Adaptive Residual Vector Quantization for Dynamic Mode Decomposition-Based CSI Feedback in MIMO Systems

1st Lingrui Zhu, 2nd Fayad Haddad, 3rd Carsten Bockelmann, 4th Armin Dekorsy
Department of Communications Engineering
University of Bremen, Germany
Email: {zhu, haddad, bockelmann, dekorsy}@ant.uni-bremen.de

***Abstract*—In multiple-antenna communication systems, it is crucial for the base station to acquire accurate downlink Channel State Information (CSI) to optimize signal transmission through beamforming. However, with the absence of the channel reciprocity, the mobile station must follow the process of channel estimation with feeding the CSI back to the base station. This can introduce a substantial overhead that increases with the number of antennas and the bandwidth. Therefore the CSI must be first compressed and quantized before reporting. In this paper we introduce a novel approach that based on combining Dynamic Mode Decomposition (DMD) with Residual Vector Quantization (RVQ). RVQ adapts the quantization accuracy based on the DMD output, namely the modes. This strategy allows the system to prioritize important feedback data and reduce the overhead bits needed for less critical data. Simulation results show that our approach can reduce the CSI feedback overhead while maintaining the target channel reconstruction accuracy.**

***Index Terms*—CSI feedback, Quantization, Time-varying channels, Dynamic Mode Decomposition.**

I. Introduction

Multi-Input Multi-Output (MIMO) techniques are fundamental to modern wireless communication networks. A key aspect of MIMO transmission is providing the Base Station (BS) with precise downlink Channel State Information (CSI), crucial for effective precoding. Typically, the CSI feedback is derived at the Mobile Station (MS) from the estimated channel matrix and then transmitted back to the BS. However, as the number of antennas and/or subcarriers increases, the overhead associated with uplink CSI feedback becomes increasingly burdensome. Hence, there is a pressing need to efficiently compress and quantize the CSI before feeding it back.

CSI compression techniques encompass a variety of approaches, including Compressive Sensing (CS), matrix decomposition, and Deep Learning (DL) methods. Traditional Compressive Sensing (CS) methods exploit the assumed sparsity of the Channel Impulse Response (CIR) in the time domain, using sparse recovery techniques like Orthogonal Matching Pursuit (OMP) [1]. These techniques focus on significant CIR taps but can suffer from inefficiencies and noise vulnerability [2]. Matrix decomposition techniques like Singular Value Decomposition (SVD) reduce the channel matrix by truncating singular values but risk losing critical information. Dynamic Mode Decomposition (DMD) [3] leverages the channel temporal correlations properties, offering robust dimension reduction and predicting future channel states to reduce frequent CSI updates [4]. Deep learning frameworks, such as Autoencoders (AEs) [5] and CsiNet [6], compress the channel matrix similarly to image compression but require extensive training data and are sensitive to noise.

After compressing CSI data, the next step involves quantization, which reduces the range of the data values to a smaller set of discrete levels. These quantized values are encoded into bits for transmission. Uniform Quantization (UQ), the most basic method, evenly spaces quantization levels across the input data range. Vector Quantization (VQ) represents data by mapping vectors to discrete symbols or codewords, offering a worthy alternative. The paper [7] utilize vector quantization and Variational Autoencoder (VQ-VAE) to compress and quantize CSI information. Residual Vector Quantization (RVQ) [8], an extension of VQ, that leverages the residual information to enhance data quantization efficiency and fidelity.

This paper integrates two powerful frameworks, DMD for compression and RVQ for quantization, into an approach called RVQ-DMD. Our strategy dynamically adjusts quantization levels based on the significance of DMD output modes. Key modes receive higher quantization levels for enhanced accuracy with more bits, while less important modes are allocated lower quantization levels.

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. $\odot$ represents the Hadamard product. $E\{.\}$ denotes the mean.

