PAPER *Special Section on Information Theory and Its Applications*

Proximal Decoding for LDPC Codes

Tadashi WADAYAMA[†a)], ***Senior Member and*** **Satoshi TAKABE**[††], ***Member***

SUMMARY This paper presents a novel optimization-based decoding algorithm for LDPC codes. The proposed decoding algorithm is based on a proximal gradient method for solving an approximate maximum a posteriori (MAP) decoding problem. The key idea of the proposed algorithm is the use of a code-constraint polynomial to penalize a vector far from a codeword as a regularizer in the approximate MAP objective function. A code proximal operator is naturally derived from a code-constraint polynomial. The proposed algorithm, called proximal decoding, can be described by a simple recursive formula consisting of the gradient descent step for a negative log-likelihood function corresponding to the channel conditional probability density function and the code proximal operation regarding the code-constraint polynomial. Proximal decoding is experimentally shown to be applicable to several non-trivial channel models such as LDPC-coded massive MIMO channels, correlated Gaussian noise channels, and nonlinear vector channels. In particular, in MIMO channels, proximal decoding outperforms known massive MIMO detection algorithms, such as an MMSE detector with belief propagation decoding. The simple optimization-based formulation of proximal decoding allows a way for developing novel signal processing algorithms involving LDPC codes.

key words: *LDPC codes, proximal gradient method, decoding algorithm*

1. Introduction

Low-density parity-check (LDPC) codes [1] are ubiquitous in practical communications and storage systems, such as mobile wireless communications, digital satellite broadcasting, optical communications, hard disks, and flash memories. Belief propagation (BP) decoding is the de facto standard for decoding LDPC codes; however, in some cases, *optimization-based decoding* algorithms have attracted research interest [2], [3]. A number of studies have been inspired by a linear programming formulation [2] of the decoding problem of LDPC codes.

A gradient descent formulation of a non-convex objective function including a penalty function for codewords has led to the concept of a gradient descent bit-flipping (GDBF) algorithm [4], which is suitable for hardware implementation requiring high-speed processing. Some of the variants of the GDBF algorithm, especially the noisy GDBF algorithm [5], provide an excellent tradeoff between decoding performance and circuit complexity. Applications of interior point methods, developed for solving convex problems, to decoding problems have also been studied [6], [7]. Recently, ADMM-based decoding algorithms for LDPC codes [8]–[11] have been proposed. These decoding algorithms have been shown to provide an excellent tradeoff between decoding complexity and decoding performance comparable to that of BP decoding.

In this study, we investigate a new direction for optimization-based decoding based on a *proximal gradient method* [12]. The proximal gradient method is a well-known iterative minimization algorithm for convex optimization problems [13]. For example, the iterative soft-thresholding algorithm (ISTA) [14], which is an efficient sparse signal recovery algorithm, is an instance of the proximal gradient method. The proposed algorithm, referred to as *proximal decoding*, is inspired by ISTA. The key idea of proximal decoding is the use of *code-constraint polynomials* to penalize a vector far from a codeword. A code proximal operator is naturally derived from a code-constraint polynomial, which is the most important component of the proposed algorithm. The proximal decoding process involves a gradient descent step for the negative log-likelihood of the channel and a proximal step based on the code proximal operator. Because the principle of proximal decoding is simple, we can naturally derive a variant of proximal decoding specific to a given channel.

The main contributions of this study are as follows: 1) presentation of a novel formulation of optimization-based decoding, i.e., proximal decoding; 2) demonstration of the behavior of proximal decoding for several non-trivial channel models such as LDPC-coded massive MIMO channels, correlated Gaussian noise channels, and nonlinear vector channels.

Derivation of a channel-specific decoding algorithm is simple and can be achieved in a coherent manner. The first step of the derivation is to formulate a decoding problem as a regularized regression or regularized least-squares problem similar to the LASSO formulation [15] for sparse signal recovery. The code-constraint polynomial acts as a regularizer in such a formulation. The proximal gradient method for solving the regularized regression problem is equivalent to proximal decoding. Thus, several ideas and techniques that have been fostered in the fields of machine learning and sparse signal recovery can be naturally applied to decoding problems.

Manuscript received February 9, 2022.
Manuscript revised May 23, 2022.
Manuscript publicized September 1, 2022.

[†] The author is with the Department of Computer Science, Nagoya Institute of Technology, Nagoya-shi, 466-8555 Japan.

