Predictive Quantization for MIMO-OFDM SVD Precoders using Reservoir Computing Framework

Agrim Gupta, Pranav Sankhe, Kumar Appaiah, Manoj Gopalkrishnan
Department of Electrical Engineering, Indian Institute of Technology Bombay
{agrim,pranavs,akumar,manojg}@ee.iitb.ac.in

***Abstract*—Precoding matrices obtained via SVD[1] of the MIMO channel matrix can be utilized at the transmitter for optimum power allocation and lower BER[2] transmissions. However, a key step enabling this improved performance is the feedback of these precoders to the transmitter from the receiver. Considering the limited bit budget for such CSI[3] feedback, the precoders need to be quantized with single digit number of bits. For a $N_T \times N_R$[4] MIMO system, this amounts to quantizing a $N_T \times N_R$ complex valued matrix. This odious task is helped by the presence of an underlying manifold structure, and temporal/frequency correlations in the precoders. In this work, we introduce a reservoir computing framework for the task of prediction of new precoders upon observation of past such precoders, by utilizing temporal correlations. This is a departure from existing methods which exploit the non linear geometry endowed by the manifold structure to perform the same. Alternately, the non-linear relation is captured in our work via the dynamical reservoir state via the training process of the reservoir. Simulations reveal reduced quantization error, which results in lower BER as well as improved achievable rate, as compared to previous work.**

I. Introduction

In MIMO wireless systems, precoding matrices are used for matrix transformation of N_S-dimensional ($N_S \leq \min(N_T, N_R)$) information vector onto the N_T-dimensional transmit vector, corresponding to the signal emanating out of the N_T antennas of the transmitter. The optimum precoding matrix, to achieve channel capacity via optimum power allocation is obtained via the SVD of the MIMO channel matrix. [1] The MIMO channel matrix is estimated at the receiver, and the transmitter has no a-priori knowledge of the same. Hence, to avail the said benefits of SVD precoders, the receiver needs to quantize and feedback the precoders it obtains from SVD of the MIMO channel matrix. To respect the bit budget imposed by limited feedback CSI schemes, the quantization needs to be performed only with a few single number of bits. Given the high dimensionality of the precoders (which form a $N_T \times N_R$ complex valued matrix, viz. $2 \times N_T \times N_R$ real numbers), effective quantization with just a few bits pose a significant challenge.

This challenge has been overcome in past work via two inherent properties of the precoders. Firstly, the precoders are not actually $2 \times N_T \times N_R$ dimensional entities, and these actually have an underlying manifold structure which allows one to work with lower dimensions if one performs operations over the manifold directly. Secondly, these precoders are correlated nicely along both time and frequency. Capturing temporal correlations allow for predictive quantization algorithms, which allow improvement of the quantization error with time. Exploiting frequency correlations allow for various interpolation algorithms, which in turn reduce the feedback overhead in a single OFDM frame, by feeding back only for certain subcarriers, and interpolating over the others. Although the underlying manifold structure allows one to work with effective lower dimensions, most of the naive linear algorithms to predict and interpolate don't work well here, owing to the inherent non-linear structure of the manifold. Hence, combining these two properties, viz. the underlying manifold structure and temporal-frequency correlations, require generalization of such linear algorithms to the manifold under consideration, by exploiting the differential geometry of the manifold. Relevant past work in this direction has been [2–8].

The manifold based approaches have provided significant performance benefits, given that they work with effective lower dimensions. However, these approaches are difficult to implement and require some familiarity with manifold geometry for a complete grasp. In this paper, we propose a reservoir computing based predictive quantizer to exploit the temporal correlations effectively. The proposed scheme offers a cleaner solution to the predictive quantization problem, which brings about both ease of implementability, and improved performance as well, when compared to [3], which proposed a manifold based method of predictive quantization.

Reservoir computing is a computational framework designed for sequential data processing. It's design takes inspiration from several frameworks of recurrent neural networks [9] including Liquid State Machines [10]. A reservoir computing system maps the inputs to spatio-temporal patterns in higher dimensional dynamical reservoir state. The pattern analysis over these spatio-temporal patterns is performed by an output coupler which maps the higher dimensional reservoir state back to the output dimension, generally having lower dimensionality than reservoir state. The major advantage of reservoir computing system is that its extremely easy to train in comparison to its counterparts like Deep Neural Networks. The input features can be efficiently read out by a simple learning algorithm because the reservoir non-linearly transforms sequential inputs into the higher dimensional reservoir state. Reservoir computing has been successfully used to solve problems like spoken digit recognition [11], handwritten digit image recognition [10] and climate prediction [12].

