Optimized Pilot Pattern for Sparse Channel Estimation: A Low-Complexity Solution for Future Networks

1st Fayad Haddad, 2nd Carsten Bockelmann, 3rd Armin Dekorsy
Department of Communications Engineering
University of Bremen, Germany
Email: {haddad, bockelmann, dekorsy}@ant.uni-bremen.de

***Abstract*—This paper introduces a pilot pattern design optimized to minimize overhead while ensuring effective channel estimation, suitable for future communication systems demanding higher data rates and enhanced spectral efficiency. Focusing on Orthogonal Frequency Division Multiplexing (OFDM) system, a key technology for 5G and a candidate for 6G, we address the challenge of reducing the pilot overhead. While sparse channel estimation is well-studied for its reduced pilot requirements compared to traditional methods, there has been limited research on optimizing pilot patterns to further cut the number of pilots. This study employs data-driven feature selection techniques, tailored for sparse channel estimation, to identify optimal pilot positions. We propose matrix decomposition methods and compare their efficacy against the deep learning-based concrete autoencoder. Our findings show that matrix decomposition offers comparable performance to the autoencoder with reduced complexity and enhanced system flexibility.**

***Index Terms*—Channel estimation, compressed sensing, pilot allocation, concrete autoencoder, matrix decomposition.**

I. Introduction

In Orthogonal Frequency Division Multiplexing (OFDM) systems, channel estimation traditionally relies on embedding reference signals, known as pilots, within the transmitted signal to enable the channel estimation at the receiver. Pilots are allocated to specific resource elements (REs) within the OFDM resource grid, ensuring efficient channel estimation while maintaining pilots orthogonality across frequency, time, and spatial domains. However, in the massive Multiple Input Multiple Output (MIMO) systems boasting numerous antennas, the pilot overhead can become burdensome. This overhead poses a challenge for future networks using the massive MIMO by consuming valuable resources and complicating the maintenance of pilot orthogonality. Therefore, minimizing pilot overhead while maintaining channel estimation accuracy is crucial for meeting system requirements.

To address this challenge, strategic placement of pilots on the OFDM grid can reduce the necessary number of pilots without sacrificing channel estimation accuracy. Traditionally, pilots' positions were determined to adhere to the Nyquist sampling theorem, resulting in significant overhead. Compressed Sensing (CS) techniques have emerged as a promising alternative leveraging the sparsity characteristics of the Channel Impulse Response (CIR) in the time domain. Compared to satisfying the Nyquist condition, applying CS requires fewer pilots. However, to leverage CS effectively for channel estimation, it is essential to carefully select pilot locations that ensure the incoherence of the measurement matrix (pilot position selection matrix) with the sparsifying basis (Fourier basis) [1]. While designing the measurement matrix randomly may offer satisfactory probability of meeting this condition, thoughtful and structural design remains crucial for its fulfillment and to enhance the incoherence. The design process can incorporate data-driven feature selection methods to identify the most informative pilot positions tailored for CS techniques.

Recent advancements in deep learning, notably with the advent of the Concrete AutoEncoder (CAE) [2], have introduced an approach for feature selection of a given data. CAE is exploited in [3] to select the most informative pilot positions for channel estimation. Alongside this, the utilization of linear algebra techniques, particularly Matrix Decomposition (MD), can offer another pathway for feature selection. Pivoted QR decomposition has been employed to extract features from training data, thereby optimizing sensor placement for image compression and reconstruction [4]. It is worth noting that both CAE and MD techniques are data-driven methods.

Additional feature selection techniques are explored in several studies such as in [5] and [6]. The approach in this context involves training a neural network to jointly design a pilot pattern alongside a channel estimator. By employing this approach, the extracted features are optimized specifically for the designed estimator, limiting their effectiveness for other estimators, including those based on CS techniques.

This paper introduces the utilization of matrix decomposition techniques for channel feature selection and their application in designing pilot patterns tailored for CS applications. Namely, we explore the use of both QR and LU [7] decomposition methods for this purpose. To our knowledge, this study marks the first utilization of QR and LU decomposition for pilot pattern design in wireless communication. The work aims also to compare the proposed technique with applying deep learning concrete autoencoder. Comparing the efficiency of applying the MD and the CAE, our findings indicate comparable performance, with MD demonstrating reduced complexity and enhanced flexibility.

