LASSO-BASED FAST RESIDUAL RECOVERY FOR MODULO SAMPLING

Shaik Basheeruddin Shah[1], *Satish Mulleti*[2]*, and Yonina C. Eldar*[1]

[1]Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Israel
[2]Department of Electrical Engineering, Indian Institute of Technology (IIT) Bombay, India
Emails: shaik-basheeruddin.shah@weizmann.ac.il, mulleti.satish@gmail.com, yonina.eldar@weizmann.ac.il

ABSTRACT

In practice, Analog-to-Digital Converter (ADC) is used to perform sampling. A practical bottleneck of ADC is its lower dynamic range, leading to loss of information. To address this issue, researchers suggested folding operation on the signal using a modulo operator before passing it as an input to ADC. Though this process preserves the signal information, an unfolding algorithm is required to get the true samples from the folded samples. Noise robustness and computational time are two key parameters of an unfolding algorithm. In this paper, we propose a fast and robust algorithm for unfolding. Specifically, we first show that the first-order difference of the residual samples (the difference between the folded and true samples) is sparse by deriving an upper bound on its sparsity, and can be recovered from its partial Fourier measurements by formulating a sparse recovery problem. We demonstrate that the proposed algorithm is robust to noise and computationally efficient compared to the existing methods.

Index Terms— Sampling, modulo sampling, dynamic range, LASSO, B^2R^2, unlimited sampling.

1. INTRODUCTION

Sampling plays a vital role in modern information processing systems. The Shannon-Nyquist theorem states that a Band-Limited (BL) signal can be perfectly recovered from its equidistant samples measured at or above the Nyquist rate, which is twice the maximum frequency component of the signal [1]. Practically, sampling is performed by an Analog-to-Digital Converter (ADC). A practical bottleneck of an ADC is its dynamic range (DR) $[-\lambda, \lambda]$. Precisely, if the given signal's DR is beyond that of ADC's, then the signal is clipped, which is undesirable in many applications [2, 3, 4]. To avoid clipping, ADC's DR should be sufficiently high, leading to high power consumption.

Modulo sampling is one of the well-known techniques used to address the DR issue. In particular, whenever the signal's amplitude goes beyond $[-\lambda, \lambda]$, a folding operation is performed using the non-linear modulo operator such that the amplitude of the resultant folded signal is within $[-\lambda, \lambda]$. Then the samples of the folded signal are measured through a conventional ADC with low DR. Reconstruction algorithms are designed to operate on true samples. Thus, in the context of modulo folding, an unfolding algorithm has to be applied to estimate true samples from the folded before reconstruction [5, 6, 7, 8, 9]. In general, unfolding algorithms depend on the ADC's

DR, sampling rate of the signals, and noise levels. Hence, designing a robust and computationally efficient unfolding algorithm that operates close to the Nyquist rate is desirable.

Bhandari et al. [5, 6] proposed the so-called unlimited sampling framework to recover the BL unfolded samples using Higher-Order Differences (HODs) of the folded samples. This algorithm requires a high sampling rate, approximately $2\pi e$ times the Nyquist rate, where e denotes the Euler's constant. The usage of HODs makes the method sensitive to noise. Note that the problem of unfolding is equivalent to phase unwrapping problem. Itoh's algorithm is the well-known phase unwrapping algorithm which is based on the first-order difference of the phase-wrapped signal [10]. Hence, by comparing with unlimited sampling in this paper, we are also comparing with a phase unwrapping approach. In [7], the authors proposed a prediction-based algorithm and showed that perfect recovery of finite-energy BL signals is possible by considering sampling above the Nyquist rate. Specifically, for a given finite-energy BL signal, the authors used the fact that beyond a certain time instant, the folded sample values are equal to the unfolded values, i.e., the residual samples, the difference between folded and unfolded samples, form a finite-length vector. Recently, in [8, 9], the authors improved upon [7] and proposed a robust recovery algorithm called B^2R^2. The performance of B^2R^2 is good compared to the existing methods; however, it is computationally expensive. Precisely, the residual samples are recovered sample by sample in B^2R^2. Moreover, to estimate each sample value, a constrained optimization problem is formulated and solved using an iterative algorithm. Thus, we need an algorithm that is robust, operates at a low sampling rate, and is computationally efficient.