Reservoir computing has also been applied to solve prob-

[1] Singular Value Decomposition [2] Bit Error Rate [3] Channel State Information [4] $N_{T(R)}$: Number of Transmit (Receive) Antennas

lems in the domain of wireless communications [9, 13, 14]. [9] used reservoir computing framework to harness channel non-linearities in order for smarter channel equalization. [13, 14] utilized the same for OFDM symbol detection while tapping in channel, as well as power amplifier non-linearities. In our work, we utilize the reservoir computing framework to discover the underlying non-linear relations between previously observed precoding matrices, in order to predict the next precoder in series. Upon vectorizing the precoding matrices, this becomes analogous to time series prediction algorithms, which the reservoir computing framework has been known to work really well for [13].

II. System Model

We consider a point-to-point MIMO-OFDM wireless system that has N_T transmit antennas and N_R receive antennas. The available bandwidth is divided into N subcarriers, so that individually for each subcarrier, the channel can be considered flat fading. The transmitter communicates an $N_s \times 1$ information data vector, where $N_s \leq \min(N_T, N_R)$, and the $N_T \times N_s$ precoding matrix maps the $N_s \times 1$ information data vector onto the $N_T \times 1$ transmit vector emanating out of the transmitter.

In this discussion, we assume that $N_T > N_R$ and $N_s = N_R$. Keeping notations consistent with [2, 3], the $N_R \times 1$ data stream received is denoted by:

$$\mathbf{y}_{i,t} = \mathbf{H}_{i,t}^{H} \tilde{\mathbf{U}}_{i,t} \mathbf{x}_{i,t} + \mathbf{w}_{i,t} \tag{1}$$

Here $\tilde{\mathbf{U}}_{i,t} \in \mathbb{C}^{N_T \times N_R}$ denotes the precoding matrix which is a function of quantized CSI fed back by the receiver, $\mathbf{y}_{i,t} \in \mathbb{C}^{N_R \times 1}$ is the received data stream, $\mathbf{x}_{i,t} \in \mathbb{C}^{N_R \times 1}$ denotes the transmitted signal, all at the i-th subcarrier of the t-th OFDM frame, $\mathbf{H}_{i,t} \in \mathbb{C}^{N_T \times N_R}$ denotes the MIMO channel matrix and $\mathbf{w}_{i,t}$ denotes the i.i.d. complex Gaussian noise with $\mathbf{w}_{i,t} \sim \mathcal{N}_{\mathbb{C}}(0, N_0 \mathbf{I}_{N_R})$, N_0 being the noise variance.

Provided an infinite bit budget for quantization, the transmitter can directly use the matrices obtained from SVD of $\mathbf{H}_{i,t}^{H}$ as precoder $\tilde{\mathbf{U}}_{i,t}$. In particular, if $\mathrm{SVD}(\mathbf{H}_{i,t}) = \mathbf{U}_{i,t}, \Sigma_{i,t}, \mathbf{V}_{i,t}$, then $\tilde{\mathbf{U}}_{i,t} = \mathbf{U}_{i,t}$. Notice that matrices $\mathbf{U}_{i,t}$ reside on the Stiefel manifold $\mathrm{St}(N_T, N_R)$, since the columns of $\mathbf{U}_{i,t}$ form a set of N_R orthogonal vectors in N_T dimensions [2, 3]. Given practical limitations, only limited feedback is available from the receiver, and the objective is to find a 'reasonable' estimate of $\mathbf{U}_{i,t}$, given by $\tilde{\mathbf{U}}_{i,t}$ using the available feedback bits.

Algorithms similar to Lloyd codebook algorithm have been studied for quantizing $\mathrm{St}(N_T, N_R)$ [15]. However, this method treats precoders close-by in time/frequency to each other as independent, since it doesn't exploit any temporal/spectral correlations for quantization. Predictive quantization algorithms [2, 3] can be utilized to reduce the quantization error with time by tapping in available correlations.