Notations: Throughout this paper, we represent matrices by uppercase boldface letters, column vectors by bold lowercase letters, scalars by italic lowercase letters and numbering by italic uppercase letters. Hadamard product is denoted with $\odot$. $(.)^T$ denotes the matrix transpose. $E\{.\}$ represent the mean.

This work was funded by the German Ministry of Education and Research (BMBF) under grants 16KISK016 (Open6GHub) and 16KIS1012 (IRLG).

II. System and Channel Models

We consider an OFDM transmission with K subcarriers. The OFDM resource grid is divided into Resource Blocks (RB),where each block encompasses all the subcarriers and spans across M OFDM symbols. Channel estimation is performed for each RB separately. Within the RB, a certain number of REs are designated as pilots. Let $K' < K$ denote the number of pilots within each subcarrier. Since we focus in this work on designing the pilots positions over the frequency, i.e. the subcarriers, we consider the designed positions to be repeated over time for all OFDM symbols within the RB. The transmitted pilot symbols can be organized in a matrix $\mathbf{X} \in \mathbb{C}^{K' \times M}$. Accounting for the channel effect, received pilots can be expressed as:

$$\mathbf{Y} = \mathbf{W} \odot \mathbf{X} + \mathbf{Z}, \tag{1}$$

where $\mathbf{W}$ and $\mathbf{Z} \in \mathbb{C}^{K' \times M}$ are the channel coefficients at pilots positions, and the additive noise with zero mean and variance σ_z^2 per element, respectively.

A. Channel Model

In wireless communication, due to the radio propagation environment, the transmitted signal reaches the receiver via multiple paths with different time delays and with different attenuation levels. For the kth subcarrier, the complex time-varying channel gain is given by:

$$h_k(m) = \sum_{l=1}^{L} \beta_{k,l} \cdot e^{j2\pi f_{k,l}(\tau_l - m)}, \tag{2}$$

where L represents the number of propagation paths between the transmit and the receive antennas. For the lth path the parameters τ_l, β_l and f_l represent the path delay, the path gain and Doppler frequency, respectively. Accordingly, the Channel Frequency Response (CFR) of the OFDM system can be represented as $\mathbf{H} \in \mathbb{C}^{K \times M}$, with $\mathbf{H} = [\mathbf{h}(1), \mathbf{h}(2), ..., \mathbf{h}(M)]$ and $\mathbf{h}(m) = [h_1(m), h_2(m), ..., h_K(m)]^T$. It is worth noting that the corresponding CIR is sparse since only few paths $S \ll L$ contribute to the wireless channel. The relationship between the CFR and the CIR can be defined as applying the Fourier transformation on time domain channel resulting in the corresponding channel in frequency domain:

$$\mathbf{H} = \mathbf{F}\mathbf{G}, \tag{3}$$

here, $\mathbf{G} \in \mathbb{C}^{L \times M}$ contains the CIR coefficients and it is sparse over the first dimension L. $\mathbf{F} \in \mathbb{C}^{K \times L}$ corresponds to the Discrete Fourier Transform (DFT) matrix if $K = L$. In cases where $K < L$, $\mathbf{F}$ is a submatrix of the DFT matrix.

It's important to highlight that according to (2), entries of the channel matrix $\mathbf{H}$ are correlated for nearby subcarriers due to similar propagation paths. Also the time-varying channels exhibit correlation within the coherence time, as discussed in [8]. These correlation aspects are exploited in our chosen estimator as will be addressed later.

III. Channel estimation and pilot design

In this work we adopt CS techniques to perform channel estimation. Starting with (1) where $\mathbf{Y}$ and $\mathbf{X}$ are available, while $\mathbf{W}$ and $\mathbf{Z}$ are unknown, the calculated channel coefficients at the pilot positions can be noisy, denoted as $\hat{\mathbf{W}}$. To account for the noisy estimation, we introduce the signal-to-noise ratio (SNR), defined as $\frac{E\{\|\mathbf{X}\|_2^2\}}{\sigma_z^2}$.

