

CSI Feedback Compression in MIMO Time-Varying Systems via Dynamic Mode Decomposition and Convolutional Autoencoder

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—In wireless communication, base stations rely on downlink Channel State Information (CSI) to perform precoding. Without channel reciprocity, the mobile station must transmit the estimated CSI back to the base station. Due to the time-varying nature of the environment, channel characteristics change constantly, requiring regular CSI feedback updates at intervals that depend on the rate of change. Thus, increasing the interval between the CSI updates, can reduce the average CSI feedback overhead. Additionally, in Multiple-Input Multiple-Output (MIMO) systems, the CSI feedback overhead grows with the number of antennas and bandwidth, leading to a potential performance bottleneck. To reduce the CSI feedback overhead and increase the intervals between CSI updates, we propose a novel method that integrates Dynamic Mode Decomposition (DMD) and Convolutional Autoencoders (CAE) to model and compress channel dynamics. DMD decomposes the channel matrix into modes that can predict the future state of the channel, thereby extending CSI feedback intervals, while CAE captures the most relevant features of these modes for further compression. Simulation results demonstrate that this method effectively reduces feedback overhead and prolongs the intervals between CSI updates.

Index Terms—Time-varying channels, MIMO systems, CSI feedback, Channel estimation, Dynamic Mode Decomposition, Convolutional Autoencoder.

I. INTRODUCTION

Multiple-Input Multiple-Output (MIMO) systems are fundamental in modern wireless communication networks. A critical aspect of MIMO transmission is obtaining accurate downlink Channel State Information (CSI) at the Base Station (BS), which is crucial for effective precoding.

In Frequency Division Duplex (FDD), the use of different frequencies for downlink and uplink prevents channel reciprocity [1], while time division duplex (TDD) theoretically allows for reciprocal channels, practical issues such as non-identical RF chains at the transmitter and receiver often negate this advantage [2]. Consequently, in both FDD and TDD, the BS cannot rely on uplink reference signals, aka pilots, to obtain downlink CSI. Instead, the Mobile Station (MS) estimates the channel based on the downlink pilots, and then reports the CSI back to the BS. However, transmitting the full channel matrix incurs significant uplink overhead. To address this, the MS compresses the estimated channel matrix to produce a low-dimensional representation that retains the key characteristics of the channel, reducing the feedback overhead while maintaining performance at a required quality level.

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

Several techniques for compressing CSI have been proposed. Traditionally, methods such as Singular Value Decomposition (SVD), which compresses the channel matrix in the frequency domain, and Compressed Sensing (CS), which assumes that wireless channels are sparse in the time domain, have been utilized. The authors in [3] compare these two methods and demonstrate that CS can outperform SVD.

Recently, machine learning (ML) techniques have emerged as promising options for channel matrix compression. The first deep learning (DL) network introduced for this purpose was presented in [4]. This architecture, known as CsiNet, exhibits significant advantages over traditional CS methods. However, CsiNet is a point estimation model that learns a single value per codeword dimension, making the reconstruction quality at the BS vulnerable to noise. Other ML models, such as Autoencoders, have also been widely used for CSI compression. The authors in [5] propose the Convolutional AutoEncoder (CAE), which estimates distribution parameters, namely the mean and variance for a Gaussian distribution, for each compressed CSI dimension. They show that CAE demonstrates noteworthy robustness against noise compared to CsiNet.

In mobile communication, the movement of terminals causes continuous changes in channel characteristics, with higher velocities result in more rapid changes [6]. This leads to a phenomenon known as channel aging. CSI updates are sent to the BS at intervals influenced by the channel aging effect. However, as user velocity changes, the rate of channel aging can also vary. Thus CSI updates can be fed back aperiodically. A study that addresses this matter through channel prediction is presented in [7], where the authors use Dynamic Mode Decomposition (DMD) [8] to compress the channel matrix and predict its future state, thereby mitigating channel aging and extending the CSI update intervals.

In this paper, we propose a method that combines DMD with CAE. DMD generates a dynamic model, that the CAE can compress to extract key features. This choice ensures a straightforward and efficient structure. Simulation results show that the proposed approach effectively reduces CSI feedback dimensionality and extends update intervals, leading to lower average CSI overhead.

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. $(\cdot)^\dagger$ represents the Moore–Penrose pseudoinverse.

