \}. \quad (26)$$

If we let \hat{x}_i^t be the sub-symbol at index i with the highest decoding probability at iteration t , then we solve only for \hat{x}_i^t using (25). Setting $\mathbf{H}^0 = \mathbf{H}$, $\mathbf{y}^0 = \mathbf{y}$, and $\mathbf{x}^0 = \mathbf{x}$, and given \hat{x}_i^t we update the ILS problem:

$$\mathbf{H}^{t+1} = [\mathbf{h}_0 \cdots \mathbf{h}_{i-1} \mathbf{h}_{i+1} \cdots \mathbf{h}_{N-1}]^t \quad (27)$$

$$\mathbf{y}^{t+1} = \mathbf{y}^t - \hat{x}_i^t \mathbf{h}_i \quad (28)$$

$$\mathbf{x}^{t+1} = [\mathbf{x}_0 \cdots \mathbf{x}_{i-1} \mathbf{x}_{i+1} \cdots \mathbf{x}_{N-1}]^t \quad (29)$$

and solve the problem:

$$\mathbf{y}^{t+1} = \mathbf{H}^{t+1} \mathbf{x}^{t+1} + \mathbf{w}. \quad (30)$$

We iterate the PILS solution for $t = (1, 2, \dots, N - 1)$ and the scheme is summarized in Algorithm 2.

Algorithm 2 Sequential Probabilistic ILS (SPILS)

- 1: **for** $t = 1$ to $N - 1$ **do**
 - 2: Apply the PILS algorithm given in Algorithm 1
 - 3: Apply (26) and (25) to recover the symbol \hat{x}_i^t with the highest decoding probability
 - 4: Reformulate the ILS problem using (27)–(30) and \hat{x}_i^t
 - 5: **end for**
-

We note that although PILS and SPILS have been formulated for an ILS problem, they can cope with any problem of the form (7) where the variables belong to a finite alphabet of values.

IV. OFDM DETECTION USING SPILS

Comparing (6) and (7), the SPILS method is easily adapted to the decoding of OFDM signals. The main point that requires some attention is the adaptation to complex variables. This adaptation is given by:

$$\begin{bmatrix} \text{Re}(\mathbf{y}) \\ \text{Im}(\mathbf{y}) \end{bmatrix} = \begin{bmatrix} \text{Re}(\mathbf{H}) & -\text{Im}(\mathbf{H}) \\ \text{Im}(\mathbf{H}) & \text{Re}(\mathbf{H}) \end{bmatrix} \begin{bmatrix} \text{Re}(\mathbf{x}) \\ \text{Im}(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} \text{Re}(\mathbf{w}) \\ \text{Im}(\mathbf{w}) \end{bmatrix}. \quad (31)$$

This formulation can support only modulation in which the real part of the possible modulation values is independent from the imaginary part. Luckily, this independence holds for all Gray-mapped QAM constellations. Aside from this adaptation, the SPILS method can be directly applied to OFDM decoding, and decoding results are presented in Section VI.

A. Extension to Coded OFDM Systems

Many communications systems use coding schemes to improve performance. Many coding schemes can benefit from soft-decision inputs for decoding. Soft-decisions typically represent the log likelihood ratios (LLR's) for each bit. The LLR's sign represents the hard decision for the bit in question, and its absolute value represents the measure of confidence with which that bit has been decoded. In this section, we show how both PILS and SPILS can produce the required soft-decisions.

1) *Coded PILS*: The PILS algorithm provides us with a simple and intuitive way to calculate the LLR. In fact, the algorithm output \mathbf{p} (see (15), (16)) already contains an estimation of the probabilities $P(x_k = a_m | \mathbf{y})$. In most cases, $M = 2^B$ and B bits are mapped for each modulated symbol. Let the modulation mapping be given by $x_k = f(b_{k,1}, \dots, b_{k,B})$. The estimated probability for $b_{k,\ell} \in \{0, 1\}$ is:

$$P(b_{k,\ell}) = \sum_{\substack{b_1, \dots, b_{\ell-1}, \\ b_{\ell+1}, \dots, b_B}} P(x_k = f(b_1, \dots, b_{\ell-1}, b_{k,\ell}, b_{\ell+1}, \dots, b_B) | \mathbf{y}) \quad (32)$$

and the LLR of this bit can be estimated by:

$$L_{k,\ell} = \log \left(\frac{P(b_{k,\ell} = 0)}{P(b_{k,\ell} = 1)} \right). \quad (33)$$

As an example, the BPSK modulation is characterized by $B = 1$ and the modulation mapping:

$$f(b) = 1 - 2b. \quad (34)$$

For this modulation, the estimated LLR is given simply by:

$$L_{k,1} = \log \left(\frac{\mathbf{p}_{k,1}}{\mathbf{p}_{k,2}} \right) \quad (35)$$

where \mathbf{p}_k is the vector of length two defined in (16), with $\mathbf{p}_{k,1} = P(x_k = 1|\mathbf{y})$ and $\mathbf{p}_{k,2} = P(x_k = -1|\mathbf{y})$.