In this paper, we propose a fast and robust algorithm by formulating the problem of recovering the finite-length residual samples as a sparse recovery problem. The key observation of the algorithm is that the first-order difference of the residual samples, denoted $\hat{z}(n)$, is sparse. Theoretically, we justify this by deriving an upper bound on the number of non-zero elements in $\hat{z}(n)$. As the first-order difference of the unfolded samples is also BL, we formulate a linear inverse problem to estimate $\hat{z}(n)$, using beyond the bandwidth frequency information of $\hat{f}_\lambda(n)$, where $\hat{f}_\lambda(n)$ denotes the first-order difference of the given folded samples. As $\hat{z}(n)$ is sparse, we solve the linear inverse problem using Least Absolute Shrinkage and Selection Operator (LASSO). Thus, the proposed algorithm is called LASSO-B^2R^2. We use the standard Iterative Soft-Thresholding Algorithm (ISTA) to solve the LASSO optimization problem.

In [11], the authors proposed a recovery algorithm, known as Fourier-Prony, for periodic BL signals, which also employs the first-order difference of the residual samples. However, LASSO-B^2R^2 differs from Fourier-Prony, a spectral estimation-based method. Fourier-Prony requires prior information about the number of fold-

ing instants in the folded signal, which is not generally available. Moreover, in the presence of noise, the algorithm uses a matrix pencil method [12] with some tuning parameter, whose details are not explored in [11]. Thus, we compare LASSO-B^2R^2 with the algorithms proposed in [5, 6, 7, 8, 9]. Through simulation results, we show that the performance of LASSO-B^2R^2 is better in comparison with HOD and prediction-based methods; and is comparable with B^2R^2. However, LASSO-B^2R^2 requires a lower computational time in comparison with B^2R^2. To address the uncertainties occurred in a hardware prototype, the authors in [13, 14, 15] proposed a generalized model called modulo-hysteresis. However, this paper deals with ideal modulo operator.

The paper is organized as follows. Section 2 introduces the problem. The upper bound on the sparsity of $\hat{z}(n)$ is derived in section 3. The proposed LASSO-B^2R^2 algorithm is discussed in Section 4. Simulation results are presented in Section 5. Section 6 draws conclusions.

We use the following notations throughout the paper. The space of continuous time band-limited signals with bandwidth $[-\omega_m, \omega_m]$ is denoted by B_{ω_m}. The first-order difference operator of a signal $f(n)$ is $\hat{f}(n) = \Delta f(n)$[1]. Symbols $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{Z}$, denote the set of real numbers, a set of all positive real numbers, and integers, respectively. We use bold lowercase letters, e.g., $\mathbf{z}$, for vectors, and bold capital letters, e.g., $\mathbf{V}$, for matrices. Symbols $||\mathbf{z}||_1$, $||\mathbf{z}||_2$, and $||\mathbf{z}||_\infty$ denote the l_1-norm, l_2-norm, and l_∞-norm of $\mathbf{z}$, respectively. The spectral norm of a matrix $\mathbf{V}$ is written as $||\mathbf{V}||_2$. Symbol $L^2(\mathbb{R})$ denotes the space of square-integrable functions on $\mathbb{R}$. For a given $x \in \mathbb{R}$ and $\lambda \in \mathbb{R}^+$, the non-linear modulo operation, $\mathcal{M}_\lambda(.)$, is defined as $\mathcal{M}_\lambda(x) = (x + \lambda) \bmod 2\lambda - \lambda$. The cumulative sum of $\mathbf{x} = [x_1, \ldots, x_N]^T$ is $\mathbf{y} = [y_1, \ldots, y_N]^T$, where $y_k = \sum_{i=1}^{k} x_i$.