II. SYSTEM AND CHANNEL MODELS

A. System Model

We consider a MIMO-Orthogonal Frequency Division Multiplexing (MIMO-OFDM) system with K subcarriers, N_t transmit antennas at the MS, and N_r receive antennas at the BS. Channel estimation is conducted over one Resource Block (RB) spanning duration T in time, resulting in a 4D estimated channel matrix denoted as $\hat{\mathbf{H}} \in \mathbb{C}^{K \times T \times N_t \times N_r}$. To find the corresponding CSI to be fed back, we reshape $\hat{\mathbf{H}}$ to 2D matrix $\mathbf{H} \in \mathbb{C}^{L \times T}$, where $L = KN_tN_r$. The CSI is then compressed before reporting to the BS. Upon receipt, the CSI at the BS undergoes decompression to reconstruct the channel matrix $\hat{\mathbf{H}}$ essential for appropriate precoding of downlink user data. However, due to the compression error, $\hat{\mathbf{H}}$ can deviate from the estimated \mathbf{H} , leading to a channel reporting error that dependent on the compression accuracy. In practice, factors like quantization and transmission can introduce additional errors to the received \mathbf{H} , but these are beyond the scope of this paper. To evaluate performance, we employ the Normalized Mean Square Error (NMSE), defined as $\text{NMSE} = \frac{E\{\|\mathbf{H} - \hat{\mathbf{H}}\|_2^2\}}{E\{\|\mathbf{H}\|_2^2\}}$.

B. Time-Varying Channel Model

In wireless mobile networks, the motion of the MS induces Doppler frequency shifts in the transmitted waves, resulting in time-varying changes to the channel characteristics in the time domain [9]. A key parameter used to characterize these time-varying channels is the coherence time τ , which indicates the duration over which the channel can be regarded as temporally correlated. This parameter can be expressed as:

$$\tau = \frac{c}{2vf_c}, \quad (1)$$

where v , f_c , and c represent the MS velocity, the carrier frequency of the signal, and the speed of light, respectively. The Doppler shift is directly proportional to the MS velocity, leading to an inverse relationship between coherence time and Doppler shift. The phenomenon of channel aging can be quantified using the channel autocorrelation function $R_H(\Delta t)$ [10, Chapter 3], as depicted in Fig. 1. As the time difference Δt increases, the temporal correlation of the channel diminishes steadily until $\Delta t = \tau$. Beyond this threshold, the correlation becomes negligible.

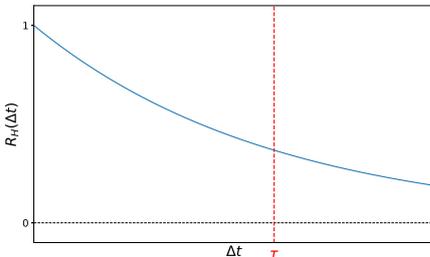


Fig. 1. Channel correlation vs. Time difference

C. Channel Sparsity

The matrix \mathbf{H} represents the frequency domain channel coefficients. Due to the radio propagation environment, it is accepted that the channel in the time domain exhibits sparsity, which aligns with the 3GPP channel model [11]. We define:

$$\mathbf{G} = \text{IDFT}(\mathbf{H}), \quad (2)$$

where, $\text{IDFT}(\cdot)$ represents the Inverse Discrete Fourier Transformation. Here $\mathbf{G} \in \mathbb{C}^{L \times T}$ denotes the Channel Impulse Response (CIR) in the time domain, which is sparse along the first dimension, indicating that only few taps are significant. According to the 3GPP model, the level of sparsity \bar{S} , that represents the number of non-zero taps of CIR, remains constant during the coherence time τ . Additionally, in MIMO systems, it is assumed that the channel support (positions of non-zero elements) is common across all MIMO channels. Consequently, the non-zero taps of the CIR can be arranged into a matrix as follows:

$$\mathbf{S} = \mathbf{G}[\mathbf{G} \neq 0], \quad (3)$$

where $\mathbf{S} \in \mathbb{C}^{S \times T}$ and the entire sparsity $S = \bar{S}N_tN_r$. To reconstruct \mathbf{H} from \mathbf{S} , we first zero-pad \mathbf{S} to form $\mathbf{G} = [\mathbf{S} \mathbf{0}]$. Then, we apply the Discrete Fourier Transform (DFT) to obtain: $\mathbf{H} = \text{DFT}(\mathbf{G})$.

