Sparse Matrix Codes: Rate-Reliability Trade-offs for URLLC

Sudarshan Adiga Ravi Tandon Tamal Bose
Department of Electrical and Computer Engineering
University of Arizona, Tucson, AZ, USA
E-mail: {*adiga, tandonr, tbose*}@email.arizona.edu

***Abstract*—In this paper, we present a new channel coding technique, namely sparse matrix codes (SMC), for URLLC applications with the goal of achieving higher reliability, and low decoding complexity. The main idea behind SMC is to map the message bits to a *structured* sparse matrix which is then multiplied by a spreading matrix and transmitted over the communication channel over time-or frequency resources. At the decoder, we recover the message from the channel output using a low-decoding complexity algorithm which is derived by leveraging and adapting tools from 2D compressed sensing. We perform various experiments to compare our approach with sparse vector code (SVC) and Polar codes for block error rate (BLER). From our experiments, we show that for a fixed code rate and reliability requirement (BLER), SMC operates at shorter blocklengths compared to Polar codes and SVC.**

I. Introduction

The goal of ultra-reliable and low latency communications (URLLC) is to support applications with low-latency (sub ms) and high-reliability requirements (order of 0.9999) [1], [2]. While LDPC, Polar, and convolutional codes are widely used and deployed in contemporary communication systems, they do not necessarily meet all the constraints required by URLLC [3]. In this paper, we thus focus on two enablers for URLLC: (i) low-complexity coding schemes to minimize latency, (ii) utilization of time, spatial, and frequency diversity techniques to enhance reliability [4].

A recent interesting work [5] proposed Sparse Vector Coding (SVC), in which message bits are encoded to a sparse vector and the locations of the non-zero entries represent the message. This sparse vector is spread into multiple resources allocated in time or frequency using a spreading sequence/matrix. At the receiver, the decoding operation involves the identification of the nonzero locations in the received signal. Therefore, compressed sensing recovery algorithms such as orthogonal matching pursuit (OMP) [6], and multipath matching pursuit (MMP) [7] can be leveraged to design low complexity decoding techniques. For low-rate regimes, it was shown in [5] that SVC outperforms polar codes in terms of block error rate (BLER). In another line of works [8], [9], Barron and Joseph propose Sparse Regression Codes (SPARC) for efficient communication over additive white Gaussian noise channels. The main idea behind SPARC is mapping the messages to sparse vectors, such that each vector can be segmented into multiple sections of equal lengths. Furthermore, every section must satisfy the constraint of having exactly one non-zero entry. Then, the codeword transmitted is the linear combination of columns of the spreading (or design) matrix. A wide variety of decoders that are asymptotically capacity-achieving have been proposed for SPARC in [10]–[13]. Moreover, the decoding objective in SPARC, also motivated by principles of compressed sensing (e.g., LASSO [14]), draws parallels with the proposed methodologies used in SVC.

To provide further capacity gains and enhance reliability, one can exploit spatial diversity using multiple-input multiple-output (MIMO). Advantages offered when SVC is used in MIMO systems have been studied in [15], in which the authors demonstrate better decoding reliability with an increase in the number of transmission antennas. In this paper, we explore a generalization of SVC by mapping the message bits to a structured sparse matrix. This in turn provides more flexibility for both encoding/decoding as well as for exploiting other resources for diversity (such as MIMO).

Main Contributions: We introduce sparse matrix codes (SMC), a novel coding technique in which every message is mapped uniquely to a sparse matrix with a fixed number of non-zero rows and columns. Furthermore, we derive a low-complexity decoding algorithm for SMC. The key aspect of this decoding algorithm is that the non-zero rows and columns in the sparse matrix can be recovered in-parallel. From our experimental results, we show that SMC operates at shorter blocklengths compared to Polar codes and SVC [5] for a fixed code-rate and reliability. Our second contribution involves the exploration of MIMO with SMC for enhancing reliability by utilizing spatial diversity.

Notations: Vectors and matrices are notated in bold lowercase and bold uppercase symbols, respectively. $\lfloor \cdot \rfloor$ denotes the floor operation. $|| \cdot ||$ denotes the l_2 norm. $\mathbf{0}$ denotes all-zero column vector. The message vector $\mathbf{m}$ is assumed to be a row vector. $\det(\mathbf{X})$ denotes the determinant of $\mathbf{X}$. $\mathbf{x_i}$ denotes the i^{th} column of $\mathbf{X}$, unless stated explicitly. $[x]$ denotes the set $\{1, 2, \cdots, x\}$; and $[x]\backslash[y]$ denotes the set of all elements in $[x]$ not in $[y]$. (M, N)-MIMO system refers to M antennas at the transmitter and N antennas at the receiver.

This work was partially supported by the Broadband Wireless Access and Applications Center (BWAC); and NSF grants CNS-1715947, CAREER 1651492 and CCF-2100013.

