Blind Unwrapping of Modulo Reduced Gaussian Vectors: Recovering MSBs from LSBs

Elad Romanov
School of CSE
Hebrew University of Jerusalem, Israel
elad.romanov@mail.huji.ac.il

Or Ordentlich
School of CSE
Hebrew University of Jerusalem, Israel
or.ordentlich@mail.huji.ac.il

Abstract—We consider the problem of recovering $n$ i.i.d.
samples from a zero mean multivariate Gaussian distribu-
tion with an unknown covariance matrix, from their
modulo wrapped measurements, i.e., measurement where
each coordinate is reduced modulo $\Delta$, for some $\Delta > 0$.
For this setup, which is motivated by quantization and
analog-to-digital conversion, we develop a low-complexity
iterative decoding algorithm. We show that if an informed
decoder that knows the covariance matrix can recover each
sample with small error probability, and $n$ is large enough,
the performance of the proposed blind recovery algorithm
closely follows that of the informed one. We complement the
analysis with numeric results that show that the algorithm
performs well even in non-asymptotic conditions.

I. INTRODUCTION

Let $X_1, \ldots, X_n \overset{i.i.d.}{\sim} N(0, \Sigma)$ be $n$ i.i.d.
realizations of a zero-mean $K$-dimensional Gaussian random
vector with covariance matrix $\Sigma \in \mathbb{R}^{K \times K}$. Let $X_i^*$
be the $K$-dimensional vector obtained by reducing each
coordinate of $X_i$ modulo $\Delta$, for some $\Delta > 0$. This
paper studies the problem of blindly reconstructing the
original $n$ samples $\{X_1, \ldots, X_n\}$ from their wrapped
counterparts $\{X_1^*, \ldots, X_n^*\}$, where the term blind
refers to without knowing the covariance matrix $\Sigma$. As the
modulo operation can be thought of as discarding the
most significant bits (MSBs) in the binary representation
of each coordinate and keeping only the least significant
bits (LSBs), this problem can be alternatively thought of
as that of blindly recovering the MSBs of the coordinates
from their LSBs.

Since the modulo operation is not invertible, it should
be clear that even in the informed case, where $\Sigma$ is
known, reconstruction of $X$ from $X^*$ can never be
guaranteed to succeed, and there is always a finite error
probability associated with the reconstruction process.

The error probability in the informed case depends on
the interplay between the covariance matrix $\Sigma$ and the
modulo size $\Delta$. As blind recovery algorithms cannot do
better than informed ones, our goal is to develop a blind
algorithm that successfully recovers $\{X_1, \ldots, X_n\}$ from
$\{X_1^*, \ldots, X_n^*\}$ with high probability, whenever $\Delta$ and
(the unknown) $\Sigma$ are such that an informed algorithm
can achieve a high successful recovery probability.

The main motivation for studying the blind unwrap-
ping problem comes from recent trends in the study of
analog-to-digital converters (ADCs). In many emerging
applications in communication and signal processing one
needs to digitize highly correlated analog processes,
where each process is observed by a separate ADC.
As a representative, but by no means exclusive, exam-
ple, consider the front-end of a massive multiple-input
multiple-output (MIMO) receiver, where the number of
antennas can be of the order of tens and even hundreds,
whereas the number of users it serves is moderate,
making the signals observed by the various antennas
highly correlated. It was recently proposed to use the so-
called modulo ADCs in such scenarios [1], [2]. A modulo
ADC is a device that first reduces its input modulo $\Delta$,
and only then quantizes it. The modulo reduction limits
the dynamic range of the resulting signal to the interval
$[-\frac{\Delta}{2}, \frac{\Delta}{2})$, which means that when $\Delta$ is small, one can
quantize the wrapped signal to within a good precision
using only a few bits. As high correlation between the
signals observed by various modulo ADCs allows for
correct reconstruction even with small $\Delta$, this archi-
tecture successfully exploits the correlation between the
signals for reducing the burden from the ADCs. Previous
works [1], [2], [3] studied the performance of modulo
ADCs for correlated signals, under the assumption that
the decoder is informed of $\Sigma$. In practice, one has no
access to $\Sigma$ and the blind setup is more appropriate for
analog-to-digital conversion. Another motivation for the

This work was supported, in part, by ISF under Grant 1791/17,
and by the GENESIS consortium and Heron Consortium via the Israel
Ministry of Economy and Industry.
blind unwrapping problem comes from the problem of communication in the presence of additive noise with unknown covariance. See [4].