D. Aperiodic Feedback

The CSI feedback update interval should be determined by evaluating the NMSE between the received CSI channel matrix $\hat{\mathbf{H}}$ and the actual channel matrix \mathbf{H} . As channel aging occurs, the NMSE increases over time. The feedback interval d can be extended until the NMSE reaches a predefined threshold γ , based on system requirements. If the MS is capable of measuring the NMSE and comparing it to the threshold γ , a simple feedback mechanism can be implemented: when $\text{NMSE} \geq \gamma$, the MS sends updated CSI to the BS [7]. The selection of the threshold γ balances system performance, measured by NMSE, and the frequency of CSI feedback, which influences the average overhead.

III. CSI FEEDBACK COMPRESSION

This section explores two distinct methods for compressing CSI feedback. The first is a mathematical data-driven approach based on DMD, which not only compresses the channel but also predicts its future state. The second is a deep learning-based method using a CAE to extract the key features of the channel matrix to reduce its dimensionality.

A. Dynamic Mode Decomposition

DMD [8] is a data-driven method for decomposing dynamical systems into spatiotemporal coherent structures that exhibit oscillations at fixed frequencies which either grow or decay at fixed rates. The method relies on collecting snapshots from a dynamical system. In the context of wireless channels, the matrix \mathbf{H} comprises T channel snapshots. Specifically, $\mathbf{H} = [\mathbf{h}_1 \mathbf{h}_2 \dots \mathbf{h}_T]$, with each $\mathbf{h}_t \in \mathbb{C}^{L \times 1}$ representing the concatenated subcarriers of all MIMO channels in one vector

and over one OFDM symbol t , $\forall t \in [1, \dots, T]$. To use DMD, the channel vectors need to be arranged into two data matrices:

$$\begin{aligned} \mathbf{H}' &= [\mathbf{h}_1 \ \mathbf{h}_2 \ \dots \ \mathbf{h}_{T-1}] \in \mathbb{C}^{L \times T-1}, \\ \mathbf{H}'' &= [\mathbf{h}_2 \ \mathbf{h}_3 \ \dots \ \mathbf{h}_T] \in \mathbb{C}^{L \times T-1}. \end{aligned} \quad (4)$$

DMD defines a linear approximation, expressing how \mathbf{H}'' evolves from \mathbf{H}' as:

$$\mathbf{H}'' \approx \mathbf{A}\mathbf{H}', \quad (5)$$

where $\mathbf{A} \in \mathbb{C}^{L \times L}$ is an approximating linear operator, determined as: $\mathbf{A} = \mathbf{H}''\mathbf{H}'^\dagger$. This solution minimizes the Frobenius norm $\|\mathbf{H}'' - \mathbf{A}\mathbf{H}'\|_F$ functioning as a linear regression of data onto the dynamics represented by \mathbf{A} . In practice, direct analysis of the matrix \mathbf{A} may be intractable, especially when the number of subcarriers or/and antennas is extensive. However, the rank of \mathbf{A} is at most $T - 1$, since it is constructed as a linear combination of the $T - 1$ columns of \mathbf{H} . Therefore, instead of solving for \mathbf{A} , DMD projects the data onto a low-rank subspace defined by at most $T - 1$ Proper Orthogonal Decomposition (POD) modes. It then solves for a low-dimensional solution evolving on these POD mode coefficients. The DMD then uses this low-dimensional solution to find the leading M eigenvectors $\Phi \in \mathbb{C}^{L \times M}$ and eigenvalues $\Lambda \in \mathbb{C}^{M \times 1}$, which are called DMD modes and dynamics, respectively. It has been demonstrated in [8] that the snapshots (channels) are recomposed as:

$$\mathbf{h}_t \approx \Phi \Lambda^t. \quad (6)$$

Here M denotes the DMD rank truncation. It indicates the number of used eigendecompositions. Equation (6) implies that the higher the M , the better the resolution of recomposed \mathbf{h}_t . However, it is important to mention that the generated eigendecompositions are sorted in descending order of significance. This implies that a few eigendecompositions contain most of the channel power. Accordingly, it may be sufficient to take just a few modes and dynamics to ensure an adequate resolution of the recomposed \mathbf{h}_t . Moreover, truncation can also contribute to noise reduction, since it removes the modes with no effect on the reconstruction and may contain only noise.