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 an 4D estimated channel matrix denoted as $\bar{\mathbf{H}} \in \mathbb{C}^{K\times T\times N_t\times N_r}$. The corresponding CSI to be fed back is derived from each $\mathbf{H}_{(n_t,n_r)}[:,:,n_t,n_r]\ \forall\ n_t \in [1,...,N_t]$ and $\forall\ n_r \in [1,...,N_r]$, then quantized and reported to the BS. To facilitate explanation of our proposed method, we initially illustrate its application on a matrix channel $\mathbf{H} \in \mathbb{C}^{K\times T}$, and subsequently extend this discussion to the MIMO channel matrix $\bar{\mathbf{H}}$.

This work was funded by the German Ministry of Education and Research (BMBF) under grants 16KISK016 (Open6GHub) and 16KISK109 (6G-ANNA).

The received CSI at the BS undergoes dequantization and decompression to reconstruct the channel matrix $\hat{\mathbf{H}}$ essential for appropriate precoding of downlink user data. However, due to the compression and quantization errors, $\hat{\mathbf{H}}$ can deviate from the estimated $\mathbf{H}$, leading to a channel reporting error that dependent on the compression degree and the quantization accuracy. To evaluate performance, we employ the Normalized Mean Square Error (NMSE), defined as

$$\text{NMSE} = \frac{E\{\|\mathbf{H} - \hat{\mathbf{H}}\|_2^2\}}{E\{\|\mathbf{H}\|_2^2\}}. \quad (1)$$

B. Time-Varying Channel Model

In practical wireless mobile networks, the movement of the MS causes Doppler frequency shifts in the radiated waves, resulting in time-varying changes to the channel in the time domain [9]. One crucial parameter used to characterize these time-varying channels is the coherence time d_c. This parameter represents the duration over which the channel remains temporally correlated and can be defined as:

$$d_c = \frac{c}{2vf_c}. \quad (2)$$

Where v, f_c and c denote the MS velocity, the signal carrier frequency and the speed of light, respectively. Essentially, the Doppler shift is directly proportional to the MS velocity, thereby making the coherence time inversely proportional to the Doppler shift.

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 [10]. We define:

$$\mathbf{H} = \text{FT}(\mathbf{G}), \quad (3)$$

where, FT(.) represents the Fourier Transformation. Here $\mathbf{G} \in \mathbb{C}^{K \times T}$ denotes the channel impulse response in the time domain. exhibits sparsity along the K dimension, indicating that only a few taps are significant. The level of the sparsity S, i.e. the number of the non-zero taps, is considered to remains constant during the correlation time d_c. Additionally, in MIMO systems, it is assumed that the channel support (positions of non-zero elements) is common across all MIMO channels [11].

III. CSI compression with DMD

In this section we explain the fundamental concept of the DMD method and how it is implemented in a time-varying channel to reduce the average CSI overhead.

Dynamic Mode Decomposition [3] 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, $\mathbf{H} = [\mathbf{h}_1 \ \mathbf{h}_2 \ ... \ \mathbf{h}_T]$, with each $\mathbf{h}_t \in \mathbb{C}^{K \times 1}$ representing the channel vector at all subcarriers over the OFDM symbol t, $\forall\, t \in [1, ..., T]$. To use DMD, the channel vectors need to be arranged into two data matrices:

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

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

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

where $\mathbf{A} \in \mathbb{C}^{K \times K}$ 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 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 $\mathbf{\Phi} \in \mathbb{C}^{K \times M}$ and eigenvalues $\mathbf{\Lambda} \in \mathbb{C}^{M \times 1}$, which are called DMD modes and dynamics, respectively. It has been demonstrated in [3] that the snapshots are recomposed as:

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

Here M denotes the DMD rank truncation. It indicates the number of used eigendecompositions. Formula (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.

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 $t = T+1, T+2, \ldots$.