We now discuss the basic mathematical model central to predictive quantization algorithms for precoding matrices. Consider that we predict based on past p observed precoding matrices, and hence $\tilde{\mathbf{U}}_{i,t-p}, \tilde{\mathbf{U}}_{i,t-p+1}, \ldots, \tilde{\mathbf{U}}_{i,t-2}, \tilde{\mathbf{U}}_{i,t-1}$ have been fed back and available at the transmitter. Using a prediction algorithm and exploiting time correlations among precoders at $t-p, t-p+1, \ldots, t-2, t-1$ time instances, the transmitter can predict $\tilde{\mathbf{U}}_{i,t}$ to be some $\mathbf{P}_{i,t} = f_P(\tilde{\mathbf{U}}_{i,t-p}, \tilde{\mathbf{U}}_{i,t-p+1}, \ldots, \tilde{\mathbf{U}}_{i,t-2}, \tilde{\mathbf{U}}_{i,t-1})$, where $f_P : \mathrm{St}(N_T, N_R)^p \rightarrow \mathrm{St}(N_T, N_R)$ is a prediction function. Now since the transmitter already knows a coarse estimate of $\tilde{\mathbf{U}}_{i,t}$ given by $\mathbf{P}_{i,t}$, the receiver just quantizes the local space of $\mathrm{St}(N_T, N_R)$ nearby $\mathbf{P}_{i,t}$ to enable a refined estimate of $\tilde{\mathbf{U}}_{i,t}$ at the transmitter. That is, the transmitter utilizes a quantization function $q_P : \mathrm{St}(N_T, N_R) \times \mathbb{N} \rightarrow \mathrm{St}(N_T, N_R)$, to obtain $\tilde{\mathbf{U}}_{i,t} \leftarrow q_P(\mathbf{P}_{i,t}, \mathrm{fb}_{i,t})$, where $\mathrm{fb}_{i,t} \in \{1, 2 \ldots 2^{\mathcal{B}}\}$ is the feedback from the receiver for precoder at i-th subcarrier and t-th time instant, considering bit budget of $\mathcal{B}$ bits. In [2–4], $\mathrm{fb}_{i,t}$ denotes the codeword index for the codebook obtained for quantizing the local tangent space at $\mathbf{P}_{i,t}$.

Since using predictive quantization scheme, quantization is on a smaller subspace as compared to the entire manifold in case of independent quantization, the quantization error is substantially lower as well. However, that being said, independent quantization algorithms are important for initialization the predictive quantization strategies. Observe that, predictive quantization algorithms for SVD precoders are typically the higher dimensional analogs of the well known approaches to reduce quantization error using Linear Prediction Codes and Delta PCM. In both LPC and Delta PCM, quantization error is reduced substantially by using a linear predictor and quantizing the scalar difference between the predicted and observed value. Analogous to this, for $\mathrm{St}(N_T, N_R)$ quantization error is reduced by quantizing a smaller subspace around predicted value, compared to the entire manifold.

In our work, we propose a reservoir computing based prediction framework, which predicts the new precoders by tapping the temporal correlations captured from past observed precoders (f_P). We reuse the q_P algorithm from [2, 3], however, by incorporating the q_P in the prediction framework, as discussed in subsequent sections, the f_P function also exploits correlations in q_P (codebook locality correlations), along with temporal correlations. Previous work [2, 3] have f_P and q_P independent of each other. The detailed discussion on reservoir computing based model for f_P and q_P is presented in Section III. The proposed reservoir computing scheme is evaluated against the temporal predictive quantization scheme in [3], and the simulation results are discussed in Section IV. Finally, Section V concludes and presents future lines of work.

III. Reservoir model for Predictive Quantization

A. Vectorizing matrices in Stiefel Manifold

The reservoir computing framework has inputs/outputs as vectors, and hence it is necessary to convert the precoder matrices to vectors. A precoding matrix $\mathbf{M} \in \mathrm{St}(N_T, N_R)$ is a $N_T \times N_R$ matrix with the property $\mathbf{M}^H \mathbf{M} = \mathbb{I}_{N_R}$. For brevity sake, let us say that such a matrix $\mathbf{M}$ is a śemi-unitaryḿatrix. A naive method of vectorizing a $N_T \times N_R$ semi-unitary matrix would be to take all it's $N_T N_R$ complex numbers individually and stack them onto a $2N_T N_R$ dimensional vector.

However, we can do this a little more efficiently, considering the $\mathbf{M}^H\mathbf{M} = \mathbb{I}_{N_R}$ property, which implies that each vector representing the columns of $\mathbf{M}$ is orthogonal to other columns. For the first column, we take the N_T complex numbers and stack them onto a $2N_T$ vector. For the second column, we take only the first $N_T - 1$ complex numbers, since the last number can be determined by utilizing orthogonality with the first vector. For the third column, we take only the first $N_T - 2$ complex numbers, and so on. Hence, we can vectorize the $N_T \times N_R$ $\mathbf{M}$ into a $2(N_T) + 2(N_T - 1) + \ldots + 2(N_T - N_R + 1) = 2N_T N_R - (N_R^2 - N_R)$ dimensional vector. For any semi-unitary matrix $\mathbf{M}$, denote by $\mathbf{m}$ the vectorized representation.

Observe here that we do pay a dimensional penalty upon vectorizing the semi-unitary $\mathbf{M}$. $\mathbf{M} \in \mathrm{St}(N_T, N_R)$, which is a $2N_T N_R - N_R^2$ dimensional manifold. We vectorize $\mathbf{M}$ to a $2N_T N_R - N_R^2 + N_R$ dimensional vector, an overhead of N_R dimensions, since we didn't exploit the fact that each column vector is a norm 1 vector as well. It looks as if we could just take $N_T - 1$ complex numbers when vectorizing the first column, instead of N_T, but this would lead to a non-unique representation, since there could then be infinite possibilities of the missing one complex number, such that the vector has a norm 1.