One important feature of DMD is its capability for future state prediction. This can be achieved by extending the application of formula (6) by growing the index t beyond T , such as $t = (T + 1), (T + 2), \dots$

Since DMD decomposes the channel matrix into modes that capture the dominant structures and their growth/decay rates, and given that the channel matrix exhibits sparsity in the time domain, the DMD modes will reflect these dominant frequencies, showing sparsity as well. This sparse representation, denoted as $\Phi_{sp} \in \mathbb{C}^{L \times M}$, is obtained by applying an IDFT as follows: $\Phi_{sp} = \text{IDFT}(\Phi)$. The non-zero taps in Φ_{sp} are then arranged in a matrix $\Psi \in \mathbb{C}^{S \times M}$, such that: $\Psi = \Phi_{sp}[\Phi_{sp} \neq 0]$. Consequently, the CSI feedback generated by DMD comprises Ψ and Λ , resulting in a total size of $(S \times M + M)$. Thereby the CSI feedback size can be adjusted by varying the rank M .

B. Convolutional Autoencoder

A Convolutional Autoencoder [12] is a neural network designed for unsupervised feature extraction, compression, and reconstruction of high-dimensional data. It consists of an *Encoder* that compresses the input into a low-dimensional latent space, and a *Decoder* that reconstructs the data. For compressing CSI feedback, the encoder maps input data $\mathbf{S} \in \mathbb{R}^{S \times T}$ to a lower-dimensional latent representation $\mathbf{z} \in \mathbb{R}^P$, where $P < S \times T$, using convolutional operations followed by activation functions.

$$\mathbf{d}_l = f(\mathbf{W}_l * \mathbf{d}_{l-1} + \mathbf{b}_l), \quad (7)$$

where \mathbf{d}_l is the output at layer l , $f(\cdot)$ is the non-linear activation function, $*$ denotes the convolution operator, \mathbf{W}_l and \mathbf{b}_l are the learnable convolutional weights and biases, respectively. The output of the encoder is a compressed latent representation \mathbf{z} , given by the final layer of the encoder. The decoder reconstructs the input data from the latent space \mathbf{z} . It applies a series of transposed convolutional layers, upsampling the latent representation back to the original input dimensions. To optimize the CAE, the objective is to minimize the reconstruction error between the input \mathbf{S} and the reconstructed output $\hat{\mathbf{S}}$. This goal is quantified using the mean squared error loss function as: $\mathcal{L}(\mathbf{S}, \hat{\mathbf{S}}) = E\{\|\mathbf{S} - \hat{\mathbf{S}}\|_2^2\}$. The CAE is trained to minimize this loss, ensuring that the compressed latent representation retains sufficient information for accurate reconstruction of the original data.

CAE-based CSI Compression: To reduce the overhead of CSI feedback, we exploit the channel's inherent sparsity in the time domain by transforming the channel matrix \mathbf{H} , as described in Section II-C. Since CAE operates on real-valued data, the resulting complex-valued matrix, \mathbf{S} , is split into its real and imaginary parts, which are then concatenated to form a real-valued input of size $2S \times T$, making it compatible with the encoder.

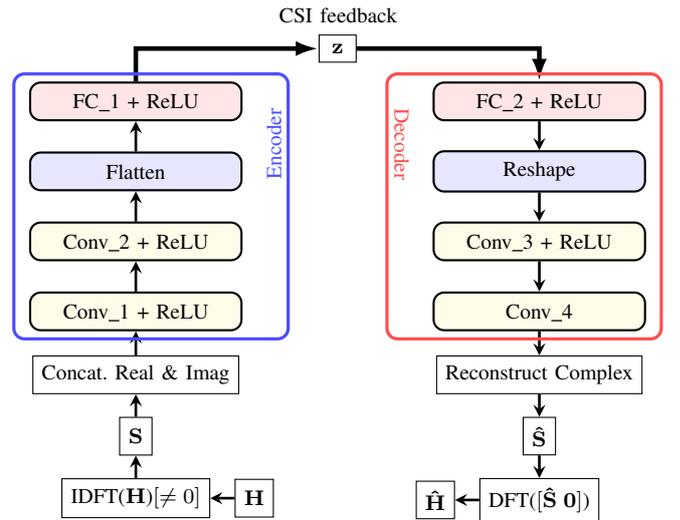


Fig. 2. CAE network architecture for CSI compression.

