

剰余サンプリングからの高効率・高精度な信号復元技術の開発

代表研究者

早川 諒

東京農工大学大学院工学研究院 准教授

1 はじめに

連続時間信号を離散時間信号に変換するサンプリングは、デジタル音楽の録音・再生、デジタル写真・ビデオの撮影に使われるなど、現代の信号処理において重要な役割を果たしている。サンプリングを行う ADC (Analog to Digital Converters: アナログデジタル変換器) には、主な性能の指標としてサンプリング周期と観測できる振幅の範囲の2つがあり、入力信号のサンプリング結果はこの2つに影響を受ける。具体的には、入力信号の最大周波数がサンプリング周期から導かれるサンプリング周波数の半分を超えると、復元の際に折りたたみ歪みが発生するため信号を正確に復元できない。また、信号の振幅が観測できる範囲を超えると信号はクリッピングされるため、元の信号を正確に復元できなくなる。サンプリング周期を短くしたり、観測できる振幅の範囲を大きくしたりするためにはより大きな電力が必要になるため、そのような観点からはよりサンプリング周期が長く、観測できる振幅の範囲が小さいADCが望ましい。観測できる振幅の範囲が小さいADCにおけるクリッピングを防ぐために、剰余サンプリングという手法が提案されている [1]。剰余サンプリングでは、ADC での測定の前に剰余演算子を用いて折りたたみ演算を行い、その結果得られる剰余信号の振幅が $[-\lambda, \lambda]$ 内に収まるようにする ($\lambda > 0$)。その後、 $[-\lambda, \lambda]$ の範囲を測定できるADCで剰余信号をサンプリングする。

剰余サンプリングによって得られた結果からもとの信号を復元するために、様々な手法が提案されている。そのためのハードウェアに関する研究 [2] では、剰余信号と共に振幅が折りたたまれた回数を記録するリセットカウンタマップを保存するアプローチが採用されているが、この手法では複雑な電子回路と追加の電力やメモリが必要となる。また、リセットカウンタマップを用いずに剰余信号を復元する方法として、信号の高次差分を用いるアルゴリズムが研究されている [1]。しかし、このアルゴリズムではサンプリング周期を短くする必要があり、推定結果はノイズの影響を受けやすい。最近では、よりノイズに頑健な手法として、 B^2R^2 (Beyond Bandwidth Residual Reconstruction) と呼ばれる、剰余信号の高周波帯域に着目したノイズに頑健な復元アルゴリズムが提案されている [3]。また、剰余信号の差分の信号がスパース性を持つことを利用して B^2R^2 を改良した LASSO- B^2R^2 (Least Absolute Shrinkage and Selection Operator- B^2R^2) も提案されている [4]。

本研究では、信号の離散性や事前情報を用いた剰余信号の復元アルゴリズムを提案する。具体的には、元信号と剰余信号の差が観測できる最大の振幅の偶数倍になるという信号の離散性と、信号の始点と終点で折りたたみが起きていないという事前情報を用いた制約を考える。離散値信号を推定するための最適化問題として提案されている SSR (Sum of Sparse Regularizers) 最適化 [5] に制約条件を加えた最適化問題である制約付き SSR 最適化を提案し、提案した最適化問題に対して、ADMM (Alternating Direction Method of Multiplier: 交互方向乗数法) [6] に基づくアルゴリズムを導出する。計算機シミュレーションによって、提案手法における NMSE (Normalized Mean Square Error: 正規化平均二乗誤差) が特定の条件下で従来手法より向上することを示す。

本稿では、実数全体の集合を \mathbb{R} 、複素数全体の集合を \mathbb{C} と書く。ベクトル $\mathbf{x} = [x_1 \cdots x_N]^T \in \mathbb{R}^N$ に対して、 $\|\mathbf{x}\|_p = \sqrt[p]{\sum_{n=1}^N |x_n|^p}$ は ℓ_p ノルムを表し ($p > 0$)、 $\|\mathbf{x}\|_\infty = \max_{1 \leq k \leq N} |x_k|$ は ℓ_∞ ノルムを表す。また、行列 $\mathbf{A} \in \mathbb{R}^{M \times N}$ に対して、 $\|\mathbf{A}\|_2 = \sigma_{\max}(\mathbf{A})$ は行列の ℓ_2 ノルムを表す。ただし、 $\sigma_{\max}(\mathbf{A})$ は \mathbf{A} の最大特異値を表す。 $\mathbf{1}$ はすべての成分が $\mathbf{1}$ であるベクトルとし、 \mathbf{I}_N は $N \times N$ の単位行列であるとする。また、 $\text{prox}_{\phi(\cdot)}$ は、関数 $\phi: \mathbb{R}^N \rightarrow \mathbb{R} \cup \{+\infty\}$ の近接写像であり、

$$\text{prox}_{\phi}(\mathbf{u}):= \arg \min_{\mathbf{v} \in \mathbb{R}^N} \left\{ \frac{1}{2} \|\mathbf{u} - \mathbf{v}\|_2^2 + \phi(\mathbf{v}) \right\} \quad (1)$$

と定義する。

2 剰余サンプリング

2-1 サンプリング

サンプリングは、連続時間信号を離散時間信号に変換する操作を指す。連続信号 $f(t)$ をサンプリングする場合、離散信号 $f[n]$ は

$$f[n] = f(nT_s) = f(t)|_{t=nT_s}$$

のように表される。ここで、 T_s はサンプリング周期である。サンプリング定理より、連続信号をサンプリングして離散信号を生成するときに、サンプリング周波数が連続信号の最大周波数の2倍以上であれば、得られた離散信号から元の連続信号を復元できる。

通常のサンプリングでは、入力信号の振幅がADCの観測できる振幅の範囲を超えると、その信号はクリッピングされる。クリッピングされた信号を $f_{cl}(t)$ とすると、 $f_{cl}(t)$ は

$$f_{cl}(t) = \begin{cases} \lambda & (f(t) > \lambda) \\ f(t) & (-\lambda \leq f(t) \leq \lambda) \\ -\lambda & (f(t) < -\lambda) \end{cases} \quad (2)$$