Since DMD is able to decompose the channel matrix into modes that capture the dominant frequencies and their growth/decay rates. And considering that the channel matrix shows sparsity in time domain, the resulting DMD modes will be sparse in the same domain, reflecting those dominant frequencies.

IV. Quantization

In this section, we introduce a quantization scheme for DMD-based CSI feedback. Although direct quantization of the DMD outputs, i.e. the modes $\mathbf{\Phi}$ is doable, we propose a further overhead reduction by exploiting the sparse nature of $\mathbf{\Phi}$ in the time domain, see II. This sparse representation,

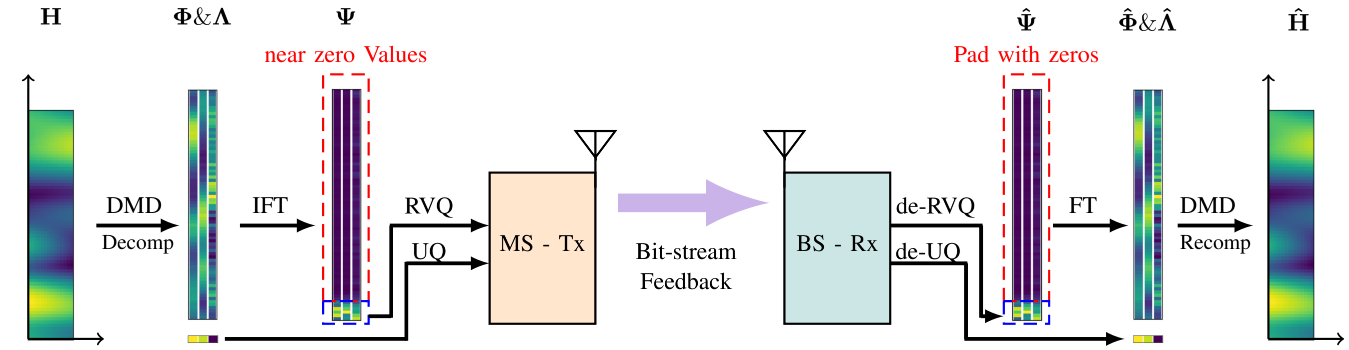

Fig. 1. The proposed framework for DMD-based CSI feedback utilizing A-RVQ

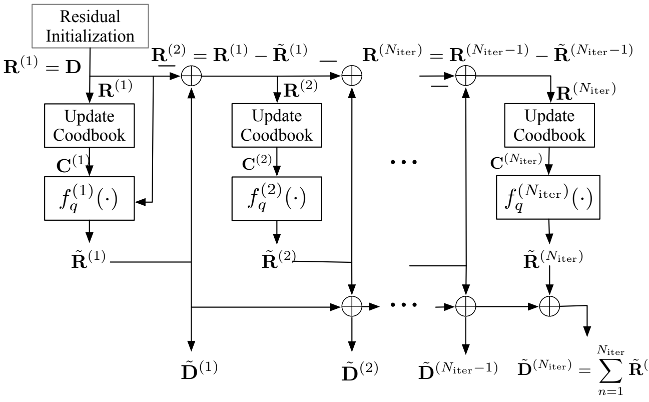
Residual Initialization
$\mathbf{R}^{(1)} = \mathbf{D}$
$\mathbf{R}^{(1)}$
Update Coodbook
$\mathbf{C}^{(1)}$
$f_q^{(1)}(\cdot)$
$\tilde{\mathbf{R}}^{(1)}$
$\mathbf{R}^{(2)} = \mathbf{R}^{(1)} - \tilde{\mathbf{R}}^{(1)}$
$\mathbf{R}^{(2)}$
Update Coodbook
$\mathbf{C}^{(2)}$
$f_q^{(2)}(\cdot)$
$\tilde{\mathbf{R}}^{(2)}$
$\mathbf{R}^{(N_{\text{iter}})} = \mathbf{R}^{(N_{\text{iter}}-1)} - \tilde{\mathbf{R}}^{(N_{\text{iter}}-1)}$
$\mathbf{R}^{(N_{\text{iter}})}$
Update Coodbook
$\mathbf{C}^{(N_{\text{iter}})}$
$f_q^{(N_{\text{iter}})}(\cdot)$
$\tilde{\mathbf{R}}^{(N_{\text{iter}})}$
$\tilde{\mathbf{D}}^{(1)}$
$\tilde{\mathbf{D}}^{(2)}$
$\tilde{\mathbf{D}}^{(N_{\text{iter}}-1)}$
$\tilde{\mathbf{D}}^{(N_{\text{iter}})} = \sum_{n=1}^{N_{\text{iter}}} \tilde{\mathbf{R}}^{(n)}$