[††] The author is with the Department of Mathematical and Computing Science, Tokyo Institute of Technology, Tokyo, 152-8550 Japan.

a) E-mail: wadayama@nitech.ac.jp

DOI: 10.1587/transfun.2022TAP0002

2. Code-Constraint Polynomial

2.1 Notation

Let n be a positive integer representing the code length. A binary matrix $\boldsymbol{H} \in \mathbb{F}_2^{m\times n}$ is a parity-check matrix, and $\tilde{C}(\boldsymbol{H})$ is the binary linear code defined by $\boldsymbol{H}$, i.e., $\tilde{C}(\boldsymbol{H}) \equiv \{\boldsymbol{b} \in \mathbb{F}_2^n \mid \boldsymbol{H}\boldsymbol{b}^T = \boldsymbol{0}\}$. A binary to bipolar transform $\beta : \mathbb{F}_2 \to \{1,-1\}$ is defined as $\beta(0) \equiv 1$ and $\beta(1) \equiv -1$. The bipolar code $C(\boldsymbol{H})$ is simply given by $C(\boldsymbol{H}) \equiv \{\beta(\boldsymbol{b}) \in \{1,-1\}^n \mid \boldsymbol{b} \in \tilde{C}(\boldsymbol{H})\}$.

The index sets $A(i)$ and $B(j)$ are defined as

$$A(i) \equiv \{j \mid j \in [n], H_{i,j} = 1\}, \quad i \in [m], \tag{1}$$

$$B(j) \equiv \{i \mid i \in [m], H_{i,j} = 1\}, \quad j \in [n], \tag{2}$$

respectively, where $H_{i,j}$ denotes the (i,j)-element of $\boldsymbol{H}$. The notation $[n]$ represents the set $\{1,2,\dots,n\}$. The multivariate Gaussian distribution with mean vector $\boldsymbol{m}$ and covariance $\boldsymbol{\Sigma}$ is denoted by $\mathcal{N}(\boldsymbol{m},\boldsymbol{\Sigma})$.

2.2 Definition of Code-Constraint Polynomial

The *code-constraint polynomial* for $C(\boldsymbol{H})$ is a multivariate polynomial defined as

$$h(\boldsymbol{x}) \equiv \sum_{j=1}^{n}(x_j^2-1)^2 + \sum_{i=1}^{m}\left(\left(\prod_{j\in A(i)} x_j\right)-1\right)^2, \tag{3}$$

where $\boldsymbol{x} \equiv (x_1,\dots,x_n) \in \mathbb{R}^n$. The first term on the right-hand side of this equation represents the bipolar constraint for $\boldsymbol{x} \in \{+1,-1\}^n$, and the second term corresponds to the parity constraint induced by $\boldsymbol{H}$, i.e., if $\boldsymbol{x} \in C(\boldsymbol{H})$, we have $\left(\prod_{j\in A(i)} x_j\right) - 1 = 0$ for any $i \in [m]$. Since the polynomial $h(\boldsymbol{x})$ has a *sum-of-squares* (SOS) form, it can be regarded as a penalty function that gives positive penalty values for non-codeword vectors in $\mathbb{R}^n$. The code-constraint polynomial $h(\boldsymbol{x})$ is inspired by the non-convex parity constraint function used in the GDBF objective function [4]. The SOS form directly implies the most important property of $h(\boldsymbol{x})$, i.e., the inequality $h(\boldsymbol{x}) \geq 0$ holds for any $\boldsymbol{x} \in \mathbb{R}^n$. The equality holds if and only if $\boldsymbol{x} \in C(\boldsymbol{H})$.

2.3 Gradient

In the following discussion, we need the gradient of $h(\boldsymbol{x})$. The first-order derivative of $h(\boldsymbol{x})$ with respect to $x_k (k \in [n])$ is given by

$$\begin{aligned}&\frac{\partial}{\partial x_k}h(\boldsymbol{x})\\ &= 4(x_k^2-1)x_k + \frac{2}{x_k}\sum_{i\in B(k)}\left(\left(\prod_{j\in A(i)} x_j\right)^2 - \prod_{j\in A(i)} x_j\right).\end{aligned} \tag{4}$$

Hence, the gradient $\nabla h(\boldsymbol{x})$ is given by

$$\nabla h(\boldsymbol{x}) = \left(\frac{\partial}{\partial x_1}h(\boldsymbol{x}),\dots,\frac{\partial}{\partial x_n}h(\boldsymbol{x})\right)^T. \tag{5}$$

The point $\boldsymbol{x} \in \mathbb{R}^n$ satisfying the equality $\nabla h(\boldsymbol{x}) = \boldsymbol{0}$ is a stationary point of h. For any codeword $\boldsymbol{x} \in C(\boldsymbol{H})$, $x_k^2 = 1$ for any $k \in [n]$ and $\prod_{j\in A(i)} x_j = 1$ holds for any $i \in [m]$. This implies that $\nabla h(\boldsymbol{x}) = \boldsymbol{0}$, i.e., a codeword vector is a stationary point of h.