のようになる。ただし、観測できる振幅の範囲を $[-\lambda, \lambda]$ とした ($\lambda > 0$)。クリッピングのイメージを図1に示す。図1からわかるように、クリッピングが起これば、観測できる振幅の範囲を超えた部分の波形は保存されない。よって、クリッピングが起これば、サンプリング定理の周波数の条件を満たした信号であっても復元が不可能になる。

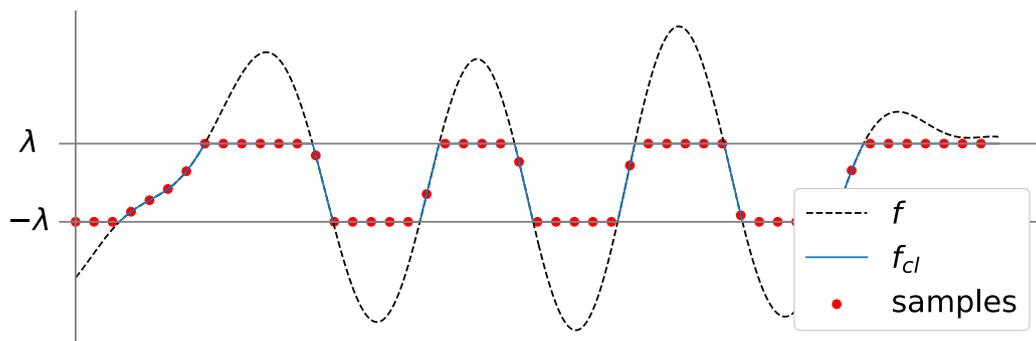


図1 クリッピング

2-2 剰余サンプリング

そこで、クリッピングを避けるために提案されたのが剰余サンプリング [1] である。剰余信号 $f_\lambda[n]$ は、入力 $f(t)$ に対して

$$M_\lambda(f(t)) = \{(f(t) + \lambda) \bmod 2\lambda\} - \lambda \quad (3)$$

という演算を施した $f_\lambda(t) = M_\lambda(f(t))$ をサンプリングすることで得られる [3]。演算 $M_\lambda(\cdot)$ は、入力信号の振幅が $[-\lambda, \lambda]$ を超えたとき、剰余演算子を用いて信号を折りたたみ、信号の振幅を $[-\lambda, \lambda]$ に収める演算となる。このとき、サンプリング後の信号 $f_\lambda[n]$ は、

$$f_\lambda[n] = f_\lambda(nT_s) \quad (4)$$

と表せる。ここで、 T_s はサンプリング周期である。

剰余サンプリングのイメージを図2に示す。図2より、剰余サンプリングではクリッピングが起これば、観測できる振幅の範囲を超えた部分の波形が保存されていることがわかる。

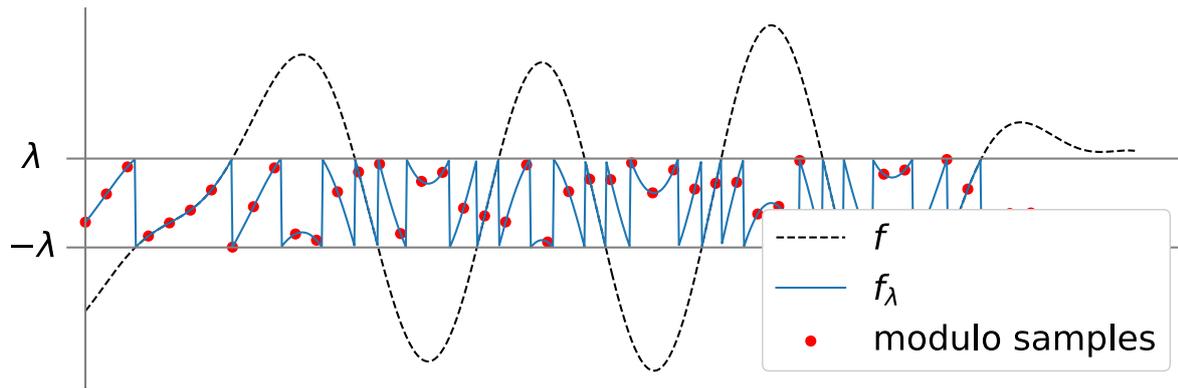


図 2 剰余サンプリング

剰余サンプリングでは折りたたみ後の信号をサンプリングするため、その結果から元の折りたたまれていない信号を復元する必要がある。剰余信号 $f_\lambda[n]$ は既知であるため、元信号 $f[n]$ の復元は折りたたみ信号

$$z[n] := f_\lambda[n] - f[n] \quad (\Leftrightarrow f_\lambda[n] = f[n] + z[n]) \quad (5)$$

を推定する問題に帰着できる。なお、剰余信号の定義より

$$f_\lambda[n] \bmod 2\lambda = f[n] \bmod 2\lambda \quad (6)$$

が成り立つので、 $z[n]$ は λ の偶数倍になる。

3 剰余信号の復元に関する従来手法

3-1 剰余サンプリングの性質

剰余サンプリングで得られた離散信号から元の信号を復元する方法として、 $z[n]$ の一次差分のスプース性を周波数領域において利用する手法である LASSO- B^2R^2 が提案されている [4]。 $z[n]$ の一次差分の例として、ある信号に対して剰余サンプリングを行ったときの元信号 $f[n]$ 、折りたたみ信号 $z[n]$ 、折りたたみ信号の一次差分 $\hat{z}[n]$ をそれぞれ図 3, 4, 5 に示す。