2. PROBLEM STATEMENT

Consider a signal $f(t) \in L^2(\mathbb{R}) \cap B_{\omega_m}$, whose amplitude is beyond the ADC's dynamic range $[-\lambda, \lambda]$. In this case, we use modulo sampling to avoid clipping. In particular, before sampling, we pass $f(t)$ through a non-linear modulo operator such that the resultant folded signal, $f_\lambda(t)$, is within the ADC's dynamic range. The folded signal is then uniformly sampled, which results in a sequence $f_\lambda(nT_s), n \in \mathbb{Z}$, where T_s is the sampling interval. The sampling rate (in rad/sec) is given as $\omega_s = \frac{2\pi}{T_s} = \text{OF} \times 2\omega_m$, where OF > 1 denotes the oversampling factor, and $2\omega_m$ is the Nyquist rate. Our aim is to design a fast and robust algorithm that operates at a low OF to recover the unfolded samples $f(nT_s)$ from the modulo (folded) samples $f_\lambda(nT_s)$. Hereafter $f_\lambda(nT_s)$ and $f(nT_s)$ are denoted as $f_\lambda(n)$ and $f(n)$, respectively. Mathematically, the folded samples can be decomposed as

$$f_\lambda(n) = f(n) + z(n), \quad n \in \mathbb{Z} \tag{1}$$

where $z(n) \in 2\lambda\mathbb{Z}$ is the residual signal. From the decomposition, we note that the problem of unfolding is equivalent to recovering the residual samples $z(n)$ from the folded samples. Hence, we focus on designing an algorithm to determine the residual sequence.

To this end, we use the following time-domain decay property of finite-energy BL signals. According to Riemann-Lebesgue lemma [7], we can show that $\lim_{|t| \to \infty} f(t) = 0$. This implies that the folded/unfolded sample values are negligible beyond a certain time

---

interval. Assume we have N samples within that time interval. Thus we can treat $z(n)$ as a finite N-length sequence. In practice, to choose N, we follow a two-step process: First, compute the energy of the given $f_\lambda(n)$. Second, consider the number of samples (N) that covers $98 - 99\%$ of that energy. Observe that the actual support set of $z(n)$, say $\mathcal{N}_\lambda \subset \{0, \cdots, N-1\}$, could be much smaller than N, i.e., $f(n) = f_\lambda(n), \forall n \notin \mathcal{N}_\lambda$. For instance, in Fig. 1, we considered a signal with $N = 1024$. In this case, $\mathcal{N}_\lambda$ consists of the values between the interval $(350, 670)$.

Applying the first-order difference operator on both sides of (1) leads to

$$\hat{f}_\lambda(n) = \hat{f}(n) + \hat{z}(n). \tag{2}$$

As $\hat{f}(n)$ is also a band-limited signal with bandwidth $[-\rho\pi, \rho\pi]$ (or $[0, \rho\pi] \cup [2\pi - \rho\pi, 2\pi)$), where $\rho = \frac{2\omega_m}{\omega_s} = \frac{1}{\text{OF}}$, we can write the following Discrete Fourier Transform (DFT) relation for $\hat{z}(n)$:

$$\hat{F}_\lambda\left(e^{\frac{j2\pi k}{N}}\right) = \sum_{n=0}^{N-1} \hat{z}(n)e^{\frac{-j2\pi kn}{N}}, \frac{2\pi k}{N} \in (\rho\pi, 2\pi - \rho\pi). \tag{3}$$

Here $\hat{F}_\lambda(e^{j\omega})$ denotes the discrete-time Fourier transform of $\hat{f}_\lambda(n)$. The above relation can be written in matrix form as

$$\left[\hat{\mathbf{F}}_\lambda\right]_{M \times 1} = [\mathbf{V}]_{M \times N}[\hat{\mathbf{z}}]_{N \times 1}, \tag{4}$$

where $\mathbf{V}_{k,n} = e^{\frac{-j2\pi kn}{N}}$, M denotes the number of discrete frequencies k that belong to $(\rho\pi, 2\pi - \rho\pi)$ and $M < N$.

Our aim is to solve the above linear inverse problem to estimate $\hat{\mathbf{z}}$. From $\hat{\mathbf{z}}$ one can easily recover $\mathbf{z}$ using the cumulative sum operator. Note that here we formulated the linear inverse problem using (2). Whereas, in [8, 9], a linear inverse problem is formulated using (1) and solved by formulating a constrained optimization problem that estimates $\mathbf{z}$ sample by sample. In the next section, we first show that $\hat{\mathbf{z}}$ is a sparse vector by deriving an upper bound on the sparsity of $\hat{\mathbf{z}}$. We then formulate the problem of estimating $\hat{\mathbf{z}}$ as a sparse recovery problem.