The matrix $\hat{\mathbf{W}}$ can be thought as projection of $\hat{\mathbf{H}}$ through a pilot selection matrix $\mathbf{C}$. By considering (3) we define:

$$\hat{\mathbf{W}} = \mathbf{C}\hat{\mathbf{H}} = \mathbf{C}\mathbf{F}\hat{\mathbf{G}}, \tag{4}$$

here, $\mathbf{C} \in \mathbb{B}^{K' \times K}$ comprises elements from Boolean domain $\mathbb{B} = \{0, 1\}$. Each row of $\mathbf{C}$ contains only one 1, indexing the selected subcarrier of the resource grid to carry the pilots. With available $\hat{\mathbf{W}}$, $\mathbf{C}$ and $\mathbf{F}$, solving (4) results in the estimated channel response $\hat{\mathbf{G}}$. But it is an underdetermined equation. However, since $\hat{\mathbf{G}}$ is sparse, a sparse recovery algorithm can be applied to detect the sparse solution. Once $\hat{\mathbf{G}}$ is obtained, it can be substituted into (3) to find the estimated channel $\hat{\mathbf{H}}$. The matrix $\mathbf{C}$ must be designed carefully to be incoherent with respect to $\mathbf{F}$, as mentioned earlier. High incoherence plays a crucial role in ensuring the success of sparse recovery based on CS.

According to [4], the number of required pilots K' must be sufficiently large, on the order of:

$$K' \approx c \cdot S \cdot \log\left(\frac{K}{S}\right), \tag{5}$$

the constant multiplier c depends on the coherence between $\mathbf{C}$ and $\mathbf{F}$. Thus, fewer pilots are required if they are less coherent. It can be also observed that K' depends also on the channel sparsity S and the number of subcarriers K.

A. Channel estimation approach

Since this work focuses on pilot pattern design, we evaluate it against a single channel estimation method. Channel estimation starts from (1), by calculating $\hat{\mathbf{W}}$. To estimate the channel coefficients between the pilots positions, we adopt using the approach Compressed Sensing Dynamic Mode Decomposition (CS-DMD) as discussed in [9]. CS-DMD leverages both the inherent sparsity and temporal correlation features of wireless time-varying channels, showing robust channel estimation performance. Within the CS component of CS-DMD, Orthogonal Matching Pursuit (OMP) [10] serves as the sparse recovery method. OMP performs an iterative greedy search to identify the most significant nonzero elements and their corresponding locations. Thus, CS-DMD is denoted here as OMP-DMD.

IV. Feature selection for compressed sensing

In compressed sensing, a careful and structural selection of a subset of relevant features can enhance the efficiency and accuracy of CS algorithms. This process involves identifying the most informative features while discarding irrelevant or redundant. In the following we discuss two different feature selection methods.

A. Feature selection with matrix decomposition

Leveraging compressed sensing with randomly selected samples demonstrates efficient reconstruction for data exhibiting sparsity in a specific basis. However, prior knowledge of the data characteristics can significantly reduce the required measurements by constructing a feature library tailored to the data. Then an appropriate matrix decomposition can be applied to the constructed feature library to extract the required features [4]. To build such a feature library, Proper Orthogonal Decomposition (POD) [11] can be employed. POD is a dimensionality reduction technique capable of capturing the dominant modes of variation within a dataset. Given a dataset of N channels arranged in a matrix $\mathbf{H}' \in \mathbb{C}^{K\times N}$, POD decomposes it into r modes $\mathbf{\Phi} \in \mathbb{C}^{K\times r}$ and their corresponding projection coefficients $\mathbf{\Lambda} \in \mathbb{C}^{r\times N}$. Here r denotes the POD truncation level. The coefficients represent the projection of the data onto subspace spanned by the POD modes. For re-composition, we use $\hat{\mathbf{H}}' = \mathbf{\Phi\Lambda}$. Accordingly from (4), we define:

$$\hat{\mathbf{W}}' = \mathbf{C}\hat{\mathbf{H}}' = \mathbf{C\Phi\Lambda}. \tag{6}$$

In both cases of (4) and (6), the objective is to optimize $\mathbf{C}$. The modes $\mathbf{\Phi}$ encapsulate the dominant patterns or structures within the channels, with each mode consisting of K features. With r dominant modes available, it becomes feasible to identify the most significant common features present in the data. One approach to achieving this is by applying a pivoting-based matrix decomposition, pivoting the rows of $\mathbf{\Phi}$ to arrange the features by importance for the re-composition. We introduce two pivoted matrix decomposition methods in the following:

1) QR decomposition is a matrix decomposition technique in linear algebra. It decomposes a given matrix $\mathbf{A} \in \mathbb{C}^{N_1\times N_2}$ into the product of an orthogonal matrix $\mathbf{Q} \in \mathbb{C}^{N_1\times N_1}$ and an upper triangular matrix $\mathbf{R} \in \mathbb{C}^{N_1\times N_2}$, such that:

$$\mathbf{A} = \mathbf{QR}. \tag{7}$$

The diagonal elements of $\mathbf{R}$ represent the magnitudes of the correlations between the features in the data. Larger diagonal elements indicate stronger correlations. So, features corresponding to larger diagonal elements are considered more important. When features are highly correlated, the matrix used in the QR decomposition can become ill-conditioned, meaning it's sensitive to numerical inaccuracies. Pivoted QR decomposition helps to mitigate this problem by reordering the columns of the matrix to reduce numerical errors and improve the stability of the decomposition process. Mathematically, it can be described as, $\mathbf{AP} = \mathbf{QR}$ where $\mathbf{P} \in \mathbb{C}^{N_2\times N_2}$ is the permutation matrix that is initialized as identity matrix and then updated iteratively to arrange the columns of $\mathbf{A}$ according the their 2-norms. At each iteration, QR column pivoting selects a pivot column with maximal 2-norm, then subtracts from every other column its orthogonal projection onto the pivot column. So at the end, the updated matrix $\mathbf{P}$ will indicates the positions of the most significant columns.

2) LU decomposition is another matrix decomposition technique that decomposes a given matrix $\mathbf{A} \in \mathbb{C}^{N_1\times N_1}$ into the product of a lower triangular matrix $\mathbf{L} \in \mathbb{C}^{N_1\times N_1}$ and an upper triangular matrix $\mathbf{U} \in \mathbb{C}^{N_1\times N_1}$, such that:

$$\mathbf{A} = \mathbf{LU}. \tag{8}$$

Although LU decomposition is typically applied to square matrices, extensions exist to accommodate rectangular matrices. Namely, the LU decomposition with partial pivoting [7], which decomposes the matrix $\mathbf{A} \in \mathbb{C}^{N_1\times N_2}$ into lower triangular $\mathbf{L} \in \mathbb{C}^{N_1\times N_2}$, upper triangular $\mathbf{U} \in \mathbb{C}^{N_2\times N_2}$, and permutation $\mathbf{P} \in \mathbb{C}^{N_1\times N_1}$ matrices, such that $\mathbf{PA} = \mathbf{LU}$. This technique ensures numerical stability for rectangular matrices, and can be used in the same way to extract the features from $\mathbf{P}$.

Using the pivoted QR decomposition as an example, we can apply it to order the features in the rows of $\mathbf{\Phi}$, as they contain the common features of the modes. This can be expressed:

$$\mathbf{\Phi}^T\mathbf{P} = \mathbf{QR}. \tag{9}$$

Here, the features are sorted within $\mathbf{P} \in \mathbb{C}^{K\times K}$. By selecting the first K' columns to indicate the position of the pilots, we obtain $\mathbf{P'} \in \mathbb{C}^{K\times K'}$, and thus $\mathbf{C} = \mathbf{P'}^T$. The same process can also be performed using pivoted LU decomposition.

B. Feature selection with concrete autoencoder

The concrete autoencoder [2] is a deep learning-based method for feature selection, which efficiently identifies a subset of the most informative features. It consists of a single concrete selector layer (encoding layer), and interpolation MultiLayer Perceptron (MLP) (decoding layers).