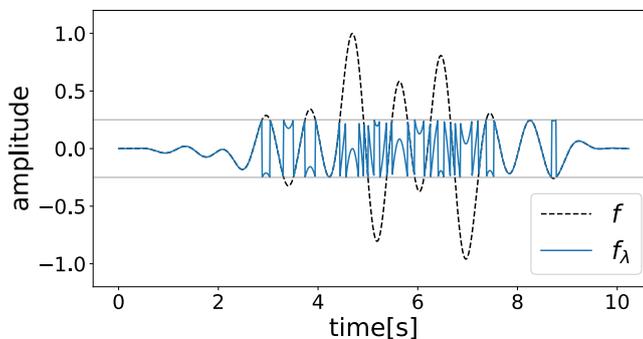


図 3 元信号

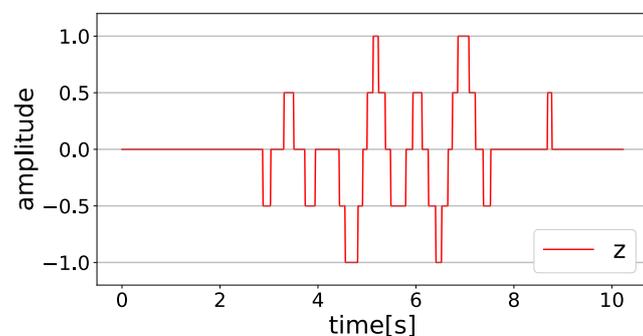


図 4 折りたたみ信号

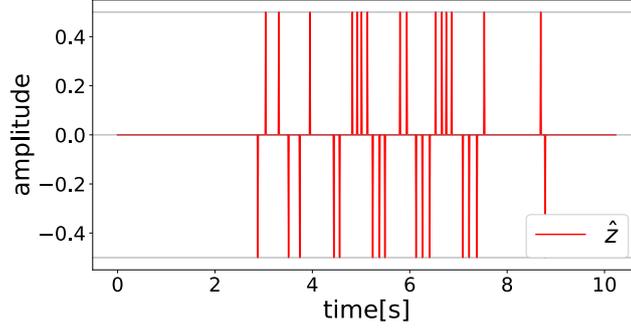


図 5 折りたたみ信号の一次差分

ここで $\hat{z}[n]$ は

$$\hat{z}[n] := z[n] - z[n-1] \quad (7)$$

と定義する。ただし、元信号と一次差分信号の長さが同じになるように、 $z[-1] = 0$ として一次差分を計算する。図 4, 5 からわかるように、 $z[n]$ は前後で同じ値をとることが多いため、 $\hat{z}[n]$ は零成分の多いスパースな信号となる。

LASSO- B^2R^2 では、 $\hat{z}[n]$ のスパース性とともに残余サンプリングの周波数領域での性質を利用する。剰余信号の復元では、信号の最大周波数 ω_m の 2 倍よりサンプリング周波数 ω_s を大きくとる必要がある。これは、信号の折りたたみが起こることにより、剰余信号の周波数スペクトルが元信号の周波数の範囲外にも生じるためである。元信号の最大周波数 ω_m の 2 倍とサンプリング周波数 ω_s の比を剰余サンプリング係数 $\text{OF} = \frac{\omega_s}{2\omega_m}$ と呼ぶ。

雑音がないと仮定すると、元信号をサンプリングした離散信号 $f[n]$ の周波数スペクトルは $[-\omega_m, \omega_m]$ の範囲に収まる。よって、

$$F(e^{j\omega T_s}) = 0 \quad (\omega_m < |\omega| < \omega_s/2) \quad (8)$$

が満たされる。ただし、各信号の離散時間フーリエ変換を、

$$F(e^{j\omega T_s}) = \sum_{n=0}^{N-1} f[n] e^{-j\omega n T_s} \quad (9)$$

のように大文字で表記する。ここで、 N は信号の長さを表す。一方、剰余サンプリングによって得られた離散信号 $f_\lambda[n]$ は、 $f[n]$ に折りたたみ信号 $z[n]$ が加わったものである。そのため、 $f_\lambda[n]$ の周波数スペクトルは $[-\omega_m, \omega_m]$ の範囲に加えて、 $[-\frac{\omega_s}{2}, -\omega_m]$ と $[\omega_m, \frac{\omega_s}{2}]$ の範囲にも存在する。具体的には、式 (5), (8) より、剰余信号 $f_\lambda[n]$ の離散時間フーリエ変換 $F_\lambda(e^{j\omega T_s})$ は

$$F_\lambda(e^{j\omega T_s}) = Z(e^{j\omega T_s}) \quad (\omega_m < |\omega| < \omega_s/2) \quad (10)$$

を満たす。ただし、 $Z(e^{j\omega T_s})$ は折りたたみ信号 $z[n]$ の離散時間フーリエ変換である。よって、 $F_\lambda(e^{j\omega T_s})$ と $z[n]$ の関係は

$$F_\lambda(e^{j\omega T_s}) = \sum_{n=0}^{N-1} z[n] e^{-j\omega n T_s} \quad (\omega_m < |\omega| < \omega_s/2) \quad (11)$$

となり、 $\omega = \frac{2\pi k}{T_s}$ と書き換えると

$$F_\lambda(e^{j2\pi k}) = \sum_{n=0}^{N-1} z[n] e^{-j2\pi kn/N} \quad \left(\frac{2\pi k}{N} \in \left(\frac{\pi}{\text{OF}}, 2\pi - \frac{\pi}{\text{OF}} \right) \right) \quad (12)$$

となる。 M を $\frac{2\pi k}{N} \in (\frac{\pi}{\text{OF}}, 2\pi - \frac{\pi}{\text{OF}})$ を満たす k の個数とし、そのような k と対応する行のみからなる部

分離散フーリエ変換行列 $\mathbf{V} \in \mathbb{C}^{M \times N}$ を考えると、その (k, n) 成分は $v_{k,n} = e^{-j2\pi kn/N}$ であり、式 (12) は

$$\mathbf{F}_\lambda = \mathbf{V}\mathbf{z} \quad (13)$$

と書ける．ここで， $\mathbf{F}_\lambda \in \mathbb{C}^M$ は剰余信号 $f_\lambda[n]$ の離散時間フーリエ変換のうち \mathbf{V} に対応する成分を並べたベクトル， $\mathbf{z} \in \mathbb{R}^N$ は折りたたみ信号 $z[n]$ を並べたベクトルである．