3. SPARSITY OF FIRST-ORDER DIFFERENCE OF THE RESIDUAL SAMPLES

The main idea behind the proposed algorithm is using the fact that $\hat{\mathbf{z}}$ in (4) is sparse. A typical $\hat{\mathbf{z}}$ is depicted in Fig. 1(c). Observe that $\hat{\mathbf{z}}$ is a sparse signal. To show that it is always sparse, we now find an upper bound on the number of non-zero values in $\hat{\mathbf{z}}$

Generally, the non-zero value at an index in $\hat{\mathbf{z}}$ represents the $(2\mathbb{Z} + 1)\lambda$ level-crossing in $f(t)$ or $2\lambda\mathbb{Z}$ level jump in $z(t)$. Let $\hat{\mathbf{z}}$ be an L-sparse vector. To find an upper bound on L, we first truncate $f(t)$ to $T = NT_s$ and denote it as $\tilde{f}(t)$. Since $f(t) \in B_{\omega_m}$, $\tilde{f}(t) \in B_{(\omega_m + \Delta\omega)}$, here $\Delta\omega$ is due to the spectral leakage that occurred by the process of truncation. Note that, beyond T, $f(t)$ has negligible energy and $f(t) = f_\lambda(t)$. Hence, we approximate the bandwidth of $\tilde{f}(t)$ to $[-\omega_m, \omega_m]$. We then construct a T-periodic signal, $f_p(t) = \sum_{l \in \mathbb{Z}} \tilde{f}(t - lT)$. The Fourier series coefficients, $F_p(k)$, of $f_p(t)$ satisfies

$$f_p(t) = \sum_{k=-K}^{K} F_p(k)e^{-jk\omega_0 t}, \tag{5}$$

where $\omega_0 = \frac{2\pi}{T}$ and $K = \lfloor \frac{\omega_m}{\omega_0} \rfloor = \lfloor \frac{N}{2\text{OF}} \rfloor$.

---

[1]Here, while computing the first-order difference, we pre-append $f(n)$ with zero to get both $f(n)$ and $\hat{f}(n)$ of the same length.
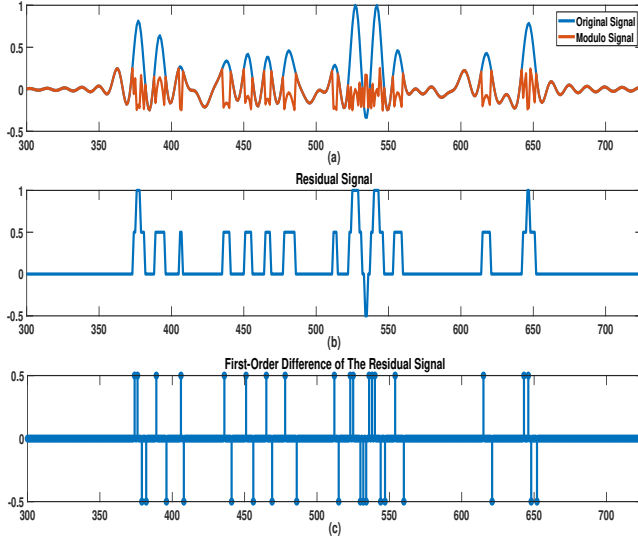
Fig. 1. (a) Samples of a band-limited signal $f(n)$ with OF = 6 and $\lambda = 0.25$, and the corresponding modulo signal $f_\lambda(n)$. (b) Residual signal, $z(n) = f_\lambda(n) - f(n)$. (c) First-order difference of the residual signal. The samples between [300, 750] are displayed for better visibility.