The main contribution of this work is an iterative algorithm, with complexity $O(n^2 \log K + n \text{poly} K)$ for the blind unwrapping problem. We derive analytic results that demonstrate that if the error probability of the benchmark informed unwrapping algorithm is small enough, and $n$ is large enough, then our blind algorithm attains roughly the same performance as those of the informed benchmark algorithm. We complement the analysis with some numerical experiments which show that our algorithm performs surprisingly well even when $n$ and the informed error probability are quite moderate. In particular, the experiments show that the number of measurements required for successful blind unwrapping is dictated by the number of strong eigenvalues in $\Sigma$, rather than the dimension of $\Sigma$. In particular, for problem of a “sparse” nature, such as massive MIMO for example, the algorithm performs well with a relatively small number of measurements.

II. PROPOSED ALGORITHM, AND MAIN RESULTS

Define the modulo operation

$$x^* = [x] \mod \Delta = x - \Delta \left\lfloor \frac{x}{\Delta} \right\rfloor \in \left[ -\frac{\Delta}{2}, \frac{\Delta}{2} \right],$$

where $[t] = \arg\min_{b \in \mathbb{Z}} \{ |t - b| \}$ is the “round” operation, which returns the closest integer to $t$, with the convention that $[a + \frac{1}{2}] = a$ for $a \in \mathbb{Z}$. For a vector $x \in \mathbb{R}^K$, we write $x^*$ for the vector obtained by applying the modulo operation on each coordinate of $x$, i.e., $x^* = [x_1^* \ldots x_K^*]^T$.

Let $\Sigma \in \mathbb{R}^{K \times K}$ be a positive definite covariance matrix, and let $\{X_1, \ldots, X_n\} \sim \mathcal{N}(0, \Sigma)$. In the Blind Unwrapping Problem we are given only the modulo reduced random vectors $\{X_1^*, \ldots, X_n^*\}$ and our goal is to recover their unfolded versions $\{X_1, \ldots, X_n\}$, without prior knowledge on the covariance matrix $\Sigma$. In particular, we are interested in devising an algorithm whose input is $\{X_1^*, \ldots, X_n^*\}$ and whose output is a set of $n$ estimates $\{\hat{X}_1, \ldots, \hat{X}_n\}$. The performance of an algorithm is measured by

$$P_e \triangleq \Pr\left( \{X_1, \ldots, X_n\} \neq \{\hat{X}_1, \ldots, \hat{X}_n\} \right).$$

For a $K$-dimensional random vector $X \sim \mathcal{N}(0, \Sigma)$ and a set $S \subset \mathbb{R}^K$, we define the truncated random vector $Y \sim \mathcal{N}(X|X \in S)$, whose probability density function (pdf) is $f_Y(y) = \frac{f_X(y)I_{y \in \mathcal{S}}}{Pr(X \in S)}$, and whose covariance matrix is $E[YY^T] = E[XX^T]|X \in S$. We propose a low-complexity algorithm for the blind unfolding problem. The algorithm is based on the following simple observation, which holds due to the fact the the modulo operation is invariant with respect to translation by integer copies of $\Delta$.

**Proposition 1**: For any integer vector $a \in \mathbb{Z}^K$ and and $x \in \mathbb{R}^K$ it holds that

$$[a^Tx^*] \mod \Delta = [a^T x] \mod \Delta.$$

The implication of Proposition 1 is that for $A \in \mathbb{Z}^{K \times K}$, we have that $[Ax^*]^T = [Ax]^T$, and consequently, if $A$ is further full-rank, and $AX \in \text{CUBE}$, where $\text{CUBE} \triangleq \{[-\frac{\Delta}{2}, \frac{\Delta}{2}]^K\}$, then $A^{-1}[(Ax)^*]^T = x$.

