# Soft Thresholding Using Moore–Penrose Inverse

Bamrung Tausiesakul, *Member, IEEE*, and Krissada Asavaskulkiet, *Member, IEEE*

***Abstract*— The acquisition of a discrete-time signal is an important part of a compressive sensing problem. A high-accuracy algorithm that could bring better signal recovery performance is often called for. In this work, two thresholding algorithms that involve a soft thresholding decision are proposed using the Moore–Penrose inverse. Numerical examples are conducted and illustrate that, in the optimal case, both proposed methods consume the computational time at the same level as the conventional soft homotopy algorithm (SHA). Under no knowledge of the optimal regularization parameter, both methods will perform better than the conventional SHA with less amount of required time for the computation. Taking the nonsparse electroencephalogram signal from a real measurement into account, all soft thresholding algorithms provide nearly the same error performance for several compression ratios, while the proposed methods consume less computational time than the conventional SHA.**

***Index Terms*— Compressive sensing, homotopy algorithm, soft thresholding.**

## I. Introduction

THIS article is an extension of the proceedings paper [1] and makes reuse of the proceedings paper [1]. In science and engineering [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], it is required to recover a desired signal $\boldsymbol{x} \in \mathbb{R}^{N\times 1}$ from a set of observation data or measurement data $\boldsymbol{b} \in \mathbb{R}^{M\times 1}$, where $M \in \mathbb{N}^{1\times 1}$ and $N \in \mathbb{N}^{1\times 1}$ are the lengths of real-valued output data and real-valued input data, respectively. The collection of data is based on a modeling matrix or transfer matrix $\boldsymbol{A} \in \mathbb{R}^{M\times N}$, which either depends on the model or can be chosen beforehand. In Fig. 1, if the perturbation $\boldsymbol{\delta}_b$ is negligible, the linear system can be written as

$$\begin{aligned}\boldsymbol{b} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\delta}_b \\ &\approx \boldsymbol{A}\boldsymbol{x}. \end{aligned} \tag{1}$$

Irrespective of signal structure, the signal of interest can be recovered by solving an optimization problem in the sense of linear least squares (LLS) [2, eq. (4.33)], i.e.,

$$\begin{aligned}\hat{\boldsymbol{x}}_{\text{LLS}} &= \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_2^2 \\ &\quad \text{s.t. } \boldsymbol{x} \in \begin{cases} \{\boldsymbol{u}|\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}\}, & \text{noiseless} \\ \{\boldsymbol{u}|\|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{b}\|_2 < \epsilon\}, & \text{noisy}\end{cases}\end{aligned} \tag{2}$$

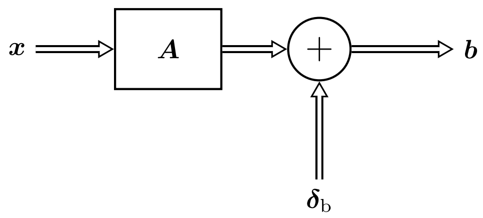

Fig. 1. Noisy linear input–output system.

where $\epsilon$ is the maximal allowable value of the square root of the noise power [13] and $\|\boldsymbol{x}\|_2$ is the $\ell_2$-norm of vector $\boldsymbol{x}$. It is fruitful to note that the matrix $\boldsymbol{A}$ can be chosen to be of full rank, i.e.,

$$\text{rank}(\boldsymbol{A}) = \min(M, N) \tag{3}$$

where $\text{rank}(\boldsymbol{A})$ is the rank of matrix $\boldsymbol{A}$. There are three possible cases for any size of the matrix $\boldsymbol{A}$. The number of provided data $M$ can be less than, equal to, or greater than the number of unknown variables $N$. The LLS problem can, thus, be classified into

$$\hat{\boldsymbol{x}}_{\text{LLS}} = \begin{cases} \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_2^2 \text{ s.t. } \boldsymbol{x} \in \{\boldsymbol{u}|\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}\}, & M < N \\ \arg_{\boldsymbol{x}} \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, & M = N \\ \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2, & M > N. \end{cases} \tag{4}$$

The closed-form solution to (4) is well known as [8, p. 204], [14, eqs. (2.3) and (2.9)], [15, p. 31], [16, p. 98], [17, eq. (12.5)]

$$\hat{\boldsymbol{x}}_{\text{LLS}} = \begin{cases} \boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{b}, & M < N \\ \boldsymbol{A}^{-1}\boldsymbol{b}, & M = N \\ \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{b}, & M > N \end{cases} \tag{5}$$

where $\cdot^{\mathsf{T}}$ is the transpose of a vector or a matrix and $\boldsymbol{X}^{-1}$ is the inverse of a square matrix $\boldsymbol{X}$. The LLS estimate exists when the matrix $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$ or $\boldsymbol{A}$ or $\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}}$ is invertible due to the full-rank assumption of $\boldsymbol{A}$. The solution in (5) has no concern with any special structure of the obtained solution [18]. The LLS estimate $\hat{\boldsymbol{x}}_{\text{LLS}}$ usually returns all nonzero entries to the desired signal $\boldsymbol{x}$.

The measurement of physiological signals, e.g., electrocardiogram (ECG) or heart rate and electromyogram (EMG) or blood pressure, plays a vital role in health monitoring [19]. Wireless body sensor networks (WBSNs) consist of multiple biosensors that create several wireless links. The WBSNs are often applied to modern health monitoring systems because wireless medical sensors allow for the mobility of a patient on a daily life basis [20]. Unfortunately, due to the restriction in

Manuscript received 11 April 2023; revised 22 May 2023; accepted 8 June 2023. Date of publication 26 June 2023; date of current version 14 July 2023. The Associate Editor coordinating the review process was Lin Xu. *(Corresponding author: Krissada Asavaskulkiet.)*

Bamrung Tausiesakul is with the Department of Electrical Engineering, Faculty of Engineering, Srinakharinwirot University, Nakhon Nayok 26120, Thailand (e-mail: bamrungt@g.swu.ac.th).

Krissada Asavaskulkiet is with the Department of Electrical Engineering, Faculty of Engineering, Mahidol University, Nakhon Pathom 73170, Thailand (e-mail: krissada.asa@mahidol.edu).

Digital Object Identifier 10.1109/TIM.2023.3289506

miniature size compared to the human body, most biosensors are subject to small batteries, and thus, limited power supply leads to low transmission rates, limited memory, and limited computational ability. Data compression and data recovery, also known together as compressed sensing, are the techniques that can overcome the above battery-related limitations. Many works, thus, are devoted to compressing the measurement data and recovering the compressed data [21], [22], [23]. In [24], a simple deterministic measurement matrix is proposed to facilitate the hardware implementation. Measured ECG and EMG biosignals are adopted to validate the recovery algorithm.

A microwave imaging system is also adopted in medical applications to detect breast cancer and other biological tissues [25]. The nonionizing radiation reveals its nondestructive testing property. To get a high spatial resolution, one must keep a small step size, which unfortunately leads to a drastic increase in data volume. Compressive sensing, thus, plays an important role in tackling the challenges of this application (see [26]). The microwave imaging system using compressive sensing spans its applications even to synthetic aperture radar (see [27]).

The measurement of current and voltage in smart grid and microgrid power systems needs to spend a long observation time in order to achieve fine frequency resolution or high harmonic estimation accuracy. Alternatively, short observation time is often preferred since it can handle dynamic measurement scenarios and enables measurement transient reduction [28]. The short observation time can be made possible by using a compressive sensing idea [29]. In [30], extensive numerical simulation is conducted for phasor measurement units that are the instruments capable of estimating the synchrophasor, frequency, and rate of change of frequency in power electronic circuitry. In [31], a compressive sensing technique using the $\ell_1$-norm is proposed for harmonic pollution source identification in smart grids. Several simulations are performed on two distribution networks that involve harmonic measurements.

In this work, we pay attention to the data recovery process. It can be coped with the applications stated above straightforwardly. Two iterative computation algorithms are proposed for signal recovery in a compressive sensing problem. Both methods adopt the Moore–Penrose inverse (MPI) in order to improve the signal acquisition error performance given by an iterative computation algorithm using a soft thresholding decision.

## II. Compressive Sensing

In this section, we consider several compressive sensing algorithms related to the $\ell_1$-norm. A least-squares criterion with an $\ell_1$-norm regularization (LS-L1R) is given by [32, Sec. 1.3.2]

$$\hat{\boldsymbol{x}}_{\text{LS-L1R}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \|\boldsymbol{x}\|_1 \tag{6}$$

where $\|\boldsymbol{x}\|_1$ is the $\ell_1$-norm of vector $\boldsymbol{x}$. In [33], an $\ell_1$-norm constrained least-squares estimate is proposed as least absolute shrinkage and selection operator (LASSO) in statistics, i.e.,

$$\hat{\boldsymbol{x}}_{\text{LASSO}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \quad \text{s.t.} \quad \boldsymbol{x} \in \{\boldsymbol{u} | \|\boldsymbol{u}\|_1 \le \tau\} \tag{7}$$

where $\tau \in \mathbb{R}_+^{1\times 1}$ is a positive number. Also known as basis pursuit denoising [18], an optimization problem similar to (7), is formulated by $\ell_1$-norm regularized least-squares (L1RLS) criterion, which is given by

$$\hat{\boldsymbol{x}}_{\text{L1RLS}} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \tag{8}$$

where $\lambda \in \mathbb{R}_+^{1\times 1}$ is a regularization parameter. The solution to (8) is the same as that of (6) when the regularization parameter $\lambda$ is 0.5. The $\ell_1$-norm minimization is given in a convex optimization framework by [34]

$$\hat{\boldsymbol{x}}_{\ell_1} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{s.t. } \boldsymbol{x} \in \{\boldsymbol{u} | \boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}\}. \tag{9}$$

The above optimization framework is also known as basis pursuit [18]. When the regularization parameter $\lambda$ tends to zero, it is found that [35, Proposition 15.1], [36]

$$\lim_{\lambda \to 0_+} \hat{\boldsymbol{x}}_{\text{L1RLS}} = \hat{\boldsymbol{x}}_{\ell_1}. \tag{10}$$

### A. Soft Thresholding Decision

A scalar minimization problem is formulated by [18, eq. (5.2)]

$$s_a(x) = \arg\min_{u} (x-u)^2 + a|u| \tag{11}$$

where $a$ is a constant and $|\cdot|$ is the absolute value operator. Its solution turns out to be a soft thresholding operator [37], given by

$$s_a(x) = \begin{cases} 0, & |x| < a \\ \text{sign}(x)(|x| - a), & \text{otherwise} \end{cases} \tag{12}$$

where sign$(\cdot)$ is the sign of a real number $\cdot$, given by

$$\text{sign}(x) = \begin{cases} \dfrac{x}{|x|}, & x \neq 0 \\ 0, & x = 0. \end{cases} \tag{13}$$

In a short form, one can see that [38]

$$\begin{aligned} s_a(x) &= \text{sign}(x)\max\{|x| - a, 0\} \\ &= \text{sign}(x)(|x| - a)_+ \end{aligned} \tag{14}$$

where $(x)_+ = \max\{x, 0\}$ is the positive-definite operator.

### B. Soft Homotopy Algorithm

If (1) is multiplied by $\boldsymbol{A}^{\mathsf{T}}$ along the left direction, one can attain

$$\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{x} \approx \boldsymbol{A}^{\mathsf{T}}\boldsymbol{b}. \tag{15}$$

If (15) is added by $\boldsymbol{x}$'s in its both sides, one can see that the result $\boldsymbol{x} + \boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{x} \approx \boldsymbol{x} + \boldsymbol{A}^{\mathsf{T}}\boldsymbol{b}$ leads to

$$\begin{aligned} \boldsymbol{x} &\approx \boldsymbol{x} + \boldsymbol{A}^{\mathsf{T}}\boldsymbol{b} - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{x} \\ &= \boldsymbol{x} + \boldsymbol{A}^{\mathsf{T}}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}). \end{aligned} \tag{16}$$

The result in (16) discloses a possible fixed-point iteration. The iterative method based on (16) is known as Landweber's algorithm [39], which is one of the simplest iterative methods for solving the normal equation in (15).

The homotopy technique is a method that tries to find a minimizer of the problem in (9). The composite gradient mapping is used to update the $i$th iterated solution from the previous iteration by [40], [41]

$$\begin{aligned}\hat{\boldsymbol{x}}[i] &= \arg\min_{\boldsymbol{x}} \langle \boldsymbol{x}, \boldsymbol{A}^{\mathsf{T}}\big(\boldsymbol{A}\hat{\boldsymbol{x}}[i-1]-\boldsymbol{b}\big)\rangle \\ &\qquad + \frac{1}{2}\|\boldsymbol{x}-\hat{\boldsymbol{x}}[i-1]\|_2^2 + \lambda_{i-1}\|\boldsymbol{x}\|_1 \\ &= \boldsymbol{s}_{\lambda_{i-1}}\big(\hat{\boldsymbol{x}}[i-1]+\boldsymbol{A}^{\mathsf{T}}\big(\boldsymbol{b}-\boldsymbol{A}\hat{\boldsymbol{x}}[i-1]\big)\big) \end{aligned} \tag{17}$$

where $\langle \boldsymbol{u}, \boldsymbol{v}\rangle = \boldsymbol{u}^{\mathsf{T}}\boldsymbol{v}$ is the inner product of $\boldsymbol{u}$ and $\boldsymbol{v}$, and $\boldsymbol{s}_a(\cdot) : \mathbb{R}^{N\times 1} \mapsto \mathbb{R}^{N\times 1}$ is the elementwise vector operator of the same size as input vector $\cdot$, which is given by

$$\boldsymbol{s}_a(\boldsymbol{x}) = \begin{bmatrix} s_a(x_1) & s_a(x_2) & \cdots & s_a(x_N)\end{bmatrix}^{\mathsf{T}} \tag{18}$$

with the soft thresholding operator $s_a(\cdot)$ discussed in (12) and (14). Note that the argument inside the soft thresholding vector in (17) is similar to the right-hand side of (16). One may imply that an iterative computation of (17) is equivalent to a soft thresholding version of the fixed-point iteration indicated by (16). Let us introduce the following:

1) the pseudo-$\ell_0$-norm of the vector $\boldsymbol{x}$ as $\lim_{p\to 0}\|\boldsymbol{x}\|_p^p$, which is equal to the number of nonzero elements in $\boldsymbol{x}$ [16, p. 4], i.e.,

$$\lim_{p\to 0}\|\boldsymbol{x}\|_p^p = |\{x_n | x_n \neq 0, n \in \{1,2,\ldots,N\}\}| \tag{19}$$

where $\|\cdot\|_p$ is the $\ell_p$-norm, given by

$$\|\boldsymbol{z}\|_p = \big(|z_1|^p + |z_2|^p + \cdots + |z_N|^p\big)^{\frac{1}{p}}. \tag{20}$$

$|\cdot|$ is the absolute value of number $\cdot$ or the cardinality of set $\cdot$;

2) the infinity norm as $\|\boldsymbol{x}\|_\infty$, which is equal to the maximum of the absolute values of the elements in $\boldsymbol{x}$, i.e.,

$$\|\boldsymbol{x}\|_\infty = \max_{n\in\{1,2,\ldots,N\}} |x_n|. \tag{21}$$

An iterative computation based on the result of (17) can be summarized in Algorithm 1.

```
Algorithm 1 SHA [41]
Input: $\boldsymbol{A} \in \mathbb{R}^{M\times N}$, $\boldsymbol{b} \in \mathbb{R}^{M\times 1}$, $K \in \mathbb{N}^{1\times 1}$, $\gamma \in (0,1)$, and
    $N_{\max} \in \mathbb{N}^{1\times 1}$
Output: $\hat{\boldsymbol{x}}_{\text{SHA}} \in \mathbb{R}^{N\times 1}$
  $\hat{\boldsymbol{x}}[0] \leftarrow \boldsymbol{0}$
  $\lambda_0 \leftarrow \|\boldsymbol{A}^{\mathsf{T}}\boldsymbol{b}\|_\infty$
  $i \leftarrow 0$
  while $\lim_{p\to 0}\|\hat{\boldsymbol{x}}[i]\|_p^p \le 2K \wedge i \le N_{\max}$ do
    $i \leftarrow i+1$
    $\hat{\boldsymbol{x}}[i] \leftarrow \boldsymbol{s}_{\lambda_{i-1}}\big(\hat{\boldsymbol{x}}[i-1] + \boldsymbol{A}^{\mathsf{T}}(\boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}[i-1])\big)$
    $\lambda_i \leftarrow \gamma\lambda_{i-1}$
  end while
  return $\hat{\boldsymbol{x}}[i]$
```

Algorithm 1 is known in [41] as a simple homotopy algorithm, which might be coined in the sequel as soft homotopy algorithm (SHA).