Equation (5) is a trigonometric polynomial of order K. We know that a trigonometric polynomial $q(t)$ of order K over a period T has a maximum of $2K$ level-crossings for level l, where $|l| \leq ||q(t)||_\infty$ [16]. Moreover, the number of $(2\mathbb{Z} + 1)\lambda$ level-crossings (the folding levels) in $f(t)$ are $2 + 2\left\lfloor \frac{||f_p(t)||_\infty - \lambda}{2\lambda} \right\rfloor$. Therefore, the number of times, say $\tilde{L}$, that $f_p(t)$ crosses the $(2\mathbb{Z} + 1)\lambda$ levels is upper bounded as

$$\tilde{L} \leq \left(2 + 2\left\lfloor \frac{||f_p(t)||_\infty - \lambda}{2\lambda} \right\rfloor \right) 2K. \tag{6}$$

This implies that the number of $(2\mathbb{Z} + 1)\lambda$ level-crossings in $f(t)$ or $2\lambda\mathbb{Z}$ level jumps in $z(t)$ is bounded as

$$\tilde{L} \leq 4K + 4K \left\lfloor \frac{||f(t)||_\infty - \lambda}{2\lambda} \right\rfloor. \tag{7}$$

From the above analysis and using the fact that $\hat{\mathbf{z}}$ is an N-length vector, we can bound the sparsity of $\hat{\mathbf{z}}$ as

$$L \leq \min\left(4K + 4K \left\lfloor \frac{||f(t)||_\infty - \lambda}{2\lambda} \right\rfloor, N \right). \tag{8}$$

For example, assume that the maximum value of $f(t)$ is normalized to 1 and $N = 1024$. Table 1 shows the upper bound on L for different λ values of a particular OF. From Table 1, we draw the following conclusions: For moderate λ and OF values, $\hat{\mathbf{z}}$ is sparse. A lower λ value increases the L value, and a higher OF value decreases the bound on L. In practice, the value of L is far less than its upper bound. For instance, we have $N = 1024$, OF = 6, and $\lambda = 0.25$ for the signal depicted in Fig. 1(c). In this case, the actual L is equal to 36, which is far less than its theoretical upper bound value of 680. An intuitive reason for this huge difference is that the actual support of $\mathbf{z}$, i.e., $\mathcal{N}_\lambda$ is much smaller than N.

λ	OF = 4		OF = 6		OF = 8	
	$2K$	Upper bound on L	$2K$	Upper bound on L	$2K$	Upper bound on L
0.75	256	512	170	340	128	256
0.5	256	512	170	340	128	256
0.25	256	1024	170	680	128	512
0.05	256	1024	170	1024	128	1024

Table 1. Upper bound on L for λ and OF values.

4. LASSO-B^2R^2 ALGORITHM

With the observation on the sparsity of $\hat{\mathbf{z}}$, we formulate the problem of estimating $\hat{\mathbf{z}}$ in equation (4) as a LASSO problem:

$$\min_{\hat{\mathbf{z}}} \frac{1}{2}||\hat{\mathbf{F}}_\lambda - \mathbf{V}\hat{\mathbf{z}}||_2^2 + \gamma||\hat{\mathbf{z}}||_1, \tag{9}$$

where γ is a regularization parameter. We solve this optimization problem using the well-known ISTA [17]. In particular, the update of $\hat{\mathbf{z}}$ at the $(i + 1)^{th}$ iteration is

$$\hat{\mathbf{z}}^{(i+1)} = S_{\gamma\tau}\left(\hat{\mathbf{z}}^{(i)} - \tau\mathbf{V}^H\left(\mathbf{V}\hat{\mathbf{z}}^{(i)} - \hat{\mathbf{F}}_\lambda \right) \right).$$

Here τ is the step size and $S_{\gamma\tau}(.)$ is the soft-thresholding operator,

$$S_{\gamma\tau}(x) = \text{sign}(x)\max\left(|x| - \gamma\tau, 0 \right).$$

As suggested in [17], for convergence we initialize $\gamma = 0.1||\mathbf{V}^H\hat{\mathbf{F}}_\lambda||_\infty$ and $\tau = \frac{1}{||\mathbf{V}||_2^2}$. Similar to B^2R^2, in this work also, we are using beyond the bandwidth frequency information to recover the residual samples by formulating a LASSO problem. Thus, we call this algorithm LASSO-B^2R^2. The summary of the proposed algorithm is presented in Algorithm 1.