B. Forward Prediction f_P function

The vector representation of semi-unitary matrices allow us to proceed with the discussion on reservoir computing framework for predictive quantization of SVD precoders. Consider the reservoir computing framework as in Fig.

The input to the reservoir i corresponding to subcarrier i, is $\tilde{\mathbf{u}}_{i,t-1}$, which is the vectorized representation of $\tilde{\mathbf{U}}_{i,t-1}$. Let $D_{\text{in}} = 2N_T N_R - N_R^2 + N_R$ be the dimension of the input vector to the dimension. Using a randomly initialized $D_{\text{in}} \times D_{\text{resv}}$ input coupler matrix $\mathbf{W}_i^{\text{in}}$, the D_{in} dimensional input is mapped to a D_{resv} dimensional vector, where D_{resv} is the dimension of the reservoir state vector $\mathbf{r}_{i,t}$. The reservoir state vector $\mathbf{r}_{i,t}$ is initialized with D_{resv} zeros, and is updated via the following equation,

$$\mathbf{r}_{i,t} = \tanh(\mathbf{A}_i \mathbf{r}_{i,t-1} + \mathbf{W}_i^{\text{in}} \tilde{\mathbf{u}}_{i,t-1}) \tag{2}$$

where $\mathbf{A}_i$ is the adjacency matrix which captures the reservoir dynamics and correlates reservoir state across time, and tanh is applied element wise. Typical choices for $\mathbf{A}_i$ have been erdos-renyi graphs with an upper bounded maximum eigen value [12, 13].

The output of the reservoir is $\mathbf{p}_{i,t}$, a vectorized representation of $\mathbf{P}_{i,t}$. Hence, the output of the reservoir is also a D_{in} dimensional vector ($D_{\text{out}} = D_{\text{in}}$). The output $\mathbf{p}_{i,t}$ is obtained via matrix transformation of the reservoir state $\mathbf{r}_{i,t}$ via the $D_{\text{resv}} \times D_{\text{out}}$ output coupler matrix $\mathbf{W}_{i,t-1}^{\text{out}}$[5], viz.

$$\mathbf{p}_{i,t} = \mathbf{W}_{i,t-1}^{\text{out}} \mathbf{r}_{i,t} \tag{3}$$

This completes the forward prediction algorithm to obtain $\mathbf{P}_{i,t}$ using just one past observed $\mathbf{U}_{i,t-1}$. That is,

$$\mathbf{P}_{i,t} = f_P(\tilde{\mathbf{U}}_{i,t-1}) = \mathbf{W}_{i,t-1}^{\text{out}} \tanh(\mathbf{A}_i \mathbf{r}_{i,t-1} + \mathbf{W}_i^{\text{in}} \tilde{\mathbf{u}}_{i,t-1})^6 \tag{4}$$

However, a complete discussion of the prediction algorithm also involves a backward pass, involving the updation of output couplers $\mathbf{W}_{i,t}^{\text{out}}$, which would follow subsequently.

C. Quantization function q_P

Before proceeding ahead, it is critical to detail out the quantization function $q_P(\mathbf{P}_{i,t}, \text{fb}_{i,t})$, which is similar to the one used in [2–4]. Central to the quantization algorithm is the fact that tangent spaces local to a point in manifold, are actually linear vector spaces. Given two points $\mathbf{X}$ and $\mathbf{Y}$ in $\mathrm{St}(N_T, N_R)$, a lifting operation $\mathbf{T}_X^Y = \texttt{lift}(\mathbf{X}, \mathbf{Y})$, $\texttt{lift} : \mathrm{St}(N_T, N_R) \times \mathrm{St}(N_T, N_R) \to \mathcal{T}_X \mathrm{St}(N_T, N_R)$ gives a tangent from X to Y, $\mathbf{T}_X^Y \in \mathcal{T}_X \mathrm{St}(N_T, N_R)$, where $\mathcal{T}_X \mathrm{St}(N_T, N_R)$ is the local tangent space at $\mathbf{X}$. A corresponding retraction operation $Y = \texttt{retract}(X, \mathbf{T}_X^Y)$, $\texttt{retract} : \mathrm{St}(N_T, N_R) \times \mathcal{T}_X \mathrm{St}(N_T, N_R) \to \mathrm{St}(N_T, N_R)$ gives back the manifold point obtained by traversing in the tangent direction given by the second argument. In this work, the chosen lifting-retraction operations are the Cayley Exponentials, elaborated in [16] and used in [2]. The Cayley exponential lifting operation maps two points in $\mathrm{St}(N_T, N_R)$, to a $N_T \times N_T$ skew hermitian matrix representing the tangent from first point to the other.