Fig. 2. The iteration Update of Residual Vector Quantization.

denoted as $\mathbf{\Phi}_{sp} \in \mathbb{C}^{K\times M}$, is obtained through an Inverse Fourier Transformation (IFT) given by:

$$\mathbf{\Phi}_{sp} = \text{IFT}(\mathbf{\Phi}) \odot \mathbf{I}_{(\max S)}, \tag{7}$$

where $\mathbf{I}_{(\max S)} \in \mathbb{C}^{K\times M}$ is a binary matrix indicates the positions of the maximum S values in each channel $\mathbf{h}_t$. The non-zero taps in $\mathbf{\Phi}_{sp}$ then arranged in a matrix $\mathbf{\Psi} \in \mathbb{C}^{S\times M}$, such that: $\mathbf{\Psi} = \mathbf{\Phi}_{sp}[\mathbf{\Phi}_{sp} \neq 0]$. Each column in $\mathbf{\Psi}$ is called a featured mode, denoted as $\boldsymbol{\psi}$.

A. Residual Vector Quantization

We employ Residual Vector Quantization (RVQ) iteratively to quantize each feature mode separately. To ensure convergence, it's important to have a suitable dataset size N_s such as $\mathbf{D} = [\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_{N_s}]$ obtained by applying DMD on various channel matrices $\mathbf{H}$. As depicted in Fig 2, the quantization process begins by initializing the residual matrix $\mathbf{R} = [\mathbf{r}_1, ..., \mathbf{r}_{N_s}]$ as $\mathbf{R} = \mathbf{D}$.

During each iteration i, the codebook $\mathbf{C}^{(i)}$ is updated as presented in Algorithm 1. The updated $\mathbf{C}^{(i)}$ is then used to quantize the residual $\mathbf{R}$ as shown in Algorithm 2. For each residual vector $\mathbf{r}_{n_s}\, \forall n_s \in [1, ..., N_s]$, the quantization process can be represented as following:

$$\tilde{\mathbf{r}}_{n_s} = f_{\text{q}}^{(i)}(\mathbf{r}_{n_s}) = \arg \min_{\mathbf{c}_{n_c} \in \mathbf{C}^{(i)}} \|\mathbf{r}_{n_s} - \mathbf{c}_{n_c}\|_2^2, \tag{8}$$

$\forall n_c \in [1, ..., N_c]$. Here $\tilde{\mathbf{r}}$ is quantized residual vector and $f_{\text{q}}^{(i)}(\cdot)$ denotes quantization function at the i-th iteration. Accordingly we can define the quantized residual matrix as:

$$\widetilde{\mathbf{R}}^{(i)} = \left[f_{\text{q}}^{(i)}(\mathbf{r}_1), \quad f_{\text{q}}^{(i)}(\mathbf{r}_2), \quad \ldots, \quad f_{\text{q}}^{(i)}(\mathbf{r}_{N_s}) \right]. \tag{9}$$