Algorithm 1 LASSO-B^2R^2 Algorithm

1: **Input:** $f_\lambda(n)$, λ, ρ, maxItr   ▷ maxItr denotes the maximum number of iterations
2: Compute N
3: Construct $[\mathbf{V}]_{M \times N}$ using (4)
4: Compute $\hat{F}_\lambda(e^{\frac{j2\pi k}{N}})$, $\forall\, k \in \mathbb{Z}$ and $\frac{2\pi k}{N} \in (\rho\pi, 2\pi - \rho\pi)$
5: **Initialize:** $\gamma = 0.1||\mathbf{V}^H\hat{\mathbf{F}}_\lambda||_\infty$, $\tau = \frac{1}{||\mathbf{V}||_2^2}$, maxItr = 1000, and $\hat{\mathbf{z}}^{(0)} \in \mathcal{N}(0, 1)$
6: **for** $i = 0 : \text{maxItr}$ **do**
7:    $\hat{\mathbf{z}}^{(i+1)} = S_{\gamma\tau}\left(\hat{\mathbf{z}}^{(i)} - \tau\mathbf{V}^H\left(\mathbf{V}\hat{\mathbf{z}}^{(i)} - \hat{\mathbf{F}}_\lambda \right) \right)$
8:    **if** $||\hat{\mathbf{z}}^{(i+1)} - \hat{\mathbf{z}}^{(i)}||_2 < 10^{-4}$ **then**
9:       $\hat{\mathbf{z}} = \hat{\mathbf{z}}^{(i+1)}$
10:       exit
11:    **end if**
12: **end for**
13: $\hat{\mathbf{z}} \leftarrow \lceil \frac{\lfloor \hat{\mathbf{z}}/\lambda \rfloor}{2} \rceil$ ▷ Rounding the residual to $2\lambda\mathbb{Z}$
14: $\mathbf{z} \leftarrow \text{cumsum}(\hat{\mathbf{z}})$ ▷ Cumulative summation operator on $\hat{\mathbf{z}}$
15: $f(n) \leftarrow f_\lambda(n) - z(n)$
16: **Output:** $f(n)$
---

5. SIMULATION RESULTS

In this section, we compare the performance of the proposed algorithm with HOD-based [5], [6], prediction-based [7], and B^2R^2 [8]
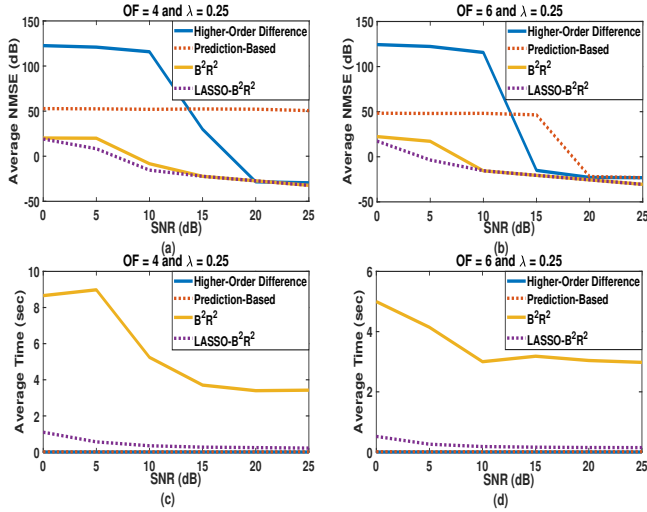
Fig. 2. Average NMSE vs SNR and average time vs SNR for OF= 4 and OF= 6 with $\lambda = 0.25$.

Fig. 3. Average NMSE vs OF for SNR= 5 dB and SNR= 10 dB.