The algorithm iterates between two basic procedures:

1) Estimating the covariance matrix $\Sigma$ of the Gaussian random vector $X$ truncated to the set $S = \text{CUBE}$, from the measurements $\{X_1^*, \ldots, X_n^*\}$. 2) Finding a full-rank integer matrix $A$, based on $\Sigma$, such that $Pr(AX \notin \text{CUBE}) < Pr(X \notin \text{CUBE})$, provided that $Pr(X \notin \text{CUBE})$ was not very small to begin with, and updating the measurements set to $\{[AX_1]^*, \ldots, [AX_n]^*\} = \{[AX_1]^T, \ldots, [AX_n]^T\}$. In order to facilitate the analysis of the algorithm in the sequel, we will restrict attention to a subset of the full-rank integer matrices, namely the group of unimodular matrices $\text{GL}_K(\mathbb{Z})$, consisting of all matrices in $\mathbb{Z}^{K \times K}$ with determinant $1$ or $-1$. Note that $A \in \text{GL}_K(\mathbb{Z})$ implies that $A^{-1} \in \text{GL}_K(\mathbb{Z})$.

Below we give the precise algorithm. The algorithm has two parameters: $d \in \mathbb{R}_+$ and $M \in \mathbb{N}$, that are chosen by the designer. The main algorithm makes use of the procedure EstTruncatedCov which will be described immediately.

**Inputs:** $(X_1^*, \ldots, X_n^*)$, $\Delta$, and two design parameters $d$ and $M$.

**Main Algorithm:**

- **Initialization**: $A = I$, $\hat{A} = 0^{K \times K}$, $\text{ctr} = 0$, $(V_1^*, \ldots, V_n^*) = (X_1^*, \ldots, X_n^*)$
- While $A \neq I$ and $\text{ctr} < M$
  1) $\hat{\Sigma} = \text{EstTruncatedCov}((V_1^*, \ldots, V_n^*), d)$
  2) Compute
     $$\hat{A} = [\hat{a}_1 \cdots \hat{a}_K]^T = \arg\max_{A \in \text{GL}_K(\mathbb{Z}), \text{det}(A) = \pm 1} \hat{\Sigma}_{kk}$$
  3) Set $V_j^* \leftarrow [\hat{A}V_j^*]^T = [\hat{A}V_j]^T$ for $j = 1, \ldots, n$, $A \leftarrow \hat{A} \cdot A$, and $\text{ctr} \leftarrow \text{ctr} + 1$

**Outputs**: $\hat{A}$ and the estimates $\hat{X}_j = A^{-1}V_j^*$, for $j = 1, \ldots, n$.

The algorithm EstTruncatedCov $((V_1^*, \ldots, V_n^*), d)$, which is used within the main algorithm, is as follows.

**Inputs:** $(V_1^*, \ldots, V_n^*)$, and a design parameter $d$.

**EstTruncatedCov Algorithm:**

1) Set $V_0^* = 0$
2) Construct a graph where each of the \( n+1 \) points \( (V_0^*, V_1^*, \ldots, V_n^*) \) is a vertex, and an edge between \( V_i^* \) and \( V_j^* \) exists if \( \|V_i^* - V_j^*\|_2 < d \). Find the connected component of \( V_0^* \) = 0, and denote it by \( T \).

4) If \( |T| < K+1 \), set \( d \leftarrow 1.1d \) and return to step 2; else

**Output:** set

\[
\Sigma = \frac{1}{|T|} \sum_{t \in T} tt^T,
\]

as the estimate of \( \mathbb{E}[VV^T | V \in \text{CUBE}] \).

For fixed \( M \) and a fixed \( d \) whose value does not decrease with \( K \), the complexity of the algorithm can be shown [4] to be \( O\left(n^2 \log K + n \log(K) + \left(\frac{5}{4}\right)K^{3/4}\right) \) if the shortest lattice basis problem in (1) is solved exactly, and \( O\left(n^2 \log K + n \log(K)\right) \) if it is approximated using the LLL algorithm [5].