The Concrete Selector Layer is based on concrete random variables that can be sampled to produce a continuous relaxation [12]. The selector layer has K' output neurons each of them is connected to all of the input features of size K. The input nodes are sampled with weights parameters $\boldsymbol{\alpha} \in \mathbb{R}^K_{>0}$, with $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_K]$, that initially specified randomly. This weights are controlled by a temperature parameter $T \in (0, \infty)$. Then, each jth sampled element from the concrete distribution is defined as:

$$m_j = \frac{\exp((\log\alpha_j + g_j)/T)}{\sum_{k=1}^{K}\exp((\log\alpha_k + g_k)/T)}, \tag{10}$$

where g is randomly sampled from a Gumbel distribution. The extent to which the vector is relaxed, is controlled by T. At the beginning of the training, and to encourage the selector layer to explore different linear combinations of input features, $\boldsymbol{\alpha}$ parameters are initialized to small positive values and the temperature parameter T is set to a large number. However, as the network is trained, the vector of the parameter weights become sparser and T tends towards zero. Because the concrete selector layer samples the input features stochastically based on $\boldsymbol{\alpha}$, any of the features with the large values in the vector $\boldsymbol{\alpha}$ may be selected. Therefore, the K' largest values in $\boldsymbol{\alpha}$ will indicate to the selected features locations, i.e. the positions of the pilots.

C. Applying to MIMO systems

To maintain pilot spatial orthogonality in MIMO systems, the pilot pattern design must be performed individually for each MIMO channel. This underscores the need to minimize design complexity and the size of training datasets, particularly in massive MIMO systems.

V. Complexity Comparison

In this section, we outline the computational complexities associated with the methods employed throughout this paper. We consider the training dataset for the utilized methods as $\mathbf{H}' \in \mathbb{C}^{K \times N}$ and the number of selected pilots is K'. Designing the pilot pattern using MD involves two steps. Initially, applying POD entails a complexity of $\mathcal{O}(NK\min(N,K))$, followed by either pivoted QR decomposition or pivoted LU decomposition, each with a complexity of $\mathcal{O}(Kr^2)$. Typically $r \ll N$, thus the total complexity remains as for POD. it is worth noting that the memory usage for pivoted QR and LU decompositions is roughly estimated as $\mathcal{O}(Kr)$ and $\mathcal{O}(K^2+Kr)$, respectively, suggesting that pivoted QR decomposition may be more feasible as it requires less memory.
For training the CAE network for pilot design, the complexity is estimated as $\mathcal{O}(EN(KK'+Q))$, with E denotes the training epochs and Q is the number of parameters in the MLP decoder. Regarding the proposed channel estimator OMP-DMD, the complexity is given as $\mathcal{O}(MK'\min(M,K'))$.

VI. Simulation Setup Results

We perform numerical simulations to evaluate pilot pattern design for sparse channel estimation in an OFDM system.

A. Simulation Setup

The number of pilots per single RB is given by $P = K'M$. We employ feature selection for pilot pattern design, utilizing methods based on either the concrete autoencoder or matrix decomposition. MD involves designing a feature library using POD with $r = K'$, followed by matrix decomposition using pivoted QR. To perform sparse channel estimation we leverage OMP-DMD, with the DMD rank $r_{dmd} = 3$.
The training datasets used for MD and CAE contain 100 and 12000 channel realization, respectively, unless otherwise stated. Additive White Gaussian Noise (AWGN) is added to the training datasets with SNR$=$ 30 dB, unless otherwise stated. For the decoder of the CAE, three layers MLP is used, where each is followed by a LeakyRelu(0.2) and a Dropout(0.1). The CAE is trained for 100 epochs.
To assess the performance, we utilize the normalized mean square error (NMSE), defined as:

$$\text{NMSE} = \frac{E\{\|\mathbf{H} - \hat{\mathbf{H}}\|_2^2\}}{E\{\|\mathbf{H}\|_2^2\}}. \tag{11}$$

To generate different realizations of channel coefficients, we employ Heterogenous Radio Mobile Simulator (HermesPy) [13]. System parameters are listed in Table I.

TABLE I
Simulation Parameters

System Parameters	Value
Channel model	COST 259 [14]
Carrier frequency	2 GHz
Receiver velocity	50 Km/h
RB size K, M	1024, 14
CIR taps in time domain L	1024
Subcarrier spacing	15 kHz

B. Simulation Results

The first simulations observation is that using different training datasets results in different pilot patterns, even with the same number of pilots. All resulting patterns achieve the same channel estimation quality, indicating that there is no single optimal pattern for pilot positions.