Correspondingly, the Voronoi set of n_c-th codeword is denoted as $\mathcal{V}(\mathbf{c}_{n_c}^{(i)})$ and defined as:

$$\mathcal{V}(\mathbf{c}_{n_c}^{(i)}) = \{\mathbf{r}_{n_s} \mid \forall\, l \neq n_c \quad ||\mathbf{r}_{n_s} - \mathbf{c}_{n_c}^{(i)}||_2^2 \leq ||\mathbf{r}_{n_s} - \mathbf{c}_l^{(i)}||_2^2\}. \tag{10}$$

Codeword $\mathbf{c}_{n_c}^{(i)}$ is updated to the mean of its Voronoi set as:

$$\mathbf{c}_{n_c}^{(i)} := \frac{1}{|\mathcal{V}(\mathbf{c}_{n_c}^{(i)})|} \sum_{\mathbf{r}_{n_s} \in \mathcal{V}(\mathbf{c}_{n_c}^{(i)})} \mathbf{r}_{n_s} \tag{11}$$

The codeword can be updated iteratively until the change of codeword is below a threshold or the maximum iterations is reached.

Algorithm 1 Codeword Update at i-th iteration

```
Inputs: Residual to quantize: $\mathbf{R} = [\mathbf{r}_1, ..., \mathbf{r}_{N_s}]$.
Initialize: Codebooks $\mathbf{C}^{(i)} = [\mathbf{c}_1^{(i)}, ..., \mathbf{c}_{N_c}^{(i)}]$.
Repeat:
    Quantize the residual (Eq. (8)).
    Update Codebook according to Eq.(11)
Until Convergence conditions
Outputs: Updated codebooks $\mathbf{C}^{(i)} = [\mathbf{c}_1^{(i)}, ..., \mathbf{c}_{N_c}^{(i)}]$.
```

RVQ uses N_{iter} iterations to update codebooks $\mathbf{C}^{(i)}$, where $i = 1, ..., N_{\text{iter}}$. After each iteration, the residual between the vector before quantization and after quantization is quantized for the next iteration, as detailed in Algorithm 2.

After obtaining the updated codebooks, the featured mode $\boldsymbol{\psi}_{n_s}$ is quantized iteratively. In each iteration, only the index, denoted as n_c, of the selected codeword $\mathbf{c}_{n_c}^{(i)}$ in the codebook $\mathbf{C}^{(i)}$ is quantized. The number of iterations for the quantization process is denoted as N'_{iter}, with $N'_{\text{iter}} \leq N_{\text{iter}}$. This means that for the quantization process, it is possible to use only a portion of the codebook obtained from Algorithm 2 to quantize the data according to the system requirements. When $N'_{\text{iter}} < N_{\text{iter}}$, fewer quantization bits are used, resulting in

Algorithm 2 RVQ Codebook Update

```
Inputs: DMD modes data $\mathbf{D} = [\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_{N_s}]$.
    Number of iterations $N_{\text{iter}}$.
Initialize: Residual $\mathbf{R}$ = Dataset $\mathbf{D}$
for $i = 1$ to $N_{\text{iter}}$ do
    Update $\mathbf{C}^{(i)}$ with residual $\mathbf{R}$ (Algorithm 1).
    Obtain quantized residual $\tilde{\mathbf{R}} = f_{\text{q}}^{(i)}(\mathbf{R})$ as Eq. (8).
    Update residual $\mathbf{R} := \mathbf{R} - \tilde{\mathbf{R}}$
end for
Outputs: Updated codebooks $\mathbf{C}^{(i)}$, for $i = 1, ..., N_{\text{iter}}$.
```

suboptimal performance compared to the case where $N'_{\text{iter}} = N_{\text{iter}}$, due to the presence of unquantized residuals. When $N'_{\text{iter}} \leq N_{\text{iter}}$, the codebooks used in the quantization process are the first N'_{iter} out of N_{iter} codebooks. This approach offers a significant advantage: as long as the number of codewords N_c remains constant, quantization processes with varying numbers of iterations can utilize the same set of codebooks. This eliminates the need to retrain codebooks for each specific parameter choice, unlike VQ-VAE, which requires retraining for different parameters.