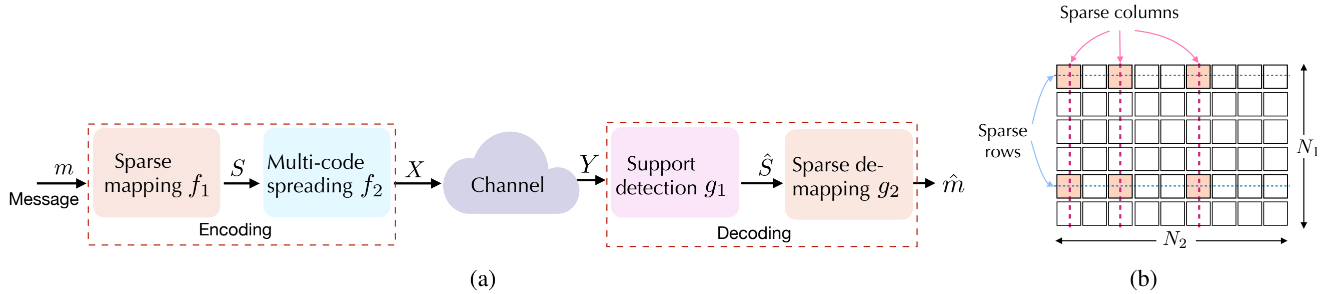


Fig. 1: (a) End-to-End block diagram for communication using sparse codes (vector/matrix) [5], (b) Example sparse matrix for parameters $N_1 = 6$, $N_2 = 9$, $K_1 = 2$, $K_2 = 3$.

II. System Model and Problem Formulation

We consider a single-user communication system as illustrated in Fig. 1(a). The codebook at the transmitter is designed as follows: each message vector $\mathbf{m} \in \mathbb{C}^{M\times 1}$ is mapped to a structured sparse matrix $\mathbf{S} \in \mathbb{C}^{N_1\times N_2}$, i.e., $\mathbf{S} = f_1(\mathbf{m})$, for some integers M, N_1, N_2, where $f_1(\cdot)$ denotes the sparse matrix mapping. We restrict $\mathbf{S}$ to be a sparse matrix with K_1 non-zero rows and K_2 non-zero columns. The input to the communication channel $\mathbf{X}$ is then designed as $\mathbf{X} = f_2(\mathbf{S}) = \mathbf{AS}$, where $\mathbf{X} \in \mathbb{C}^{L\times N_2}$, where, $\mathbf{A} \in \mathbb{C}^{L\times N_1}$ is a spreading matrix used to spread $\mathbf{S}$ into multiple time or frequency resources. The entries of $\mathbf{A}$ are sampled from either a Gaussian or Bernoulli distribution. From the compressed sensing literature [6], it has been shown that the columns of $\mathbf{A}$ are orthogonal with high probability, and this property allows us to design efficient decoding approaches inspired by the compressed sensing recovery algorithms. The receiver receives the channel output $\mathbf{Y}$ corrupted with additive white Gaussian noise $\mathbf{Z} \in \mathbb{C}^{L\times N_2}$. Specifically, the receiver receives $\mathbf{Y} = \mathbf{H}f_2(\mathbf{S}) + \mathbf{Z}$, where $\mathbf{Y} \in \mathbb{C}^{L\times N_2}$. $\mathbf{H} \in \mathbb{C}^{L\times L}$ is the channel matrix, which is an identity matrix when there is no fading; and in presence of fading the entries of $\mathbf{H}$ are assumed to be i.i.d across time and sampled from Rayleigh distribution with an identity covariance matrix. For sparse matrix codes (SMC), the block-length (ρ) and the code-rate (γ) are as follows:

$$\rho = LN_2 \qquad \gamma = \frac{\left\lfloor \log_2\left(\binom{N_1}{K_1}\binom{N_2}{K_2}\right)\right\rfloor}{LN_2} \tag{1}$$

The goal of the receiver is to estimate $\mathbf{m}$ using the channel output. The receiver comprises of support detection block $g_1(\cdot)$ which outputs $\hat{\mathbf{S}}$ (an estimate of $\mathbf{S}$) from $\mathbf{Y}$. Our goal is to design a low-complexity decoder g_1 for SMC. Subsequently, we obtain $\hat{\mathbf{m}}$ (an estimate of $\mathbf{m}$) by sparse de-mapping, such that $\hat{\mathbf{m}} = g_2(\hat{\mathbf{S}})$. The performance of any code is characterized by its rate and reliability (measured by the block error rate, $\Pr(\hat{\mathbf{m}} \neq \mathbf{m})$).

III. Sparse Matrix Codes

In this section, we present the encoding and decoding algorithms for SMC.