### C. Soft Homotopy Algorithms Using Moore–Penrose Inverse

Instead of multiplying (1) by $\boldsymbol{A}^{\mathsf{T}}$ along the left direction and yielding (15), one may multiply (1) by $\boldsymbol{A}^{\mathsf{MP}}$, which results in

$$\boldsymbol{A}^{\mathsf{MP}}\boldsymbol{A}\boldsymbol{x} \approx \boldsymbol{A}^{\mathsf{MP}}\boldsymbol{b} \tag{22}$$

where $\boldsymbol{A}^{\mathsf{MP}}$ is the MPI, which is given by

$$\boldsymbol{A}^{\mathsf{MP}} = \boldsymbol{A}^{\mathsf{T}}\big(\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}}\big)^{-1}. \tag{23}$$

Note that the data on both sides of (22) are weighted by $(\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}})^{-1}$ compared to the data on both sides of (15). The weighting method would bring more numerical stability. If one adds $\boldsymbol{x}$ to both sides of (22), the outcome $\boldsymbol{x} + \boldsymbol{A}^{\mathsf{MP}}\boldsymbol{A}\boldsymbol{x} \approx \boldsymbol{x} + \boldsymbol{A}^{\mathsf{MP}}\boldsymbol{b}$ leads to

$$\begin{aligned}\boldsymbol{x} &\approx \boldsymbol{x} + \boldsymbol{A}^{\mathsf{MP}}\boldsymbol{b} - \boldsymbol{A}^{\mathsf{MP}}\boldsymbol{A}\boldsymbol{x} \\ &= \boldsymbol{x} + \boldsymbol{A}^{\mathsf{MP}}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}).\end{aligned} \tag{24}$$

The result of (24) offers another possible fixed-point iteration. It serves as the basis for the algorithm design in what follows. In [1], the term $\boldsymbol{A}^{\mathsf{T}}$ of the updating solution $\hat{\boldsymbol{x}}[i] \leftarrow \boldsymbol{s}_{\lambda_{i-1}}(\hat{\boldsymbol{x}}[i-1]+\boldsymbol{A}^{\mathsf{T}}(\boldsymbol{b}-\boldsymbol{A}\hat{\boldsymbol{x}}[i-1]))$ in Algorithm 1 is replaced by the MPI in (23). Similar to Algorithm 1, an iterative computation is proposed as follows.

```
Algorithm 2 SHA-MPI [1]
Input: $\boldsymbol{A} \in \mathbb{R}^{M\times N}$, $\boldsymbol{b} \in \mathbb{R}^{M\times 1}$, $K \in \mathbb{N}^{1\times 1}$, $\gamma \in (0,1)$, and
    $N_{\max} \in \mathbb{N}^{1\times 1}$
Output: $\hat{\boldsymbol{x}}_{\text{SHA-MPI}} \in \mathbb{R}^{N\times 1}$
  $\hat{\boldsymbol{x}}[0] \leftarrow \boldsymbol{0}$
  $\lambda_0 \leftarrow \|\boldsymbol{A}^{\mathsf{T}}\boldsymbol{b}\|_\infty$
  $i \leftarrow 0$
  while $\lim_{p\to 0}\|\hat{\boldsymbol{x}}[i]\|_p^p \le 2K \wedge i \le N_{\max}$ do
    $i \leftarrow i+1$
    $\hat{\boldsymbol{x}}[i] \leftarrow \boldsymbol{s}_{\lambda_{i-1}}\big(\hat{\boldsymbol{x}}[i-1] + \boldsymbol{A}^{\mathsf{MP}}(\boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}[i-1])\big)$
    $\lambda_i \leftarrow \gamma\lambda_{i-1}$
  end while
  return $\hat{\boldsymbol{x}}[i]$
```

The above algorithm can be denoted by SHA with MPI (SHA-MPI). Furthermore, it is worth noting that one of the stopping criteria in the SHA is $\lim_{p\to 0}\|\boldsymbol{x}[i]\|_p^p \le 2K$. In the theoretical case, the number of nonzero elements should satisfy

$$\lim_{p\to 0}\|\boldsymbol{x}_0\|_p^p = K \tag{25}$$

where $\boldsymbol{x}_0$ is the true value of $\boldsymbol{x}$. Therefore, a relaxed stopping criterion could be $\lim_{p\to 0}\|\hat{\boldsymbol{x}}[i]\|_p^p \le K$, which inspires the following algorithm.

Note that the soft minimal homotopy algorithm with the MPI (SMHA-MPI) should have less computational burden than the SHA-MPI due to a relaxed stopping criterion $\lim_{p\to 0}\|\hat{\boldsymbol{x}}[i]\|_p^p \le K$.

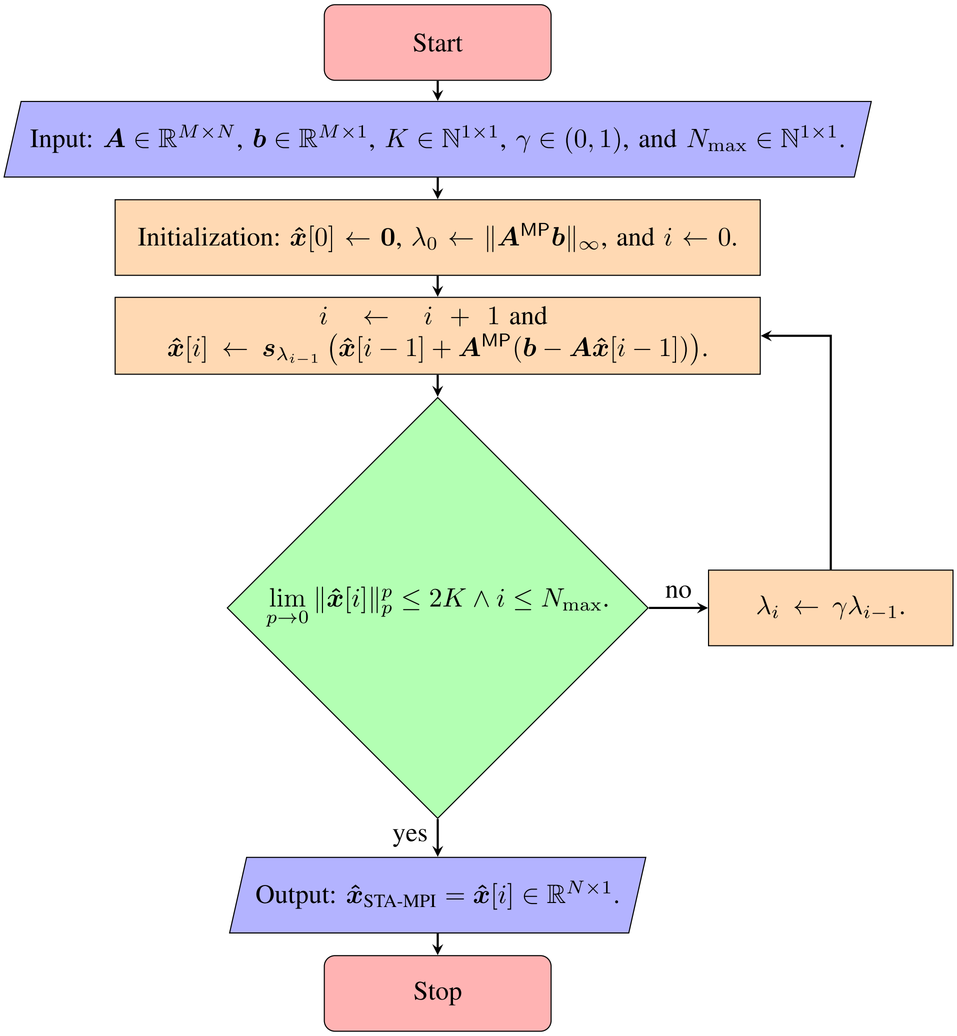
Start
Input: $\boldsymbol{A} \in \mathbb{R}^{M\times N}$, $\boldsymbol{b} \in \mathbb{R}^{M\times 1}$, $K \in \mathbb{N}^{1\times 1}$, $\gamma \in (0,1)$, and $N_{\max} \in \mathbb{N}^{1\times 1}$.
Initialization: $\hat{\boldsymbol{x}}[0] \leftarrow \boldsymbol{0}$, $\lambda_0 \leftarrow \|\boldsymbol{A}^{\mathsf{MP}}\boldsymbol{b}\|_\infty$, and $i \leftarrow 0$.
$i \leftarrow i + 1$ and
$\hat{\boldsymbol{x}}[i] \leftarrow \boldsymbol{s}_{\lambda_{i-1}}\left(\hat{\boldsymbol{x}}[i-1] + \boldsymbol{A}^{\mathsf{MP}}(\boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}[i-1])\right)$.
$\lim_{p\to 0} \|\hat{\boldsymbol{x}}[i]\|_p^p \leq 2K \wedge i \leq N_{\max}$.
no
$\lambda_i \leftarrow \gamma\lambda_{i-1}$.
yes
Output: $\hat{\boldsymbol{x}}_{\text{STA-MPI}} = \hat{\boldsymbol{x}}[i] \in \mathbb{R}^{N\times 1}$.
Stop

Fig. 2. STA-MPI framework corresponding to Algorithm 4.

**Algorithm 3** SMHA-MPI [1]

```
Input: $\boldsymbol{A} \in \mathbb{R}^{M\times N}$, $\boldsymbol{b} \in \mathbb{R}^{M\times 1}$, $K \in \mathbb{N}^{1\times 1}$, $\gamma \in (0,1)$, and
  $N_{\max} \in \mathbb{N}^{1\times 1}$
Output: $\hat{\boldsymbol{x}}_{\text{SMHA-MPI}} \in \mathbb{R}^{N\times 1}$
  $\hat{\boldsymbol{x}}[0] \leftarrow \boldsymbol{0}$
  $\lambda_0 \leftarrow \|\boldsymbol{A}^{\mathsf{T}}\boldsymbol{b}\|_\infty$
  $i \leftarrow 0$
  while $\lim_{p\to 0} \|\hat{\boldsymbol{x}}[i]\|_p^p \leq K \wedge i \leq N_{\max}$ do
    $i \leftarrow i + 1$
    $\hat{\boldsymbol{x}}[i] \leftarrow \boldsymbol{s}_{\lambda_{i-1}}\left(\hat{\boldsymbol{x}}[i-1] + \boldsymbol{A}^{\mathsf{MP}}(\boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}[i-1])\right)$
    $\lambda_i \leftarrow \gamma\lambda_{i-1}$
  end while
  return $\hat{\boldsymbol{x}}[i]$
```

**Algorithm 4** STA-MPI

```
Input: $\boldsymbol{A} \in \mathbb{R}^{M\times N}$, $\boldsymbol{b} \in \mathbb{R}^{M\times 1}$, $K \in \mathbb{N}^{1\times 1}$, $\gamma \in (0,1)$, and
  $N_{\max} \in \mathbb{N}^{1\times 1}$
Output: $\hat{\boldsymbol{x}}_{\text{STA-MPI}} \in \mathbb{R}^{N\times 1}$
  $\hat{\boldsymbol{x}}[0] \leftarrow \boldsymbol{0}$
  $\lambda_0 \leftarrow \|\boldsymbol{A}^{\mathsf{MP}}\boldsymbol{b}\|_\infty$
  $i \leftarrow 0$
  while $\lim_{p\to 0} \|\hat{\boldsymbol{x}}[i]\|_p^p \leq 2K \wedge i \leq N_{\max}$ do
    $i \leftarrow i + 1$
    $\hat{\boldsymbol{x}}[i] \leftarrow \boldsymbol{s}_{\lambda_{i-1}}\left(\hat{\boldsymbol{x}}[i-1] + \boldsymbol{A}^{\mathsf{MP}}(\boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}[i-1])\right)$
    $\lambda_i \leftarrow \gamma\lambda_{i-1}$
  end while
  return $\hat{\boldsymbol{x}}[i]$
```

### *D. Soft Thresholding Algorithms Using Moore–Penrose Inverse*

In this work, two additional algorithms are proposed by invoking the MPI also at the initialization of the regularization parameter $\lambda$.

The flowchart of the STA-MPI is shown in Fig. 2.

Similar to Algorithm 3, we propose a relaxed version of Algorithm 4 as follows.

The flowchart of the SMTA-MPI is given in Fig. 3.

## III. Numerical Examples

All numerical simulation in this work is conducted using Python language.

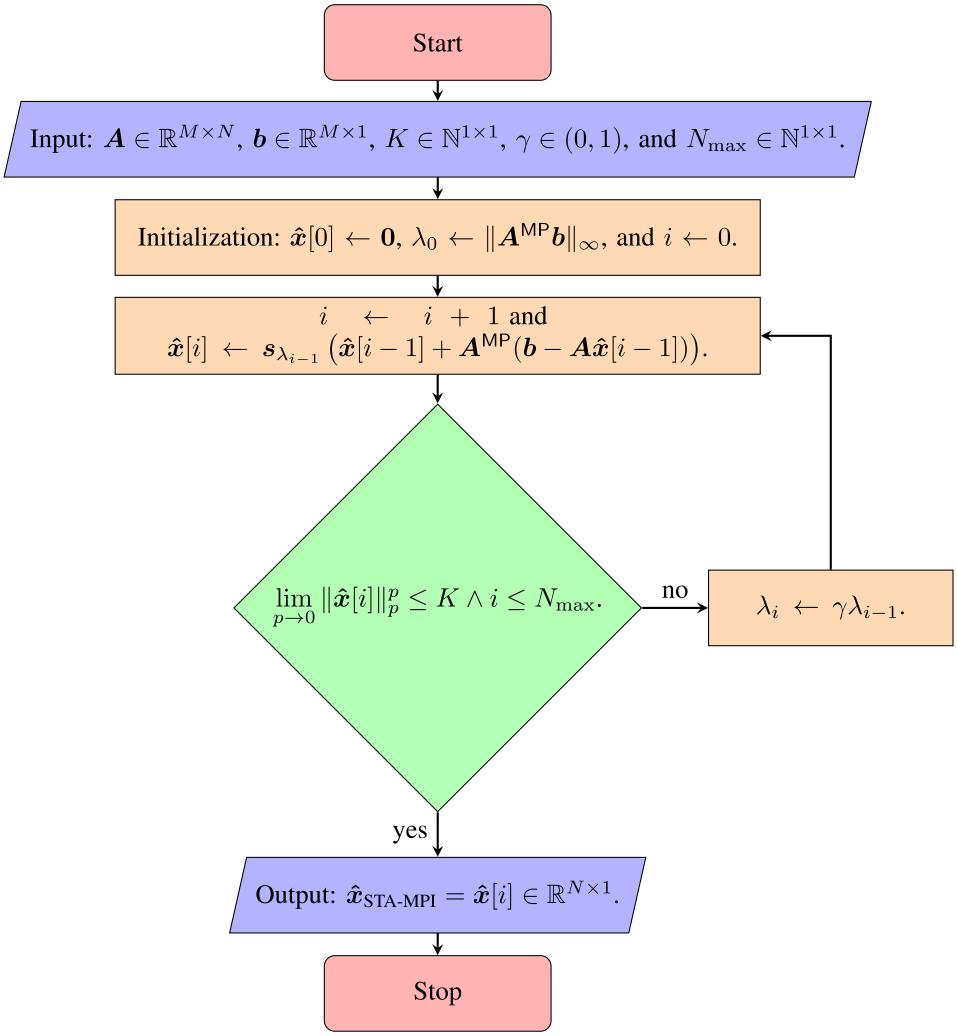
Start
Input: $\boldsymbol{A} \in \mathbb{R}^{M\times N}$, $\boldsymbol{b} \in \mathbb{R}^{M\times 1}$, $K \in \mathbb{N}^{1\times 1}$, $\gamma \in (0,1)$, and $N_{\max} \in \mathbb{N}^{1\times 1}$.
Initialization: $\hat{\boldsymbol{x}}[0] \leftarrow \boldsymbol{0}$, $\lambda_0 \leftarrow \|\boldsymbol{A}^{\mathsf{MP}}\boldsymbol{b}\|_\infty$, and $i \leftarrow 0$.
$i \leftarrow i + 1$ and
$\hat{\boldsymbol{x}}[i] \leftarrow \boldsymbol{s}_{\lambda_{i-1}}\left(\hat{\boldsymbol{x}}[i-1] + \boldsymbol{A}^{\mathsf{MP}}(\boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}[i-1])\right)$.
$\lim_{p\to 0} \|\hat{\boldsymbol{x}}[i]\|_p^p \le K \wedge i \le N_{\max}$.
no
$\lambda_i \leftarrow \gamma\lambda_{i-1}$.
yes
Output: $\hat{\boldsymbol{x}}_{\text{STA-MPI}} = \hat{\boldsymbol{x}}[i] \in \mathbb{R}^{N\times 1}$.
Stop

Fig. 3. SMTA-MPI framework corresponding to Algorithm 5.