Our analysis requires several assumptions on the underlying covariance matrix, which we shall now specify. We start by introducing some notation. For a symmetric positive definite matrix \( \Sigma \in \mathbb{R}^{K \times K} \), \( \Sigma > 0 \), denote the smallest eigenvalue by \( \lambda_K(\Sigma) \), and let \( \Sigma^{-1/2} \) be such that \( \Sigma^{-1} = \Sigma^{-1/2} \Sigma^{-1/2} \). For a full-rank matrix \( G \in \mathbb{R}^{K \times K} \) we define the lattice \( \Lambda(G) = G \cdot \mathbb{Z}^K \).

The packing radius of a lattice \( \Lambda(G) \) is defined as

\[
r_0(\Lambda(G)) = \frac{1}{2} \min_{g \in \mathbb{Z}^K \setminus \{0\}} \|Gg\|
\]

and the effective radius \( r_{\text{eff}}(\Lambda(G)) \) is defined as the radius of a \( K \)-dimensional Euclidean ball whose volume is \( |G| \), i.e.,

\[
V_{K, \text{eff}}(\Lambda(G)) = |G|, \text{ where } V_K \text{ is the volume of a } K\text{-dimensional unit ball.}
\]

Now, suppose that \( X \sim \mathcal{N}(0, \Sigma) \), and let \( 0 < \epsilon < 1 \), \( \tau_{\text{min}} > 0 \), \( 0 < P < 1 \) and \( 0 < \rho_{\text{pack}} < 1 \) be parameters; the guarantees we give on the performance of our algorithm will be given in terms of these. We say that \( \Sigma \) satisfies assumption A\( i \), \( i = 1, \ldots, 4 \), if

A1 : \( \min_t \{\Pr(AX \notin \text{CUBE}) : A \in \text{GL}_K(\mathbb{Z})\} \leq \epsilon \)

A2 : \( \lambda_K(\Sigma) \geq \tau_{\text{min}}^2 \)

A3 : \( \Pr(X \notin \text{CUBE}) \leq P \)

A4 : \( \frac{r_0(\Lambda(\Sigma^{-1/2}))}{r_{\text{eff}}(\Lambda(\Sigma^{-1/2}))} \geq \rho_{\text{pack}}. \)

Assumption A1 simply means that the informed benchmark integer-forcing decoder [1], \( X_{\text{IF}} = A_{\text{opt}}^{-1}[A_{\text{opt}} X]^* \), attains error probability

\[
\Pr(X_{\text{IF}} \neq X) = \Pr(A_{\text{opt}} X \notin \text{CUBE}) \leq \epsilon,
\]

where \( A_{\text{opt}} = \min_{A \in \text{GL}_K(\mathbb{Z})} \Pr(AX \notin \text{CUBE}) \). Assumption A2 limits the “spikiness” of \( \Sigma \), assumption A3 requires that the initial conditions of the algorithm, corresponding to \( A = I \) are not too bad, and assumption A4 is rather technical.

Let \( Q(t) = \frac{1}{\sqrt{2\pi}} \int_t^{\infty} e^{-\frac{x^2}{2}} dx \) be the Q-function, and \( Q^{-1}(t) \) be its inverse. We break the analysis of our algorithm into three parts: (1) We first show that if the procedure EstTruncatedCov is replaced with any method (“Genie”) that always returns a sufficiently accurate estimate of \( \mathbb{E}[VV^T | V \in \text{CUBE}] \), then the main algorithm converges to a near optimal \( A \in \text{GL}_K(\mathbb{Z}) \).

This is the content of Theorem 1. (2) We then show that given enough samples \( X_1, \ldots, X_n \), a single call to the specific procedure EstTruncatedCov we propose produces, with high probability, a good estimate for the truncated covariance matrix. This is done in Theorem 2. (3) Lastly, in Theorem 3, we combine the previous results to give a guarantee for (a slightly modified version of) the entire recovery algorithm.