Assume that a non-codeword bipolar vector $\boldsymbol{x} \in \{1,-1\}^n$ satisfying $\boldsymbol{x} \notin C(\boldsymbol{H})$ is given. In such a case, the parity constraints are violated as $\prod_{j\in A(i)} x_j = -1$ for some i. This leads to the nonzero gradient due to $\left(\prod_{j\in A(i)} x_j\right)^2 - \prod_{j\in A(i)} x_j = 1-(-1) = 2$. This implies that $\boldsymbol{x}$ is not a stationary point.

The above-mentioned argument can be summarized as follows. A codeword $\boldsymbol{x} \in C(\boldsymbol{H})$ is a stationary point of h and a non-codeword bipolar vector $\boldsymbol{x} \in \{1,-1\}^n, \boldsymbol{x} \notin C(\boldsymbol{H})$ cannot be a stationary point. A stationary point that is a codeword of $C(\boldsymbol{H})$ is referred to as a *codeword stationary point*. However, the opposite is not true, i.e., a stationary point is not necessarily a bipolar codeword in $C(\boldsymbol{H})$. For example, the zero vector $\boldsymbol{0} \in \mathbb{R}^n$ is not a bipolar codeword but it is a stationary point.

The code-constraint polynomials as multivariate functions of $\boldsymbol{x}$ are non-convex and have several local minima and maxima in general. However, if the initial point is sufficiently close to a codeword stationary point, then a gradient descent process produces a convergent point sequence to the corresponding codeword. This *pull-in property* is of great importance in proximal decoding.

3. Principle of Proximal Decoding

3.1 Approximate Maximum a Posteriori (MAP) Decoding

Assume that a sender transmits a codeword of $C(\boldsymbol{H})$ to a given channel. The channel is defined by a probability density function (PDF), $p(\boldsymbol{y}|\boldsymbol{x})(\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^n)$. The negative log-likelihood is defined as $L(\boldsymbol{x};\boldsymbol{y}) \equiv -\ln p(\boldsymbol{y}|\boldsymbol{x})$. The MAP decoding rule is expressed as $\hat{\boldsymbol{x}} \equiv \mathrm{argmax}_{\boldsymbol{x}\in\mathbb{R}^n}\, p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$, where $p(\boldsymbol{x})$ is the prior PDF on the input space. It is natural to make the equal probability assumption on $C(\boldsymbol{H})$, which is given by

$$p(\boldsymbol{x}) \equiv \frac{1}{|C(\boldsymbol{H})|}\sum_{\boldsymbol{c}\in C(\boldsymbol{H})}\delta(\boldsymbol{x}-\boldsymbol{c}), \tag{6}$$

where δ is Dirac's delta function. Instead of the true $p(\boldsymbol{x})$ above, here, we assume a prior PDF with the form $\tilde{p}(\boldsymbol{x}) \equiv \frac{1}{Z}\exp(-\gamma h(\boldsymbol{x}))$, where Z is the normalizing constant and γ is a positive constant. Note that, at the limit $\gamma \longrightarrow \infty$, we have

$$\tilde{p}(\boldsymbol{x}) = \frac{1}{Z}\exp(-\gamma h(\boldsymbol{x})) \to \frac{1}{|C(\boldsymbol{H})|}\sum_{\boldsymbol{c}\in C(\boldsymbol{H})}\delta(\boldsymbol{x}-\boldsymbol{c}). \tag{7}$$

This justifies the use of $\tilde{p}(\boldsymbol{x})$ as an approximation of $p(\boldsymbol{x})$. By using this result, we immediately have the approximation

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}) \simeq p(\boldsymbol{y}|\boldsymbol{x})\tilde{p}(\boldsymbol{x}) \\ &= \exp\left(-L(\boldsymbol{x};\boldsymbol{y}) - \gamma h(\boldsymbol{x})\right). \end{aligned} \tag{8}$$

Hence, the approximate MAP rule considered here is given by

$$\hat{\boldsymbol{x}} \equiv \mathrm{argmin}_{\boldsymbol{x}\in\mathbb{R}^n}\ L(\boldsymbol{x};\boldsymbol{y}) + \gamma h(\boldsymbol{x}). \tag{9}$$

The problem can be regarded as a regression problem with a regularizer given by $h(\boldsymbol{x})$. Note that the minimization problem (9) is similar to the LASSO problem [15] for sparse signal recovery. The ISTA is derived from the LASSO formulation. It is natural to consider a counterpart of the ISTA for (9), i.e., proximal decoding, which will be presented in the next subsection.