**Algorithm 5** Soft Minimal Thresholding Algorithm With MPI

```
Input: $\boldsymbol{A} \in \mathbb{R}^{M\times N}$, $\boldsymbol{b} \in \mathbb{R}^{M\times 1}$, $K \in \mathbb{N}^{1\times 1}$, $\gamma \in (0,1)$, and
    $N_{\max} \in \mathbb{N}^{1\times 1}$
Output: $\hat{\boldsymbol{x}}_{\text{SMTA-MPI}} \in \mathbb{R}^{N\times 1}$
    $\hat{\boldsymbol{x}}[0] \leftarrow \boldsymbol{0}$
    $\lambda_0 \leftarrow \|\boldsymbol{A}^{\mathsf{MP}}\boldsymbol{b}\|_\infty$
    $i \leftarrow 0$
    while $\lim_{p\to 0} \|\hat{\boldsymbol{x}}[i]\|_p^p \le K \wedge i \le N_{\max}$ do
      $i \leftarrow i + 1$
      $\hat{\boldsymbol{x}}[i] \leftarrow \boldsymbol{s}_{\lambda_{i-1}}\left(\hat{\boldsymbol{x}}[i-1] + \boldsymbol{A}^{\mathsf{MP}}(\boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}[i-1])\right)$
      $\lambda_i \leftarrow \gamma\lambda_{i-1}$
    end while
    return $\hat{\boldsymbol{x}}[i]$
```

### A. Random Number Generation

The true value $\boldsymbol{x}_0$ is assumed to follow an identical and independent real-valued Gaussian distribution with zero mean and normalized variance, i.e.,

$$\boldsymbol{x}_0 \sim \mathcal{N}_{\mathbb{R}}\left(\boldsymbol{0}, \frac{1}{\sqrt{K}}\boldsymbol{I}_N\right). \tag{26}$$

The transfer matrix $\boldsymbol{A}$ is assumed to have an identical and independent real-valued Gaussian distribution with zero mean and normalized variance, i.e.,

$$\boldsymbol{A} = \begin{bmatrix}\boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_N\end{bmatrix} \tag{27a}$$

$$\boldsymbol{a}_n \sim \mathcal{N}_{\mathbb{R}}\left(\boldsymbol{0}, \frac{1}{\sqrt{M}}\boldsymbol{I}_M\right), \quad n \in \{1, 2, \ldots, N\}. \tag{27b}$$

As $K < N$ holds for a sparse signal vector, the $K$ locations of all $K$ nonzero elements in $\boldsymbol{x}_0$ are assumed to have an equal probability on all $N$ possible positions.

The random signal generation in this manner can cover a specific signal class, e.g., the ECG signal [42], the electroencephalogram (EEG) signal [43], and random combination of sine waves [44].

### B. Algorithmic Comparison

The methods for numerical comparison involve the following:

1) LLS, the estimate given by (5);
2) LS-L1R, the estimate formulated in (6) and solved by an open-source Python-embedded modeling language for

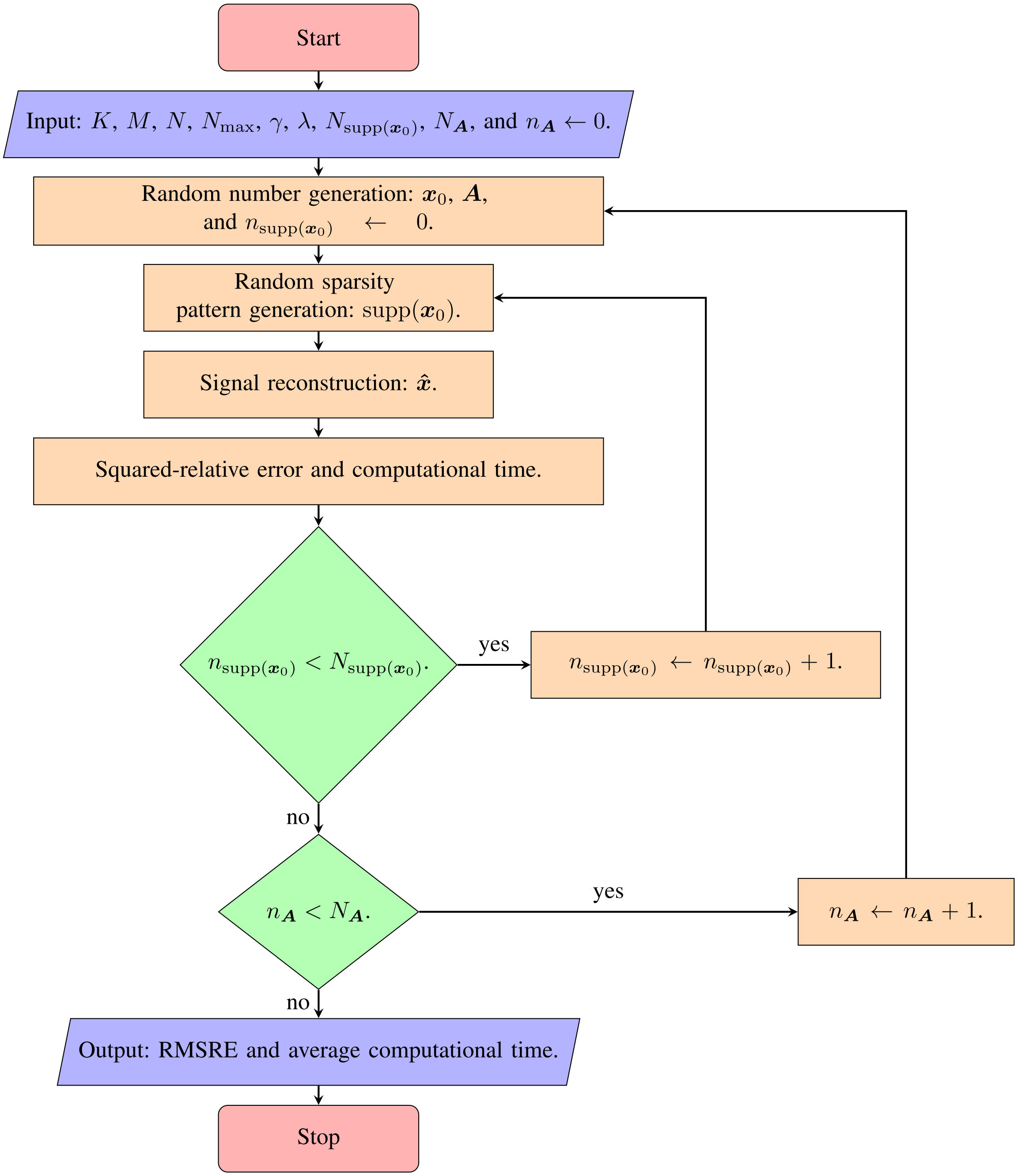
Start
Input: $K$, $M$, $N$, $N_{\max}$, $\gamma$, $\lambda$, $N_{\text{supp}(\boldsymbol{x}_0)}$, $N_{\boldsymbol{A}}$, and $n_{\boldsymbol{A}} \leftarrow 0$.
Random number generation: $\boldsymbol{x}_0$, $\boldsymbol{A}$, and $n_{\text{supp}(\boldsymbol{x}_0)} \leftarrow 0$.
Random sparsity pattern generation: $\text{supp}(\boldsymbol{x}_0)$.
Signal reconstruction: $\hat{\boldsymbol{x}}$.
Squared-relative error and computational time.
$n_{\text{supp}(\boldsymbol{x}_0)} < N_{\text{supp}(\boldsymbol{x}_0)}$.
yes
$n_{\text{supp}(\boldsymbol{x}_0)} \leftarrow n_{\text{supp}(\boldsymbol{x}_0)} + 1$.
no
$n_{\boldsymbol{A}} < N_{\boldsymbol{A}}$.
yes
$n_{\boldsymbol{A}} \leftarrow n_{\boldsymbol{A}} + 1$.
no
Output: RMSRE and average computational time.
Stop

Fig. 4. Diagram of the experimental setup for the noiseless case.

convex optimization problems, also known as ConVeX PYthon (CVXPY) [45];

3) L1RLS, the estimate formulated in (8) and solved by CVXPY [45];
4) SHA, the estimate given by Algorithm 1;
5) SHA-MPI [1], the estimate given by Algorithm 2;
6) SMHA-MPI [1], the estimate given by Algorithm 3;
7) STA-MPI, the estimate given by Algorithm 4;
8) SMTA-MPI, the estimate given by Algorithm 5.

It is found that the closed-form expression of the LLS estimate in (5) provides exactly the same error performance as that given by solving (4) with the CVXPY [45].

Figs. 2 and 3 and other algorithms that we have taken for comparison in this work are in the same kind of signal reconstruction, which does not involve the treatment of $\boldsymbol{A}$. They need to know $\boldsymbol{A}$ earlier. There are some works that propose the algorithms to design $\boldsymbol{A}$ for a specific purpose (see [46] and [47]). Those works do not lie exactly in the same area as this work.

Fig. 4 shows the structure of numerical simulation, which is applied to every algorithm. Figs. 2 and 3 and other algorithms in this work are implicitly inside Fig. 4. The numerical simulation has to supply the matrix $\boldsymbol{A}$ to each algorithm. For averaging the error performance, the numerical simulation will supply different random values of the matrix $\boldsymbol{A}$ to every algorithm. All algorithms in this work do not deal with the design of the matrix $\boldsymbol{A}$ as same as [46] and [47]. Similar to most works, we pay attention to only the reconstruction

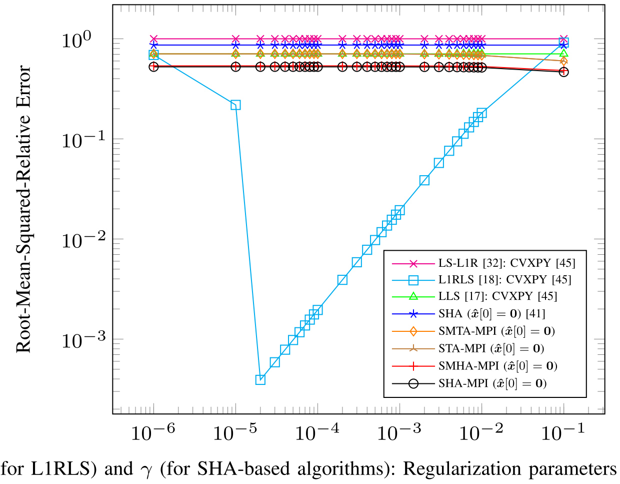

Fig. 5. RMSRE as a function of regularization and shrinking parameters, such as $\lambda$ for L1RLS and $\gamma$ for SHA-based algorithms with $K = 6$.

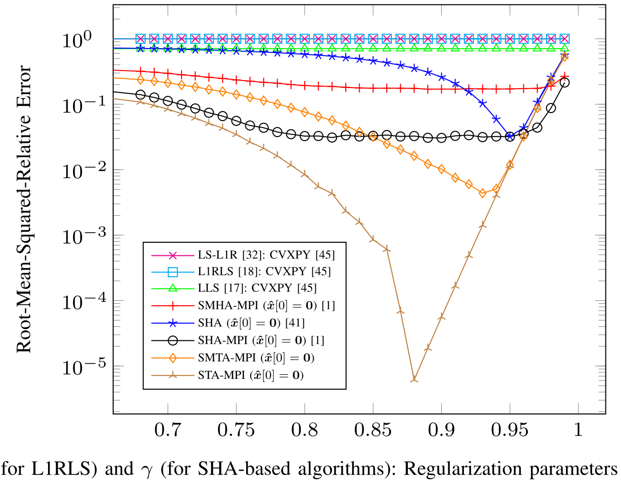

Fig. 6. RMSRE as a function of regularization and shrinking parameters, such as $\lambda$ for the L1RLS and $\gamma$ for the SHA-based algorithms with $K = 6$.

technique for a given $\boldsymbol{A}$ rather than the design of the matrix $\boldsymbol{A}$.

### C. Performance Metric

The simulation result will not be fair if one relies on only a single realization of cascaded random variables. It is general or in various works to average the simulation results from a large number of realizations. See the outer loop in Fig. 4. The mean value given by this large number of simulations will suppress the fluctuation due to the randomness and is more reasonable than the result given by a single realization. Therefore, the matrix $\boldsymbol{A}$ has to be different for each realization. Rather than invoking a specific set of values for the random variables, the randomization technique, sometimes known as Monte Carlo simulation, is commonly used for fair results.

In other words, the matrix $\boldsymbol{A}$ is fixed when it is plugged into an algorithm. The matrix $\boldsymbol{A}$ does not change inside the algorithm. However, the matrix $\boldsymbol{A}$ is set to be changed randomly to other values in the next round when it is inserted into the algorithm.

The root mean square relative error (RMSRE), denoted by

$$\begin{aligned}\text{RMSRE} &= \sqrt{\mathcal{E}_{\boldsymbol{A},\boldsymbol{x}_0,\text{supp}(\boldsymbol{x}_0),\boldsymbol{\delta}_b}\left\{\frac{\left\|\hat{\boldsymbol{x}}-\boldsymbol{x}_0\right\|_2^2}{\|\boldsymbol{x}_0\|_2^2}\right\}}\\ &= \sqrt{\mathcal{E}_{\boldsymbol{A},\boldsymbol{x}_0,\text{supp}(\boldsymbol{x}_0),\boldsymbol{\delta}_b}\left\{\left(\frac{\left\|\hat{\boldsymbol{x}}-\boldsymbol{x}_0\right\|_2}{\|\boldsymbol{x}_0\|_2}\right)^2\right\}} \qquad (28)\end{aligned}$$

is the metric for the performance evaluation of each algorithm. It is computed by the square root of the probabilistic average of the square of the normalized estimation error, where $\mathcal{E}_{\boldsymbol{A},\boldsymbol{x}_0,\text{supp}(\boldsymbol{x}_0),\boldsymbol{\delta}_b}\{\cdot\}$ is the expectation with respect to all randomization.