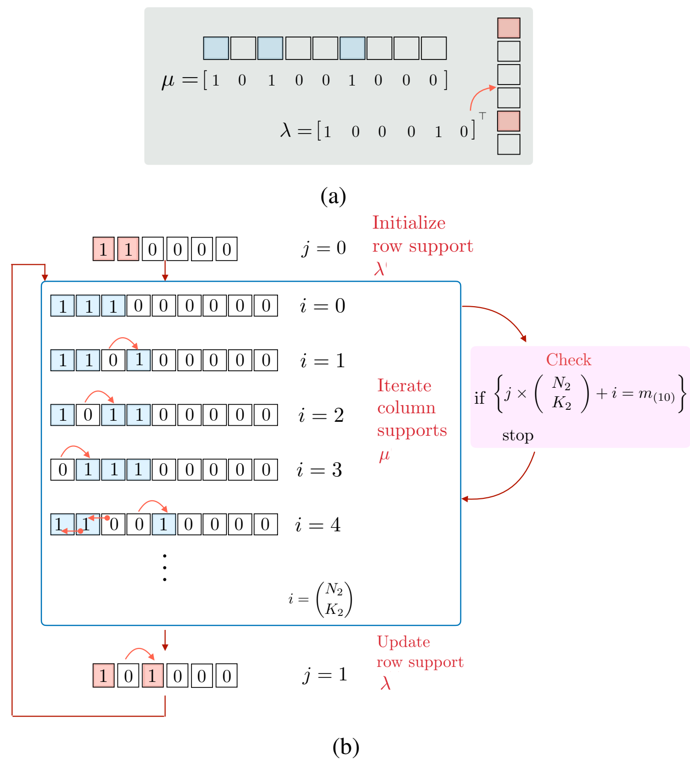


Fig. 2: (a) Row support ($\boldsymbol{\lambda}$) and column support ($\boldsymbol{\mu}$) vector for sparse matrix codes depicted in Fig. 1(b); (b) Algorithm to generate the codebook by mapping every message $\mathbf{m}$ to a unique structured sparse matrix $\mathbf{S}$.

A. Sparse Mapping

We now describe the sparse mapping function $f_1(\cdot)$ that maps each message $\mathbf{m}$ to a structured sparse matrix $\mathbf{S}$. For SMC with parameters N_1, K_1, N_2, K_2; we can encode $\lfloor \log_2\left(\binom{N_1}{K_1}\binom{N_2}{K_2}\right)\rfloor$ bits of information. We denote the row- and column-support of $\mathbf{S}$ by $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, respectively. Here, $\boldsymbol{\lambda}$ is a column vector and $\boldsymbol{\mu}$ is a row vector. Furthermore, we notate the set of indices of non-zero entries in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ by $\mathcal{C}$ and $\mathcal{R}$, respectively.

We first initialize $\boldsymbol{\lambda}$ by assigning exactly K_1 entries to 1, and the remaining $N_1 - K_1$ entries to 0. Likewise, $\boldsymbol{\mu}$ is initialized by assigning K_2 entries to 1, and the remaining $N_2 - K_2$ entries to 0. The sparse mapping algorithm generates the sparse matrices by iterating through all possible $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. That is, for each bit shift in $\boldsymbol{\lambda}$, we iterate through all possible $\boldsymbol{\mu}$. The cardinality of the set of all possible $\boldsymbol{\mu}$ given N_2, K_2 is $\binom{N_2}{K_2}$. We repeat this until the sum of bit-shift operations in both $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ equals $\mathbf{m}_{(10)}$ (the value of $\mathbf{m}$ in base-10). This yields $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ of the sparse matrix $\mathbf{S}$ that corresponds to message $\mathbf{m}$. The sparse mapping algorithm for SMC for parameters $N_1 = 6$, $K_1 = 2$, $N_2 = 9$, $K_2 = 3$ is illustrated in Fig. 2(b).

B. Sparse Matrix Recovery

We now derive the recovery algorithm for SMC to recover $\mathbf{m}$ from the channel output $\mathbf{Y} = [\mathbf{y_1}\ \mathbf{y_2} \cdots \mathbf{y_{N_2}}]$. We first express the channel output $\mathbf{Y}$ as a function of the columns in $\mathbf{S}$ and $\mathbf{Z}$ denoted by $\mathbf{s_i}$ and $\mathbf{z_i}$, respectively; where $i = \{1, 2, \cdots, N_2\}$.

$$\mathbf{Y} = \left[\boldsymbol{\phi}\mathbf{s_1} + \mathbf{z_1} \quad \boldsymbol{\phi}\mathbf{s_2} + \mathbf{z_2} \ \cdots \ \boldsymbol{\phi}\mathbf{s_{N_2}} + \mathbf{z_{N_2}}\right] \tag{2}$$

where, $\boldsymbol{\phi} = \mathbf{HA}$. Consider the set of indices of non-zero rows $\mathcal{R} = \{r_1, r_2, \cdots, r_{K_1}\}$ and the set of indices of non-zero columns $\mathcal{C} = \{c_1, c_2, \cdots, c_{K_2}\}$. Then, we have $\mathbf{s_i} = \boldsymbol{\lambda}$, $\forall i \in \mathcal{C}$, and $\mathbf{s_i} = \mathbf{0}$, $\forall i \in [N_2]\backslash\mathcal{C}$. The channel output $\mathbf{Y}$ in (2) transforms to:

$$\mathbf{y_j} = \begin{cases} \sum\limits_{i \in \mathcal{R}} \boldsymbol{\phi_i} + \mathbf{z_j}, & j \in \mathcal{C} \\ \mathbf{z_j}, & \text{otherwise} \end{cases} \tag{3}$$

Our proposed recovery algorithm leverages the form of channel output expressed in (3) to recover the non-zero rows and columns of $\mathbf{S}$. The recovery algorithm comprises of: 1) Initialization, 2) Column recovery, 3) Row recovery; and we next describe these steps.