The quantized index set is defined as $\mathcal{I} = \{\iota_1, ..., \iota_{N'_{\text{iter}}}\}$. ι_i represents the selected index at i-th iteration from the codebook $\mathbf{C}^{(i)}$, ranging from 1 to N_{c}. The reconstructed featured mode can be obtained using the following equation:

$$\hat{\boldsymbol{\psi}}^{(N'_{\text{iter}})} = \sum_{i=1}^{N'_{\text{iter}}} \mathbf{c}_{\iota_i}^{(i)}, \tag{12}$$

and $\hat{\mathbf{D}}^{(N'_{\text{iter}})} = [\boldsymbol{\psi}_1^{(N'_{\text{iter}})}, ..., \boldsymbol{\psi}_{N_s}^{(N'_{\text{iter}})}]$ as depicted in Fig. 2. After getting $\hat{\boldsymbol{\psi}}$, DFT and zero-padding will be performed to get reconstruction of mode, $\hat{\boldsymbol{\phi}}$, as depicted in Fig. 1. Then, the channel matrix can be reconstructed using DMD, as descried in (6). The quantization of dynamics $\boldsymbol{\Lambda}$ is assumed to be done as a scalar quantization utilizing the uniform quantization, since the number scalars is limited.

B. Feedback Overhead and Complexity Analysis

Given the number of quantization iterations N'_{iter} and the number of codebooks for each iteration N_{c}, the number of quantization bits for single mode can be calculated as $N'_{\text{iter}} \cdot \lceil \log_2(N_{\text{c}}) \rceil$. Additionally, the CSI feedback includes the positions of non-zero elements in $\boldsymbol{\Phi}_{sp}$. These positions are defined as integer values within a range of length K, and thus can be measured with $S \cdot \lceil \log_2(K) \rceil$ bits.

For the complexity of RVQ, each iteration involves finding the nearest codeword to the current residual from N_c codewords. This results in a complexity of $\mathcal{O}(N_c \cdot S)$, where S is the dimension of residuals. Therefore, Therefore, the total complexity over N'_{iter} iterations is $\mathcal{O}(N'_{\text{iter}} \cdot N_c \cdot S)$.

C. Adaptive RVQ for DMD

Building upon the insights from DMD (see Section III), where modes exhibit varying levels of importance, we employ Adaptive RVQ (A-RVQ) with different iteration counts N'_{iter} tailored to each mode. Modes of higher significance undergo more iterations N'_{iter}, resulting in increased bit overhead but enhanced reconstruction accuracy. Conversely, less critical modes are quantized with fewer iterations, reducing overhead and sacrificing some accuracy. This approach optimizes quantization accuracy by allocating more resources to modes crucial for reconstruction while minimizing overall bit overhead.

D. Implementation in MIMO Systems

Considering the temporal correlation of channels discussed in Section III, DMD operates effectively within a correlation time d_c, leveraging the temporal properties of the channel. Furthermore, dominant modes exhibit sparsity in the time domain. Additionally, as discussed in Section II, channel sparsity remains consistent throughout d_c and across all channels in a MIMO system. To capitalize on these observations, we can reduce CSI feedback overhead by transmitting position information of non-zero taps once for all modes and channels. Instead of transmitting $Mn_tn_rS\lceil \log_2(K) \rceil$ bits to represent the CIR taps positions for the entire matrix $\bar{\mathbf{H}}$, we reduce it to $S\lceil \log_2(K) \rceil$ bits. This technique efficiently transmits the channel information while minimizing feedback redundancy.

V. Simulation results

A. Simulation Settings

In this section, we perform numerical simulations to evaluate the performance of the DMD-based CSI feedback scheme combined with the RVQ quantization scheme in an adaptive manner. The proposed method is compared with various CSI compression and quantization techniques. For the simulations, we employ Heterogenous Radio Mobile Simulator (HermesPy) [12] to generate the channel coefficients. System parameters are enumerated in Table I.