2) *Coded SPILS*: In the case of SPILS, for each iteration, we determine the LLR for the B bits associated with the variable with the highest confidence according to (26). Thus, if the SPILS algorithm chooses the sub-symbol index i as having the highest probability of successful decoding at iteration t , then we solve for the LLR of only the bits $b_{i,\ell}$, $\ell = 1 \dots B$ at iteration t as:

$$L_{i,\ell} = \log \left(\frac{P(b_{i,\ell} = 0)}{P(b_{i,\ell} = 1)} \right). \quad (36)$$

V. PILS AND SPILS WITH REDUCED COMPLEXITY

A. Efficient Implementation of PILS

In its basic form, and due to the symmetry of \mathbf{A} , the complexity of PILS is dominated by the $(N)(N-1)/2$ matrix inversions required as seen in (8). SPILS iterates the PILS algorithm $N-1$ times, where the number of matrix inversions decreases by one at each iteration, meaning that it requires:

$$\sum_{n=2}^N \frac{(n)(n-1)}{2} = \frac{N(N^2-1)}{6}$$

matrix inversions. Thus, PILS and SPILS have computational complexities of $\mathcal{O}(N^5)$ and $\mathcal{O}(N^6)$, respectively. Keller [9] already showed that both algorithms can be implemented with only N matrix inversions, greatly reducing the computational burden. They did this by avoiding the direct calculation of the subspace pairwise projection operators $\{\mathbf{P}_{k,r}\}$, (8). Instead, they first computed the set of unit norm vectors $\{\mathbf{u}_i\}_{i=0}^{N-1}$ such that:

$$\begin{aligned} \langle \mathbf{u}_i, \mathbf{h}_k \rangle &= 0 & \forall i \neq k \\ \|\mathbf{u}_i\| &= 1 & \forall i. \end{aligned} \quad (37)$$

A vector \mathbf{u}_i is computed by applying the unary projection operator \mathbf{P}_i to \mathbf{h}_i :

$$\mathbf{u}_i = \frac{\mathbf{P}_i \mathbf{h}_i}{\|\mathbf{P}_i \mathbf{h}_i\|} \quad (38)$$

where \mathbf{P}_i is the orthogonal projection of space spanned by $\{\mathbf{h}_k\}$ onto the complement of the subspace spanned by $\{\mathbf{h}_k | k \neq i\}$. Similarly to (8), \mathbf{P}_i is computed by:

$$\mathbf{P}_i = \mathbf{I} - \mathbf{B}_i (\mathbf{B}_i^T \mathbf{B}_i)^{-1} \mathbf{B}_i^T \quad (39)$$

where \mathbf{B}_i is the matrix obtained from \mathbf{H} by omitting the i column. They show that the projections in (10) are equal to:

$$\mathbf{P}_{k,r} \mathbf{y} = \langle \mathbf{u}_k, \mathbf{y} \rangle \mathbf{u}_k + \langle \mathbf{u}_r^*, \mathbf{y} \rangle \mathbf{u}_r^* \quad (40)$$

$$\mathbf{P}_{k,r} \mathbf{h}_k = \langle \mathbf{u}_k, \mathbf{h}_k \rangle \mathbf{u}_k + \langle \mathbf{u}_r^*, \mathbf{h}_k \rangle \mathbf{u}_r^* \quad (41)$$

$$\mathbf{P}_{k,r} \mathbf{h}_r = \langle \mathbf{u}_r^*, \mathbf{h}_r \rangle \mathbf{u}_r^* \quad (42)$$

where \mathbf{u}_r^* is given by:

$$\mathbf{u}_r^* = \frac{\mathbf{u}_r - \langle \mathbf{u}_k, \mathbf{u}_r \rangle \mathbf{u}_k}{\sqrt{1 + \langle \mathbf{u}_k, \mathbf{u}_r \rangle^2}}. \quad (43)$$

Applying (40)–(42) avoids directly computing $\mathbf{P}_{k,r}$. Therefore, PILS requires only N matrix inversions in (39).

We now show that it is possible to implement the algorithms using a single matrix inversion, thus reducing the complexity due to the matrix inversions to $\mathcal{O}(N^3)$. We first note that (39) computes the inverse of $\mathbf{B}_i^T \mathbf{B}_i$, where \mathbf{B}_i is a replica of \mathbf{H} without its i -th column. Using the inverse of a partitioned matrix [14] and the matrix inversion lemma [15], we can use the matrix $\tilde{\mathbf{G}} \triangleq (\mathbf{H}^T \mathbf{H})^{-1}$ to calculate $\mathbf{G}_i \triangleq (\mathbf{B}_i^T \mathbf{B}_i)^{-1}$ for all i without any additional matrix inversions. Thus, we can write (39) as:

$$\mathbf{P}_i = \mathbf{I} - \mathbf{B}_i \mathbf{G}_i \mathbf{B}_i^T. \quad (44)$$

Since \mathbf{G}_i can be determined for all i directly from $\tilde{\mathbf{G}}$, the N matrix inversions required in (39) are no longer needed. The algorithm for calculating \mathbf{G}_i is summarized in Algorithm 3.

Algorithm 3 Updating matrix inverse

- 1: $\tilde{\mathbf{G}} = (\mathbf{H}^T \mathbf{H})^{-1}$ % single matrix inversion
 - 2: $d = \tilde{\mathbf{G}}(i, i)$
 - 3: $\mathbf{v} = [1 \dots i-1 \ i+1 \dots N]$ % all indices excluding i
 - 4: $\mathbf{q} = \tilde{\mathbf{G}}(\mathbf{v}, i)$ % the i -th column of $\tilde{\mathbf{G}}$ excluding $\tilde{\mathbf{G}}(i, i)$
 - 5: $\mathbf{G}_i = \tilde{\mathbf{G}}(\mathbf{v}, \mathbf{v}) - \mathbf{q}\mathbf{q}^T/d$ % updated inverse
-