1) *Initialization*: We obtain the initial estimates of $\mathcal{R}$, $\mathcal{C}$ using the fact that the columns of $\boldsymbol{\phi}$ are orthogonal to each other with high probability (this follows from the fact that columns of $\boldsymbol{A}$ are orthogonal to each other with high probability). Then, on multiplying every column in $\mathbf{Y}$ with the columns of $\boldsymbol{\phi}$ normalized, we obtain $\hat{\mathbf{S}}_{\text{ini}}(i,j) = \boldsymbol{\phi_i}^\top \mathbf{y_j}/||\boldsymbol{\phi_i}||^2$, such that,

$$\hat{\mathbf{S}}_{\text{ini}}(i,j) = \begin{cases} 1 + \dfrac{\boldsymbol{\phi_i}^\top}{||\boldsymbol{\phi_i}||^2}\mathbf{z_j}, & i \in \mathcal{R}, j \in \mathcal{C} \\ \dfrac{\boldsymbol{\phi_i}^\top}{||\boldsymbol{\phi_i}||^2}\mathbf{z_j}, & \text{otherwise} \end{cases} \tag{4}$$

where, i, j are the indices of the columns in $\boldsymbol{\phi}$ and $\mathbf{Y}$, respectively. $\hat{\mathbf{S}}_{\text{ini}}(i,j)$ denotes the entry corresponding to the i^{th} row and j^{th} column of matrix $\hat{\mathbf{S}}_{\text{ini}} \in \mathbb{C}^{N_1 \times N_2}$. (4) shows that the indices in $\hat{\mathbf{S}}_{\text{ini}}$ with large values correspond to the sparse entries in $\mathbf{S}$. From $\hat{\mathbf{S}}_{\text{ini}}$, we obtain $\hat{\boldsymbol{\lambda}}$ and $\hat{\boldsymbol{\mu}}$ (also $\hat{\mathcal{C}}$ and $\hat{\mathcal{R}}$), which are the estimates of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, respectively.

2) *Column recovery*: We now use maximum likelihood estimation to update our initial estimate $\hat{\mathcal{C}}$. In other words, we choose $\hat{\mathcal{C}}$ (an estimate of $\mathcal{C}$) that maximizes the probability of the channel output $\mathbf{Y}$. That is,

$$\hat{\mathcal{C}} = \arg\max_{\bar{\mathcal{C}}} P(\mathbf{Y}|\bar{\mathcal{C}}) \tag{5}$$

The channel noise $\mathbf{z_j}$ in (3) is L-dimensional Gaussian distributed as $\mathbf{z_j} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, and is i.i.d across time. Here, $\boldsymbol{\Sigma}$ denotes the covariance matrix. Likewise, $\forall j \in \mathcal{C}$, we have $\boldsymbol{\phi}\mathbf{s_j} + \mathbf{z_j} \sim \mathcal{N}(\boldsymbol{\phi\lambda}, \boldsymbol{\Sigma})$. Therefore, one can obtain $\hat{\mathcal{C}}$ when we maximize $P(\mathbf{Y}|\bar{\mathcal{C}})$ which is written as follows,

$$P(\mathbf{Y}|\bar{\mathcal{C}}) = \prod_{i \in \bar{\mathcal{C}}} \frac{e^{-\frac{1}{2}(\mathbf{y_i} - \boldsymbol{\phi}\hat{\boldsymbol{\lambda}})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y_i} - \boldsymbol{\phi}\hat{\boldsymbol{\lambda}})}}{\sqrt{(2\pi)^L \det(\boldsymbol{\Sigma})}} \prod_{j \in [N_2]\backslash\bar{\mathcal{C}}} \frac{e^{-\frac{1}{2}(\mathbf{y_j})^\top \boldsymbol{\Sigma}^{-1}\mathbf{y_j}}}{\sqrt{(2\pi)^L \det(\boldsymbol{\Sigma})}} \tag{6}$$

On further simplifying (6), and substituting in (5), $\hat{\mathcal{C}}$ can be obtained by solving the following:

$$\arg\max_{\bar{\mathcal{C}}} \log P(\mathbf{Y}|\bar{\mathcal{C}}) = \arg\min_{\bar{\mathcal{C}}} \sum_{i \in \bar{\mathcal{C}}} \left(\tilde{\mathbf{y}}_{\mathbf{i}}^\top \boldsymbol{\Sigma}^{-1} \tilde{\mathbf{y}}_{\mathbf{i}} - \mathbf{y_i}^\top \boldsymbol{\Sigma}^{-1} \mathbf{y_i}\right) \tag{7}$$

where, $\tilde{\mathbf{y}}_{\mathbf{i}} = \mathbf{y_i} - \boldsymbol{\phi}\hat{\boldsymbol{\lambda}}$. That is, we can obtain $\hat{\mathcal{C}}$ by simply picking K_2 of the N_2 columns in $\mathbf{Y}$ which yields lower values of the term $\tilde{\mathbf{y}}_{\mathbf{i}}^\top \boldsymbol{\Sigma}^{-1} \tilde{\mathbf{y}}_{\mathbf{i}} - \mathbf{y_i}^\top \boldsymbol{\Sigma}^{-1} \mathbf{y_i}$, for $i \in [N_2]$. The most interesting aspect of (7) is its low decoding-complexity which is utmost $\mathcal{O}(N_2 \log N_2)$. We next use $\hat{\mathcal{C}}$ to update $\hat{\mathcal{R}}$ which is the estimate of $\mathcal{R}$.