上記の議論は信号の一次差分に対しても同様に成り立つ．信号 $f[n]$ の一次差分 $\hat{f}[n]$ を $\hat{f}[n] := f[n] - f[n-1]$ ，剰余信号 $f_\lambda[n]$ の一次差分を $\hat{f}_\lambda[n] := f_\lambda[n] - f_\lambda[n-1]$ と定義すると，式 (5) の両辺の一次差分をとった式は

$$\hat{f}_\lambda[n] = \hat{f}[n] + \hat{z}[n] \quad (14)$$

となる．このとき， $\hat{f}[n]$ は $f[n]$ と同様に帯域制限されるので，

$$\hat{\mathbf{F}}_\lambda = \mathbf{V}\hat{\mathbf{z}} \quad (15)$$

も成り立つ．

3-2 ISTA を用いた剰余信号の復元

LASSO- B^2R^2 では，まず式 (15) と $\hat{\mathbf{z}}$ のスパース性を用いて $\hat{\mathbf{z}}$ の推定を行う．その結果の累積和をとることで \mathbf{z} を求め，式 (5) に基づいて元の信号 $f[n]$ を復元する．ただし $\hat{\mathbf{F}}_\lambda$ や \mathbf{V} の成分は複素数であるため，実数領域での復元アルゴリズムを構成できるように本稿では式 (15) を実数領域の式へ書き換えて考える． $\hat{\mathbf{F}}_\lambda$ や \mathbf{V} の実部と虚部をそれぞれ並べた行列

$$\tilde{\mathbf{F}}_\lambda = \begin{bmatrix} \text{Re}\hat{\mathbf{F}}_\lambda \\ \text{Im}\hat{\mathbf{F}}_\lambda \end{bmatrix}, \tilde{\mathbf{V}} = \begin{bmatrix} \text{Re}\mathbf{V} \\ \text{Im}\mathbf{V} \end{bmatrix} \quad (16)$$

を用いて式 (15) を書き直すと，

$$\tilde{\mathbf{F}}_\lambda = \tilde{\mathbf{V}}\hat{\mathbf{z}} \quad (17)$$

となる．

$\hat{\mathbf{z}}$ の推定では，スパースな未知ベクトルの推定手法である LASSO [7], [8] に基づいて最適化問題

$$\text{minimize}_{\hat{\mathbf{z}} \in \mathbb{R}^N} \left\{ \frac{1}{2} \|\tilde{\mathbf{F}}_\lambda - \tilde{\mathbf{V}}\hat{\mathbf{z}}\|_2^2 + \gamma \|\hat{\mathbf{z}}\|_1 \right\} \quad (18)$$

を解く．式 (18) の第一項は観測値 $\tilde{\mathbf{F}}_\lambda$ と $\tilde{\mathbf{V}}\hat{\mathbf{z}}$ の差であり，第二項は $\hat{\mathbf{z}}$ のスパース性を活用するための正則化項である． γ (> 0) は正則化項の重みを表すパラメータである．

式 (18) の最適化問題に対して ISTA (Iterative Shrinkage Thresholding Algorithm) [7], [9] を用いることで得られる復元アルゴリズムを Algorithm 1 に示す．ISTA は，

$$\hat{\mathbf{z}}^{(i+1)} = S_{\gamma\tau} \left(\hat{\mathbf{z}}^{(i)} - \tau \tilde{\mathbf{V}}^\top (\tilde{\mathbf{V}}\hat{\mathbf{z}}^{(i)} - \tilde{\mathbf{F}}_\lambda) \right) \quad (19)$$

を繰り返すことによって最終的な推定値を得るアルゴリズムである．ここで， i は反復のインデックスを表す．また， $S_{\gamma\tau}$ はソフト閾値関数であり，

$$S_{\gamma\tau}(x) = \text{sign}(x) \max(|x| - \gamma\tau, 0) \quad (20)$$

と定義される．ベクトルが入力になるときは，成分ごとに式 (20) の関数を作用させる．また， τ はステップサイズと呼ばれるパラメータである．[4] では， $\tau = \frac{1}{\|\tilde{\mathbf{V}}\|_2}$ および $\gamma = 0.1 \|\tilde{\mathbf{V}}^\top \tilde{\mathbf{F}}_\lambda\|_\infty$ としている．

式 (19) を反復することで $\hat{\mathbf{z}}$ の推定値が求まるが， $\hat{\mathbf{z}}$ は λ の偶数倍の値しかとらないため，

$$\left\lceil \frac{\lfloor \hat{\mathbf{z}}/\lambda \rfloor}{2} \right\rceil \cdot 2\lambda \quad (21)$$

と丸めたものを最終的な $\hat{\mathbf{z}}$ の推定値とする． \mathbf{z} の各成分は， $z[-1] = 0$ としていることから $z[0] = \hat{z}[0]$, $z[1] = z[0] + \hat{z}[1]$, ..., $z[N-1] = z[N-2] + \hat{z}[N-1]$ のように順番に求められる．Algorithm 1 では，この操作を $\text{cumsum}(\hat{\mathbf{z}})$ で表している．また， $\hat{\mathbf{z}}^{(i)}$ の初期値 $\hat{\mathbf{z}}^{(0)}$ は，[7] では平均 0，分散 1 の正規分布に従う乱数としている．

Algorithm 1 LASSO- B^2R^2 [4]