### D. Simulation Results

In the interest of fairness, numerical simulation in [1] is conducted by always keeping the same value for both $\lambda$, which is required by the L1RLS criterion, and $\gamma$, which is required by the SHA and our homotopy and thresholding methods. In this work, we find that this equality setup might not be justifiable for each algorithm because each method is constructed from different criteria.

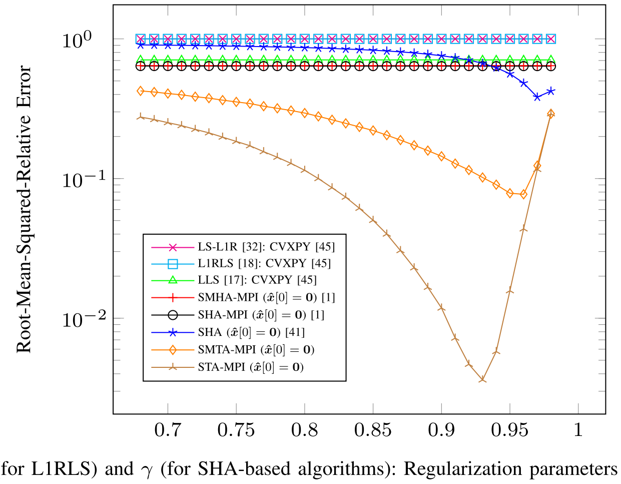

Fig. 7. RMSRE as a function of regularization and shrinking parameters, such as $\lambda$ for the L1RLS and $\gamma$ for the SHA-based algorithms with $K = 16$.

Assume that the optimality lies in the sense of minimal RMSRE. Therefore, it would rather be fruitful to look for the optimal values that might be different for each approach, given a circumstance, e.g., for $K = 6$ or $K = 16$. The simulation setup for the noiseless case is provided in Fig. 4.

In Fig. 5, we search for the optimal value for the regularization parameter $\lambda$ for the L1RLS approach. The parameters for the simulation of Figs. 5 and 6 include $N_R = 10\,00\,002$ independent runs, $M = 128$, $N = 256$, and $N_{\max} = 100$. In Figs. 6 and 7, the optimal values of soft thresholding parameter $\gamma$ are explored for five algorithms for $K = 6$ and $K = 16$, respectively, while corresponding time consumption in the computation by each method is shown in Fig. 8 for the case of $K = 16$. The parameters for the simulation of Figs. 7 and 8 include $N_R = 1\,00\,000$ independent runs, $M = 128$, $N = 256$, and $N_{\max} = 100$.

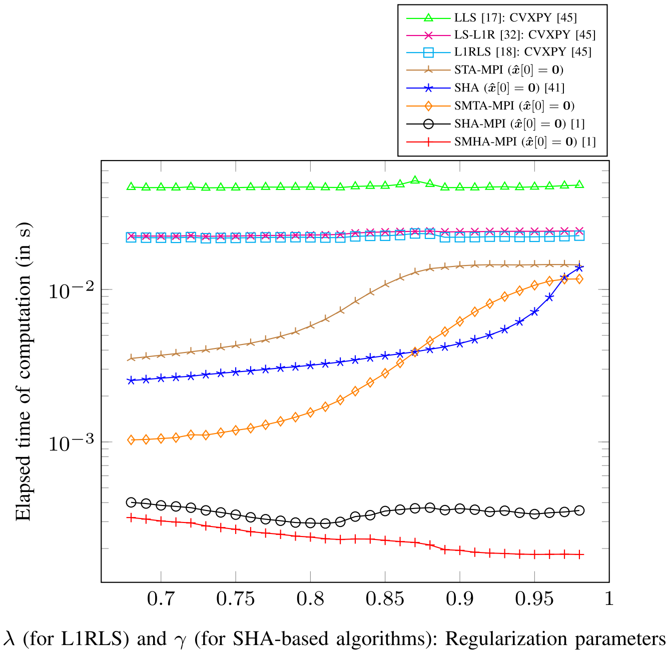

Fig. 8. Elapsed time of computation as a function of regularization and shrinking parameters, such as $\lambda$ for the L1RLS and $\gamma$ for the SHA-based algorithms with $K = 16$.

Under a finite grid search in Fig. 5, we find that the value $\lambda = 10^{-3}$ brings the minimal RMSRE to the L1RLS method when the number of nonzero elements of $\boldsymbol{x}_0$ is 6. It is worth noting that, from our simulation experience, the L1RLS framework may suffer from the nonconvergence, especially in the noisy signal case, when $\lambda$ is less than or equal to $10^{-4}$. The tiny value region in $(10^{-6}, 10^{-1})$ does not provide any low RMSRE to other methods.

In Fig. 6, the finite grid search comes up with a set of optimal values, such as $\gamma = 0.95$ for the SHA, $\gamma = 0.9$ for the SHA-MPI, $\gamma = 0.91$ for the SMHA-MPI, $\gamma = 0.88$ for the STA-MPI, and $\gamma = 0.93$ for the SMTA-MPI.

Comparing to the results given by Fig. 6, one may find that, in Fig. 7, the optimal values of $\gamma$ for the proposed homotopy and thresholding methods deviate from those derived from Fig. 6 when the number nonzero elements increase from $K = 6$ to $K = 16$. This variation implies that there are no fixed optimal values of $\gamma$ for all homotopy and thresholding algorithms under different sparsity circumstances.

In Fig. 8, one can see that the relaxed stopping criterion can reduce the complexity of the proposed homotopy and thresholding approaches. In addition, the homotopy and thresholding algorithms, including the SHA, need less computational time than the convex optimization methods, such as the LS-L1R, the L1RLS, and the LLS.

Assume that we are interested in a very sparse signal, such as that of $K = 6$. The regularization parameters for Figs. 9 and 10 include $\gamma = 0.95$ for the SHA, $\gamma = 0.9$ for the SHA-MPI, $\gamma = 0.91$ for the SMHA-MPI, $\gamma = 0.88$ for the STA-MPI, $\gamma = 0.93$ for the SMTA-MPI, and $\lambda = 10^{-3}$ for the L1RLS. Other parameters encompass $N_R = 100\,002$ independent runs, $K = 6$, $M = 128$, and $N = 256$. Fig. 9 reveals that only the STA-MPI and the SMTA-MPI can provide lower RMSRE than the L1RLS method for $N_{\max} \geq 40$ and $N_{\max} \geq 70$, respectively. Fig. 10 indicates that all the homotopy and thresholding algorithms, including the SHA, perform almost identically and have the advantage of low computational time over those convex optimization methods.

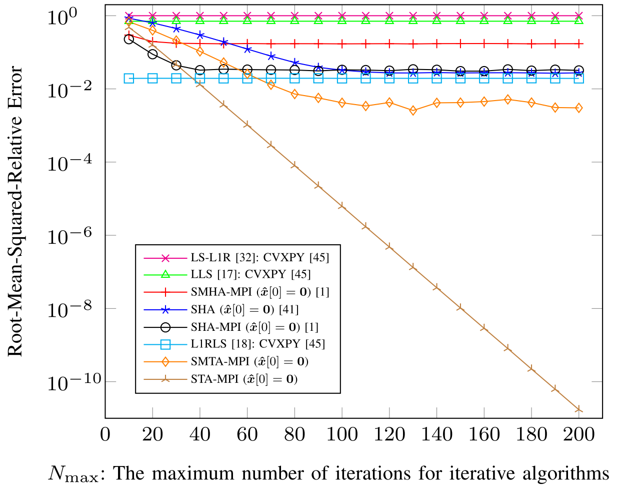

Fig. 9. RMSRE as a function of the maximum number of iterations.

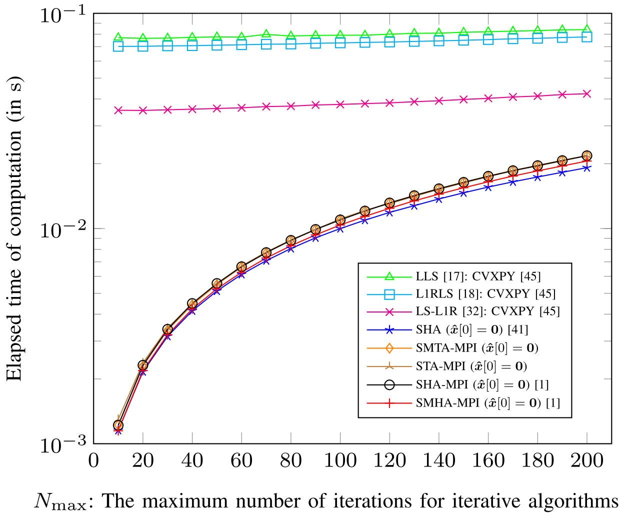

Fig. 10. Elapsed time of computation as a function of the maximum number of iterations.

The regularization parameters for Figs. 11 and 12 are $\gamma = 0.95$ for the SHA, $\gamma = 0.9$ for the SHA-MPI, $\gamma = 0.91$ for the SMHA-MPI, $\gamma = 0.88$ for the STA-MPI, $\gamma = 0.93$ for the SMTA-MPI, and $\lambda = 10^{-4}$ for the L1RLS. Other parameters comprise of $N_R = 100\,000$ independent runs, $M = 128$, $N = 256$, and $N_{\max} = 100$. For the sake of various signal sparsity, Fig. 11 shows that the homotopy and thresholding algorithms can outperform the L1RLS approach when the signal contains a tiny amount of nonzero elements. Only the STA-MPI can perform better than the L1RLS method for every level of signal sparsity.

Fig. 12 implies that the SHA-MPI and the SMHA-MPI take significantly lower computational time than the SHA, the STA-MPI, and the SMTA-MPI.

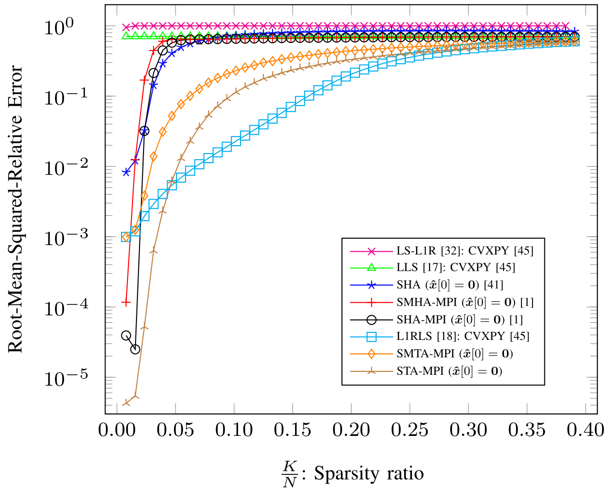

Fig. 11. RMSRE as a function of sparsity ratio ($K/N$).

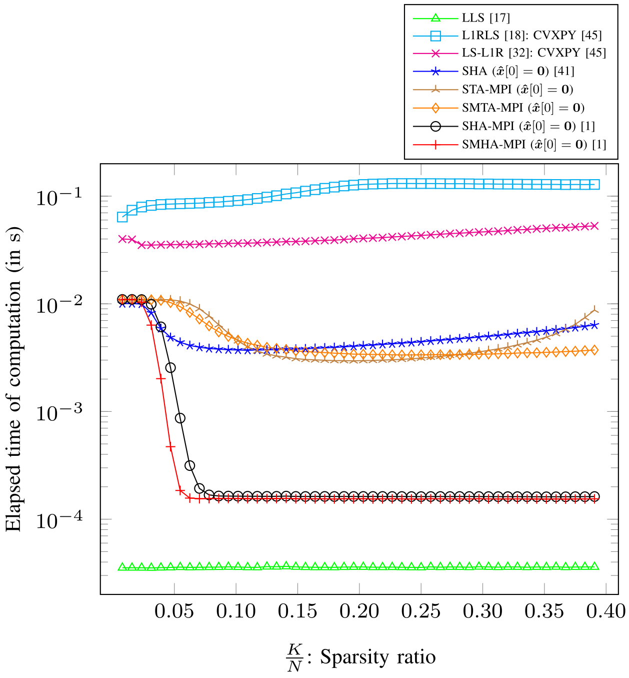

Fig. 12. Elapsed time of computation as a function of sparsity ratio.

All previous simulation results are conducted under the situation where there was no additive noise embedded in the output vector $\boldsymbol{b}$. The simulation setup that accounts for the additive noise is provided in Fig. 13.

In the next aspect, we investigate the effect of system output uncertainty in Figs. 14 and 15, i.e., there exists additive noise according to Fig. 1.

Let us introduce the signal-to-noise ratio (SNR) in decibel (dB) as

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\mathcal{E}_{\boldsymbol{A},\boldsymbol{x}_0,\mathrm{supp}(\boldsymbol{x}_0)}\left\{\|\boldsymbol{A}\boldsymbol{x}_0\|_2^2\right\}}{\mathcal{E}_{\boldsymbol{\delta}_b}\left\{\|\boldsymbol{\delta}_b\|_2^2\right\}}\right). \tag{29}$$

The regularization parameters for Figs. 14 and 15 incorporate $\gamma = 0.95$ for the SHA, $\gamma = 0.9$ for the SHA-MPI, $\gamma = 0.91$ for the SMHA-MPI, $\gamma = 0.88$ for the STA-MPI, $\gamma = 0.93$ for the SMTA-MPI, and $\lambda = 10^{-2}$ for the L1RLS. Other parameters for the simulation cover $N_{\boldsymbol{A}} = N_{\boldsymbol{x}_0} = 6$, $N_{\mathrm{supp}(\boldsymbol{x}_0)} = 17$, $N_{\boldsymbol{\delta}_b} = 100$, and $N_R = N_{\boldsymbol{A}} N_{\mathrm{supp}(\boldsymbol{x}_0)} N_{\boldsymbol{\delta}_b} = 1\,00\,200$ independent runs, $M = 128$, $N = 256$, and $N_{\max} = 100$. In Fig. 14, for SNR in the range $(-10, 35)$ dB, the SHA outperforms the others, including the proposed methods. However, when the received signal tends to be noiseless, e.g., for SNR greater than 55 dB, all homotopy and thresholding methods start to exhibit their accuracy. This result implies that, when the shrinking parameter $\gamma$ is chosen to be optimal from the noiseless case, the proposed methods in this work are quite sensitive to the additive noise.

In Fig. 15, one can see that, although it gives the lowest RMSRE in the practical noisy case, e.g., for the SNR in the range $(-10, 35)$ dB, the SHA consumes more computational time than the proposed MPI methods. The convex optimization frameworks take much more time than all homotopy and thresholding approaches.

Next, we present a blind scenario where there is no knowledge of the optimal regularization parameters available to any algorithm. The parameters for the simulation of Figs. 16 and 17 include $N_{\boldsymbol{A}} = N_{\boldsymbol{x}_0} = 16$, $N_{\mathrm{supp}(\boldsymbol{x}_0)} = 63$, $N_{\boldsymbol{\delta}_b} = 100$, $N_R = N_{\boldsymbol{A}} N_{\mathrm{supp}(\boldsymbol{x}_0)} N_{\boldsymbol{\delta}_b} = 1\,00\,800$ independent runs, $K = 16$, $M = 32$, $N = 64$, $N_{\max} = 100$, and $\gamma = \lambda = 0.5$. Figs. 16 and 17 demonstrate that, for the SNR greater than 0 dB, the proposed homotopy and thresholding methods using the MPI can achieve lower RMSRE and take lower computational time than the previous methods.