3) *Row recovery*: For recovering $\mathcal{R}$, one can in principle leverage any sparse vector recovery algorithm using one of the following two approaches:

a) *Aggregate Decoding:* In this approach, we directly apply sparse vector recovery on $\mathbf{y_a}$ which is the concatenation of columns in $\mathbf{Y}$ corresponding to $\hat{\mathcal{C}} = \{\hat{c}_1, \hat{c}_2, \cdots, \hat{c}_{K_2}\}$, such that,

$$\mathbf{y_a} = \begin{bmatrix} \mathbf{y_{\hat{c}_1}} \\ \mathbf{y_{\hat{c}_2}} \\ \vdots \\ \mathbf{y_{\hat{c}_{K_2}}} \end{bmatrix} \tag{8}$$

b) *Reduced Variance Decoding:* Sparse vector recovery is applied on $\mathbf{y}_{\text{av}}$, which is the average across entries of columns in $\mathbf{Y}$ whose indices correspond to the set $\hat{\mathcal{C}}$. That is,

$$\mathbf{y}_{\text{av}} = \frac{1}{K_2} \sum_{j \in \hat{\mathcal{C}}} \mathbf{y_j} \tag{9}$$

This method provides better reliability due to the power gain of the combined symbols as demonstrated for SVC in [5]. Some of the known approaches for vector recovery are orthogonal matching pursuit (OMP) [6], multi-path matching pursuit (MMP) [7], Compressive sampling (CoSaMP) [16], Basis pursuit [16], Iterative hard thresholding (IHT) [17].

Remark 1. *Impact of the sequence of steps in Algorithm 1: Since we can get the initial estimates $\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\lambda}}$ in Step 1, we can also perform row-recovery in step 2 and column recovery in step 3. We study the impact of ordering of row-and column-recovery in Algorithm 1 on decoding performance in Section IV.*

Remark 2. *Choice of row-recovery approach: The reduced variance decoding offers two advantages over aggregate decoding because: (i) it has a lower decoding complexity; (ii)* $\mathbf{y}_{av}$ *has a lower noise variance. In Section IV we discuss the impact of choice of the row-recovery approach on the overall performance of our proposed approach.*

Remark 3. *We note that the initialization step in Algorithm 1 is computationally cheap and is independent of the number of non-zero rows and columns in* $\mathbf{S}$*. However, the performance of this step relies heavily on the coherence*[1] *of the matrix* ϕ *being small and is therefore not the most effective way of row-and column-recovery for SMC.*

Algorithm 1 Sparse Matrix Recovery.

Input to the algorithm: $\mathbf{Y}$, ϕ, N_1, N_2, K_1, K_2

1. Initialization (generate $\hat{\mathbf{S}}_{\text{ini}}$, an initial estimate $\mathbf{S}$)

for i = 1 to number of columns in ϕ **do**

  for j = 1 to number of columns in $\mathbf{Y}$ **do**

    Compute:

$$\hat{\mathbf{S}}_{\text{ini}}(i,j) = \frac{\boldsymbol{\phi_i}^\top \mathbf{y_j}}{||\boldsymbol{\phi_i}||^2}$$

  end for

end for

Threshold entries in $\hat{\mathbf{S}}_{\text{ini}}$ to determine initial $\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\lambda}}$.

2. Column recovery Determine $\hat{\mathcal{C}}$ by solving:

$$\arg\min_{\bar{\mathcal{C}}} \sum_{i \in \bar{\mathcal{C}}} \left(\tilde{\mathbf{y}}_{\mathbf{i}}^\top \boldsymbol{\Sigma}^{-1} \tilde{\mathbf{y}}_{\mathbf{i}} - \mathbf{y_i}^\top \boldsymbol{\Sigma}^{-1} \mathbf{y_i} \right)$$

3. Row recovery Apply vector recovery algorithm on either of the following: (a) $\mathbf{y_a}$, (b) $\mathbf{y}_{\text{av}}$.

Enhancing reliability using MIMO for URLLC: We consider a (M_1, M_2)-MIMO system with M_1 antennas at the transmitter and M_2 antennas at the receiver as shown in Fig. 3. We have channel input designed as $\mathbf{X} = f_2(\mathbf{S})$, where $\mathbf{S} \in \mathbb{C}^{N_1 \times N_2}$, $\mathbf{X} \in \mathbb{C}^{M_1 L \times N_2}$. The channel output $\mathbf{Y} \in \mathbb{C}^{M_2 L \times N_2}$ at the receiver is:

$$\underbrace{\begin{bmatrix} \mathbf{Y_1} \\ \mathbf{Y_2} \\ \vdots \\ \mathbf{Y_L} \end{bmatrix}}_{=\mathbf{Y}} = \underbrace{\begin{bmatrix} \mathbf{H_1} & & & \\ & \mathbf{H_2} & & \\ & & \ddots & \\ & & & \mathbf{H_L} \end{bmatrix}}_{=\mathbf{H}} \underbrace{\begin{bmatrix} \mathbf{X_1} \\ \mathbf{X_2} \\ \vdots \\ \mathbf{X_L} \end{bmatrix}}_{=\mathbf{X}} + \underbrace{\begin{bmatrix} \mathbf{Z_1} \\ \mathbf{Z_2} \\ \vdots \\ \mathbf{Z_L} \end{bmatrix}}_{=\mathbf{Z}} \tag{10}$$

where, $\forall i \in [L]$, $\mathbf{X_i} \in \mathbb{C}^{M_1 \times N_2}$ and $\mathbf{Y_i} \in \mathbb{C}^{M_2 \times N_2}$. $\mathbf{Z_i} \in \mathbb{C}^{M_2 \times N_2}$ is the additive noise at the receiver, and $\mathbf{H_i} \in \mathbb{C}^{M_2 \times M_1}$ is the channel matrix. We directly apply Algorithm 1 to recover the row and column supports of $\mathbf{S}$.