Require: $f_\lambda[n](0 \leq n < N)$, λ , maxItr , OF , N **Ensure:** $f[n]$

- 1: Construct $\tilde{\mathbf{V}}$
 - 2: Compute $\hat{\mathbf{F}}_\lambda(e^{j\omega T_s})$
 - 3: **Initialize:** $\gamma = 0.1 \left\| \tilde{\mathbf{V}}^\top \hat{\mathbf{F}}_\lambda \right\|_\infty$, $\tau = \frac{1}{\|\tilde{\mathbf{V}}\|_2^2}$, $\text{maxItr}=1000$, and $\hat{\mathbf{z}}^{(0)} \in \mathcal{N}(0, 1)$
 - 4: **for** $i = 0$: maxItr **do**
 - 5: $\hat{\mathbf{z}}^{(i+1)} = S_{\gamma\tau}(\hat{\mathbf{z}}^{(i)} - \tau \tilde{\mathbf{V}}^\top (\tilde{\mathbf{V}} \hat{\mathbf{z}}^{(i)} - \hat{\mathbf{F}}_\lambda))$
 - 6: **if** $\|\hat{\mathbf{z}}^{(i+1)} - \hat{\mathbf{z}}^{(i)}\|_2 < 10^{-4}$ **then**
 - 7: $\hat{\mathbf{z}} = \hat{\mathbf{z}}^{(i+1)}$
 - 8: **exit**
 - 9: **end if**
 - 10: **end for**
 - 11: $\hat{\mathbf{z}} \leftarrow \left\lceil \frac{|\hat{\mathbf{z}}/\lambda|}{2} \right\rceil \cdot 2\lambda$
 - 12: $\mathbf{z} \leftarrow \text{cumsum}(\hat{\mathbf{z}})$
 - 13: $f[n] \leftarrow f_\lambda[n] - z[n]$
-

4 提案復元手法

4-1 制約付き SSR 最適化

剰余サンプリングによって得られた折りたたみ信号の一次差分 $\hat{\mathbf{z}}$ の各成分は、 λ の偶数倍の離散的な値しかとらず、そのほとんどが $0, \pm 2\lambda$ となる。また、 $\hat{\mathbf{z}}$ の要素の合計は、

$$\begin{aligned} \sum_{n=0}^{N-1} \hat{\mathbf{z}}[n] &= \sum_{n=0}^{N-1} z[n] - \sum_{n=0}^{N-1} z[n-1] \\ &= z[N-1] - z[-1] \end{aligned} \quad (22)$$

となる。[10] と同様に元信号が有限エネルギー信号であることを仮定すると、有限エネルギー信号の時間領域減衰特性より、 $\lim_{t \rightarrow \infty} f(t) = 0$ が導ける。このことから、元信号の終端が折りたたまれていない、つまり $[-\lambda, \lambda]$ の範囲内にあることを仮定すると、 $z[N-1] = 0$ となる。また、 $z[-1] = 0$ としているため、終端が折りたたまれていないという仮定のもとでは

$$\sum_{n=0}^{N-1} \hat{\mathbf{z}}[n] = 0 \quad (23)$$

となり、 $\hat{\mathbf{z}}$ の要素の合計が 0 になることがわかる。しかし、従来の手法では、 $\hat{\mathbf{z}}$ の成分に 0 が多いという情報しか用いておらず、 $\hat{\mathbf{z}}$ の成分の離散性や $\hat{\mathbf{z}}$ の成分の総和が 0 になるという性質を考慮していない。

本研究では、これらの事前情報を活用した $\hat{\mathbf{z}}$ の復元手法として、離散値ベクトルの復元のための SSR 最適化問題 [5] のアイデアを応用した制約付き SSR 最適化を提案する。SSR 最適化問題で提案されている正則化項と、事前情報を用いた制約を用いて $\hat{\mathbf{z}}$ の推定問題を定式化すると、

$$\text{minimize}_{\hat{\mathbf{z}} \in \mathbb{R}^N} \left\{ \frac{1}{2} \|\tilde{\mathbf{F}}_\lambda - \tilde{\mathbf{V}} \hat{\mathbf{z}}\|_2^2 + \sum_{l=1}^3 q_l h_l(\hat{\mathbf{z}} - r_l \mathbf{1}) \right\} \quad (24)$$

$$\text{subject to } \mathbf{1}^\top \hat{\mathbf{z}} = 0$$

となる。ここで、関数 $h_l(\cdot)$ はスパース正則化の関数であり、 $h_l(\cdot)$ の近接写像は効率的に計算できるもの

とする. r_l は \hat{z} の要素がとりうる値であり, 差分の多くは 0 か $\pm 2\lambda$ になることから, 今回は $r_1 = 0, r_2 = 2\lambda, r_3 = -2\lambda$ とする. q_l は各正則化項に対する重みを表す. 従来の手法では, \hat{z} のスパース性に注目しているため正則化項は $h_l(\hat{z})$ のみを用いていたが, 提案手法では \hat{z} の離散性も活用するため正則化項として $h_l(\hat{z} - r_l \mathbf{1})$ の和を用いる. これにより, $\hat{z} - r_l \mathbf{1}$ がいくつかの零成分を持つという性質を活用する. 制約条件の $\mathbf{1}^\top \hat{z} = 0$ は, 式 (23) に基づく制約を表している.

4-2 制約付き SSR 最適化の ADMM による解法

式 (24) の最適化問題に対する ADMM [11] に基づくアルゴリズムを導出する. まず, 制約条件を目的関数へ移動させて ADMM を適用できる形にするため, 閉凸集合を

$$C := \{\hat{z} \mid \mathbf{1}^\top \hat{z} = 0\} \quad (25)$$

と定義し指示関数

$$\iota_C(\hat{z}) = \begin{cases} 0 & (\hat{z} \in C) \\ \infty & (\hat{z} \notin C) \end{cases} \quad (26)$$

を導入する. 指示関数を用いると, 式 (24) は

$$\underset{\hat{z} \in \mathbb{R}^N}{\text{minimize}} \left\{ \frac{1}{2} \|\tilde{\mathbf{F}}_\lambda - \tilde{\mathbf{V}}\hat{z}\|_2^2 + \sum_{l=1}^3 q_l h_l(\hat{z} - r_l \mathbf{1}) + \iota_C(\hat{z}) \right\} \quad (27)$$