In addition, a realistic signal is taken into account. We downloaded a realistic EEG signal from [48], which is a part of CHB-MIT Scalp EEG Database [49]. The measured signal from the file “chb18_18.edf” contains 7000 samples. Unfortunately, due to a limit amount of memory, the computer cannot process all 7000 samples simultaneously.

Therefore, we process only $N = 7000$ samples of the whole 9 21 600 samples. To complete all 9 21 600 samples, this computation is repeated to the next data blocks until it reaches and covers the last data point. It means that the last data block will be of the length $N = 4600$.

Let $y(t)$ be the raw data of 7000 data points from the file “chb18_18.edf” in [48]. It can be expressed in the form of a vector as

$$\boldsymbol{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_{7000} \end{bmatrix}^{\mathsf{T}} \tag{30}$$

where $y_n$ is the $n$th sample from $y(t)$, i.e.,

$$y_n = y(nT_s) \tag{31}$$

with $T_s$ being a sampling period. The vector $\boldsymbol{y}$ is shown in Fig. 18, while one of its portions is illustrated in Fig. 19.

The signal in Figs. 18 and 19 is very weak, i.e., on the order of $10^{-6}$–$10^{-5}$ due to the low voltage induced by the mechanism inside the human brain.

In this work, the discrete cosine transform (DCT) is preferred because it involves only a real-valued number per data point, whereas the discrete Fourier transform contrives a complex-valued number per data point, which incurs double memory for data storage. Furthermore, the discrete wavelet transform is more suitable for 2-D data, such as an image.

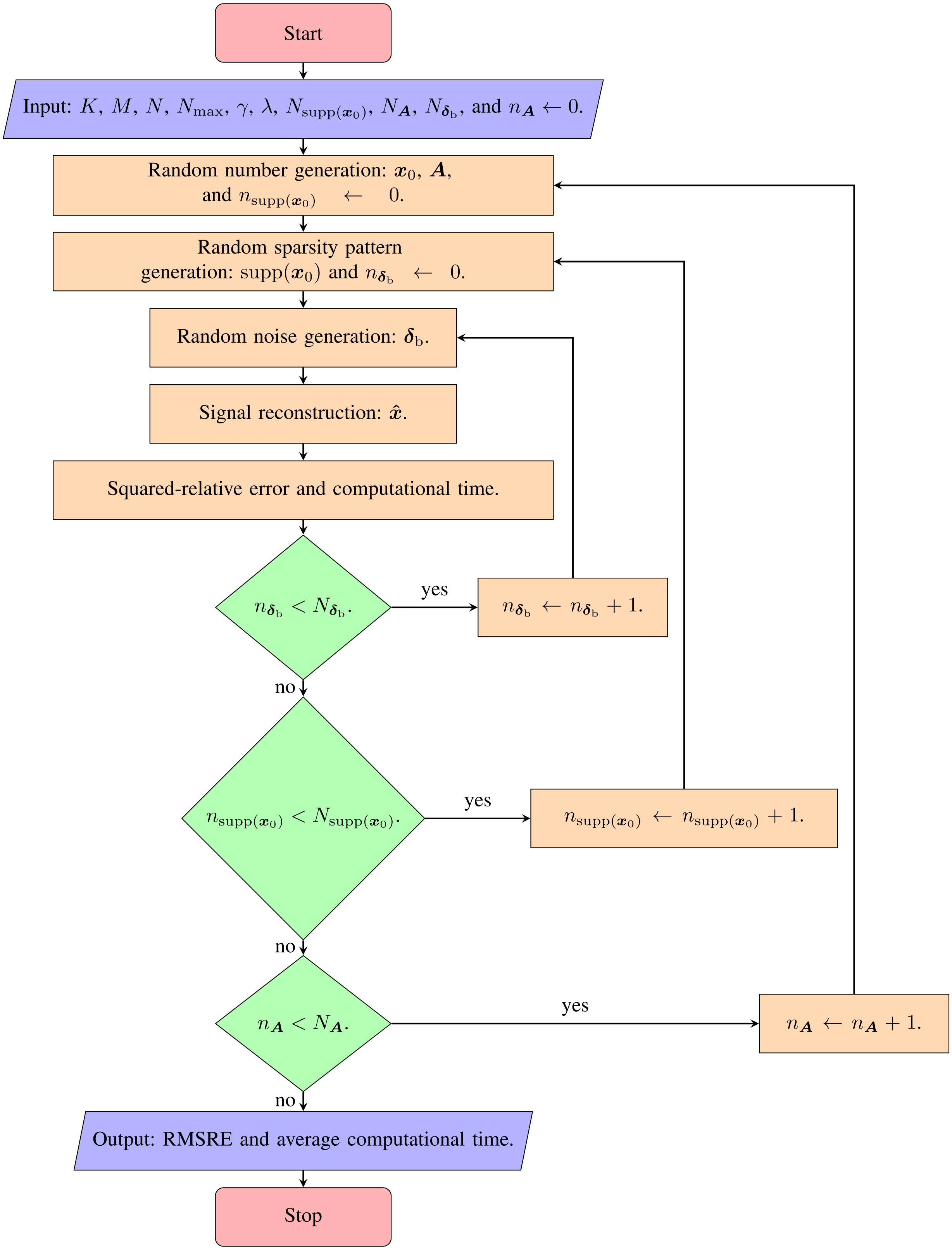
Start
Input: $K$, $M$, $N$, $N_{\max}$, $\gamma$, $\lambda$, $N_{\mathrm{supp}(\boldsymbol{x}_0)}$, $N_{\boldsymbol{A}}$, $N_{\boldsymbol{\delta}_{\mathrm{b}}}$, and $n_{\boldsymbol{A}} \leftarrow 0$.
Random number generation: $\boldsymbol{x}_0$, $\boldsymbol{A}$, and $n_{\mathrm{supp}(\boldsymbol{x}_0)} \leftarrow 0$.
Random sparsity pattern generation: $\mathrm{supp}(\boldsymbol{x}_0)$ and $n_{\boldsymbol{\delta}_{\mathrm{b}}} \leftarrow 0$.
Random noise generation: $\boldsymbol{\delta}_{\mathrm{b}}$.
Signal reconstruction: $\hat{\boldsymbol{x}}$.
Squared-relative error and computational time.
$n_{\boldsymbol{\delta}_{\mathrm{b}}} < N_{\boldsymbol{\delta}_{\mathrm{b}}}$.
yes
$n_{\boldsymbol{\delta}_{\mathrm{b}}} \leftarrow n_{\boldsymbol{\delta}_{\mathrm{b}}} + 1$.
no
$n_{\mathrm{supp}(\boldsymbol{x}_0)} < N_{\mathrm{supp}(\boldsymbol{x}_0)}$.
yes
$n_{\mathrm{supp}(\boldsymbol{x}_0)} \leftarrow n_{\mathrm{supp}(\boldsymbol{x}_0)} + 1$.
no
$n_{\boldsymbol{A}} < N_{\boldsymbol{A}}$.
yes
$n_{\boldsymbol{A}} \leftarrow n_{\boldsymbol{A}} + 1$.
no
Output: RMSRE and average computational time.
Stop

Fig. 13. Diagram of the experimental setup for the noisy case.

Let the DCT of $y_n$ for $n \in \{1, 2, \ldots, 7000\}$ be [50, eq. (8.158)], [51, Sec. 5.6]

$$y_{\cos}[k] = \alpha_k \sum_{n=0}^{N-1} \underbrace{y_n}_{=y(nT_s)} \cos\left(\frac{1}{2N}(2n+1)\pi k\right) \tag{32}$$

where $\alpha_k$ is given by

$$\alpha_k = \begin{cases} \dfrac{1}{\sqrt{N}}, & k = 0 \\ \sqrt{\dfrac{2}{N}}, & k \in \{1, 2, \ldots, N-1\} \end{cases} \tag{33}$$

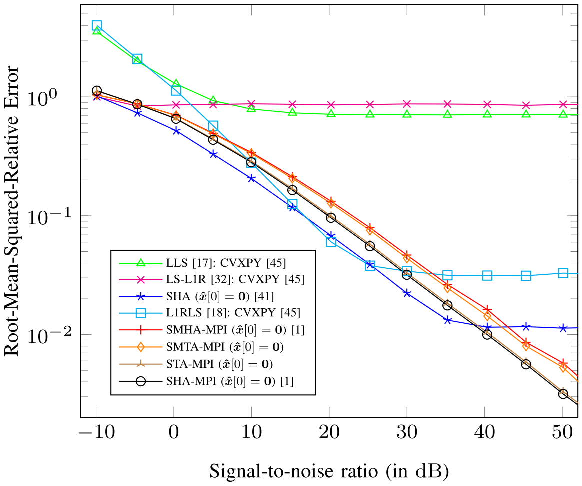

Fig. 14. RMSRE as a function of SNR for $K = 6$.

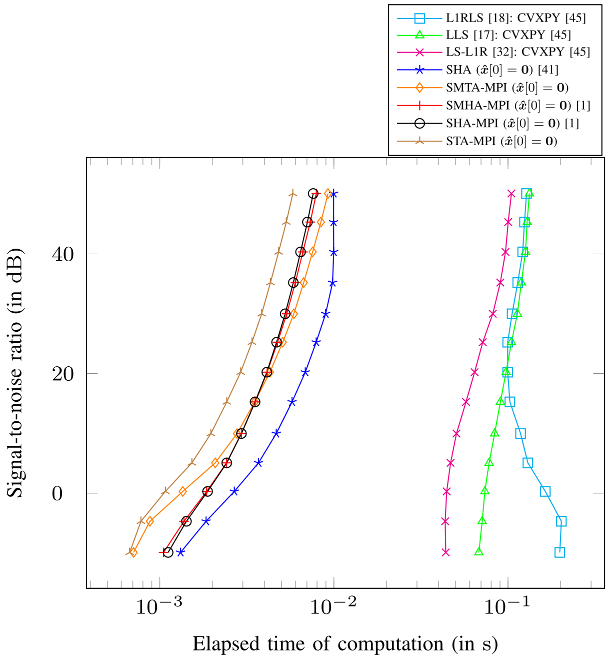

Fig. 15. SNR as a function of the elapsed time of computation for $K = 6$.

and $N = 7000$. In Fig. 20, the signal along the time is transformed to the domain of a cosine function by using the DCT. It can be seen that the transformed signal in the cosine domain does not exhibit a sparsity in the spectrum. The EEG signal from the above database represents an example of a nonsparse signal in nature. If we assume that the DCT signal $y_{\cos}[k]$ in (32) for $k \in \{1, 2, \ldots, 7000\}$ is the desired information $\boldsymbol{x}$ in (1), we have

$$\boldsymbol{x} = \begin{bmatrix} y_{\cos}[1] & y_{\cos}[2] & \cdots & y_{\cos}[7000] \end{bmatrix}^{\mathsf{T}}. \tag{34}$$

If $\mathcal{C}\{\cdot\}$ is the DCT operator of a vector or a matrix $\cdot$, it follows from (32) that

$$\boldsymbol{x} = \mathcal{C}\{\boldsymbol{y}\}. \tag{35}$$

In Fig. 21, we randomly pick up only $M = 3430$ samples from all $N = 7000$ samples shown in Figs. 18 and 19.

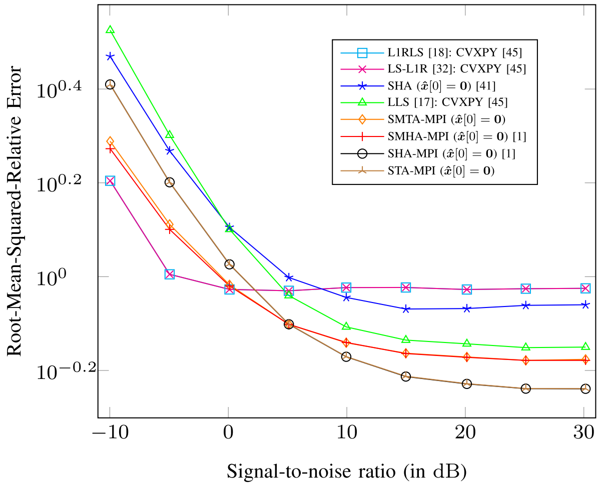

Fig. 16. RMSRE as a function of SNR for $K = 16$.

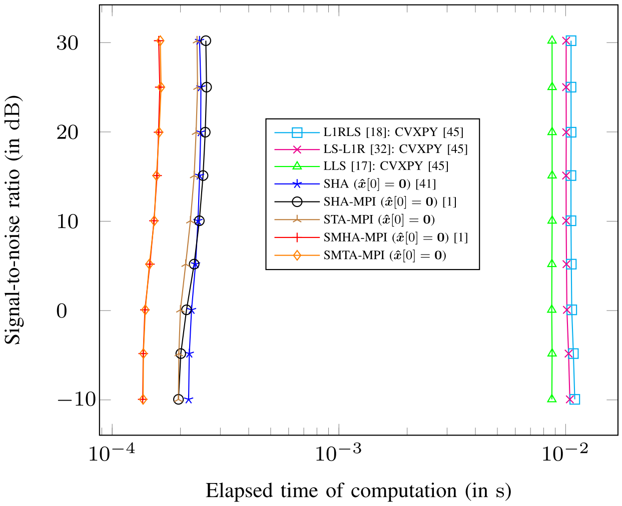

Fig. 17. SNR as a function of the elapsed time of computation for $K = 16$.

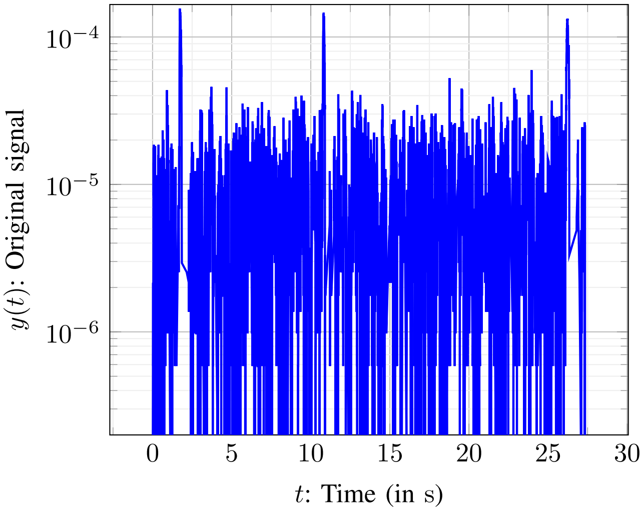

Fig. 18. EEG signal that is taken from the first $N = 7000$ samples of the whole 921 600 samples.

The signal of a smaller size can be denoted by $b(t)$, which can be written in a discrete-time form as

$$\boldsymbol{b} = \begin{bmatrix} b_1 & b_2 & \cdots & b_{3430} \end{bmatrix}^{\mathsf{T}}. \tag{36}$$

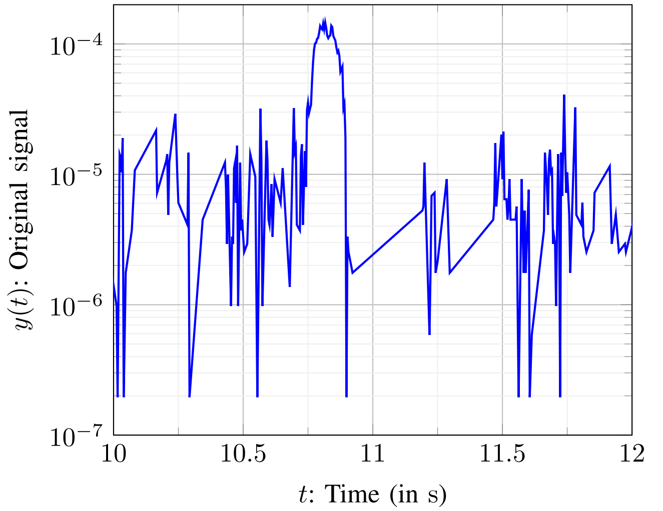

Fig. 19. Zoomed-in view of the original signal in Fig. 18 from $t = 10$ to $t = 12$ s.