B. Efficient Implementation of SPILS

To make use of the efficient implementation of PILS described in the previous section, Keller [9] showed that we can update the set of vectors $\{\mathbf{u}_i\}_{i=0}^{N-1}$ from (38) for each iteration in the following manner. We let $\{\mathbf{u}_i^t\}$ denote the set $\{\mathbf{u}_i\}_{i=0}^{N-t}$ at iteration t , and j be the index of the sub-symbol decoded at that iteration. The set $\{\mathbf{u}_i^{t+1}\}$ can be computed by first orthonormalizing \mathbf{u}_i^t with respect to \mathbf{u}_j^t :

$$\hat{\mathbf{u}}_i^t = \frac{\mathbf{u}_i^t - \langle \mathbf{u}_i^t, \mathbf{u}_j^t \rangle \mathbf{u}_j^t}{\sqrt{1 + \langle \mathbf{u}_i^t, \mathbf{u}_j^t \rangle^2}} \quad (45)$$

and then computing \mathbf{u}_i^{t+1} using:

$$\begin{aligned} \hat{\mathbf{u}}_i^{t+1} &= \hat{\mathbf{u}}_i^t \langle \mathbf{h}_i, \hat{\mathbf{u}}_i^t \rangle + \mathbf{u}_j^t \langle \mathbf{h}_i, \mathbf{u}_j^t \rangle \\ \mathbf{u}_i^{t+1} &= \hat{\mathbf{u}}_i^{t+1} / \|\hat{\mathbf{u}}_i^{t+1}\|. \end{aligned} \quad (46)$$

This efficient updating scheme saves us from computing the set $\{\mathbf{u}_i^t\}$ at each iteration using (38) and (39).

C. Quick SPILS

We have seen that the SPILS algorithm solves for all remaining variables at a given iteration, and then selects the one with the highest confidence. This leads to calculating the values

in a large matrix \mathbf{A} which is computationally intensive. In this subsection we present a new method termed Quick SPILS (QSPILS), a version of SPILS which only utilizes the pairwise posteriors of Q variables per iteration.

We first need to decide which variables to solve for at a given iteration. We accomplish this by defining a set of indices S^t of size Q for each iteration t . At iteration t , instead of creating a matrix of size $(vM)^2 \times (vM)^2$ (as in SPILS) where $v \triangleq N - t + 1$ is the number of variables left to solve for and M is the alphabet size, we compute a smaller matrix of size $(QM)^2 \times (QM)^2$ which involves only the variables in the Q -sized set S^t :

$$\mathbf{A} = \begin{bmatrix} \mathbf{C}_{0,0} & \cdots & \mathbf{C}_{0,Q-1} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{Q-1,0} & \cdots & \mathbf{C}_{Q-1,Q-1} \end{bmatrix}. \quad (47)$$

Defining $\kappa \triangleq S^t(k)$ and $\rho \triangleq S^t(r)$, the submatrices $\mathbf{C}_{k,r} \forall k, r$ are defined as:

$$\mathbf{C}_{k,r} = \begin{bmatrix} \mathbf{P}_{\kappa,1}\mathbf{P}_{\rho,1} & \cdots & \mathbf{P}_{\kappa,1}\mathbf{P}_{\rho,M} \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{\kappa,M}\mathbf{P}_{\rho,1} & \cdots & \mathbf{P}_{\kappa,M}\mathbf{P}_{\rho,M} \end{bmatrix}. \quad (48)$$

Similarly to the regular SPILS algorithm, the off-diagonal blocks are calculated using (14) and:

$$\mathbf{C}_{k,r}[m, \ell] = P(x_\kappa = a_m, x_\rho = a_\ell | \mathbf{y}) \quad \forall k \neq r \quad (49)$$

and diagonal blocks are calculated using:

$$\mathbf{C}_{k,k}[m, \ell] = P(x_\kappa = a_m)P(x_\kappa = a_\ell). \quad (50)$$

This implies that the reduced matrix represents the pairwise posteriors for only Q out of the v variables remaining to be decoded. We then find the leading eigenvector of the matrix as in PILS. Out of the Q variables solved for, the best one is selected according to (26), and the ILS problem is updated using (27)–(30). S^{t+1} is created by simply removing the index of the variable solved for from S^t and inserting a new index according to a pre-determined order. In general, the channel matrix is close to a banded matrix, where the ‘width’ of the diagonal band depends on the Doppler spread. Thus, cross-talk occurs mostly between neighboring sub-symbols, and the most relevant information can be gleaned by utilizing the pairwise posteriors of neighboring sub-symbols. We therefore utilize a simple numerical order whereby $S^1 = \{1, 2, \dots, Q\}$ and S^t is created by removing the index solved for in the previous iteration and adding the index $Q + t - 1$. When only Q variables remain to be solved for ($v = Q$), SPILS continues regularly. The QSPILS algorithm is summarized in Algorithm 4.

Algorithm 4 Quick SPILS (QSPILS)

- 1: Initialize the set of indices to solve for, S^1
- 2: Calculate $\{\mathbf{u}_i\}_{i=0}^{N-1}$ using (38) and (39)
- 3: **for** $t = 1$ to $N - Q$ **do**

- 4: Compute the matrix \mathbf{A} for the Q variables found in S^t
 - 5: Compute the leading eigenvector of \mathbf{A}
 - 6: Apply (26) and (25) to recover the symbol \hat{x}_i^t with the highest decoding probability
 - 7: Reformulate the ILS problem using (27)–(30) and \hat{x}_i^t
 - 8: Update the set $\{\mathbf{u}_i^{t+1}\}$ using (45) and (46)
 - 9: Create S^{t+1} by removing the index i from S^t and adding a new index according to a preset order
 - 10: **end for**
 - 11: **for** $t = N - Q + 1$ to $N - 1$ **do**
 - 12: Run the regular SPILS algorithm to solve for the remaining variables
 - 13: **end for**
-