The architecture of the CAE employed in this study is illustrated in Fig. 2. This proposed network is designed to be basic and straightforward, as the primary objective of this paper is not to create a sophisticated compression mechanism but to apply a compression technique to the DMD modes and assess the performance variations with and without compression.

In the encoder component of the CAE, the primary building blocks consist of two consecutive Convolutional layers (Conv), denoted as Conv_1 and Conv_2. These layers are specifically focused on extracting local features from the input data, capturing essential spatial or temporal patterns. The convolutional operations performed by these layers allow the network to learn and identify relevant features effectively, which is crucial for the subsequent steps in the compression process. Following the convolutional layers, the output data is flattened into a one-dimensional vector to prepare it for further processing.

To facilitate adjustments to the latent space size, we incorporate a Fully Connected (FC) layer, referred to as FC_1. The addition of FC layers in a CAE is useful as they enable effective control on the dimensionality reduction after feature extraction, retaining only the most relevant information for efficient feedback transmission.

Transitioning to the decoder section of the CAE, we begin with another fully connected layer, FC_2. This layer reshapes the data output from the encoder to make it compatible with the subsequent Conv_3 layer. The use of convolutional layers in the decoder, Conv_3 and Conv_4, allows the network to reconstruct the input data from the compressed latent representation.

Furthermore, throughout the architecture, we utilize Rectified Linear Units (ReLU) as activation functions in the deep layers. The ReLU activation function introduces non-linearity to the network, enhancing the learning capacity.

Overall, the design of the CAE in this study emphasizes simplicity and functionality, allowing for a focused analysis of the impact of compression on DMD modes.

C. DMD-CAE-Based CSI Compression

Incorporating the DMD framework into the CAE network introduces a novel approach to compressing CSI feedback. The DMD technique enables extracting essential modes that capture the underlying structures of the channel. By applying IDFT to the DMD modes, we transform into the time domain representation, reflecting the temporal sparsity features of the channel, as mentioned in Section III-A. The resulting data, denoted as Ψ , consists of complex-valued modes. This transformation ensures that the CAE processes a compact and relevant representation of the channel's dominant structures. Unlike traditional CAE methods that directly utilize the channel state information, the DMD-CAE leverages the more informative DMD modes as input. Fig 3 details the entire process for the proposed DMD-CAE network. This shift results in a higher Compression Ratio (CR), as the size of Ψ is $S \times M$, while the original input size, without applying DMD, is \mathbf{H} is of size $S \times T$, with $M < T$. Additionally, utilizing DMD compression offers the advantage of enabling predictions of the future state of the channel at the BS. Algorithm 1 shows the process steps for the proposed approach DMD-CAE encoder that is performed at the MS side. The opposite process is carried out at the BS side.

Algorithm 1: Pseudocode for the DMD-CAE Encoder

- 1: **Input:** Channel matrix \mathbf{H} , DMD rank M
 - 2: **Output:** Compressed representation \mathbf{z}
 - 3: **Step 1:** Apply DMD to \mathbf{H}
 - 4: Compute DMD modes Φ
 - 5: **Step 2:** Apply IDFT Φ
 - 6: Compute $\Psi \leftarrow \text{IDFT}(\Phi)_{[\neq 0]}$
 - 7: **Step 4:** Concatenate real and imaginary parts
 - 8: Reform $\Psi \leftarrow \begin{bmatrix} \Psi_{\text{real}} \\ \Psi_{\text{imag}} \end{bmatrix}$
 - 9: **Step 5:** Compress using the CAE Encoder
 - 10: Compute the compressed representation $\mathbf{z} \leftarrow \text{CAE}(\Psi)$
-