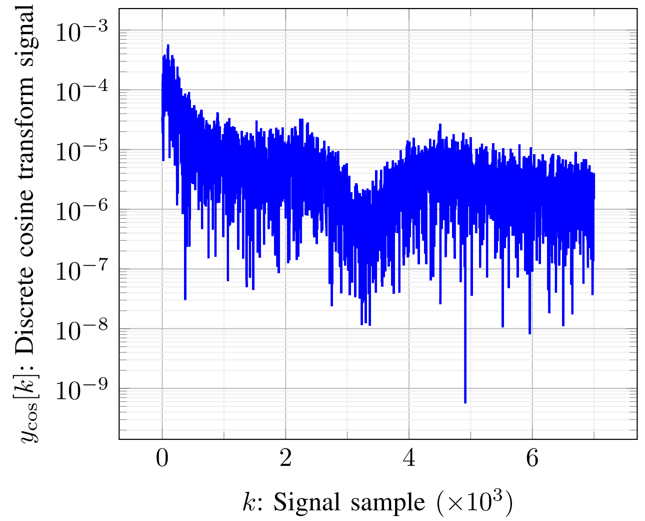

Fig. 20. DCT signal $y_{\cos}[k]$ in (32) for $k \in \{1, 2, \ldots, 7000\}$.

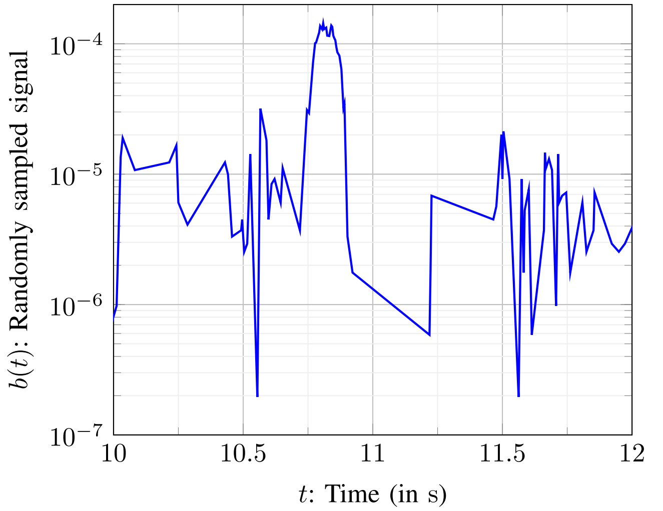

Fig. 21. Zoom from $t = 10$ to $t = 12$ s for the sampled signal that is randomly chosen from 49% samples of the original signal in Fig. 18.

From 7000 samples to 3430 samples, there exists a sampling matrix $\mathbf{\Phi} \in \mathbb{R}^{3430\times 7000}$ such that

$$\boldsymbol{b} = \mathbf{\Phi}\boldsymbol{y}. \tag{37}$$

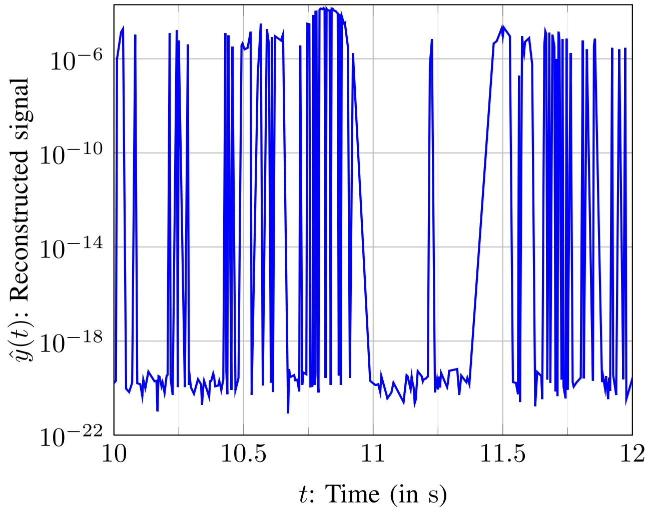

Fig. 22. Zoom from $t = 10$ to $t = 12$ s for the EEG signal reconstructed by the LLS method.

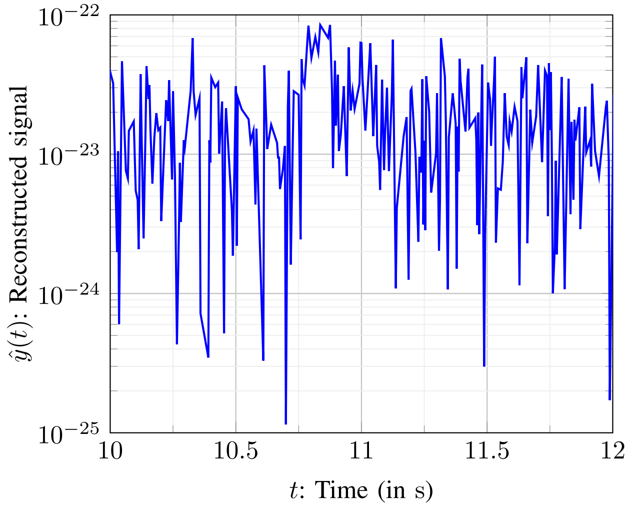

Fig. 23. Zoom from $t = 10$ to $t = 12$ s for the EEG signal reconstructed by the LSL1R method.

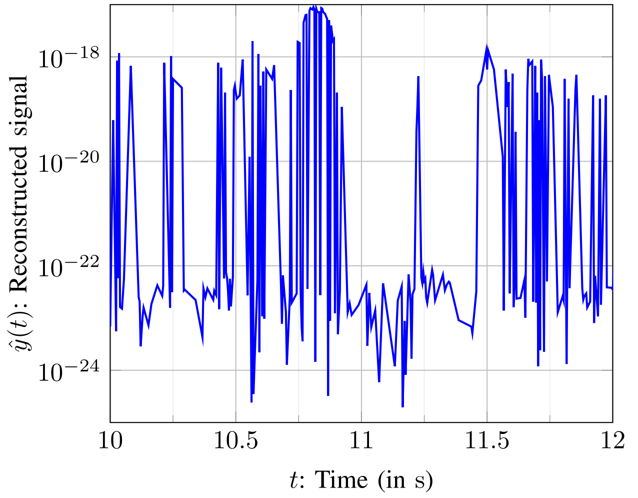

Fig. 24. Zoom from $t = 10$ to $t = 12$ s for the EEG signal reconstructed by the L1RLS method using $\gamma_{\text{L1RLS}} = 10^{-4}$.

In practice, the sampling matrix $\mathbf{\Phi}$ does not need to be generated. Let $\mathcal{C}^{-1}\{\cdot\}$ be the inverse DCT (IDCT) operator of a vector or a matrix $\cdot$. To comply with the compressive

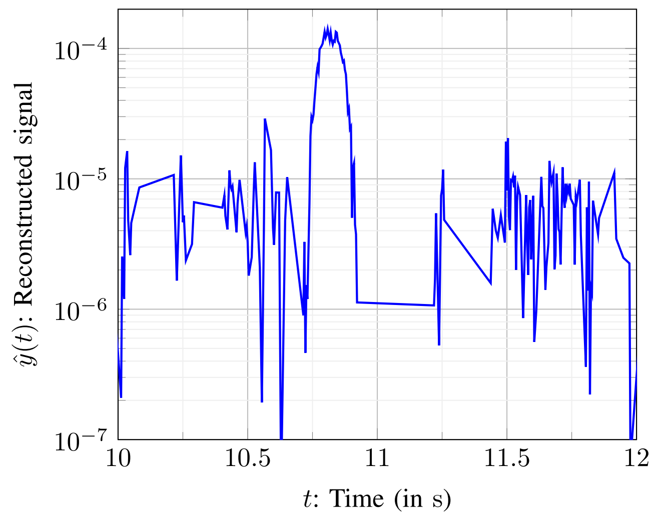

Fig. 25. Zoom from $t = 10$ to $t = 12$ s for the EEG signal reconstructed by the SHA method using $\gamma_{\text{SHA}} = 0.95$.

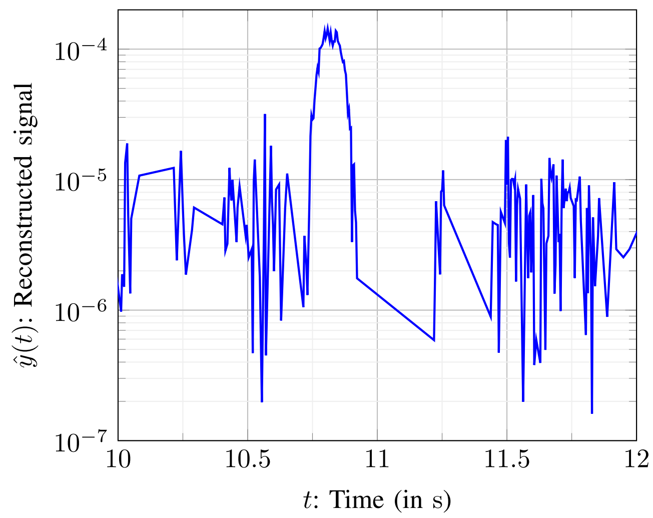

Fig. 26. Zoom from $t = 10$ to $t = 12$ s for the EEG signal reconstructed by the SHAMPI method using $\gamma_{\text{SHAMPI}} = 0.9$.

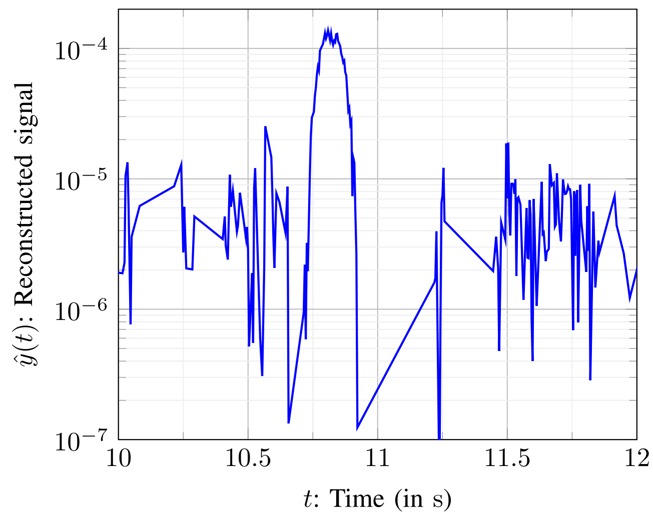

Fig. 27. Zoom from $t = 10$ to $t = 12$ s for the EEG signal reconstructed by the SMHAMPI method using $\gamma_{\text{SMHAMPI}} = 0.91$.

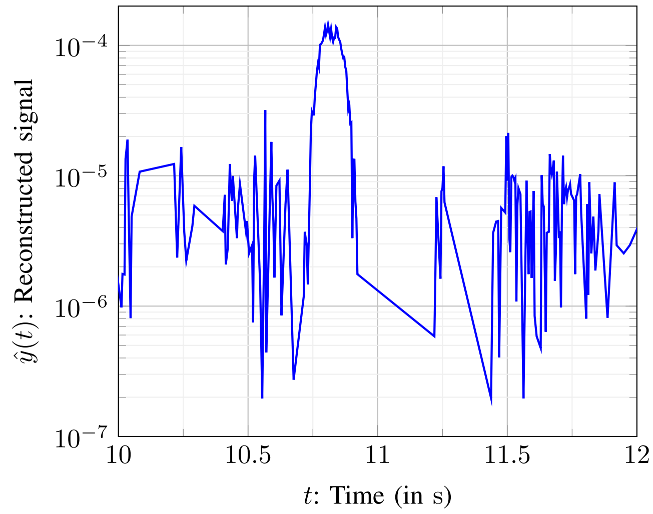

Fig. 28. Zoom from $t = 10$ to $t = 12$ s for the EEG signal reconstructed by the STAMPI method using $\gamma_{\text{STAMPI}} = 0.88$.

sensing concept [52], [53], the linear sampling model in (37) can be written as

$$\begin{aligned}\boldsymbol{b} &= \boldsymbol{\Phi}\mathcal{C}^{-1}\left\{\underbrace{\mathcal{C}\{\boldsymbol{y}\}}_{=\boldsymbol{x},\ (35)}\right\}\\ &= \boldsymbol{\Phi}\underbrace{\mathcal{C}^{-1}\{\boldsymbol{I}_N\}}_{\boldsymbol{\Psi}}\boldsymbol{x}\\ &= \underbrace{\boldsymbol{\Phi}\boldsymbol{\Psi}}_{\boldsymbol{A},\ (1)}\boldsymbol{x}\\ &= \boldsymbol{A}\boldsymbol{x} \qquad (38)\end{aligned}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{N\times N}$ is the IDCT matrix of the identity matrix, which is given by [51, Sec. 5.6]

$$\boldsymbol{\Psi} = \mathcal{C}^{-1}\{\boldsymbol{I}_N\} \qquad (39)$$

and the sensing matrix $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ is given by

$$\boldsymbol{A} = \boldsymbol{\Phi}\boldsymbol{\Psi}. \qquad (40)$$

The sampling matrix $\boldsymbol{\Phi}$ is implicit in the selection of only $M$ rows from all $N$ rows of $\boldsymbol{\Psi}$ to be $\boldsymbol{A}$. The signal $b(t)$ shown in Fig. 21 is due to only a single realization of sampling randomization. In Fig. 21, the sampled signal is of length $M = 3430$. It is randomly chosen under sub-Nyquist sampling from the original signal of length $N = 7000$. Hence, the reconstructed results, in terms of

$$\hat{\boldsymbol{y}} = \mathcal{C}^{-1}\{\hat{\boldsymbol{x}}\} \qquad (41)$$

from Figs. 22 to 29 are derived from only a single realization of random sampling in Fig. 21.

Figs. 22 to 29 adopt $\hat{\boldsymbol{x}}[0] = \boldsymbol{0}$, $N_{\text{max}} = 100$, and the compression ratio $(1/N)(N - M) = 0.51$. All methods provide the reconstructed signals of length $N = 7000$ and the performance in Table I.

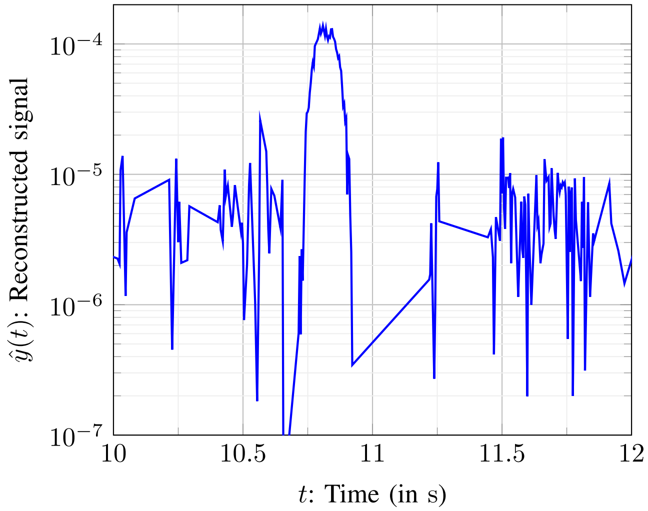

Fig. 29. Zoom from $t = 10$ to $t = 12$ s for the EEG signal reconstructed by the SMTAMPI method using $\gamma_{\text{SMTAMPI}} = 0.93$.