D. Computational Complexity and Complexity Reduction

1) *PILS*: The computational complexity of the PILS scheme stems from the construction of the matrix \mathbf{A} and computing its leading eigenvector \mathbf{p} . For the sake of simplicity, we divide the computational burden into three elements, β_1 , β_2 , and β_3 , where the complexity of PILS will be:

$$C_{PILS} = \beta_1 + \beta_2 + \beta_3. \quad (51)$$

β_1 will represent the complexity due to the calculation of the projection operators $\mathbf{P}_{k,r}$ in (8) or \mathbf{P}_i in (39). β_2 will denote the complexity of computing the matrix \mathbf{A} . Last, β_3 will be the number of operations required to compute the leading eigenvector \mathbf{p} in (15). If the length of the vector we are decoding is N and the alphabet is of size M , it follows from (8) that the construction of \mathbf{A} requires $\mathcal{O}(N^2)$ matrix inversions where a matrix inversion requires $\mathcal{O}(N^3)$ operations. Additionally, (18) and (19) imply that there is an order of $\mathcal{O}(N^2 M^2)$ values of \mathbf{A} to be computed, requiring $\mathcal{O}(N)$ operations for each entry. Finally, computing the leading eigenvector of \mathbf{A} can be done using the Power Iteration [13] with T iterations, where each iteration requires $\mathcal{O}(NM)$ operations. Considering that $M \ll N$ and $T = 10$ is a reasonable number of power iterations, the general complexity of PILS is:

$$\begin{aligned} C_{PILS_1}(T, N, M) &= \underbrace{\mathcal{O}(N^2)\mathcal{O}(N^3)}_{\beta_1} + \underbrace{\mathcal{O}(N^2 M^2)\mathcal{O}(N)}_{\beta_2} + \underbrace{(T)\mathcal{O}(NM)}_{\beta_3} \\ &= \mathcal{O}(N^5). \end{aligned} \quad (52)$$

The complexity of PILS is dominated by the many matrix inversions of β_1 . After the efficient implementation of Keller described in Section V-A, only N matrix inversions were required. Thus, the complexity of PILS was improved to:

$$\begin{aligned} C_{PILS_2}(T, N, M) &= \underbrace{(N)\mathcal{O}(N^3)}_{\beta_1} + \underbrace{\mathcal{O}(N^2 M^2)\mathcal{O}(N)}_{\beta_2} + \underbrace{(T)\mathcal{O}(NM)}_{\beta_3} \\ &= \mathcal{O}(N^4). \end{aligned} \quad (53)$$

We then showed how PILS can be implemented using only one matrix inversion. Thus, the complexity of PILS is improved to:

$$\begin{aligned} C_{PILS_3}(T, N, M) &= \underbrace{(1)\mathcal{O}(N^3)}_{\beta_1} + \underbrace{\mathcal{O}(N^2M^2)\mathcal{O}(N)}_{\beta_2} + \underbrace{(T)\mathcal{O}(NM)}_{\beta_3} \\ &= \mathcal{O}(N^3M^2) \end{aligned} \quad (54)$$

which is approximately $\mathcal{O}(N^3)$ since $M \ll N$.

2) *SPILS*: In the case of SPILS, the PILS scheme is run $N - 1$ times, with the signal length n getting reduced by one at each iteration. Additionally, aside from its initialization using the single matrix inversion in the first step of Algorithm 3 ($\beta_1 = \mathcal{O}(N^3)$), the set $\{\mathbf{u}_i\}_{i=0}^{n-1}$ must be updated at the end of each iteration (see Section V-B), requiring $\mathcal{O}(nN)$ operations. Therefore:

$$C_{SPILS}(T, N, M) = \mathcal{O}(N^3) + \sum_{n=2}^N \mathcal{O}(n^2M^2)\mathcal{O}(N) + (T)\mathcal{O}(nM) + \mathcal{O}(nN). \quad (55)$$

The summations in (55) can be calculated using closed formulas [16] and the resulting complexity is:

$$C_{SPILS} = \mathcal{O}(N^4M^2). \quad (56)$$

By comparison, the MMSE-SIC method has a complexity of $\mathcal{O}(N^4)$, which is about the same as SPILS since $M \ll N$.

3) *QSPILS*: We now analyze the QSPILS algorithm's complexity. We recall that QSPILS involves solving for only Q variables per SPILS iteration for as long as $Q \leq N - t$. Therefore, the complexity equations can be divided into two parts, one where the number of variables solved for is the constant Q , and one where the number of variables solved for decreases with each iteration. We thus have:

$$C_{QSPILS}(T, N, M, Q) = \mathcal{O}(N^3) + \sum_{i=1}^2 C_{QSPILS_i}(T, N, M, Q) + \sum_{n=2}^N \mathcal{O}(nN) \quad (57)$$

where $\mathcal{O}(N^3)$ is due to the single matrix inversion, the last term is due to the updates of $\{\mathbf{u}_i\}_{i=0}^{n-1}$ that do not depend on Q , and:

$$\begin{aligned} C_{QSPILS_1}(T, N, M, Q) &= \sum_{n=Q}^N [\mathcal{O}(Q^2M^2)\mathcal{O}(N) + (T)\mathcal{O}(QM)] \end{aligned} \quad (58)$$

$$\begin{aligned} C_{QSPILS_2}(T, N, M, Q) &= \sum_{n=2}^{Q-1} [\mathcal{O}(n^2M^2)\mathcal{O}(N) + (T)\mathcal{O}(nM)]. \end{aligned} \quad (59)$$

First, we note that for $Q = N$, the complexity of C_{QSPILS} in (57) reduces to that of C_{SPILS} in (55). For the more

general case where $Q < N$, the resulting complexity of QSPILS is:

$$C_{QSPILS} = \mathcal{O}(N^2Q^2M^2). \quad (60)$$

We thus lower the complexity by a factor of N^2/Q^2 as compared to SPILS.