Fig. 4. (a)-(b) Average NMSE and average L for different OF and λ pair values, respectively.

algorithms. First, we compare the performance in terms of robustness. The given $f_\lambda(n)$ (with certain OF and λ values) is added to the noise of a particular Signal to Noise Ratio (SNR) value. Here, noise follows the Gaussian distribution with mean zero and variance v, where v is computed using the given SNR value. We then apply different algorithms to estimate $f(n)$. Let $\bar{f}(n)$ be the estimated unfolded signal using a particular algorithm. The Normalized Mean Square Error (NMSE) is computed as NMSE $= \frac{||f(n)-\bar{f}(n)||_2^2}{||f(n)||_2^2}$. The average NMSE is computed by repeating this process for 250 independent and identically distributed noise realizations. Fig. 2(a) and Fig. 2(b) depict the average NMSE results obtained by varying SNR from 0 to 25 dB, for OF= 4 and OF= 6, respectively. Fig. 2(c) and Fig. 2(d) depict the average time (again averaged over 250 noise realizations for a fixed SNR value) required by each algorithm for estimating the true unfolded signal. Note that for the simulations in this section, we considered signals with $N = 1024$ and the maximum amplitude value of each signal is normalized to 1. Simulations are executed using python version 3.9 on AMD Ryzen 7 3700x 8-core processor with 16 GB RAM. From these results, we conclude that the LASSO-B^2R^2 is more robust in comparison with the HOD-based and prediction-based methods, and it requires slightly higher computational time. The robustness of LASSO-B^2R^2 is comparable with B^2R^2, but it requires a lower computational time. In B^2R^2, we recover $z(n)$ sample by sample. In particular, to estimate a sample value, we solve a constrained optimization problem using a iterative algorithm, and this process is repeated for all the samples in $z(n)$. In contrast, using LASSO-B^2R^2, we recover the complete residual signal.

Next, we compare the performance of LASSO-B^2R^2 in terms of the required OF value. Here we considered the modulo signal with a particular SNR value and $\lambda = 0.25$. For each OF value, we generated 250 signal realizations and computed the average NMSE. Fig. 3(a) and Fig. 3(b) depict the average NMSE results obtained by varying the OF values from 2 to 22, for SNR values 5 dB and 10 dB, respectively. From these results, we conclude that LASSO-B^2R^2 requires a lower OF value compared with higher-order difference and prediction-based methods, whereas it is comparable with B^2R^2. **Uniqueness guarantees:** According to compressed sensing theory
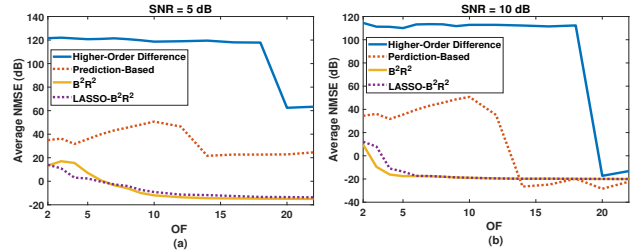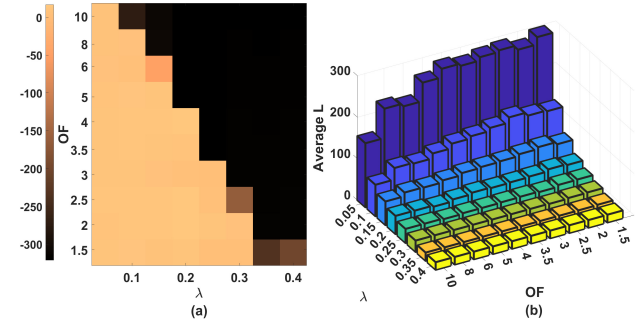
[18, 19, 20], if $M > CL \log\left(\frac{N}{\epsilon}\right)$, then $\hat{z}$ be the unique solution to the LASSO problem formulated in (9) with probability at least $1 - \epsilon$, where C is a constant. This implies that for a unique solution, M must satisfy the following condition: $CL \log\left(\frac{N}{\epsilon}\right) \le M < N$. Here, the parameters M and L are controlled by the values of OF and λ, respectively. For example, in the above analysis, with $N = 1024$ and $\lambda = 0.25$, the number of discrete frequencies, for OF = 6, that are within $[0, \rho\pi] \cup [2\pi - \rho\pi, 2\pi)$ are 170, i.e., $2K = 170$, this implies $M = N - 2K = 854$. Assume $C = 1$ and $\epsilon = 10^{-4}$, then $\lfloor CL \log\left(\frac{N}{\epsilon}\right) \rfloor = 581 < M$. As noted earlier, the smaller the λ, the less sparse the $\hat{z}$ will be. Hence, sufficient oversampling is required to satisfy the above condition; the same can be observed in Fig 4. Fig. 4(a) and Fig. 4(b) depict the average NMSE and the corresponding average L, respectively, for different OF and λ pairs. Here, for each OF and λ pair, the average NMSE and average L are computed by using 100 signal realizations. For low λ values, we observed that the non-zero values in $\hat{z}$ are forming groups. Thus, one may address this oversampling issue by formulating the problem of residual signal recovery as a group LASSO or sparse group LASSO.