In the next section, we present a comprehensive set of experiments to study how the performance (i.e., decoding success) of SMC scales with an increase in transmitting as well as receiving antennas.

[1]Coherence of matrix ϕ is defined as $\max_{i \neq j} \frac{|\langle \boldsymbol{\phi_i}, \boldsymbol{\phi_j} \rangle|}{||\boldsymbol{\phi_i}||||\boldsymbol{\phi_j}||}$

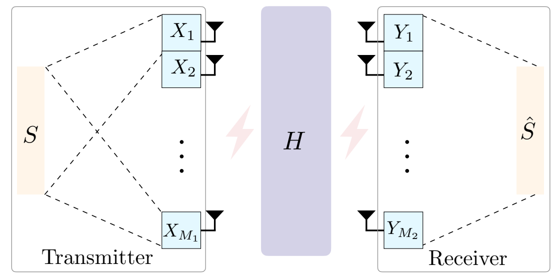
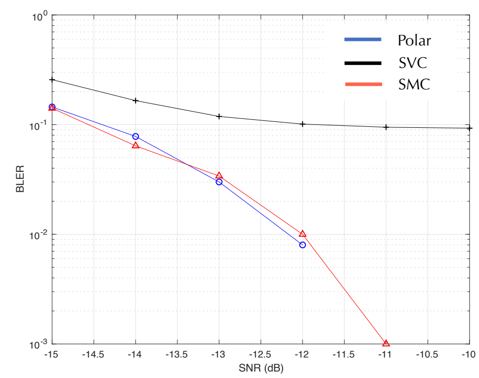
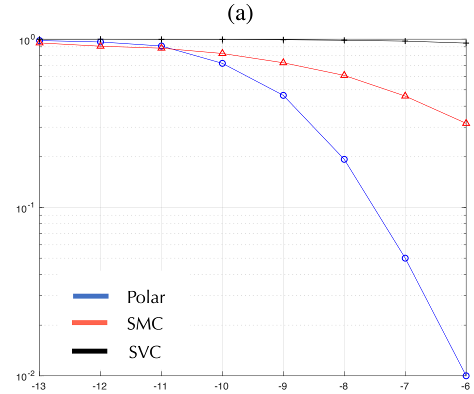


Fig. 3: SMC in (M_1,M_2)-MIMO system for URLLC.

IV. Experiments



(a)



(b)

Fig. 4: Performance comparison of SMC with Polar and SVC. Shorter block-lengths are used for SMC compared to Polar and SMC in both (a), (b), but at the same rate. In plot (a) we set $\rho_{\text{svc}} = 1024$, $\rho_{\text{polar}} = 1024$ and $\rho_{\text{smc}} = 800$. In (b) we set $\rho_{\text{svc}} = 128$, $\rho_{\text{polar}} = 256$ and $\rho_{\text{smc}} = 104$.

A. Performance comparison with other codes

To show the advantages offered by the proposed sparse matrix codes (SMC), we first compare with polar codes. In Fig. 4(a), we use the code rate $\gamma \approx 0.01$ across all codes. The block-lengths for SVC, and Polar codes are $\rho_{\text{svc}} = 1024$, and $\rho_{\text{polar}} = 1024$, respectively. For SMC, the block-length is $\rho_{\text{smc}} = 800$.

The parameters of SMC are set to $N_1 = 72$, $K_1 = 1$, $N_2 = 4$, $K_2 = 2$; and for SVC the length of the sparse vector $N = 72$ with $K = 2$. The performance of SVC is evaluated for 2 repetitions. For polar codes, we set information block-length to 11.

We use multi-path matching pursuit (MMP) [7] with reduced-variance approach for row-recovery in SMC. For SVC, we directly apply multi-path matching pursuit algorithm to recover the sparse entries from the channel output. *As seen from Fig. 4(a), SMC gives similar performance to Polar codes at a given rate but at shorter block-lengths.* In addition, SMC outperforms SVC in terms of block error rate across all SNRs. In the second result presented in Fig. 4(b), we set $\rho_{\text{smc}} = 104$, $\rho_{\text{polar}} = 256$, and $\rho_{\text{svc}} = 128$. The code rate $\gamma \approx 0.10$ for both the codes. The parameters of SMC are set to $N_1 = 512$, $K_1 = 1$, $N_2 = 4$, $K_2 = 2$. For SVC, we consider $N = 16$ and $K = 8$. For polar codes, the information length is 12 with CRC length 16. From these experiments, we note that the SMC operates at much shorter block-lengths.