TABLE I
Simulation Parameters

System Parameters	**Value**
Channel model	COST 259 [13]
Carrier frequency f_c	2 GHz
MS velocity v	50 Km/h
Number of antenna N_t &N_r	4 &4
Subcarrier spacing	15 KHz
$\mathbf{H}$ size $K \times T$	72×14
Sparsity S	8
Channel estimating error	AWGN

B. Result and Discussion

We compare RVQ-DMD to other CSI feedback compression and quantization algorithms. The method UQ-DMD quantizes the DMD featured modes $\boldsymbol{\Psi}$ uniformly on a scalar scale. While vector quantization variational autoencoder [7] uses an autoencoder to compress the channel matrix and trains a codebook aligned with the encoder and decoder to quantize the latent variables. Additionally, we compare the proposed adaptive approach, A-RVQ-DMD, with its non-adaptive variant, namely RVQ-DMD.

In Fig. 3, the reconstruction NMSEs of channel matrix are compared. For RVQ-DMD, each iterations, the number of codewords is set to $N_{\text{c}} = 256$ (8 bits) and here we set number of iterations $N'_{\text{iter}} = 4$. The number of modes to reconstruct channel matrix $M = 4$. Therefore, the total number of quantization bits is 128 bits for each channel matrix. A-RVQ-DMD employs the same configuration as RVQ-DMD. The number of iterations for the four modes is set to 5 iterations to the first mode, 4 iterations for the second and third modes, 3 iterations for the fourth mode. Therefore, the total number of quantization bits is still 128. Similarly, the numbers of quantization bits for VQ-VAE and UQ-DMD (4 bits form each element in the four modes) are also 128. We find that DMD-based methods perform better than VQ-VAE in lower-SNR areas since DMD shows robustness against noise, as discussed III. For higher SNRs, RVQ-DMD achieves lower MSE, and A-RVQ-DMD further improves the performance.

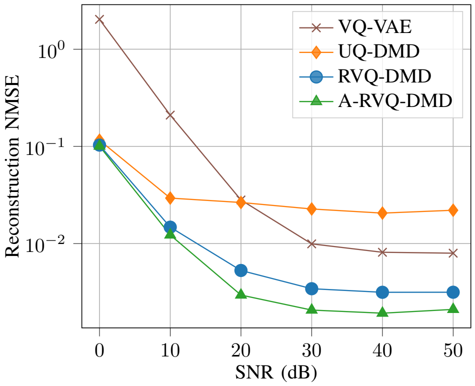

Fig. 3. Comparison of channel reconstruction NMSE for different algorithms. RVQ-DMD achieves the lower NMSE compared to VQ-VAE and UQ-DMD. A-RVQ-DMD can further improve the accuracy for high SNRs.

Figure 4 compares the RVQ and A-RVQ methods with iteration counts of $N'_{\text{iter}} = 2$ and $N'_{\text{iter}} = 4$ at SNRs of 10 dB and 40 dB. The x-axis shows the quantization bits per iteration, with 4 bits corresponding to $N_c = 16$ codewords and 8 bits to $N_c = 256$. Points marked with red circles indicate that 2 iterations with 8 bits per iteration outperform 4 iterations with 4 bits per iteration even though the total numbers of bits keep the same, the reason is that 2 iterations with 8 bits utilizes $2 \times 2^8 = 512$ codewords but 4 iterations with 4 bits uses only $4 \times 2^4 = 64$ codewords. In addition, At 10 dB, the performance difference between 2 and 4 iterations is less due to channel estimation noise, while at 40 dB, where quantization error dominates, additional iterations offer greater improvement. Thus, fewer iterations are suitable for low SNR, and more iterations are beneficial for high SNR.