VI. NUMERICAL RESULTS

In this section, we evaluate the performance of the PILS and SPILS schemes via computer simulation. We report the results for both coded and uncoded OFDM systems with $N = 64$ subcarriers and QPSK modulation. Both the CP length and number of paths are set to 6. We simulate a wide-sense stationary uncorrelated scattering tapped-delay-line channel with a uniform power profile, where each channel tap is independently generated using Clarke's [17] (Jakes' [18]) Doppler spectrum. The normalized autocorrelation function of the channel due to Doppler spread is given by the zero order Bessel function of the first kind:

$$R(\tau) = J_0(2\pi f_d \tau) \quad (61)$$

where τ is the delay and f_d is the maximum Doppler shift, caused by movement in the system. We assume that the receiver is able to estimate the channel matrix with no error.

A. Uncoded Results

For the uncoded simulations, we compare our results with the well known ZF and MMSE schemes, as well as their sequential counterparts, ZF-SIC and MMSE-SIC respectively. A good summary of these algorithms is presented in [3]. It is important to keep in mind that based on the results shown in Section V-C, both SPILS and MMSE-SIC are of the same order of complexity.

Fig. 1(a) shows the bit error rate (BER) for a normalized Doppler frequency of $f_n = 0.1$. It shows that the SPILS scheme outperforms MMSE-SIC by close to 2.5 dB, which is a significant improvement. In Fig. 1(b), we analyze the results for a normalized Doppler of $f_n = 0.15$, and SPILS shows an even greater improvement over MMSE-SIC. We note that the PILS scheme on its own is not particularly successful. The true strength of the spectral method manifests itself in SPILS, where only one sub-symbol is decoded per iteration.

We now analyze the QSPILS scheme and describe the trade-off between speed and accuracy. As expected, the BER gets slightly worse as Q gets smaller, as seen in Fig. 2(a). This is due to the algorithm not using all of the available information. On the other hand, Fig. 2(b) shows the gains in speed achieved by using QSPILS. The graph shows the percentage of SPILS runtime as a function of Q . Starting at the right, the baseline for comparison is regular SPILS ($Q = 128$) which has a runtime percentage of 100%. Moving left, we reduce the speed to 60% for $Q = 64$ and around 38% for $Q = 32$.

It is interesting to look at how the SPILS algorithm compares to the NP-hard ML solution for a smaller number of subcarriers. Fig. 3 shows simulation results for an 8 subcarrier system with

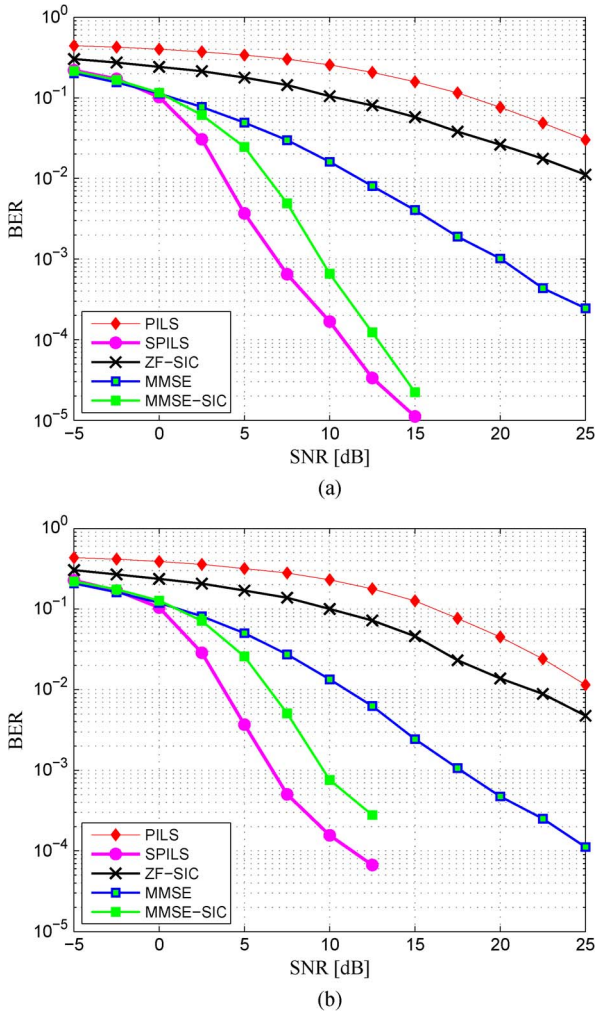


Fig. 1. BER for several algorithms where the normalized Doppler frequency is (a) $f_n = 0.1$ and (b) $f_n = 0.15$.

QPSK modulation and a normalized Doppler of $f_n = 0.1$. We see that for the lower SNR range, SPILS performs very similar to the optimal ML solution.

As noted in Section III-A, the prior probabilities can be utilized to create the pairwise probability matrix. This capability is very useful in iterative decoding, where each decoding step produces an estimate of the prior probabilities to be used in the next step. To demonstrate the capability of the PILS and SPILS algorithms to take advantage of the prior probabilities, Fig. 4 shows the results of a simulation in which the bits were chosen at random with a probability distribution of $\{0.9, 0.1\}$. The figure shows the BER as a function of SNR for the PILS and SPILS schemes. The higher BER curves were obtained by decoders that ignored the prior knowledge and assumed a distribution of $\{0.5, 0.5\}$. The better BER curves were obtained by decoders that made proper use of the prior knowledge. Thus, we note that the algorithms dramatically improve when making use of the prior information.