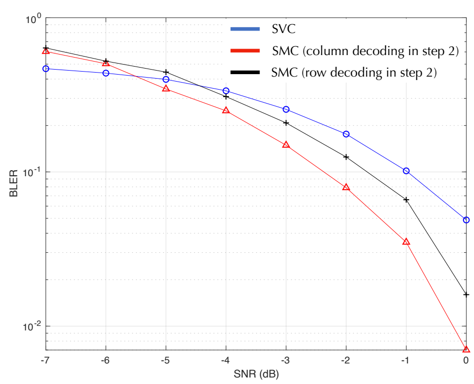
Remark 4. *From the experimental results, we note that the performance of SMC is better than Polar codes for smaller SNR, particularly with shorter block-lengths but at the same code-rate. SMC outperforms SVC with 2 repetitions (used to get better power gain of the combined symbol [5] at half the code-rate). Furthermore, we note that SMC can be viewed as a principled method to combining repetition coding (i.e., along the non-zero columns) with SVC without compromising on the code-rate significantly.*

Next, we study the impact of: (i) ordering of steps in Algorithm 1, (ii) different row-recovery approaches, on the performance of the recovery algorithms for SMC.

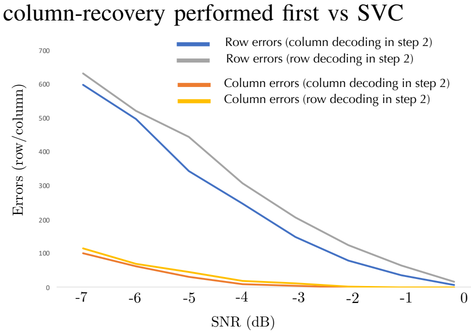
B. Performance analysis of Algorithm 1

For the scope of analysis of performance of SMC we set $N_1 = 8$, $K_1 = 4$, $N_2 = 4$, $K_2 = 2$. We use SVC with $N = 15$, $K = 9$. In addition, for SVC we use 2 repetitions to obtain enhanced power gains as demonstrated in [5]. The block-lengths of SMC, SVC are $\rho_{\text{smc}} = 120$, $\rho_{\text{svc}} = 180$, respectively. The code rate $\gamma = 0.0667$ for both the codes. We use the MMP algorithm [7] for the row-recovery in SMC and vector recovery in SVC. As seen from Fig. 5(a), we note that Algorithm 1 yields a lower block error rate when column recovery is performed in step 2. In addition, for this approach the row-errors and column-errors are lower as seen in Fig. 5(b).

In Fig. 6(a), we use the same parameters of SMC as Fig. 5. We observe that the performance of Algorithm 1 is better when a reduced variance row recovery approach is applied.

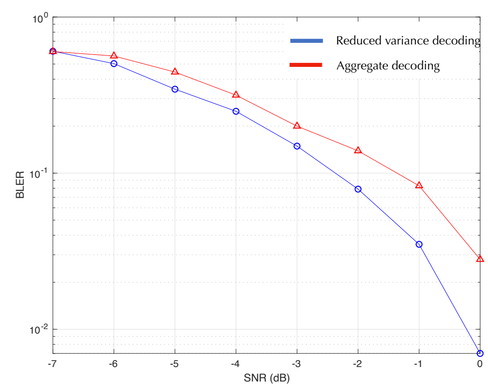


(a) SMC with row-recovery performed first and column-recovery performed first vs SVC

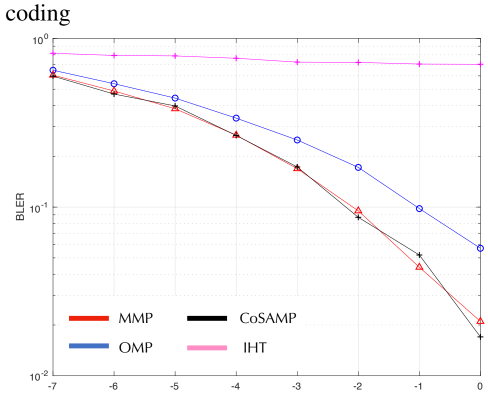


(b) Row-recovery and column-recovery errors in SMC with column-and row-recovery vs SNR.

Fig. 5: Impact of performing row-recovery first vs column recovery first (i.e. in step 2 of Algorithm 1).



(a) Reduced-variance decoding vs Aggregate decoding



(b) Performance of MMP [7], OMP [6], CoSAMP [16], and IHT [17] for row-recovery.

Fig. 6: Impact of choice of row-recovery approach on decoding performance.

In Fig. 6(b), we note that the performance of the recovery algorithm is similar when MMP [7] or CoSAMP [16], and this outperforms the block error rate obtained using OMP [6] or IHT [17]. However, there is more flexibility with MMP in terms of decoding (i.e., selection of child nodes) [7] .

C. Enhancing Reliability using MIMO

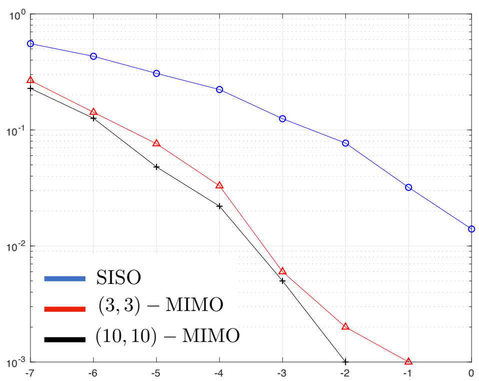


Fig. 7: Leveraging spatial diversity with MIMO to enhance reliability of SMC.