**Theorem 1 (Guarantee on Genie-Aided Algorithm):** Let \( X \sim \mathcal{N}(0, \Sigma) \), where \( \Sigma \) satisfies assumptions A1 and A3, and \( \epsilon \) satisfies \( Q^{-1}(\frac{\epsilon}{2}) \geq 6\sqrt{K} \). Consider a genie-aided version of the main algorithm, where the procedure EstTruncatedCov\((V_1^*, \ldots, V_n^*)\) is replaced with a genie that returns a matrix \( \Sigma \) that satisfies \( (1 - \beta) \Sigma_{\text{V.trunc}} \preceq \Sigma \preceq (1 + \beta) \Sigma_{\text{V.trunc}} \), where \( \Sigma_{\text{V.trunc}} \triangleq \mathbb{E}[VV^T | V \in \text{CUBE}] \), for some \( 0 \leq \beta \leq 0.1 \). Then, after

\[
M = \left[ \frac{\log(180)}{\log \left(\frac{1}{3\epsilon}\right)} + 2 \right]
\]

iterations, the matrix \( A \) found by the algorithm must satisfy

\[
\Pr(AX \notin \text{CUBE}) \leq K \cdot Q \left(0.99 \sqrt{\frac{1 - \beta}{1 + \beta}} \cdot Q^{-1}(\frac{\epsilon}{2})\right).
\]

The proof can be found in [4]. It is based on showing, via the recently proved Gaussian Correlation Inequality, that if \( V \sim \mathcal{N}(0, \Sigma) \), then

\[
f(\Pr(X \notin \text{CUBE})) \cdot \Sigma \preceq \Sigma_{\text{V.trunc}} \preceq \Sigma,
\]

for some decreasing mapping \( f : [0, 1] \rightarrow [0, 1] \) which we explicitly specify. Consequently, by A1, we know that the matrix \( A \) calculated in step 2 of the algorithm will be such that \( \max_{k \in [K]} \text{diag}(A^2 \Sigma A) \) is small. Leveraging this fact along with a Gaussian extremal inequality due to Latała and Oleszkiewicz [6], shows that the probability of missing CUBE decreases from iteration to iteration.

The next result shows that given enough samples and that assumption A1 holds with small \( \epsilon \), the algorithm EstTruncatedCov indeed produces an accurate estimate of \( \Sigma_{\text{V.trunc}} \) from the measurements \( \{V_1^*, \ldots, V_n^*\} \). In [4] we give precise guarantees on the accuracy of the estimate. As the exact result is rather cumbersome, here we only state the asymptotic version.

**Theorem 2 (Guarantee on truncated covariance es-**


**timation**: Let $\Sigma \in \mathbb{R}^{K \times K}$ satisfy assumptions A1, A2 and A3, where $\epsilon$ is such that $\kappa_{\epsilon} \triangleq \frac{Q^{-1}(\frac{\epsilon}{2})}{\sqrt{n}} > 1$. Suppose that the procedure $\text{EstTruncatedCov}(\{V_1^{*}, \ldots, V_n^{*}\}, d)$ is run with a distance parameter $d = 2n\sqrt{K} \cdot \tau_{\min} \cdot \kappa_{\epsilon}$, for some $\eta \in (0, 1 - \frac{1}{\epsilon})$, and $V_1^{*}, \ldots, V_n^{*}$ are $n$ independent wrapped samples from $\mathcal{N}(0, \Sigma)$. Then, with probability at least $1 - \rho_{\text{est-err}}$, it holds that

$$0.999\Sigma_{\text{V},\text{trunc}} \leq \Sigma \leq 1.001\Sigma_{\text{V},\text{trunc}},$$

where for fixed $K$ and $\eta$ taking $\epsilon \to 0$ and $n = \epsilon^{-\zeta}$ for some $\zeta < (1 - \eta)^2$ we have that $\rho_{\text{est-err}} < n^{-0.99(1-\eta)^2}$.

The proof is given in [4]. Assume $\epsilon$ is small. The main observation is that for two points $V_i, V_j$ such that $V_i = V_j$ but $V_j \neq V_j$, it must hold that $\|V_i - V_j\| > d$ with high probability, for $d = 2\eta\sqrt{K} \cdot \tau_{\min} \cdot \kappa_{\epsilon}$, where the probability decreases with $\eta$. Thus, with high probability the connected component of 0 in the graph constructed by the procedure $\text{EstTruncatedCov}$ contains only points that were not folded. If $n$ is large enough, the points will be dense and almost all points that were not folded will land in the connected component.