のように書き換えられる. 次に, 式 (27) を新たな変数 $\mathbf{u}_1, \dots, \mathbf{u}_4$ を用いて

$$\underset{\hat{z}, \mathbf{u}_l \in \mathbb{R}^N}{\text{minimize}} \left\{ \frac{1}{2} \|\tilde{\mathbf{F}}_\lambda - \tilde{\mathbf{V}}\hat{z}\|_2^2 + \sum_{l=1}^3 q_l h_l(\mathbf{u}_l - r_l \mathbf{1}) + \iota_C(\mathbf{u}_4) \right\} \quad \text{subject to} \quad \hat{z} = \mathbf{u}_l (l = 1, \dots, 4) \quad (28)$$

のように変形する. さらに $\mathbf{u} = [\mathbf{u}_1^\top \ \dots \ \mathbf{u}_4^\top]^\top \in \mathbb{R}^{4N}$, $\Phi = [\mathbf{I}_N \ \mathbf{I}_N \ \mathbf{I}_N \ \mathbf{I}_N]^\top$, $f(\hat{z}) = \frac{1}{2} \|\tilde{\mathbf{F}}_\lambda - \tilde{\mathbf{V}}\hat{z}\|_2^2$, $g(\mathbf{u}) = \sum_{l=1}^3 q_l h_l(\mathbf{u}_l - r_l \mathbf{1}) + \iota_C(\mathbf{u}_4)$ と置くと,

$$\underset{\hat{z} \in \mathbb{R}^N, \mathbf{u} \in \mathbb{R}^{4N}}{\text{minimize}} \{f(\hat{z}) + g(\mathbf{u})\} \quad \text{subject to} \quad \Phi \hat{z} = \mathbf{u} \quad (29)$$

となり, ADMM を適用できる形になる. 式 (29) に対する ADMM の反復式は

$$\hat{z}^{(i+1)} = \arg \min_{\hat{z} \in \mathbb{R}^N} \left\{ f(\hat{z}) + \frac{\rho}{2} \|\Phi \hat{z} - \mathbf{u}^{(i)} + \mathbf{y}^{(i)}\|_2^2 \right\} \quad (30)$$

$$\mathbf{u}^{(i+1)} = \text{prox}_{\frac{1}{\rho}g}(\Phi \hat{z}^{(i+1)} + \mathbf{y}^{(i)}) \quad (31)$$

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + \Phi \hat{z}^{(i+1)} - \mathbf{u}^{(i+1)} \quad (32)$$

となる. ここで, ρ は ADMM のパラメータであり, $\mathbf{y}^{(i)} \in \mathbb{R}^{4N}$ である.

式 (30) の更新式を導出するため,

$$\frac{\partial}{\partial \hat{z}} \left\{ f(\hat{z}) + \frac{\rho}{2} \|\Phi \hat{z} - \mathbf{u}^{(i)} + \mathbf{y}^{(i)}\|_2^2 \right\} = 0 \quad (33)$$

を \hat{z} について解くと,

$$\hat{z}^{(i+1)} = (4\rho \mathbf{I}_N + \tilde{\mathbf{V}}^\top \tilde{\mathbf{V}})^{-1} \left(\rho \sum_{l=1}^4 (\mathbf{u}_l^{(i)} - \mathbf{y}_l^{(i)}) + \tilde{\mathbf{V}}^\top \tilde{\mathbf{F}}_\lambda \right) \quad (34)$$

となる. ここで $\mathbf{u}_l^{(i)}, \mathbf{y}_l^{(i)} \in \mathbb{R}^N$ はそれぞれ $\mathbf{u}^{(i)}, \mathbf{y}^{(i)}$ の l 番目の部分ベクトルを表す.

また, 式 (31) の更新式は, 一般に関数 $\psi(\mathbf{x}) = \phi(\mathbf{x} - \mathbf{z})$ の近接写像が $\text{prox}_\psi(\mathbf{x}) = \mathbf{z} + \text{prox}_\phi(\mathbf{x} - \mathbf{z})$ となるという近接写像の並行移動に関する性質 [12] を用いて,

$$\mathbf{u}^{(i+1)} = \begin{bmatrix} r_1 \mathbf{1} + \text{prox}_{\frac{q_1 h_1}{\rho}}(\mathbf{z}^{(i+1)} + \mathbf{y}_1^{(i)} - r_1 \mathbf{1}) \\ r_2 \mathbf{1} + \text{prox}_{\frac{q_2 h_2}{\rho}}(\mathbf{z}^{(i+1)} + \mathbf{y}_2^{(i)} - r_2 \mathbf{1}) \\ r_3 \mathbf{1} + \text{prox}_{\frac{q_3 h_3}{\rho}}(\mathbf{z}^{(i+1)} + \mathbf{y}_3^{(i)} - r_3 \mathbf{1}) \\ \text{prox}_{\frac{1}{\rho} \iota_C}(\mathbf{z}^{(i+1)} + \mathbf{y}_4^{(i)}) \end{bmatrix} \quad (35)$$

と書ける. 例えば, $h_l(\cdot)$ として $\|\cdot\|_1$ を用いる場合は, 近接写像 $\text{prox}_{\tau\|\cdot\|_1}(\cdot)$ ($\tau > 0$) は式 (20) のソフト閾値関数を用いた処理

$$S_\tau(x) = \text{sign}(x) \max(|x| - \tau, 0) \quad (36)$$

を要素ごとに作用させたものになる. また, 式 (35) の最後の指示関数の近接写像は

$$\text{prox}_{\frac{1}{\rho} \iota_C}(\mathbf{x}) = \arg \min_{\mathbf{y} \in \mathbb{R}^N} \left\{ \frac{\rho}{2} \|\mathbf{x} - \mathbf{y}\|^2 \right\} \quad (37)$$

subject to $\mathbf{1}^\top \mathbf{y} = 0$

と書ける. 式 (37) をラグランジュの未定乗数法で解くことにより,

$$\text{prox}_{\frac{1}{\rho} \iota_C}(\mathbf{x}) = \left(\mathbf{I}_N - \frac{1}{N} \mathbf{1} \mathbf{1}^\top \right) \mathbf{x} \quad (38)$$

が得られる.