B. Coded Results

For a coded OFDM system, we use an LDPC code of length 32400 taken from the DVB-S.2 standard. We first compare

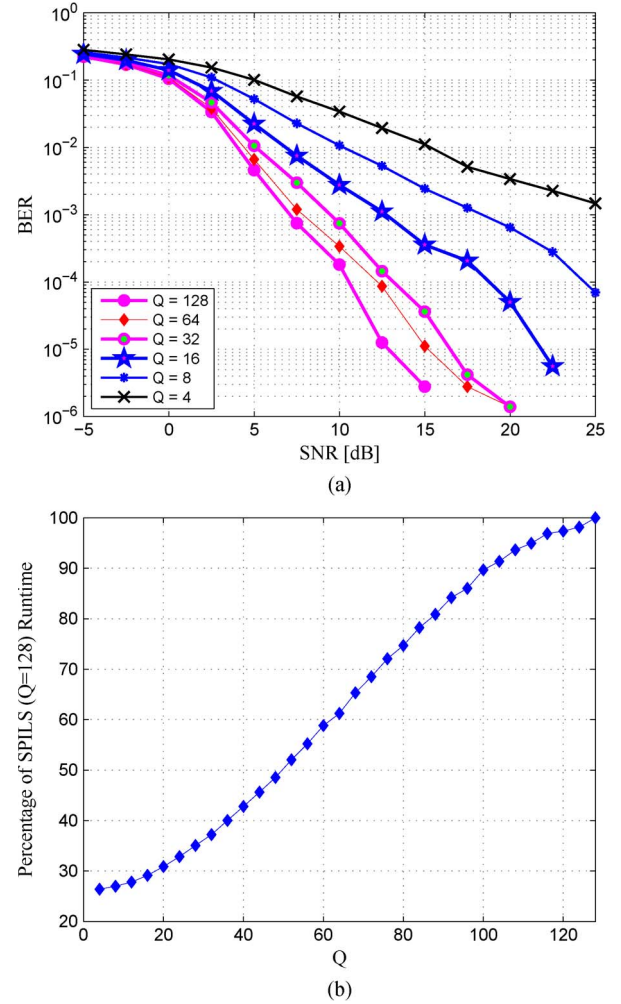


Fig. 2. (a) BER for the QSPILS algorithm with several values of Q , where the normalized Doppler frequency is $f_n = 0.1$. $Q = 128$ represents the standard SPILS scheme and baseline for comparison. (b) The runtime for the standard SPILS scheme is set to 100% and the runtimes for smaller values of Q are plotted as percentages of that.

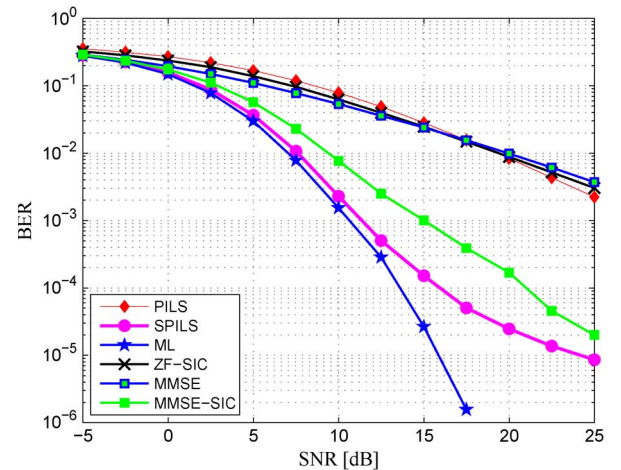


Fig. 3. BER for several algorithms compared to maximum likelihood solution for an 8 subcarrier OFDM system with a normalized Doppler of $f_n = 0.1$.

our coded PILS and coded SPILS algorithms to the MMSE-SIC coded scheme where the extrinsic LLR is calculated as described in [19]. Coded PILS refers to the method discussed

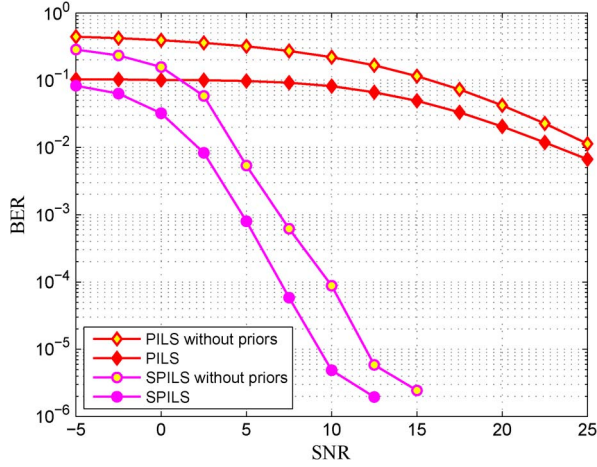


Fig. 4. BER for PILS and SPILS both utilizing and ignoring prior information, with a normalized Doppler of $f_n = 0.25$.