The quantization algorithm exploits the linear vector space property of the tangent space and quantizes the local tangent space at the predicted precoder $\mathbf{P}_{i,t}$, viz $\mathcal{T}_{\mathbf{P}_{i,t}} \mathrm{St}(N_T, N_R)$. The codebook for $\mathcal{T}_{\mathbf{P}_{i,t}} \mathrm{St}(N_T, N_R)$ basically corresponds to a collection of codewords representing the different directions (tangents) in $\mathcal{T}_{\mathbf{P}_{i,t}} \mathrm{St}(N_T, N_R)$. This codebook is also referred to as the base codebook (base^C) for the quantization function. The feedback from the receiver indicates the optimum tangent in the base^C, which the transmitter can choose to get closest (in terms of chordal distance metric, $d_s(\mathbf{X}, \mathbf{Y})$ for $\mathbf{X}, \mathbf{Y} \in \mathrm{St}(N_T, N_R)$ detailed in [2, 3]) to the actual value $\mathbf{U}_{i,t}$. With the optimum direction chosen, the next step is to determine how much to move in that particular direction, viz. the length of the chosen tangent direction. For this, we adopt the strategy in [3], which controls the magnitude of tangent steps, by having two codebooks T_p^C, T_m^C of different spreads s_p, s_m, but same $2^{\mathcal{B}-1}$ base vectors from base^C, i.e. $T_{\{p/m\}}^C = s_{\{p/m\}} \text{base}^C$ (All codewords $\in \text{base}^C$ multiplied by $s_{\{p/m\}}$ individually). The two codebooks are concatenated to form a $2^{\mathcal{B}}$ length codebook T^C. The receiver finds the optimal index $\text{fb}_{i,t} \in \{1, 2, \ldots, 2^{\mathcal{B}}\}$ in T^C by comparing the chordal distance metric (d_s), of each codeword to the actual precoder $\mathbf{U}_{i,t}$ obtained from the SVD of the channel matrix, using (5). The receiver then feeds back $\text{fb}_{i,t}$ to the transmitter using $\mathcal{B}$ bits. The transmitter uses the fed back $\text{fb}_{i,t}$ and (6) to calculate $\tilde{\mathbf{U}}_{i,t}$,

$$\text{fb}_{i,t} \leftarrow \text{argmin}_{i \in \{1,2,\ldots,2^{\mathcal{B}}\}} \left(d_s \left(\mathbf{U}_{i,t}, \text{retract}(\mathbf{P}_{i,t}, T^C[i]) \right) \right) \tag{5}$$

$$\tilde{\mathbf{U}}_{i,t} = q_P(\mathbf{P}_{i,t}, \text{fb}_{i,t}) = \text{retract}(\mathbf{P}_{i,t}, T^C[\text{fb}_{i,t}]) \tag{6}$$

$\text{fb}_{i,t}$ is also used to update the spread of the codebooks

[5] Why it is $t-1$ and not t would surface up in Section III-D

[6] For Brevity sake, $\mathbf{p}_{i,t}$ and $\mathbf{P}_{i,t}$ are treated to be one and the same

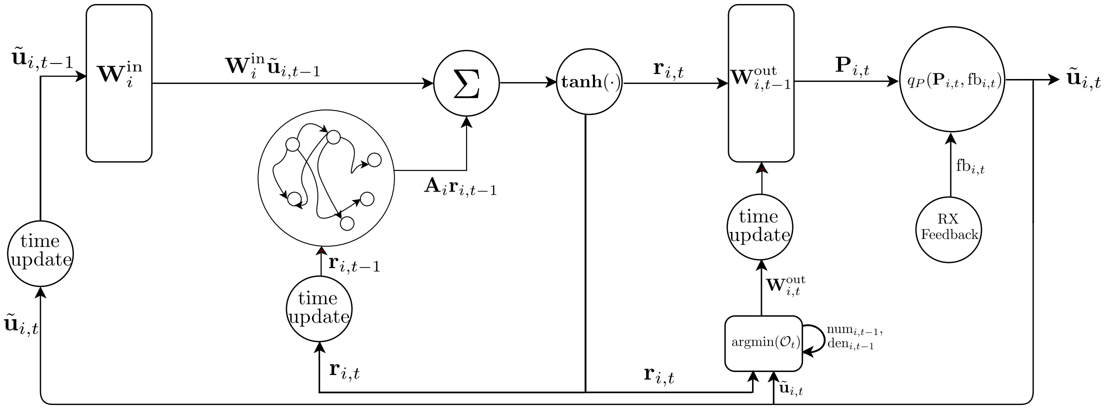
$\tilde{\mathbf{u}}_{i,t-1}$
$\mathbf{W}_i^{\text{in}}$
$\mathbf{W}_i^{\text{in}}\tilde{\mathbf{u}}_{i,t-1}$
$\sum$
tanh(·)
$\mathbf{r}_{i,t}$
$\mathbf{W}_{i,t-1}^{\text{out}}$
$\mathbf{P}_{i,t}$
$q_P(\mathbf{P}_{i,t}, \text{fb}_{i,t})$
$\tilde{\mathbf{u}}_{i,t}$
$\mathbf{A}_i\mathbf{r}_{i,t-1}$
$\text{fb}_{i,t}$
RX Feedback
time update
time update
time update
$\mathbf{r}_{i,t-1}$
$\mathbf{W}_{i,t}^{\text{out}}$
$\tilde{\mathbf{u}}_{i,t}$
argmin($\mathcal{O}_t$)
$\text{num}_{i,t-1}$, $\text{den}_{i,t-1}$
$\mathbf{r}_{i,t}$
$\mathbf{r}_{i,t}$
$\tilde{\mathbf{u}}_{i,t}$