At this point, one might naively think that Theorems 1 and 2 immediately imply that the main algorithm successfully recovers all the points, with high probability, provided that $\epsilon$ is small enough and $n$ large enough. The situation, however, is more subtle: when we use, in each iteration, the entire sample to estimate the truncated covariance, we are inadvertently creating inter-dependencies between the samples as we are multiplying by $A$. This means that in the following iteration the samples we use are no longer i.i.d, hence we cannot apply Theorem 2 as is. To mitigate this difficulty, we introduce a slight modification of the algorithm. The twist is as follows: given the design parameter $M$, we partition $[n]$ into $M$ disjoint index sets $I_1, \ldots, I_M$, each of size $n/M$. In the main algorithm, in step 2 of the $m$th iteration, instead of applying $\text{EstTruncatedCov}(\{V_1^{*}, \ldots, V_n^{*}\}, d)$ we apply $\text{EstTruncatedCov}(\{V_{i_m}^{*}, \ldots, V_{i_m,n/M}^{*}\}, d)$, where $I_m = \{i_{m,1}, \ldots, i_{m,n/N,M}\}$. Thus, every sample only participates once (or never participates) in $\text{EstTruncatedCov}$. 

**Theorem 3 (Success probability for the (modified) main algorithm: recovering $X_1, \ldots, X_n$:** Let $X \sim \mathcal{N}(0, \Sigma)$, where $\Sigma$ satisfies assumptions A1, A3 and A4. Suppose that the modified main algorithm is run with parameter $M$ satisfying (3), and distance parameter

$$d = 0.01 \cdot 2^{\sqrt{K}} \cdot \frac{\Delta}{\chi_1(\epsilon; P, K)},$$

where

$$\chi_1(\epsilon; P, K) \triangleq \frac{K^{2K - \frac{1}{2}} \cdot V_K \cdot (Q^{-1}(\frac{\epsilon}{2}))^{K}}{2^K \cdot \rho_{\text{pack}}^{K} \cdot (Q^{-1}(\frac{P}{2}))^{K-1}}.$$ 

Suppose that $n = \epsilon^{-\zeta}$ for some $\zeta < (0.99)^2$. Then there is some $\epsilon^* = \epsilon^*(K, P, \rho_{\text{pack}}, \zeta) > 0$ such that if $\epsilon < \epsilon^*$, then $P_\epsilon \leq n^{0.96}$.

**Remark 1:** The absence of assumption A2 here seems odd at first sight. Instead, we need to use A4 to be able to control $\lambda_K(\Sigma^{(l)})$ throughout the run of the algorithm.

### III. Numerical Results

In all experiments below, we consider $K \times K$ covariance matrices of the form

$$\Sigma = I_K + \text{snr HH}^T,$$

where $H \in \mathbb{R}^{K \times \text{rank}}$ for some integer rank, and $\text{snr} > 0$ is a parameter. Such covariance matrices correspond to the output of a narrowband MIMO channel $Y_t = HS_t + Z_t$, where $H \in \mathbb{R}^{K \times \text{rank}}$ is the channel matrix, $Z_t$ is additive white Gaussian noise, and $S_t$ is the vector of communication symbols transmitted at time $t$ (which are assumed to be Gaussian i.i.d.). If one then applies a modulo ADC with modulo size $\Delta$ on the output of each receive antenna, the resulting $K$-dimensional vector would be of the form $X_t^\gamma$, where $X_t = Y_t + Q_t$, and $Q_t$ is the vector of quantization noises incurred by the modulo ADCs. Further making the simplifying assumption that $Q_t$ is additive white Gaussian noise, we obtain that $X_t \sim \mathcal{N}(0, \Sigma)$, where $\text{snr}$ depends on the channel noise variance and the quantization noise variance.