6. CONCLUSION

In this work, we proposed a fast and robust algorithm to recover the residual signal in modulo sampling. In the process, we have initially shown that the residual signal's first-order difference is a sparse vector by deriving an upper bound on its sparsity. Then, the problem of recovering the residual signal is formulated as a sparse recovery problem. Precisely, we formulated a LASSO problem and solved it by using the ISTA algorithm. Through simulations, we demonstrated that the proposed algorithm is fast and robust compared to existing methods.

7. REFERENCES

[1] Y. C. Eldar, *Sampling Theory: Beyond Bandlimited Systems*, Cambridge University Press, 2015.

[2] P. E. Debevec and J. Malik, "Recovering High Dynamic Range Radiance Maps from Photographs," in *ACM Siggraph*, 1997, p. 369–378.

[3] K. Yamada, T. Nakano, and S. Yamamoto, "A vision sensor having an expanded dynamic range for autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 47, no. 1, pp. 332–341, 1998.

[4] J. Zhang, J. Hao, X. Zhao, S. Wang, L. Zhao, W. Wang, and Z. Yao, "Restoration of clipped seismic waveforms using projection onto convex sets method," *Nature Sci. Rep.*, vol. 6, no. 39056, 2016.

[5] A. Bhandari, F. Krahmer, and R. Raskar, "On unlimited sampling," in *Proc. Int. Conf. Sampling Theory Appl.*, Jul. 2017, pp. 31–35.

[6] A. Bhandari, F. Krahmer, and R. Raskar, "On Unlimited Sampling and Reconstruction," *IEEE Trans. Signal Process.*, vol. 69, pp. 3827–3839, 2021.

[7] E. Romanov and O. Ordentlich, "Above the Nyquist Rate, Modulo Folding Does Not Hurt," *IEEE Signal Process. Lett.*, vol. 26, no. 8, pp. 1167–1171, 2019.

[8] E. Azar, S. Mulleti, and Y. C. Eldar, "Residual Recovery Algorithm for Modulo Sampling," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 5722–5726.

[9] E. Azar, S. Mulleti, and Y. C. Eldar, "Robust unlimited sampling beyond modulo," *arXiv:2206.14656*, 2022.

[10] K. Itoh, "Analysis of the phase unwrapping algorithm," *Appl. Opt.*, vol. 21, no. 14, pp. 2470–2470, 1982.

[11] A. Bhandari, F. Krahmer, and T. Poskitt, "Unlimited Sampling From Theory to Practice: Fourier-Prony Recovery and Prototype ADC," *IEEE Trans. Signal Process.*, vol. 70, pp. 1131–1141, 2022.

[12] Y. Hua and T. K. Sarkar, "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 38, no. 5, pp. 814–824, 1990.

[13] D. Florescu, F. Krahmer, and A. Bhandari, "Unlimited sampling with hysteresis," in *Proc. 55th Asilomar Conf. Signals, Syst. Comput.*, 2021, pp. 831–835.

[14] D. Florescu and A. Bhandari, "Unlimited sampling with local averages," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 5742–5746.

[15] D. Florescu and A. Bhandari, "Unlimited sampling via generalized thresholding," in *Intl. Symp. Inf. Theory*, 2022, pp. 1606–1611.

[16] M. J. D. Powell, *Approximation Theory and Methods*, Cambridge University Press, 1981.

[17] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.

[18] H. Rauhut, "Compressive Sensing and Structured Random Matrices," *Theoret. Found. Numer. Methods Sparse Recov.*, pp. 1–92, 2010.

[19] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[20] H. Rauhut, "Random sampling of sparse trigonometric polynomials," *Appl. Comput. Harmon. Anal.*, vol. 22, no. 1, pp. 16–42, 2007.