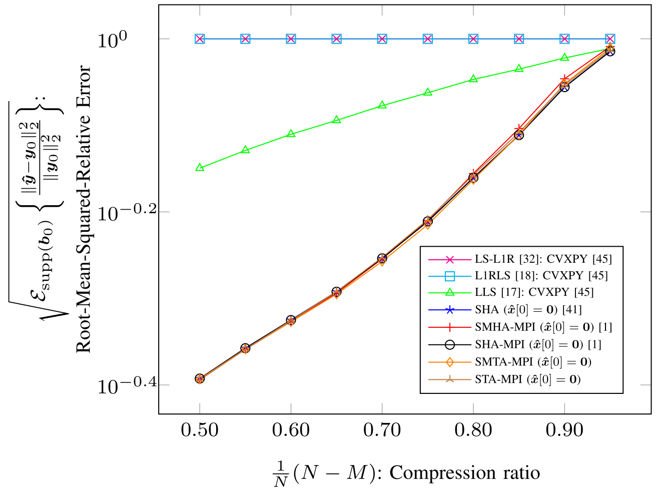

Fig. 30. RMSRE as a function of compression ratio $(1/N)(N - M)$ from $N_R = 100$ independent runs for $K = M$, $N = 7000$, and $N_{\max} = 100$.

TABLE I

Normalized Error and Computational Time of EEG Signal Recovery Given by Each Algorithm Compression Ratio of 0.51

| Methods | Normalized Error | Computational Time (in s) |
|---|---|---|
| LLS [17] | 0.7139 | 500.17 |
| LS-L1R [32] | 1.0 | 9,788.64 |
| L1RLS [18] | 1.0 | 15,396.74 |
| SHA [41] | 0.2880 | 118.73 |
| SHA-MPI | 0.2874 | 111.02 |
| SMHA-MPI | 0.2876 | 56.44 |
| STA-MPI | 0.2874 | 104.76 |
| SMTA-MPI | 0.2870 | 72.57 |

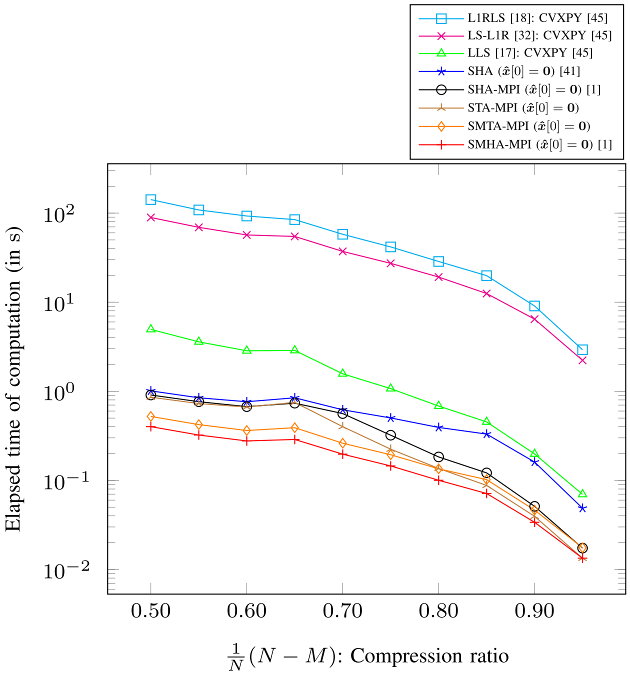

Fig. 31. Elapsed time of computation as a function of compression ratio $(1/N)(N - M)$ from $N_R = 100$ independent runs for $K = M$, $N = 7000$, and $N_{\max} = 100$.

TABLE II

Normalized Error and Computational Time of Wind Speed Signal Recovery Given by Each Algorithm for $N_R = 100$ Independent Runs and a Compression Ratio of 0.8

| Methods | Normalized Error | Computational Time (in s) |
|---|---|---|
| LLS [17] | 0.8943 | 1.31 |
| LS-L1R [32] | 0.2434 | 36.21 |
| L1RLS [18] | 0.2551 | 68.35 |
| SHA [41] | 0.2426 | 199.42 |
| SHA-MPI | 0.2417 | 93.12 |
| SMHA-MPI | 0.2418 | 68.87 |
| STA-MPI | 0.2413 | 70.90 |
| SMTA-MPI | 0.2409 | 89.99 |

In Figs. 30 and 31, we average the EEG signal reconstruction error and the computational time from 100 realizations of random sampling. The regularization parameters for Figs. 30 and 31 consist of $\gamma = 0.95$ for the SHA, $\gamma = 0.9$ for the SHA-MPI, $\gamma = 0.91$ for the SMHA-MPI, $\gamma = 0.88$ for the STA-MPI, $\gamma = 0.93$ for the SMTA-MPI, and $\lambda = 10^{-4}$ for the L1RLS. One can see from Fig. 30 that, for a range of the compression ratio, all soft thresholding algorithms achieve nearly the same recovery error, while, from Fig. 31, the soft thresholding algorithms using the MPI consume less computational time than the conventional thresholding method. The flowchart of the experimental setup for this EEG signal is given in Fig. 32.

In Fig. 33, the wind speed of a wind turbine system was measured for 302.75 days [54, Ch. 2]. The measurement started on October 7, 2014, at 02:00 and finished on October 7, 2015, at 00:00. This result is taken from a real measurement and available in terms of a support file. All regularization parameters are chosen to be the same as in Figs. 22–29. The length of the data in Fig. 33 is $N = 7266$.

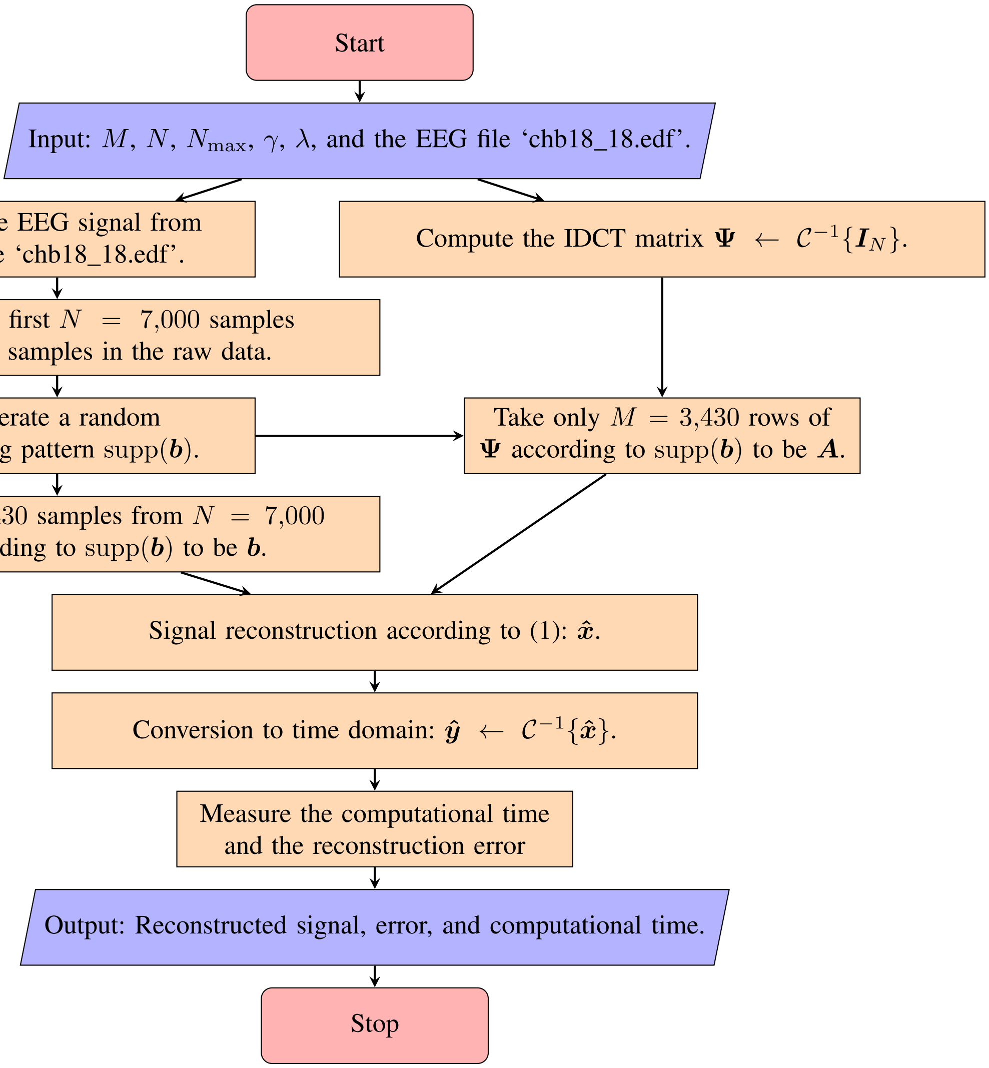
Start
Input: $M$, $N$, $N_{\max}$, $\gamma$, $\lambda$, and the EEG file 'chb18_18.edf'.
Load the EEG signal from the file 'chb18_18.edf'.
Compute the IDCT matrix $\boldsymbol{\Psi} \leftarrow \mathcal{C}^{-1}\{\boldsymbol{I}_N\}$.
Pick up only the first $N = 7{,}000$ samples from all 7,000 samples in the raw data.
Generate a random sampling pattern $\mathrm{supp}(\boldsymbol{b})$.
Take only $M = 3{,}430$ rows of $\boldsymbol{\Psi}$ according to $\mathrm{supp}(\boldsymbol{b})$ to be $\boldsymbol{A}$.
Take the $M = 3{,}430$ samples from $N = 7{,}000$ samples according to $\mathrm{supp}(\boldsymbol{b})$ to be $\boldsymbol{b}$.
Signal reconstruction according to (1): $\hat{\boldsymbol{x}}$.
Conversion to time domain: $\hat{\boldsymbol{y}} \leftarrow \mathcal{C}^{-1}\{\hat{\boldsymbol{x}}\}$.
Measure the computational time and the reconstruction error
Output: Reconstructed signal, error, and computational time.
Stop

Fig. 32. Diagram of the experimental setup for the EEG signal.

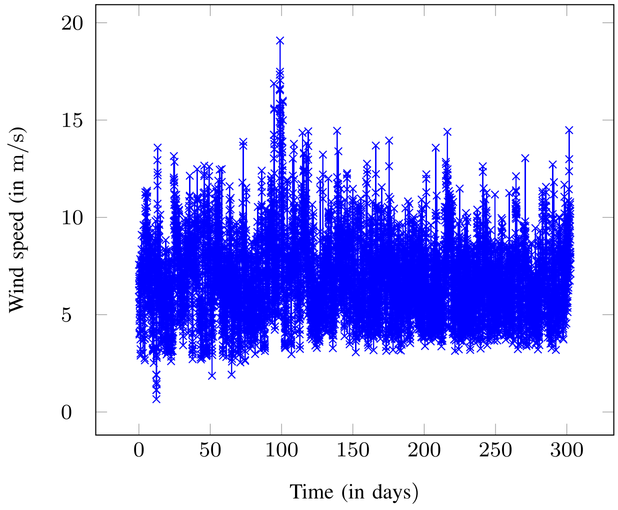

Fig. 33. Wind speed signal.

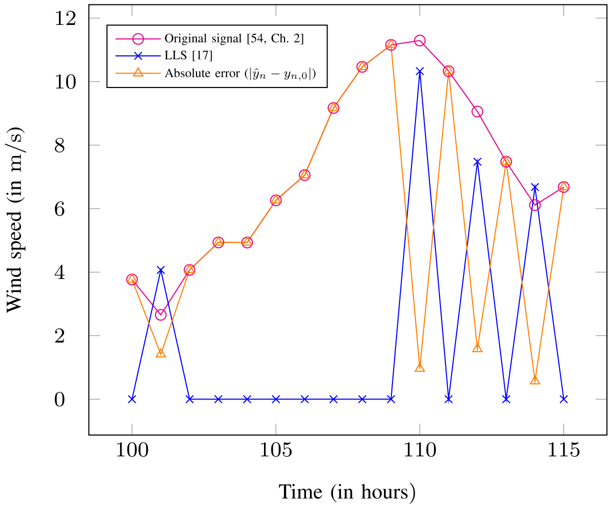

Fig. 34. Wind speed signal reconstructed by the LLS.

From Figs. 34 to 41, the wind speed signal is reconstructed with the compression ratio of 0.8 for a single realization of the sampling pattern. These results are shown only from $y_{100}$ to $y_{115}$ for the sake of clarity. It can be seen that all algorithms recover the desired signal with nearly the same error, except for the LLS in Fig. 34, which is the worst.

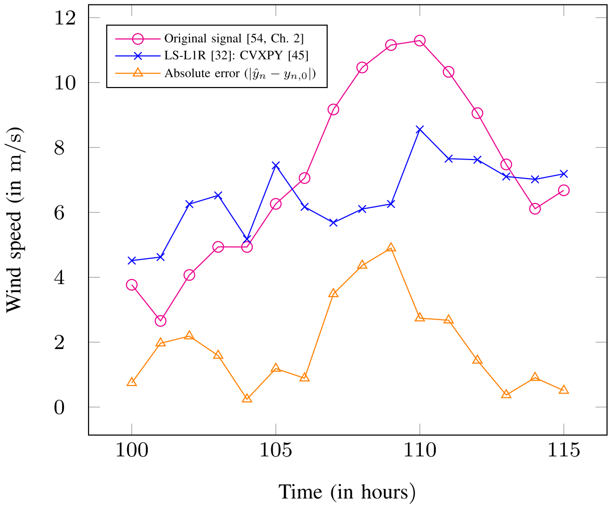

Fig. 35. Wind speed signal reconstructed by the LS-L1R.

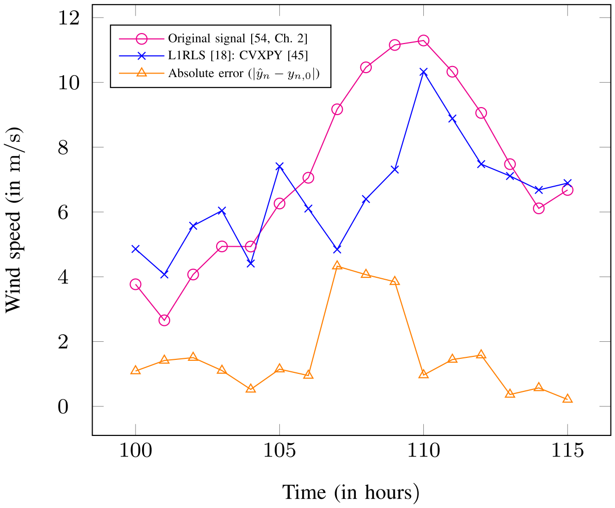

Fig. 36. Wind speed signal reconstructed by the L1RLS.

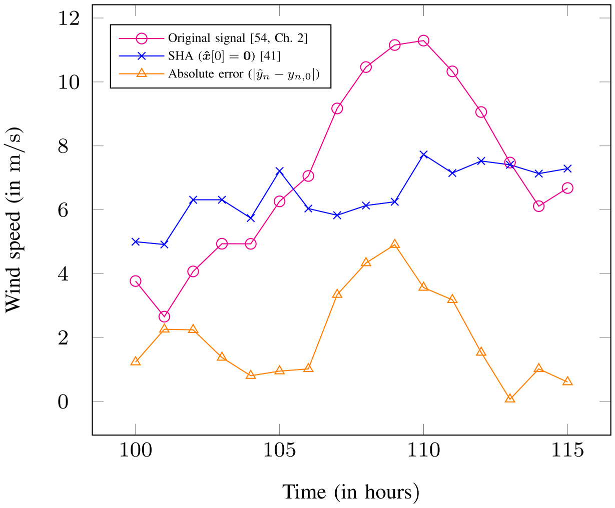

Fig. 37. Wind speed signal reconstructed by the SHA.