Fig. 1: An overview of the reservoir computing framework proposed for predictive quantization of $\mathbf{u}_{i,t}$

T_p^C, T_m^C. Depending on whether $\text{fb}_{i,t} \geq 2^{\mathcal{B}-1}$, i.e. whether the optimum codeword is in T_p^C or T_m^C, the scale parameter $s[k]$, which in turn controls values of s_p, s_m, is updated in the following manner,

$$s_p = g^{\min(s[k-1]+1,0)}, s_m = g^{s[k-1]-1}$$

$$s[k] = \begin{cases} \min(s[k-1]+1, 0), & \text{for } \text{fb}_{i,t} \in T_p^C \\ s[k-1]-1, \text{otherwise} \end{cases}$$

with $s[0] = 0$. Intuitively, the algorithm reduces/increases the spread of the codebook till the operation of reduction/increase is no longer beneficial, i.e. the optimum codeword lies in the higher/lower spread codebook instead. The scheme used by us to obtain base^C is similar to the one presented in [2], which performs a k-means ($k = 2^{\mathcal{B}-1}$) clustering on a collection of tangents to obtain an isotropic collection of $2^{\mathcal{B}-1}$ tangent codewords. This completes the discussion on the quantization function q_P to obtain $\tilde{\mathbf{U}}_{i,t} = q_P(\mathbf{P}_{i,t}, \text{fb}_{i,t})$ from $\mathbf{P}_{i,t}, \text{fb}_{i,t}$.

D. Reservoir Training Procedure, the backward pass

Armed with the descriptions of f_P, q_P we are now ready to describe the training process for the reservoir. Recall that to obtain $\tilde{\mathbf{U}}_{i,t}$ from $\tilde{\mathbf{U}}_{i,t-1}$, we have,

$$\tilde{\mathbf{U}}_{i,t} = q_P(\mathbf{P}_{i,t}, \text{fb}_{i,t}) = q_P(f_P(\tilde{\mathbf{U}}_{i,t-1}), \text{fb}_{i,t}) \quad (7)$$

The key idea here is that as the predicted matrix $\mathbf{P}_{i,t}$, which is basically the coarse estimate for $\mathbf{U}_{i,t}$, gets closer to the refined estimate $\tilde{\mathbf{U}}_{i,t}$, the receiver has to quantize even smaller subspaces and can thus provide an even refined estimate via it's feedback $\text{fb}_{i,t}$. Hence, for the time instant $t \geq 1$, we train the reservoir output coupler such that the following objective function is optimized,

$$O_t = \sum_{s=1}^{t} \frac{1}{\lambda^{t-s}} ||\mathbf{p}_{i,t} - \tilde{\mathbf{u}}_{i,t}||^2 = \sum_{s=1}^{t} \frac{1}{\lambda^{t-s}} ||W_{i,t}^{\text{out}}\mathbf{r}_{i,t} - \tilde{\mathbf{u}}_{i,t}||^2 \quad (8)$$

here $\lambda > 1$ is the history parameter. We wish to perform the following optimization in order to find the optimum $\mathbf{W}_{i,t}^{\text{out}}$ from whatever we have observed about $\tilde{\mathbf{u}}_{i,t}$ uptill now,

$$W_{i,t}^{\text{out}} \leftarrow \text{argmin}_{W_{i,t}^{\text{out}}}(O_t = \sum_{s=1}^{t} \frac{1}{\lambda^{t-s}} ||W_{i,t}^{\text{out}}\mathbf{r}_{i,t} - \tilde{\mathbf{u}}_{i,t}||^2) \quad (9)$$