We evaluate the performance of the algorithm for various values of $K$, rank, and $\text{snr}$. As our benchmark informed decoder, we take the integer-forcing decoder [1]. In the experiments, we find it convenient to set $\Delta = \Delta(\Sigma, \epsilon)$ such that the error probability of the integer-forcing decoder is fixed to some value, say $\epsilon$. Thus, $\Delta$ may vary across different realizations of $\Sigma$, but the (per-sample) error probability of the integer-forcing decoder remains fixed to $\epsilon$ throughout.

For covariance matrices of the form (4), we have that all eigenvalues of $\Sigma$ are at least 1. Thus, we have that assumption A2 holds with $\tau_{\min} = 1$, for all covariance matrices we consider here. Furthermore, by our choice of $\Delta(\Sigma, \epsilon)$ we have that assumption A1 holds. Thus, guided by our analysis we set $d = \eta\sqrt{K} \tau_{\min} Q^{-1}(\frac{\epsilon}{2})$, with $\eta = 0.5$, and $\tau_{\min} = 1$ throughout all experiments. Furthermore, $M$ was set to 30 in all experiments.

For fixed $H$, $\text{snr}$, $\epsilon$, and $n$, let $\Sigma$ be as in (4) and define $P_{\text{err}}^\text{blind}(H, \text{snr}, \epsilon, n)$ as the probability that the estimates $\{X_1, \ldots, X_n\}$ produced by our algorithm are not all identical to $\{X_1, \ldots, X_n\}$, where the modulo size is $\Delta(\Sigma, \epsilon)$. Furthermore, for fixed $K$ and rank we assume the entries of $H$ are i.i.d. $\mathcal{N}(0, 1)$, and define

$$P^\text{blind}(K, \text{rank, snr, } \epsilon, n) \triangleq \mathbb{E}[P_{\text{err}}^\text{blind}(H, \text{snr, } \epsilon, n)],$$

where $\mathbb{E}$ denotes the expectation.
where the expectation is with respect to the randomness in \( \mathbf{H} \). We estimate \( P_e^{\text{blind}}(K, \text{rank}, \text{snr}, \epsilon, n) \) using Monte-Carlo simulations. In particular, for fixed \( K, \text{rank}, \text{snr}, \epsilon \) and \( n \) we draw \( r = 1000 \) different realizations of \( \mathbf{H} \), and for each one of those compute \( \Sigma \) according to \( (4) \).

We then draw \( \mathbf{X}_1, \ldots, \mathbf{X}_n \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \Sigma) \), compute \( \{\mathbf{X}_1^*, \ldots, \mathbf{X}_n^*\} \), where \( \Delta = \Delta(\Sigma, \epsilon) \), and apply the proposed algorithm that yield estimates \( \{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \).

Our estimate for \( P_e^{\text{blind}}(K, \text{rank}, \text{snr}, \epsilon, n) \) is taken as the fraction of realizations of \( \mathbf{H} \) for which \( \{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \neq \{\mathbf{X}_1^*, \ldots, \mathbf{X}_n^*\} \).

Figure 1 shows the Monte-Carlo estimate of \( P_e^{\text{blind}}(K, \text{rank}, \text{snr}, \epsilon, n) \) for \( n = 1000 \), \( \epsilon = 9.6 \cdot 10^{-7} \), and various values of \( K, \text{rank} \) and \( \text{snr} \).

![Graphs showing Monte-Carlo evaluation](image)

**Fig. 1.** Monte-Carlo evaluation of \( P_e^{\text{blind}}(K, \text{rank}, \text{snr}, \epsilon, n) \) for \( n = 1000 \), \( \epsilon = 9.6 \cdot 10^{-7} \), such that \( 1 - (1 - \epsilon)^n \approx 9.6 \cdot 10^{-4} \). and various values of \( K, \text{rank} \) and \( \text{snr} \).

The results show that even for moderate values of \( n \) and \( \epsilon \), at moderate \( \text{snr} \) the algorithm performs close to the informed benchmark provided that \( \text{rank} \), rather than \( K \) is small. The dependence on the \( \text{rank} \) rather than the problem’s dimension is encouraging, because in many problems of practical interest where modulo ADCs are advantageous, the dimension is large, but the covariance matrix has a small number of significant eigenvalues.

**REFERENCES**