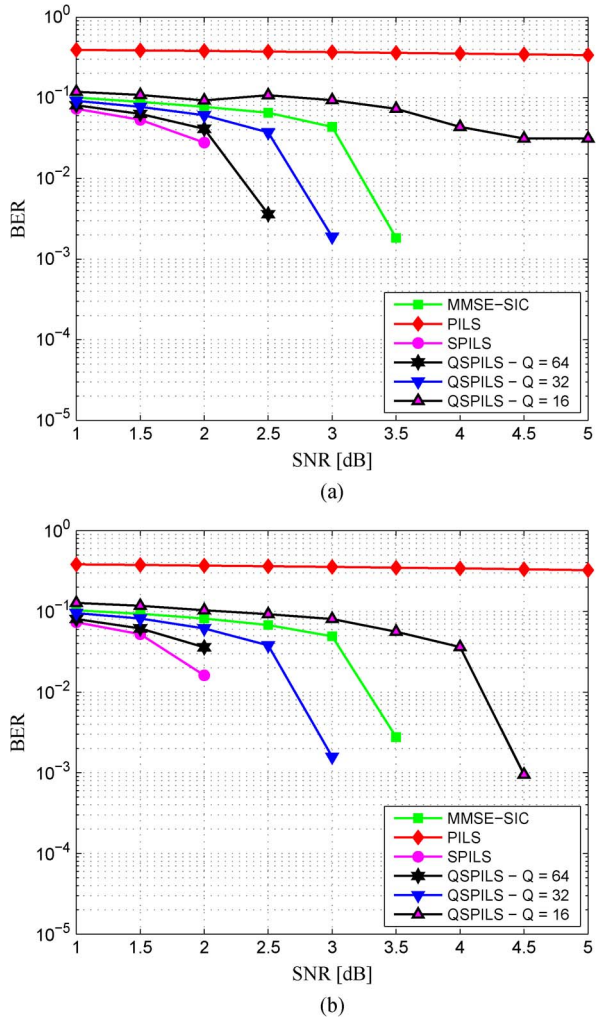


Fig. 5. BER for the coded QSPILS scheme for several values of Q where the coderate is $R = 3/4$ and the normalized Doppler frequency is (a) $f_n = 0.1$ and (b) $f_n = 0.5$. “SPILS” represents the basic SPILS scheme where $Q = 128$.

in Section IV-A1 and Coded SPILS implies that a single LLR value is determined per SPILS iteration as described in Section IV-A2. In addition, we show results for coded QSPILS

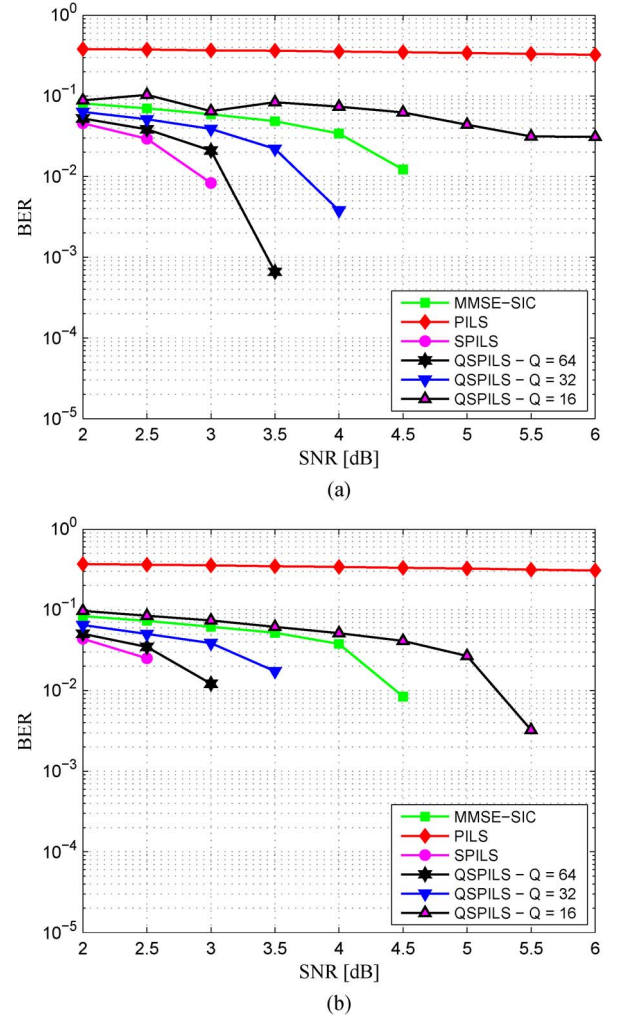


Fig. 6. BER for the coded QSPILS scheme for several values of Q where the coderate is $R = 5/6$ and the normalized Doppler frequency is (a) $f_n = 0.1$ and (b) $f_n = 0.5$. “SPILS” represents the basic SPILS scheme where $Q = 128$.

for various values of Q . We compare the algorithms for coderates of $R = 3/4$ and $R = 5/6$.

Fig. 5(a) and (b) show the BER measurements for a coderate of $R = 3/4$ and normalized Doppler frequencies of $f_n = 0.1$ and $f_n = 0.15$, respectively. We see that the coded SPILS scheme outperforms MMSE-SIC by approximately 1.5 dB. As expected based on the uncoded results, coded PILS on its own does not perform well.

In Fig. 6(a) and (b), we report the decoding accuracy for $R = 5/6$ and $f_n = 0.1$ and $f_n = 0.15$, respectively. As expected, the SNR required to achieve any error probability is higher than before, due to the higher coderate. Here, SPILS has approximately a 1.5–2 dB advantage over MMSE-SIC.

The QSPILS scheme can be used in the coded system to achieve a trade-off between computational complexity and decoding performance. Fig. 5(a) and (b), show a comparison of the regular coded SPILS algorithm and its QSPILS alternatives for a coderate of $R = 3/4$. It follows that for $Q = 64$ and $Q = 32$, we achieve a significant gain over MMSE-SIC, albeit a smaller one than that of standard SPILS. The same is true for a coderate of $R = 5/6$, as can be seen in Fig. 6(a) and (b).

It is interesting to note that the accuracy of SPILS actually improves as the Doppler spread gets larger. Note that the BER of MMSE-SIC in Fig. 6(a) and (b) is approximately identical for the two Doppler spreads $f_n = 0.1$ and $f_n = 0.15$. SPILS, on the other hand, is actually more successful for $f_n = 0.15$. The same phenomenon is evident in the uncoded results in Fig. 1(a) and (b). The reason for this result is that at high Doppler spreads, the channel creates a greater spreading of the signals, implying that each element in the measurements vector contains more information about each other sub-symbol. Apparently, the performance of the SPILS algorithm improves as the pairwise posteriors contain more information on the different sub-symbols.