Computing the gradients $\frac{\partial O_t}{\partial W_{i,t}^{\text{out}}}$ and setting it to null gives,

$$W_{i,t}^{\text{out}} = \frac{\sum_{s=1}^{t} \frac{1}{\lambda^{t-s}} \texttt{outer}(\tilde{\mathbf{u}}_{i,s}, \mathbf{r}_{i,s})}{\sum_{s=1}^{t} \frac{1}{\lambda^{t-s}} ||\mathbf{r}_{i,s}||^2} \quad (10)$$

where $\texttt{outer}(\mathbf{x}, \mathbf{y})$ is the outer product between N_x dimensional vector $\mathbf{x}$ and N_y dimensional vector $\mathbf{y}$ to give a $N_x \times N_y$ matrix. Let $\sum_{s=1}^{t} \frac{1}{\lambda^{t-s}} \texttt{outer}(\tilde{\mathbf{u}}_{i,s}, \mathbf{r}_{i,s})$ be $\text{num}_{i,t}$ and $\sum_{s=1}^{t} \frac{1}{\lambda^{t-s}} ||\mathbf{r}_{i,s}||^2$ be $\text{den}_{i,t}$, with $W_{i,t}^{\text{out}} = \frac{\text{num}_{i,t}}{\text{den}_{i,t}}$. Observe that,

$$W_{i,t}^{\text{out}} = \frac{\frac{\text{num}_{i,t-1}}{\lambda} + \texttt{outer}(\tilde{\mathbf{u}}_{i,t}, \mathbf{r}_{i,t})}{\frac{\text{den}_{i,t-1}}{\lambda} + ||\mathbf{r}_{i,t}||^2} \quad (11)$$

(11) makes training of the reservoir easy to implement, and of low complexity as well. We just need to store the matrix $\text{num}_{i,t-1}$, a number $\text{den}_{i,t-1}$, and using the current reservoir state $\mathbf{r}_{i,t}$ and obtained $\tilde{\mathbf{u}}_{i,t}$ from (7), one can calculate $W_{i,t}^{\text{out}}$ for forward calculation of $\tilde{\mathbf{u}}_{i,t+1}$. This completes our technical discussion for the complete reservoir computing based framework for predictive quantization illustrated in Fig.

To conclude this section, now summarize and touch upon the novel points of the reservoir computing based framework proposed for predictive quantization of SVD precoding. The first key advantage of the proposed framework, over the past work in this line [2–4] is that the prediction function and quantization function are not independent of each other. To support the above statement, observe (2),(7) to obtain $\mathbf{W}_{i,t}^{\text{out}}$ from $\tilde{\mathbf{u}}_{i,t}$, and (4) to obtain $\mathbf{P}_{i,t+1}$ combinedly,

$$W_{i,t}^{\text{out}} = \frac{\frac{\text{num}_{i,t-1}}{\lambda} + \texttt{outer}(q_P(f_P(\tilde{\mathbf{u}}_{i,t-1}), \text{fb}_{i,t}), \mathbf{r}_{i,t})}{\frac{\text{den}_{i,t-1}}{\lambda} + ||\mathbf{r}_{i,t}||^2}$$

$$\mathbf{P}_{i,t+1} = f_P(\tilde{\mathbf{U}}_{i,t}) = \mathbf{W}_{i,t}^{\text{out}} \tanh(\mathbf{A}_i\mathbf{r}_{i,t} + \mathbf{W}_i^{\text{in}}\tilde{\mathbf{u}}_{i,t})$$

The dependence between f_P and q_P in our framework is in the sense that the backward pass of the framework ensures that prediction function ‘’learns’ the correlations brought about by q_P, in addition to temporal correlations. This is done by optimizing $W_{i,t}^{\text{out}}$ such that the prediction output to be as close to the output from q_P (Objective function O_t). The updated $W_{i,t}^{\text{out}}$ is then used in forward pass to predict $\mathbf{P}_{i,t+1}$, which is hopefully closer to $\tilde{\mathbf{U}}_{i,t+1}$ than $\mathbf{P}_{i,t}$ was to $\tilde{\mathbf{U}}_{i,t}$ due to optimization of O_t.

The second key advantage of the proposed scheme comes from the ease of training the reservoir, entailed by (2). By storing just one matrix $\text{num}_{i,t}$ and a number $\text{den}_{i,t}$, we can capture the enitre history of the precoding matrices, with past values weighted by λ. This is a departure from previous schemes [2–4] which store the past n (which has to be pre-decided) precoders in a n sized cyclic buffer kind-of arrangement. Reservoir computing, thus proposes an easier to implement data centric scheme, which is a rarity for typical data centric ML based schemes.