We consider $(M_1,\ M_2)$-MIMO system on Rayleigh fading channel. We vary both the transmitting antennas M_1 and the receiving antennas M_2. We consider SMC with $N_1 = 8$, $K_1 = 4$, $N_2 = 4$, $K_2 = 2$ with code rate $\gamma = 0.0667$ and block-length $\rho_{\text{smc}} = 120$. As demonstrated in Fig. 7, we note that spatial diversity using multiple antennas at both the transmitter and receiver helps enhance the reliability compared to the single-input single-output (SISO) setting.

V. Discussions and Future Directions

We propose a novel coding technique in SMC that provides a gain in channel utilization per message at a fixed rate and reliability requirement. To the best of our knowledge, this is the first work exploring the use of sparse matrices for transmission in the scope of URLLC. We also show the applicability of our approach in the MIMO setting and demonstrated enhancements in reliability obtained using multiple antennas at both transmitter and the receiver. There are several directions for future work, and some of them are listed next.

One can explore adaptations of SMC to enhance its rate. The message can be encoded in both position and symbols in the sparse matrix similar to the work in [18] for SVC. We know that incorporating the advantages of modulation techniques to enhance reliability has been studied in [19]. One can easily adopt the same for SMC to get better decoding reliability. Furthermore, the use of deep-learning techniques to perform row-and column-recovery remains to be explored. The efficacy of deep learning techniques has already been demonstrated in [20] for SVC.

References

[1] M. Alsenwi, N. H. Tran, M. Bennis, A. K. Bairagi, and C. S. Hong, "eMBB-URLLC Resource Slicing: A Risk-Sensitive Approach," *IEEE Communications Letters*, vol. 23, no. 4, pp. 740–743, 2019.

[2] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View," *IEEE Access*, vol. 6, pp. 55 765–55 779, 2018.

[3] M. Shirvanimoghaddam, M. S. Mohammadi, R. Abbas, A. Minja, C. Yue, B. Matuz, G. Han, Z. Lin, W. Liu, Y. Li *et al.*, "Short Block-Length Codes for Ultra-Reliable Low Latency Communications," *IEEE Communications Magazine*, vol. 57, no. 2, pp. 130–137, 2018.

[4] P. Popovski, Č. Stefanović, J. J. Nielsen, E. De Carvalho, M. Angjelichinoski, K. F. Trillingsgaard, and A.-S. Bana, "Wireless Access in Ultra-Reliable Low-Latency Communication (URLLC)," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5783–5801, 2019.

[5] H. Ji, S. Park, and B. Shim, "Sparse Vector Coding for Ultra Reliable and Low Latency Communications," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6693–6706, 2018.

[6] J. A. Tropp and A. C. Gilbert, "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[7] S. Kwon, J. Wang, and B. Shim, "Multipath Matching Pursuit," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2986–3001, 2014.

[8] A. Joseph and A. R. Barron, "Least Squares Superposition Codes of Moderate Dictionary Size Are Reliable at Rates up to Capacity," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 2541–2557, 2012.

[9] A. Joseph and A. Barron, "Fast Sparse Superposition Codes Have Near Exponential Error Probability for $\mathcal{R} < \mathcal{C}$," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 919–942, 2013.

[10] S. Cho and A. Barron, "Approximate Iterative Bayes Optimal Estimates for High-Rate Sparse Superposition Codes," in *Sixth Workshop on Information-Theoretic Methods in Science and Engineering*, 2013.

[11] A. R. Barron and S. Cho, "High-rate sparse superposition codes with iteratively optimal estimates," in *2012 IEEE International Symposium on Information Theory Proceedings*. IEEE, 2012, pp. 120–124.

[12] R. Venkataramanan, S. Tatikonda, and A. Barron, "Sparse Regression Codes," *arXiv preprint arXiv:1911.00771*, 2019.

[13] J. Barbier and F. Krzakala, "Approximate Message-Passing Decoder and Capacity Achieving Sparse Superposition Codes," *IEEE Transactions on Information Theory*, vol. 63, no. 8, pp. 4894–4927, 2017.

[14] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[15] R. Zhang, B. Shim, Y. Lou, S. Jia, and W. Wu, "Sparse Vector Coding Aided Ultra-Reliable and Low-Latency Communications in Multi-User Massive MIMO Systems," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 1019–1024, 2020.

[16] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic Decomposition by Basis Pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[17] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[18] W. Kim, S. K. Bandari, and B. Shim, "Enhanced Sparse Vector Coding for Ultra-Reliable and Low Latency Communications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5698–5702, 2020.

[19] X. Zhang, D. Zhang, B. Shim, G. Han, D. Zhang, and T. Sato, "Sparse Superimposed Coding for Short Packet URLLC," *IEEE Internet of Things Journal*, 2021.

[20] W. Kim, H. Ji, H. Lee, Y. Kim, J. Lee, and B. Shim, "Sparse Vector Transmission: An Idea Whose Time Has Come," *IEEE Vehicular Technology Magazine*, vol. 15, no. 3, pp. 32–39, 2020.