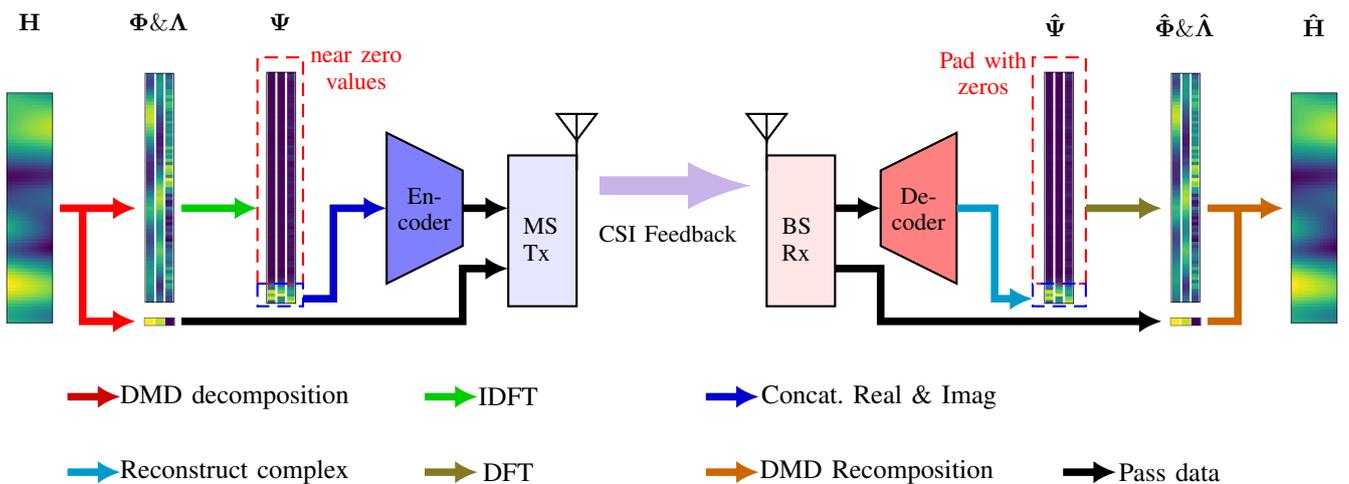


Fig. 3. The proposed framework for DMD-based CSI feedback utilizing CAE.

IV. SIMULATION SETUP AND RESULTS

In this section, we perform numerical simulations to evaluate the performance of the dynamical model CSI feedback scheme and compare its components with CAE and compare the performance for different scenarios.

A. Simulations Setup

We employ Heterogenous Radio Mobile Simulator (HermesPy) [13] to generate the channel coefficients. The system parameters used are listed in Table I.

TABLE I
SIMULATION PARAMETERS

System Parameters	Value
Channel model	COST 259 [14]
Carrier frequency f_c	2 GHz
MS velocity v	50 Km/h
No. of BS and MS antenna N_r, N_t	2, 2
Subcarrier spacing	15 KHz
RB size $K \times T$	72×14
CIR sparsity \bar{S}	8
Channel estimating error	AWGN

Utilizing (1), the coherence time is approximated by $\tau \approx 5.4$ ms. With a subcarrier spacing of 15 KHz, the duration of one OFDM symbol is $66.7 \mu\text{s}$. The number of OFDM symbols within the correlation time T' is calculated as $T' = \lceil \frac{5.4 \cdot 10^{-3}}{66.7 \cdot 10^{-6}} \rceil = 80$. This value will be used later when discussing the channel prediction. The channel matrix \mathbf{H} is of size (228×14) , whereas the CIR matrix \mathbf{S} is of size (32×14) complex coefficients.

The size of DMD-based CSI with no further compression is based on the DMD rank M , as mentioned in the Section III-A. For $M = 1$ the CSI size is $32 \times 1 + 1 = 33$. Whereas, when $M = 3$ the CSI size is $32 \times 3 + 3 = 99$ complex coefficients. For the CAE network, the kernel size for all Conv layers is 3, with stride equals 2. The (in_channel, out_channel) sizes for Conv_1, Conv_2, Conv_3, and Conv_4 are (2, 16), (16, 32), (32, 16) and (32, 2), respectively. We trained the CAE network using a dataset of 5000 realizations. The training dataset is subjected to noise with SNR values ranging between 10 and 30 dB. The training process spanned 100 epochs with a batch size of 100.

B. Simulation Results

We begin by evaluating the performance of DMD and CAE independently, without any combinations between them. Fig. 4 compares the performance of DMD and CAE for varying CSI sizes, with each method indicated in the legend along with its corresponding output size, measured in complex coefficients. In this analysis, an output size of 9 coefficients corresponds to a DMD rank of $M = 1$, while an output size of 27 coefficients corresponds to $M = 3$. The CAE-based CSI size can be adjusted to match the DMD output size by tuning the FC layer parameters. The x-axis represents the channel estimation error expressed as an SNR value in dB.