制約付き SSR 最適化に基づく提案復元アルゴリズムを Algorithm 2 に示す.

Algorithm 2 Constrained SSR (proposed)

Require: $f_\lambda[n]$ ($0 \leq n < N$), λ , maxItr , OF , N

Ensure: $f[n]$

- 1: Construct $\tilde{\mathbf{V}}$
 - 2: Compute $\hat{F}_\lambda(e^{j\omega T_s})$
 - 3: **Initialize:** $\rho > 0$, maxItr , and $\hat{\mathbf{z}}^{(0)} \in \mathcal{N}(0, 1)$
 - 4: **for** $i = 0$: maxItr **do**
 - 5: $\hat{\mathbf{z}}^{(i+1)} = (4\rho \mathbf{I}_N + \tilde{\mathbf{V}}^\top \tilde{\mathbf{V}})^{-1} (\rho \sum_{l=1}^4 (\mathbf{u}_l^{(i)} - \mathbf{y}_l^{(i)}) + \tilde{\mathbf{V}}^\top \tilde{\mathbf{F}}_\lambda)$
 - 6: $\mathbf{u}_l^{(i+1)} = r_l \mathbf{1} + \text{prox}_{\frac{q_l h_l}{\rho}}(\mathbf{z}^{(i+1)} + \mathbf{y}_l^{(i)} - r_l \mathbf{1})$ ($l = 1, 2, 3$)
 - 7: $\mathbf{u}_4^{(i+1)} = \text{prox}_{\iota_C}(\mathbf{z}^{(i+1)} + \mathbf{y}_4^{(i)})$
 - 8: $\mathbf{y}_l^{(i+1)} = \mathbf{y}_l^i + \hat{\mathbf{z}}^{(i+1)} - \mathbf{u}_l^{(i+1)}$ ($l = 1, \dots, 4$)
 - 9: **end for**
 - 10: $\hat{\mathbf{z}} = \hat{\mathbf{z}}^{(\text{maxItr})}$
 - 11: $\hat{\mathbf{z}} \leftarrow \lceil \frac{\lfloor \hat{\mathbf{z}} / \lambda \rfloor}{2} \rceil \cdot 2\lambda$
 - 12: $\mathbf{z} \leftarrow \text{cumsum}(\hat{\mathbf{z}})$
 - 13: $f[n] \leftarrow f_\lambda[n] - z[n]$
-

5 計算機シミュレーション

5-1 シミュレーション設定

最大周波数 $1/T_s \times 1/\text{OF}$, 最大振幅が 1 のランダムな信号に幅 λ での剰余サンプリングを行い, その結果から元の信号を復元した際の精度を評価する. また, 信号には各 SNR (Signal to Noise Ratio: 信号対

雑音比) に基づいて生成した平均 0 のガウス雑音信号を付加する. このノイズ付加信号に対する復元を 250 回行い, 元信号と復元信号の NMSE

$$NMSE = \frac{\|f - f_{\text{est}}\|_2^2}{\|f\|_2^2} \quad (39)$$

の平均を評価する. 信号やアルゴリズムのパラメータの値は $\lambda = 0.25, N = 1024, T_s = 0.01, \text{maxItr} = 150, q_1 = \frac{40}{3}, q_2 = \frac{10}{3}, q_3 = \frac{10}{3}, \rho = 20$ とした. また, シミュレーションではスパース正則化項として l_1 ノルムを用いる.

5-2 シミュレーション結果

まず, $OF = 4, 6$ にそれぞれ固定し, SNR を $0 \sim 35$ dB にそれぞれ変更したときの復元結果の NMSE を図 6, 7 に示す. 図 6, 7 より, SNR が 15 dB 付近より大きいとき, つまり, ノイズが小さいときに NMSE の値は提案手法の方が小さくなる. これは, 新しく加えた正則化項や制約が機能した結果であると考えられる. しかし, $OF = 4$ のときに比べて, $OF = 6$ のときは従来手法と提案手法の NMSE の値の差が小さい.