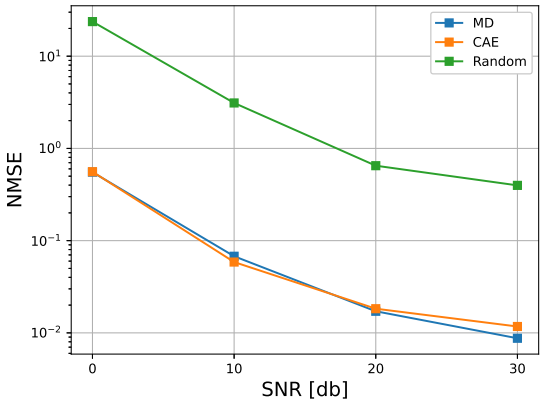

Fig. 1. Channel NMSE in terms of SNR, $P = 112$, $S = 8$.

The results in Fig. 1 highlight the impact of designing the pilot position on enhancing channel estimation. It's evident that the NMSE associated with randomly selected pilot positions is higher compared to that achieved with pilot pattern design with MD or CAE. The NMSE performed by applying CAE and MD is comparable. However, MD's training dataset is 120 times smaller than CAE's, indicating its efficiency in requiring fewer channel realizations. When comparing the NMSE depicted in Fig. 1 to the NMSE in Fig. 3 from [9], it is observable that our obtained NMSE is significantly higher, despite utilizing the same estimator, OMP-DMD. This difference can be attributed to our choice of $K' = 8$, whereas [9] used $K' = 102$.

The bar graph presented in Fig. 2 illustrates the performance of NMSE across various number of pilots P. With a low number of pilots ($P = 84$), designing the pilots positions yields only small enhancements compared to random selection. This can be attributed to the limited number of pilots, which may inadequately capture channel features. Also, with a high number of pilots ($P = 224$), the benefits of pilot pattern design are marginal, as random selection is more likely to encapsulate channel information. Hence, we refrain from choosing $K' = 102$, as in [9], since random selection can perform comparably at this high value. For a moderate number of pilots (112 and 140 pilots), the utilization of feature selection yields significant improvements for the sparse channel estimation.

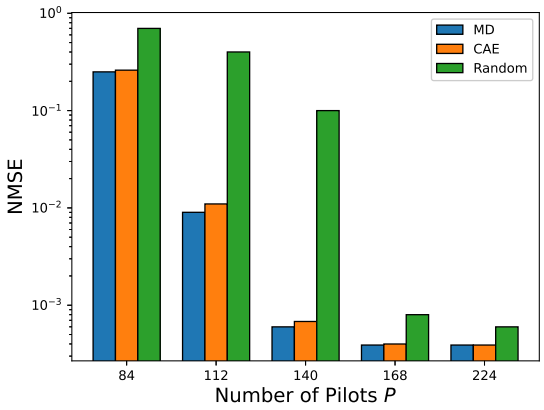

Fig. 2. Channel NMSE in terms of number of pilots, SNR $= 30$ dB, $S = 8$.

Fig. 3 provides a comparative analysis of the performance of two pilot pattern design methods, MD and CAE, across varying training SNRs and dataset size N. Each method is denoted in the legend along with its corresponding N. In Fig. 3(a), with a training SNR of 30 dB, increasing the dataset size for both MD and CAE does not significantly enhance the NMSE. However, in Fig. 3(b), with a training SNR of 10 dB, expanding the dataset size for the MD method shows potential for performance improvement, leading to reduced NMSE. This is because the POD is able to capture more dominant modes and the truncation of the modes can reduce the noise. Conversely, enlarging the dataset size for CAE in the same scenario leads to degraded performance and increased NMSE. This decline stems from high noise within the training data, which introduces irrelevant patterns that the model may erroneously learn, resulting in overfitting. Hence, careful consideration must be given to the training dataset size for the CAE approach.