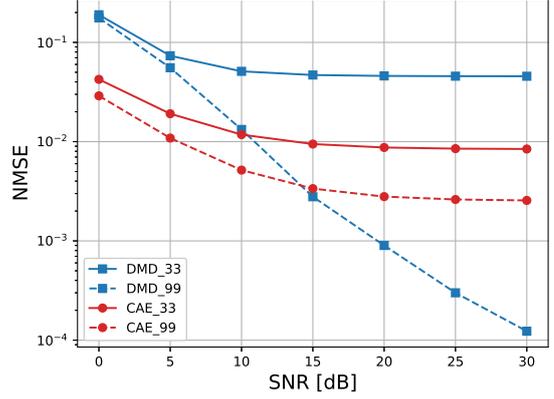


Fig. 4. Comparison of CSI compression performance for CAE and DMD, in terms of channel SNR and for two output sizes.

By integrating DMD and CAE, we achieve further reductions in CSI feedback overhead through the compression of DMD modes. Fig. 5 illustrates the NMSE performance of the proposed method across three different compression ratios of $1/4$, $1/8$, and $1/16$. The compressed modes are also subjected to noise with varying estimation errors. At lower SNR values, the NMSE for the different CRs is nearly identical, which can be attributed to the high noise levels introduced by the estimation error compared to the CAE compression error. In contrast, at higher SNR values, the NMSE performance diverges, with $CR=1/4$ demonstrating the best performance. This can be explained by the fact that the noise introduced from channel estimation is comparable to the noise added during the CAE compression process.

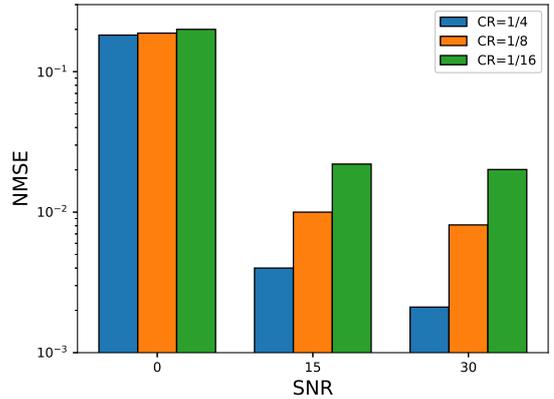


Fig. 5. Comparison of DMD modes compression performance for CAE with different CR values, in terms of channel SNR.

We now leverage the prediction capabilities of DMD to evaluate its performance after compressing the DMD modes with the CAE. When using prediction, CSI feedback is only required once the quality of the predicted channel matrix falls below a predefined threshold γ , leading to an aperiodic CSI

update, as discussed in [7]. The key point is that better prediction quality results in longer intervals between CSI updates and a reduction in the average CSI feedback overhead. In this simulation, we assume perfect channel estimation to isolate the prediction performance. Fig. 6 presents a comparison of NMSE performance for different channel prediction strategies using DMD in a time-varying channel scenario. The x-axis represents the future RB sequence. The NMSE performance is compared between Three scenarios:

- **No Prediction** (Curve ‘A’): In this scenario, the estimated channel matrix at RB#0 is simply repeated for future RBs. The performance is poor, as expected, due to channel aging effects and time-varying properties.
- **DMD Prediction without Compression** (Curve ‘B’): Here, DMD is applied to predict the future channel matrices without compressing the DMD modes. This method performs the best among all curves since since no compression error is introduced. However, prediction error accumulates over time, leading to a gradual increase in NMSE.
- **DMD Prediction with Compression** (Curves ‘C’, ‘D’ and ‘E’): These curves show the performance of DMD prediction after compressing the DMD modes with a CAE at different compression ratios, specifically CR=1/4, 1/8 and 1/16, respectively. This comparison highlights the trade-off between compression efficiency and prediction accuracy. As expected, higher CR values introduce more error, but the prediction remains relatively robust, particularly in the short term (between RB#1 and RB#3).