VII. CONCLUSION

In this work, we showed that the recently introduced SPILS scheme is applicable for OFDM detection in the presence of ICI and demonstrated that this novel OFDM detection scheme outperforms the state-of-the-art MMSE-SIC approach. We then presented an efficient computation method that significantly lowers the computational complexity of the PILS and SPILS schemes to $\mathcal{O}(N^3)$ and $\mathcal{O}(N^4)$, respectively, which is the same order of complexity as MMSE and MMSE-SIC. To further lower the complexity we proposed a novel fast suboptimal decoding scheme termed QSPILS, which allows a trade-off between speed and accuracy. We also extended the SPILS method to support soft decision communications systems, by using the estimated marginal posteriors to compute the LLR as an input to a soft-decision coding system such as LDPC. Numerical results show that for a wide range of Doppler spreads, the proposed SPILS scheme outperforms the MMSE-SIC algorithm by a large margin. Additionally, a surprising conclusion stemming from this work is that the SPILS decoding improves as the Doppler spread becomes more significant. We attribute this phenomenon to SPILS's ability to capitalize on the increased cross-talk between distant sub-symbols within the OFDM symbol. Finally, we showed that SPILS maintains its advantage in the soft-decision coding scenario.

REFERENCES

- [1] X. Cai and G. B. Giannakis, "Bounding performance and suppressing intercarrier interference in wireless mobile OFDM," *IEEE Trans. Commun.*, vol. 51, no. 12, pp. 2047–2056, Dec. 2003.
- [2] W. G. Jeon, K. H. Chang, and Y. S. Cho, "An equalization technique for orthogonal frequency-division multiplexing systems in time-variant multipath channels," *IEEE Trans. Commun.*, vol. 47, no. 1, pp. 27–32, Jan. 1999.
- [3] Y. S. Choi, P. J. Voltz, and F. A. Cassara, "On channel estimation and detection for multicarrier signals in fast and selective Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1375–1387, Aug. 2001.
- [4] L. Rugini, P. Banelli, and G. Leus, "Simple equalization of time-varying channels for OFDM," *IEEE Commun. Lett.*, vol. 9, no. 7, pp. 619–621, Jul. 2005.
- [5] P. Schniter, "Low-complexity equalization of OFDM in doubly selective channels," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 1002–1011, Apr. 2004.
- [6] X. Huang and H. C. Wu, "Robust and efficient intercarrier interference mitigation for OFDM systems in time-varying fading channels," *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 2517–2528, Sep. 2007.
- [7] B. Li, S. Zhou, M. Stojanovic, L. Freitag, and P. Willett, "Non-uniform Doppler compensation for zero-padded OFDM over fast-varying underwater acoustic channels," in *Proc. IEEE OCEANS*, 2007, pp. 1–6.
- [8] S. Yerramalli and U. Mitra, "On optimal resampling for OFDM signaling in doubly-selective underwater acoustic channels," in *Proc. IEEE OCEANS*, 2008, pp. 1–6.
- [9] Y. Keller, A Spectral Probabilistic Approach to Integer Least Squares.
- [10] P. van Emde Boas, "Another NP-complete partition problem and the complexity of computing short vectors in a lattice," Dept. Math., Univ. Amsterdam, Amsterdam, The Netherlands, Tech. Rep. 81-04, 1981.
- [11] J. Goldberger and A. Leshem, "Iterative tomographic solution of integer least squares problems with applications to MIMO detection," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 8, pp. 1486–1496, Dec. 2011.
- [12] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA, USA: SIAM, 2001.
- [13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1996.
- [14] S. G. Glisic, *Advanced Wireless Communications and Internet: Future Evolving Technologies*, 3rd ed. Hoboken, NJ, USA: Wiley, 2011.
- [15] W. Hager, "Updating the inverse of a matrix," *SIAM Rev.*, vol. 31, no. 2, pp. 221–239, 1989.
- [16] I. Gradshteyn and Z. Ryzhik, *Table of Integrals, Series, Products*, 7th ed. New York, NY, USA: Academic Press, 2007.
- [17] R. H. Clarke, "A statistical theory of mobile-radio reception," *Bell Syst. Tech. J.*, pp. 957–1000, Jul./Aug. 1968.
- [18] W. C. Jakes, *Microwave Mobile Communications*. New York, NY, USA: Wiley, 1974.
- [19] A. Matache, C. Jones, and R. Wesel, "Reduced complexity MIMO detectors for LDPC coded systems," in *Proc. IEEE MILCOM*, 2004, vol. 2, pp. 1073–1079.



Avi Septimus was born in New York in 1984. He received the B.Sc. degree in electrical engineering from the Technion—Israel Institute of Technology, Haifa, Israel, in 2009 and the M.Sc. degree in electrical engineering from Bar-Ilan University, Ramat Gan, Israel, in 2012.

Between 2012–2013, he worked at Teledyne LeCroy where he researched signal processing algorithms for high-bandwidth digital oscilloscopes.



Yosi Keller received the B.Sc. degree in electrical engineering from the Technion—Israel Institute of Technology, Haifa, Israel, in 1994, and the M.Sc. and Ph.D. degrees in electrical engineering from Tel-Aviv University, Tel-Aviv, Israel, in 1998 and 2003, respectively.

From 2003 to 2006 he was a Gibbs Assistant Professor with the Department of Mathematics, Yale University, New Haven, CT, USA. He is an Associate Professor at the Faculty of Engineering in Bar Ilan University, Ramat-Gan, Israel.



Itsik Bergel received the B.Sc. degrees in electrical engineering and in physics from Ben Gurion University, Beer-Sheva, Israel, in 1993 and 1994, respectively, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Tel Aviv, Tel-Aviv, Israel, in 2000 and 2005, respectively.

From 2001 to 2003, he was a Senior Researcher at INTEL Communications Research Lab. In 2005, he was a Postdoctoral Researcher at the Dipartimento di Elettronica of Politecnico di Torino, Torino, Italy. He is currently faculty member in the faculty of engineering at Bar-Ilan University, Ramat-Gan, Israel.