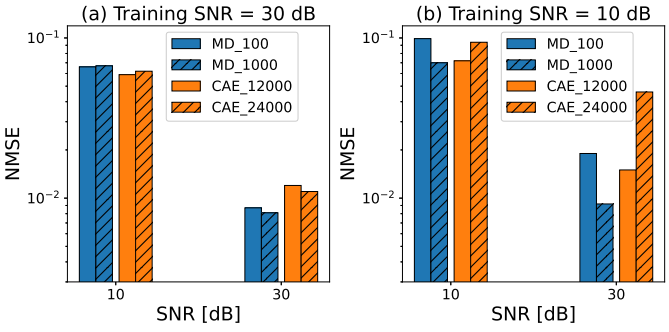

Fig. 3. Channel NMSE in terms of two distinct training SNR values and two varying training dataset sizes. $P = 112$, $S = 8$.

In Table II, we illustrate the relationship between sparsity S and the number of pilots P needed to achieve NMSE $\leq 10^{-3}$. It shows that as sparsity increases, more pilot symbols are required, aligning with the expectations set by equation (5). Notably, when designing the pilot position using feature selection, the resulting NMSE matches that of random selection but with a reduced number of required pilots.

TABLE II
REQUIRED PILOTS TO PERFORM NMSE $\leq 10^{-3}$

		Pilot Design Method	
		Feature Selection	Random
Sparsity S	**3**	70	98
	5	98	126
	8	140	168

VII. CONCLUSION

In conclusion, our study delves into the optimization of pilot patterns in OFDM systems, crucial for efficient channel estimation with minimized overhead. We explored feature selection techniques, particularly tailored for CS applications, in identifying informative pilot positions, leveraging linear algebra and deep learning methods. Our research compared the performances of MD and CAE. Comparative analysis revealed comparable performance levels. However, MD exhibited advantages in terms of data efficiency, requiring fewer channel realizations, and boasted reduced computational complexity. Additionally, the utilization of POD to construct the feature library demonstrated better performance in scenarios involving noisy data. This underscores the potential of the used MD method in enhancing the efficacy of pilot pattern design for sparse channel estimation in OFDM systems.

REFERENCES

[1] R. Baraniuk, "Compressive sensing [lecture notes], ieee signal process," *Mag*, vol. 24, no. 4, pp. 118–121, 2007.
[2] M. F. Balın, A. Abid, and J. Zou, "Concrete autoencoders: Differentiable feature selection and reconstruction," in *International conference on machine learning*, pp. 444–453, PMLR, 2019.
[3] M. Soltani, V. Pourahmadi, and H. Sheikhzadeh, "Pilot pattern design for deep learning-based channel estimation in ofdm systems," *IEEE Wireless Communications Letters*, vol. 9, no. 12, pp. 2173–2176, 2020.
[4] K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, "Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns," *IEEE Control Systems Magazine*, vol. 38, no. 3, pp. 63–86, 2018.
[5] H. Fu, W. Si, and I.-M. Kim, "Deep learning-based joint pilot design and channel estimation for ofdm systems," *IEEE Transactions on Communications*, vol. 71, no. 8, pp. 4577–4590, 2023.
[6] J. Guo, T. Chen, S. Jin, G. Y. Li, X. Wang, and X. Hou, "Deep learning for joint channel estimation and feedback in massive mimo systems," *Digital Communications and Networks*, vol. 10, no. 1, pp. 83–93, 2024.
[7] G. Shabat, Y. Shmueli, Y. Aizenbud, and A. Averbuch, "Randomized lu decomposition," *Applied and Computational Harmonic Analysis*, vol. 44, no. 2, pp. 246–272, 2018.
[8] F. Haddad, C. Bockelmann, and A. Dekorsy, "A dynamical model for csi feedback in mobile mimo systems using dynamic mode decomposition," in *ICC 2023-IEEE International Conference on Communications*, pp. 5265–5271, IEEE, 2023.
[9] F. Haddad, C. Bockelmann, and A. Dekorsy, "Channel estimation and pilot overhead reduction in ofdm systems using compressed sensing dynamic mode decomposition," *IEEE Communications Letters*, 2024.
[10] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
[11] J. Weiss, "A tutorial on the proper orthogonal decomposition," in *AIAA aviation 2019 forum*, p. 3333, 2019.
[12] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv preprint arXiv:1611.00712*, 2016.
[13] "HermesPy." https://www.barkhauseninstitut.org/en/results/hermespy.
[14] "ETSI TR 125 943." https://www.etsi.org/deliver/etsitr/125900125999/125943/06.00.0060/tr125943v060000p.pdf.