VI. Conclusion

In this paper, we tackle the challenge of substantial CSI feedback overhead in MIMO systems with a novel approach that combines DMD and RVQ adaptively. Our method leverages the channel's temporal correlation and dominant mode sparsity to adjust quantization accuracy based on mode significance. This prioritizes critical feedback data and reduces bits for less important data, effectively lowering CSI feedback overhead. Simulations show our strategy maintains channel reconstruction accuracy while significantly reducing feedback, offering a promising solution for optimizing MIMO signal transmission.

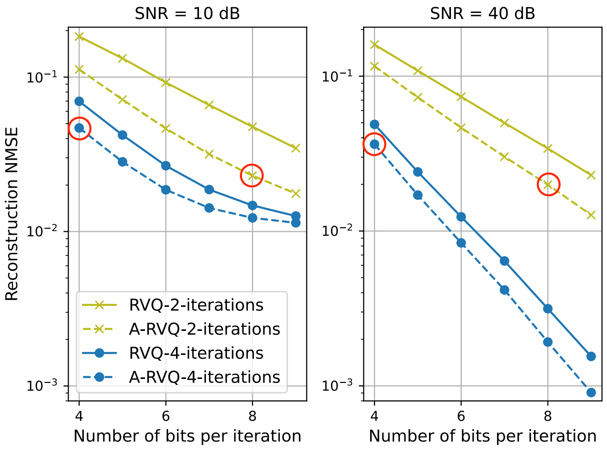

Fig. 4. Comparison of channel reconstruction NMSE for different number of iterations of RVQ-DMD and A-RVQ-DMD. As the number of iteration increasing, NMSE is reduced. A-RVQ-DMD is able to reconstruct channel matrices more accurately.

References

[1] 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.

[2] Khosravy, Mahdi, et al. ”Recovery in compressive sensing: a review.” Compressive sensing in healthcare (2020): 25-42.

[3] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, Dynamic mode decomposition: data-driven modeling of complex systems. SIAM, 2016.

[4] Haddad, Fayad, Carsten Bockelmann, and Armin Dekorsy. ”A Dynamical Model for CSI Feedback in Mobile MIMO Systems Using Dynamic Mode Decomposition.” ICC 2023-IEEE International Conference on Communications. IEEE, 2023.

[5] So, Jinhyun, and Hyukjoon Kwon. ”Universal Auto-Encoder Framework for MIMO CSI Feedback.” GLOBECOM 2023-2023 IEEE Global Communications Conference. IEEE, 2023.

[6] Wen, Chao-Kai, Wan-Ting Shih, and Shi Jin. ”Deep learning for massive MIMO CSI feedback.” IEEE Wireless Communications Letters 7.5 (2018): 748-751.

[7] Shin, Junyong, Yujin Kang, and Yo-Seb Jeon. ”Vector Quantization for Deep-Learning-Based CSI Feedback in Massive MIMO Systems.” arXiv preprint arXiv:2403.07355 (2024).

[8] Liu, Shicong, Junru Shao, and Hongtao Lu. ”Generalized residual vector quantization and aggregating tree for large scale search.” IEEE Transactions on Multimedia 19.8 (2017): 1785-1797.

[9] Rappaport, Theodore S. Wireless communications: principles and practice. Cambridge University Press, 2024.

[10] “ETSI TR 138 901“.[Online]. Available: https://www.etsi.org/deliver/etsi_tr/138900_138999/138901/16.01.00_60/tr_138901v160100p.pdf.

[11] Peng, Xingxing, et al. ”Channel estimation for extremely large-scale massive MIMO systems in hybrid-field channel.” 2023 IEEE/CIC International Conference on Communications in China (ICCC). IEEE, 2023.

[12] “HermesPy.” https://www.barkhauseninstitut.org/en/results/hermespy.

[13] “ETSI TR 125 943.” https://www.etsi.org/deliver/etsitr/125900125999/125943/06.00.0060/tr125943v060000p.pdf.