IV. Simulation Results

A. Simulation setting considered

We compare the results of the proposed reservoir computing framework with the predictive quantization scheme presented in [3], which captures the temporal correlations to enable reduced quantization error. We simulate both our predictive quantization algorithms and the one presented in [3] for the IEEE Pedestrian-A channel, with $N_T = 4$, $N_R = 2$. We simulated our algorithms by using the IT++ library through the Python wrapper py-itpp [17], with the channels being generated using Jakes model for realistic simulations. Although in both our work and [3], only temporal correlations are exploited, we compare the BER, quantization error and logsum rate results for the MIMO-OFDM setting with number of subcarriers $N = 64$, and evenly spaced 8 feedback subcarriers, indexed from 0, viz. $\{0, 9, \ldots, 63\}$ ($\{9k, k \in \{0, 1, \ldots, 7\}\}$). The precoding matrices at non fedback subcarriers ($i \neq \{9k\}$) are estimated via the Cayley Interpolation method presented in [2]. Feedback is available for $i = \{9k\}$ subcarriers (i.e. transmitter obtains $\text{fb}_{\{9k\},t}$) for predictive quantization. The setting is visually illustrated in Fig. BER, quantization error results have been simulated for $f_D T_s = 3.5 \times 10^{-4}$, whereas for achievable rate we compare results for $f_D T_s = 10^{-4}, 10^{-2}, 10^{-1}$.

B. Quantization Error, BER and Achievable Rate Results

Considering mitigation against randomly appearing drastic events in wireless channels, like sudden interferences, and dynamic environment, only 100 channel evolutions are considered for the simulations.

V. Conclusions and Future Work

References

[1] D. J. Love, R. W. Heath, V. K. Lau, D. Gesbert, B. D. Rao, and M. Andrews, “An overview of limited feedback in wireless communication systems,” *IEEE J. Sel. Areas Commun.*, vol. 26, no. 8, 2008.

[2] A. Gupta, K. Appaiah, and R. Vaze, “Predictive quantization and joint Time-Frequency interpolation technique for MIMO-OFDM precoding,” in *2019 IEEE International Conference on Communications (ICC): Wireless Communications Symposium (IEEE ICC’19 - WC Symposium)*, Shanghai, P.R. China, May 2019.

[3] S. Schwarz and M. Rupp, “Predictive Quantization on Stiefel Manifold,” *IEEE Signal Process. Lett.*, vol. 22, no. 2, pp. 234–238, Feb 2015.

[4] S. Schwarz, R. W. Heath, and M. Rupp, “Adaptive Quantization on a Grassmann-Manifold for Limited Feedback Beamforming Systems,” *IEEE Trans. Signal Process.*, vol. 61, no. 18, pp. 4450–4462, Sept 2013.

[5] Y. Chou and T. Sang, “Efficient interpolation of precoding matrices in MIMO-OFDM systems,” in *SPAWC, 2010*, June 2010, pp. 1–4.

[6] T. Li, F. Li, and C. Li, “Manifold-based predictive precoding for the time-varying channel using differential geometry,” *Wireless Networks*, vol. 22, no. 8, pp. 2773–2783, Nov 2016.

[7] N. Khaled, B. Mondal, R. W. Heath, G. Leus, and F. Petré, “Quantized multi-mode precoding for spatial multiplexing MIMO-OFDM systems,” in *IEEE 62nd Vehic. Technol. Conf.*, vol. 2. IEEE, 2005, pp. 867–871.

[8] R. T. Krishnamachari and M. K. Varanasi, “On the geometry and quantization of manifolds of positive semi-definite matrices,” *IEEE Trans. Signal Process.*, vol. 61, no. 18, pp. 4587–4599, 2013.

[9] H. Jaeger and H. Haas, “Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication,” *science*, vol. 304, no. 5667, pp. 78–80, 2004.

[10] A. Jalalvand, G. Van Wallendael, and R. Van de Walle, “Real-time reservoir computing network-based systems for detection tasks on visual contents,” in *2015 7th International Conference on Computational Intelligence, Communication Systems and Networks*. IEEE, 2015, pp. 146–151.

[11] D. Verstraeten, B. Schrauwen, D. Stroobandt, and J. Van Campenhout, “Isolated word recognition with the liquid state machine: a case study,” *Information Processing Letters*, vol. 95, no. 6, pp. 521–528, 2005.

[12] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, “Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 12, p. 121102, 2017.

[13] S. Mosleh, L. Liu, C. Sahin, Y. R. Zheng, and Y. Yi, “Brain-inspired wireless communications: Where reservoir computing meets mimo-ofdm,” *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–15, 2017.

[14] R. Shafin, L. Liu, J. Ashdown, J. Matyjas, M. Medley, B. Wysocki, and Y. Yi, “Realizing green symbol detection via reservoir computing: An energy-efficiency perspective,” in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[15] R. Pitaval and O. Tirkkonen, “Joint Grassmann-Stiefel Quantization for MIMO Product Codebooks,” *IEEE Trans. Wireless Commun.*, vol. 13, no. 1, pp. 210–222, January 2014.

[16] R. Chakraborty and B. C. Vemuri, “Statistics on the (compact) Stiefel manifold: Theory and Applications,” *CoRR*, vol. abs/1708.00045, 2017. [Online]. Available: http://arxiv.org/abs/1708.00045

[17] V. Saxena, https://github.com/vidits-kth/py-itpp.