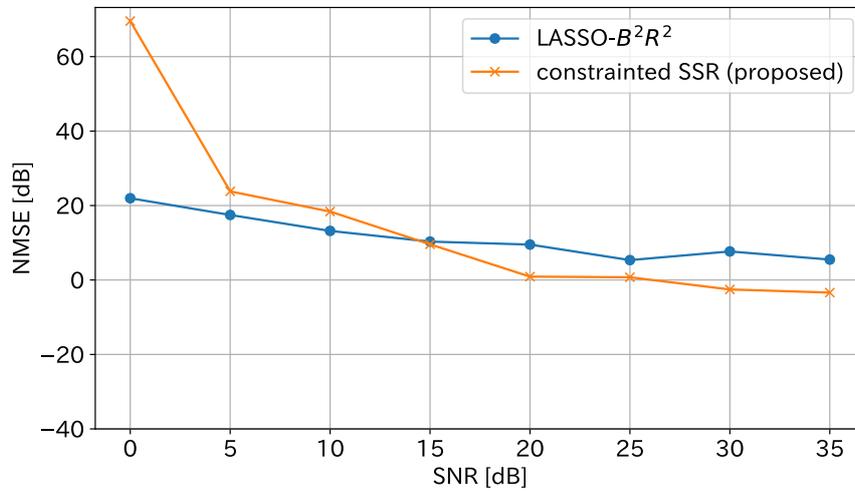


図 6 $OF = 4$ のときの復元精度

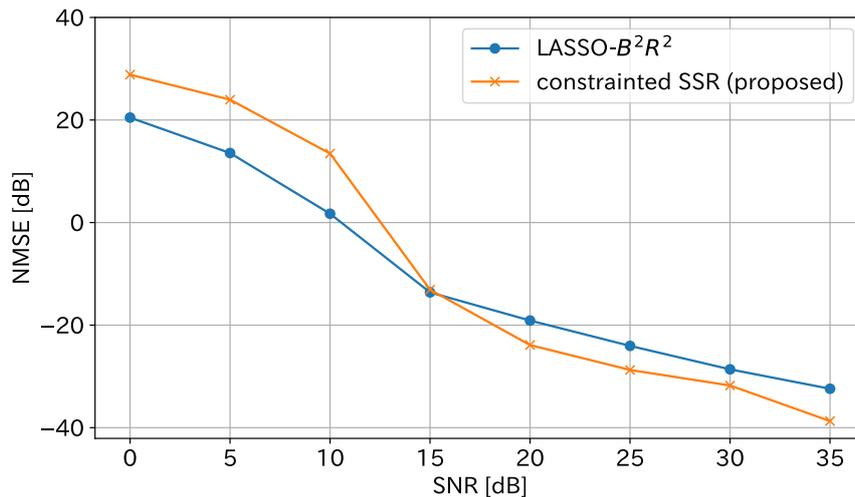


図 7 $OF = 6$ のときの復元精度

どのような OF で提案手法の復元精度が従来手法よりも高いのかを調べるため, SNR = 20 dB に固定して

OF を 3～8 とした場合の NMSE を図 8 に示す. 図 8 より, この場合は OF が 4 から 6 の範囲で提案手法の NMSE の値の方が小さくなり, 優れた復元精度を達成していることがわかる.

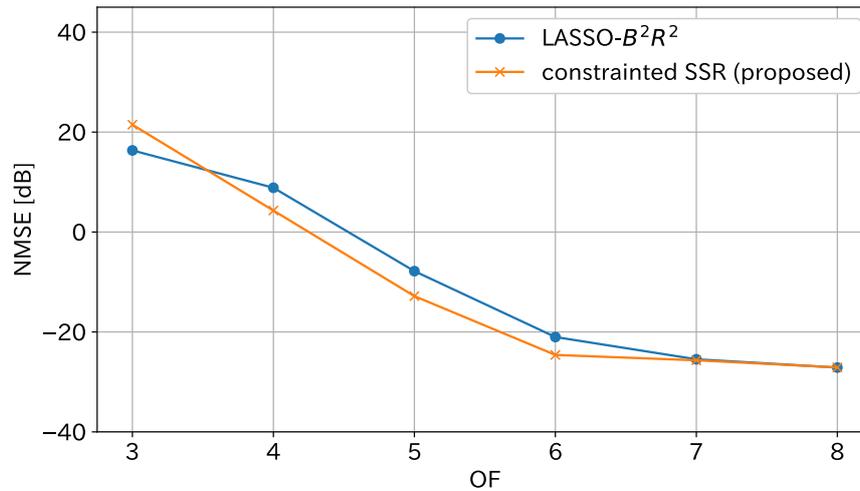


図 8 SNR = 20 のときの復元精度

6 まとめ

本研究では, 剰余サンプリングからの信号の復元において, 剰余信号に関する制約と離散性を活用した最適化問題を提案した. また, ADMM に基づく復元アルゴリズムを導出した. 計算機シミュレーションにより, 提案手法の復元精度は SNR が 15 dB より大きく OF が 4 から 6 の範囲で従来の手法より優れていることがわかった.

【参考文献】

- [1] A. Bhandari, F. Krahmer, and R. Raskar, “On Unlimited Sampling and Reconstruction,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 3827–3839, 2021.
- [2] K. Sasagawa, T. Yamaguchi, M. Haruta, Y. Sunaga, H. Takehara, H. Takehara, T. Noda, T. Tokuda, and J. Ohta, “An Implantable CMOS Image Sensor With Self-Reset Pixels for Functional Brain Imaging,” *IEEE Transactions on Electron Devices*, vol. 63, no. 1, pp. 215–222, Jan. 2016.
- [3] E. Azar, S. Mulleti, and Y. C. Eldar, “Residual Recovery Algorithm for Modulo Sampling,” in *ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022.
- [4] S. B. Shah, S. Mulleti, and Y. C. Eldar, “Lasso-Based Fast Residual Recovery For Modulo Sampling,” in *ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023.
- [5] R. Hayakawa and K. Hayashi, “Discrete-Valued Vector Reconstruction by Optimization with Sum of Sparse Regularizers,” in *2019 27th European Signal Processing Conference (EUSIPCO)*. A Coruna, Spain, Sep. 2019.
- [6] S. Boyd, “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.
- [7] 永原正章, “スパースモデリングのための凸最適化—近接勾配法による高速アルゴリズム,” *システム/制御/情報*, vol. 61, no. 1, pp. 20–28, 2017.
- [8] 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, Jan. 1996.

- [9] P. L. Combettes and V. R. Wajs, “Signal Recovery by Proximal Forward-Backward Splitting,” *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, Jan. 2005.
- [10] E. Romanov and O. Ordentlich, “Above the Nyquist Rate, Modulo Folding Does Not Hurt,” *IEEE Signal Processing Letters*, vol. 26, no. 8, pp. 1167–1171, Aug. 2019.
- [11] 小野峻佑, “近接分離による分散凸最適化—交互方向乗数法に基づくアプローチを中心として—,” *計測と制御*, vol. 55, no. 11, pp. 954–959, 2016.
- [12] P. L. Combettes and J.-C. Pesquet, “Proximal Splitting Methods in Signal Processing,” in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds. New York, NY: Springer New York, 2011, vol. 49, pp. 185–212.

〈発表資料〉

題名	掲載誌・学会名等	発表年月
剰余サンプリングからの信号復元のための制約付き離散値ベクトル再構成	電子情報通信学会 信号処理研究会	2024年8月
剰余サンプリングのための非凸最適化に基づく制約付き離散値ベクトル再構成	信号処理シンポジウム	2024年12月