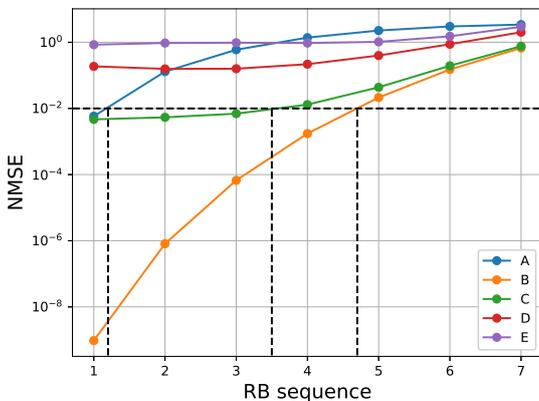


Fig. 6. Channel MSE evolution over time across different scenarios: ‘A’ denotes no prediction techniques used, instead repeating the estimated channel values. ‘B’ corresponds to the application of DMD with $M = 3$, without any additional compression. ‘C’, ‘D’, and ‘E’ illustrate the compression of DMD modes using a CAE with CRs of 1/4, 1/8, and 1/16, respectively.

Consider a system where the received CSI channel matrix $\hat{\mathbf{H}}$ must achieve an NMSE of $\gamma = 10^{-2}$, accounting for Doppler effects corresponding to a given velocity. Simply repeating the CSI (curve ‘C’) meets γ for the first future RB, while prediction without compression (curve ‘B’) ensures four future RBs meet the γ threshold without a CSI update.

Compressing DMD modes can still meet the system requirements at CR=1/4, predicting the channel for three future RBs. However, higher CR values fail to meet the required accuracy for future predictions.

V. CONCLUSION

In this work, we introduced a method that combines DMD and CAE to reduce the CSI feedback overhead in MIMO systems. The first step involves using DMD to decompose the channel matrix into modes and dynamics, with only the most dominant modes and their dynamics being utilized for CSI feedback. In the second step, CAE is employed to further compress the dimensionality of these modes. This technique effectively reduces the CSI overhead while maintaining acceptable reconstruction accuracy at the base station. Additionally, by exploiting the predictive capabilities of DMD, the frequency of CSI updates is minimized, as future channel states can be effectively predicted. This results in longer intervals between CSI feedback updates and a significant reduction in average overhead. Our simulations demonstrate that the proposed approach improves feedback efficiency, addressing the growing challenges of CSI feedback in future systems with increasing antenna arrays and bandwidth.

REFERENCES

- [1] T. Laas, J. A. Nossek, S. Bazzi, and W. Xu, “On reciprocity in physically consistent tdd systems with coupled antennas,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6440–6453, 2020.
- [2] H. Asplund, J. Karlsson, F. Kronstedt, E. Larsson, D. Astely, P. von Butovitsch, T. Chapman, M. Frenne, F. Ghasemzadeh, M. Hagström, et al., *Advanced antenna systems for 5G network deployments: bridging the gap between theory and practice*. Academic Press, 2020.
- [3] R. Ahmed, E. Visotsky, and T. Wild, “Explicit csi feedback design for 5g new radio phase ii,” in *WSA 2018; 22nd International ITG Workshop on Smart Antennas*, pp. 1–5, VDE, 2018.
- [4] C.-K. Wen, W.-T. Shih, and S. Jin, “Deep learning for massive mimo csi feedback,” *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, 2018.
- [5] S. Yao, D. Ying, and C. Li, “Autoencoder-based csi feedback with adjustable feedback payload size,” in *ICC 2024-IEEE International Conference on Communications*, pp. 3579–3585, IEEE, 2024.
- [6] G. Matz and F. Hlawatsch, “Time-varying communication channels: Fundamentals, recent developments, and open problems,” in *2006 14th European Signal Processing Conference*, pp. 1–5, IEEE, 2006.
- [7] 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.
- [8] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.
- [9] T. S. Rappaport, *Wireless communications: principles and practice*. Cambridge University Press, 2024.
- [10] G. Matz and F. Hlawatsch, “Fundamentals of time-varying communication channels,” in *Wireless communications over rapidly time-varying channels*, pp. 1–63, Elsevier, 2011.
- [11] “Etsi tr 138 901.” https://www.etsi.org/deliver/etsi_tr/138900/138999/138901/16.01.00_60/tr_138901v160100p.pdf, 2016. [Online].
- [12] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, “Stacked convolutional auto-encoders for hierarchical feature extraction,” in *Artificial Neural Networks and Machine Learning-ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*, pp. 52–59, Springer, 2011.
- [13] “HermesPy.” <https://www.barkhauseninstitut.org/en/results/hermespy>.
- [14] “ETSI TR 125 943.” <https://www.etsi.org/deliver/etsitr/125900/125999/125943/06.00.0060/tr125943v060000p.pdf>.