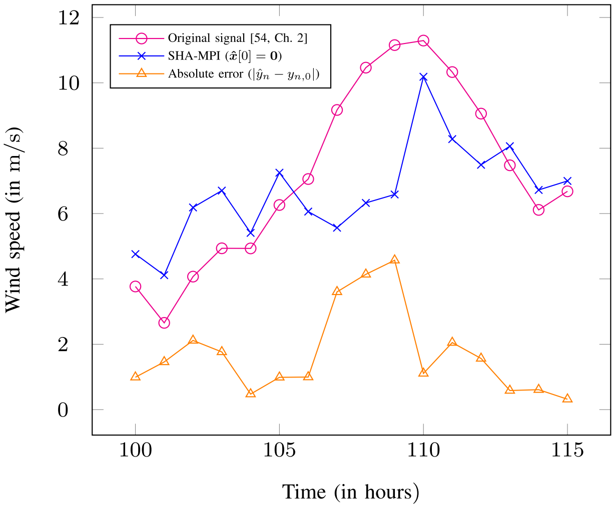

Fig. 38. Wind speed signal reconstructed by the SHA-MPI.

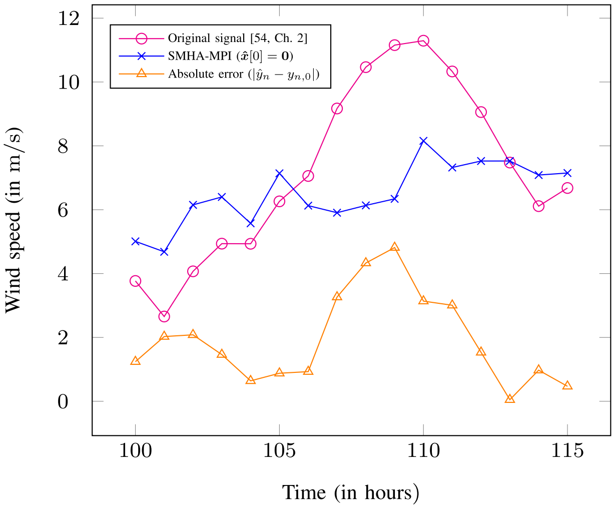

Fig. 39. Wind speed signal reconstructed by the SMHA-MPI.

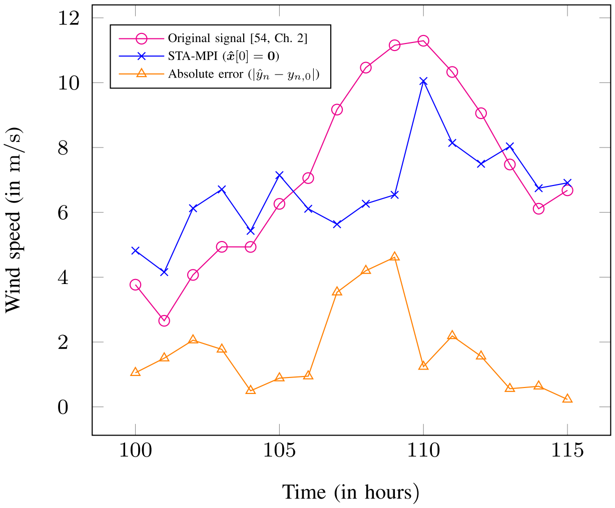

Fig. 40. Wind speed signal reconstructed by the STA-MPI.

If we reconstruct the wind speed signal as in Figs. 34–41 for 100 possible sampling patterns, the average error performance, which appears to be almost identical, is shown in the first column of Table II. However, one can see in the second column of Table II that the computational time consumed by the methods using the MPI is relatively low compared to the former SHA.

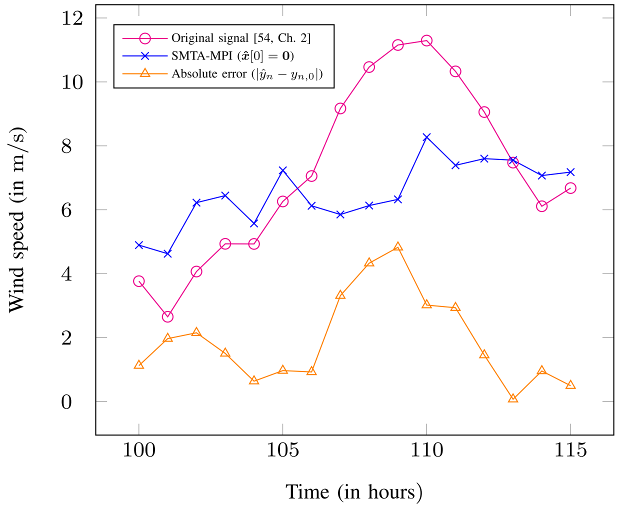

Fig. 41. Wind speed signal reconstructed by the SMTA-MPI.

## IV. Conclusion

Apart from the two previously proposed SHAs in [1], two thresholding algorithms that involve a soft decision are proposed in this work by means of the MPI. The additional complexity is relatively minimal since the necessary matrix inverse $(\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}})^{-1}$ and the matrix multiplication $\boldsymbol{A}^{\mathsf{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}})^{-1}$ can be done before the iteration starts. Numerical examples have been conducted to illustrate the improved error performance from the optimal values of the shrinking parameter $\gamma$ in the noiseless case. It is found that, in the noisy case, the proposed MPI methods require less computational time than the SHA at the expense of signal acquisition accuracy. Under a blind setup where no optimal values of the regularization and shrinking parameters are available, the proposed homotopy and thresholding algorithms can outperform the former methods in terms of both higher accuracy and lower computational time.

## References

[1] B. Tausiesakul, "Soft homotopy through Moore-penrose inverse," in *Proc. IEEE 9th Int. Conf. Comput. Intell. Virtual Environments Meas. Syst. Appl. (CIVEMSA)*, Chemnitz, Germany, Jun. 2022, pp. 1–5.

[2] V. M. Patel and R. Chellappa, *Sparse Representations and Compressive Sensing for Imaging and Vision*. New York, NY, USA: Springer Science, 2013.

[3] M. H. Conde, *Compressive Sensing for the Photonic Mixer Device: Fundamentals, Methods and Results*. Wiesbaden, Germany: Springer Fachmedien, 2017.

[4] A. K. Mishra and R. S. Verster, *Compressive Sensing Based Algorithms for Electronic Defence* (Signals and Communication Technology), W.-S. Gan, C.-C. J. Kuo, T. F. Zheng, and M. Barni, Eds. Cham, Switzerland: Springer International, 2017.

[5] M. Testa, D. Valsesia, T. Bianchi, and E. Magli, *Compressed Sensing for Privacy-Preserving Data Processing* (Signal Processing), W.-S. Gan, C.-C. J. Kuo, T. F. Zheng, and M. Barni, Eds. Singapore: Springer Nature, 2019.

[6] M. Amin, *Compressive Sensing for Urban Radar*. Boca Raton, FL, USA: CRC Press, 2015.

[7] A. D. Maio, Y. C. Eldar, and A. M. Haimovich, *Compressed Sensing in Radar Signal Processing*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[8] C. Chen, *Compressive Sensing of Earth Observations* (Signal and Image Processing of Earth Observations), C. Chen, Ed. Boca Raton, FL, USA: CRC Press, 2017.

[9] Z. Han, H. Li, and W. Yin, *Compressive Sensing for Wireless Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[10] L. Kong, B. Wang, and G. Chen, *When Compressive Sensing Meets Mobile Crowdsensing*. Singapore: Springer Nature, 2019.

[11] R. M. Thanki, V. J. Dwivedi, and K. R. Borisagar, *Multibiometric Watermarking With Compressive Sensing Theory: Techniques and Applications* (Signals and Communication Technology). Cham, Switzerland: Springer International, 2018.

[12] M. Khosravy, N. Dey, and C. A. Duque, *Compressive Sensing in Healthcare* (Advances in Ubiquitous Sensing Applications for Healthcare), vol. 11, N. Dey, A. S. Ashour, and S. J. Fong, Eds. London, U.K.: Academic Press, 2020.

[13] D. Malioutov, M. Çetin, and A. S. Willsky, "An affine scaling methodology for best basis selection," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3010–3022, Aug. 2005.

[14] J. A. R. Blais, "Least squares for practitioners," *Math. Problems Eng.*, vol. 2010, pp. 1–19, Sep. 2010.

[15] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[16] H. Boche, R. Calderbank, G. Kutyniok, and J. Vybíral, *Compressed Sensing and Its Applications: MATHEON Workshop* (Applied and Numerical Harmonic Analysis), J. J. Benedetto, Ed. Cham, Switzerland: Birkhäuser, 2015.

[17] S. Boyd and L. Vandenberghe, *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 2018.

[18] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, Jan. 1998.

[19] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Trans. Syst., Man, Cybern., C Appl. Rev.*, vol. 40, no. 1, pp. 1–12, Jan. 2010.

[20] R. Yan, H. Sun, and Y. Qian, "Energy-aware sensor node design with its application in wireless sensor networks," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 5, pp. 1183–1191, May 2013.

[21] K. Luo, J. Li, and J. Wu, "A dynamic compression scheme for energy-efficient real-time wireless electrocardiogram biosensors," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 9, pp. 2160–2169, Sep. 2014.

[22] A. Ravelomanantsoa, H. Rabah, and A. Rouane, "Simple and efficient compressed sensing encoder for wireless body area network," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 12, pp. 2937–2982, Dec. 2014.

[23] J. Zhang, Z. L. Yu, Z. Gu, Y. Li, and Z. Lin, "Multichannel electrocardiogram reconstruction in wireless body sensor networks through weighted $\ell_{1,2}$ minimization," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 9, pp. 2024–2034, Sep. 2018.

[24] A. Ravelomanantsoa, H. Rabah, and A. Rouane, "Compressed sensing: A simple deterministic measurement matrix and a fast recovery algorithm," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 12, pp. 3405–3413, Dec. 2015.

[25] R. Chandra, H. Zhou, I. Balasingham, and R. M. Narayanan, "On the opportunities and challenges in microwave medical sensing and imaging," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 7, pp. 1667–1682, Jul. 2015.

[26] C. Tang, G. Tian, S. Boussakta, and J. Wu, "Feature-supervised compressed sensing for microwave imaging systems," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 8, pp. 5287–5297, Aug. 2020.

[27] X. Yang, Y. R. Zheng, M. T. Ghasr, and K. M. Donnell, "Microwave imaging from sparse measurements for near-field synthetic aperture radar," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 10, pp. 2680–2692, Oct. 2017.

[28] W. Premerlani, B. Kasztenny, and M. Adamiak, "Development and implementation of a synchrophasor estimator capable of measurements under dynamic conditions," *IEEE Trans. Power Del.*, vol. 23, no. 1, pp. 109–123, Jan. 2008.

[29] M. Bertocco, G. Frigo, C. Narduzzi, and F. Tramarin, "Resolution enhancement by compressive sensing in power quality and phasor measurement," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 10, pp. 2358–2367, Oct. 2014.

[30] G. Frigo, P. A. Pegoraro, and S. Toscani, "Enhanced support recovery for PMU measurements based on Taylor–Fourier compressive sensing approach," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 9004211.

[31] D. Carta, C. Muscas, P. A. Pegoraro, A. V. Solinas, and S. Sulis, "Compressive sensing-based harmonic sources identification in smart grids," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.

[32] M. Andersen, J. Dahl, Z. Liu, and L. Vandenberghe, *Interior-Point Methods for Large-Scale Cone Programming* (Optimization for Machine Learning), S. Sra, S. Nowozin, and S. J. Wright, Eds. Cambridge, MA, USA: MIT Press, Sep. 2011.

[33] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc., B Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[34] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[35] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing* (Applied and Numerical Harmonic Analysis), J. J. Benedetto, Ed. New York, NY, USA: Birkhäuser, 2013.

[36] D. A. Lorenz, M. E. Pfetsch, and A. M. Tillmann, "Solving basis pursuit: Heuristic optimality check and solver comparison," *ACM Trans. Math. Softw.*, vol. 41, no. 2, pp. 1–29, Feb. 2015.

[37] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[38] L. Xiao and T. Zhang, "A proximal-gradient homotopy method for the $\ell_1$-regularized least-squares problem," in *Proc. 29th Int. Conf. Mach. Learn. (ICML)*, Edinburgh, Scotland, Jul. 2012, pp. 1–8.

[39] L. Landweber, "An iteration formula for Fredholm integral equations of the first kind," *Amer. J. Math.*, vol. 73, no. 3, pp. 615–624, Jul. 1951.

[40] Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Program.*, vol. 140, no. 1, pp. 125–161, Aug. 2013.

[41] L. Zhang, T. Yang, R. Jin, and Z.-H. Zhou, "A simple homotopy algorithm for compressive sensing," in *Proc. 18th Int. Conf. Artif. Intell. Stat. (AISTATS)*, vol. 38. San Diego, CA, USA: PMLR, May 2015, pp. 1116–1124.

[42] D. Mitra, H. Zanddizari, and S. Rajan, "Investigation of kronecker-based recovery of compressed ECG signal," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 6, pp. 3642–3653, Jun. 2020.

[43] V. Gupta and R. B. Pachori, "FB dictionary based SSBL-EM and its application for multi-class SSVEP classification using eight-channel EEG signals," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–8, 2022.

[44] S. Roy, D. P. Acharya, and A. K. Sahoo, "Incremental Gaussian elimination approach to implement OMP for sparse signal measurement," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 4067–4075, Jul. 2020.

[45] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2909–2913, Jan. 2016.

[46] H. Bai, G. Li, S. Li, Q. Li, Q. Jiang, and L. Chang, "Alternating optimization of sensing matrix and sparsifying dictionary for compressed sensing," *IEEE Trans. Signal Process.*, vol. 63, no. 6, pp. 1581–1594, Mar. 2015.

[47] X. Ding, W. Chen, and I. J. Wassell, "Joint sensing matrix and sparsifying dictionary optimization for tensor compressive sensing," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3632–3646, Jul. 2017.

[48] A. L. Goldberger et al., "Physiobank, Physiotoolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000. [Online]. Available: https://physionet.org/content/chbmit/1.0.0/

[49] A. H. Shoeb, "Application of machine learning to epileptic seizure onset detection and treatment," Ph.D. dissertation, Harvard Univ., MIT Division Health Sci. Technol., Cambridge, MA, USA, 2009. [Online]. Available: https://dspace.mit.edu/handle/1721.1/54669

[50] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.

[51] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2008.

[52] 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, Feb. 2006.

[53] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[54] Y. Ding, *Data Science for Wind Energy*. Boca Raton, FL, USA: CRC Press, 2020.

**Bamrung Tausiesakul** (Member, IEEE) received the B.Eng. and M.Eng. degrees in electrical engineering from Chulalongkorn University, Bangkok, Thailand, in 2001 and 2004, respectively, and the Dr.-Ing. degree in communication technology from Leibniz University Hannover, Hanover, Germany, in 2010.

He was a Post-Doctoral Researcher with the School of Telecommunication Engineering, University of Vigo, Vigo, Spain, from 2011 to 2012, and a Research Associate with the School of Science and Technology, Department of Engineering, City, University of London, London, U.K., from 2013 to 2014. Since 2015, he has been a Lecturer with the Department of Electrical Engineering, Faculty of Engineering, Srinakharinwirot University, Bangkok. His research interests lie in signal processing and wireless communications.

**Krissada Asavaskulkiet** (Member, IEEE) received the B.Eng. and M.Eng. degrees in electrical engineering and the Ph.D. degree from the Department of Electrical Engineering, Chulalongkorn University, Bangkok, Thailand, in 2001, 2004, and 2011, respectively.

He was appointed as a Lecturer and an Assistant Professor at the Department of Electrical Engineering, Mahidol University, Nakhon Pathom, Thailand, in 2011 and 2014, respectively. His current research interests are in image and video processing, signal processing, and digital signal processing (DSP) in telecommunication.