Note that the value of γ controls the landscape of the objective function:

$$f(\boldsymbol{x}) \equiv L(\boldsymbol{x};\boldsymbol{y}) + \gamma h(\boldsymbol{x}). \tag{10}$$

If we choose a large γ, many undesirable stationary points, including local minima, appear in the landscape. This means that a decoding algorithm based on a gradient descent method tends to fail successful decoding due to these stationary points. We thus need to keep the value of γ being a particular finite value although MAP decoding is achieved when $\gamma \to \infty$. In practice, the value of γ should be adjusted according to the decoding performance.

3.2 Proximal Decoding

Solving the approximate MAP problem (9) can be regarded as a non-convex minimization problem. To solve the approximate MAP problem efficiently, we will use the proximal gradient method [12]. The proximal operator of $f : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\mathrm{prox}_f(\boldsymbol{v}) \equiv \mathrm{argmin}_{\boldsymbol{x}\in\mathbb{R}^n}\left(f(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}\|^2\right), \tag{11}$$

where $\|\cdot\|$ represents the Euclidean norm. The proximal operators can be regarded as a generalized projection. It is known that a proximal operator can be well approximated by a gradient descent step (page 126 of [12]). Thus, the proximal operator $\mathrm{prox}_{\gamma h}(\boldsymbol{x})$ can be approximated by

$$\mathrm{prox}_{\gamma h}(\boldsymbol{x}) \simeq \boldsymbol{x} - \gamma\nabla h(\boldsymbol{x}), \tag{12}$$

where the approximated proximal operator is said to be the *code-proximal operator*.

The proximal decoding proposed in this paper is given by the following iterative process:

$$\boldsymbol{r}^{(k+1)} = \boldsymbol{s}^{(k)} - \omega\nabla L(\boldsymbol{s}^{(k)};\boldsymbol{y}) \tag{13}$$

$$\boldsymbol{s}^{(k+1)} = \boldsymbol{r}^{(k+1)} - \gamma\nabla h(\boldsymbol{r}^{(k+1)}), \tag{14}$$

for $k = 0, 1, 2, \ldots$, where ω is a positive real number representing the step-size parameter of a gradient descent process in (13). The step indicated by (13) is referred to as the gradient descent step, and the step indicated by (14) is referred to as the code-proximal step. The entire procedure of proximal decoding is summarized in Algorithm 1.

Algorithm 1 Proximal decoding (general form)

```
$\boldsymbol{s}^{(0)} := \mathbf{0}$
for $k := 0$ to $K-1$ do
    $\boldsymbol{r}^{(k+1)} := \boldsymbol{s}^{(k)} - \omega\nabla L(\boldsymbol{s}^{(k)};\boldsymbol{y})$
    Compute $\nabla h(\boldsymbol{r}^{(k+1)})$ according to (4).
    $\boldsymbol{s}^{(k+1)} := \boldsymbol{r}^{(k+1)} - \gamma\nabla h(\boldsymbol{r}^{(k+1)})$
    $\hat{\boldsymbol{x}} := \mathrm{sign}(\boldsymbol{s}^{(k+1)})$
    If $\hat{\boldsymbol{x}}$ passes the parity-check condition, break the loop.
end for
Output $\hat{\boldsymbol{x}}$
```

Let $B_\eta \equiv [-\eta, \eta]^n$ where η is a positive constant slightly larger than one, be the n-dimensional hypercube, where $[a, b] \equiv \{x \in \mathbb{R} \mid a \le x \le b\}$. The gradient norm $\|\nabla h(\boldsymbol{x})\|$ tends to be extremely large if $\boldsymbol{x} \notin B_\eta$ owing to a property of the code-constraint polynomial. In the proximal decoding process defined above, this may cause numerical instability (oscillation or divergent behavior) in some cases. In such cases, we can use

$$\boldsymbol{s}^{(k+1)} = \Pi_\eta\left(\boldsymbol{r}^{(k+1)} - \gamma\nabla h(\boldsymbol{r}^{(k+1)})\right) \tag{15}$$

instead of (14) to prevent numerical instability. The projection operator $\Pi_\eta : \mathbb{R}^n \to B_\eta$ represents the projection onto B_η, i.e., the projection operator is defined by

$$\Pi_\eta(\boldsymbol{x}) = \arg\min_{\boldsymbol{x}'\in B_\eta}\|\boldsymbol{x}-\boldsymbol{x}'\|. \tag{16}$$

Let us discuss the time complexity per iteration of the proximal operation. In the following argument, we assume an LDPC code, i.e., the number of ones in $\boldsymbol{H}$ is $O(n)$. For evaluating $\boldsymbol{r}^{(k+1)} - \gamma\nabla h(\boldsymbol{r}^{(k+1)})$, we require the gradient of $h(\boldsymbol{x})$. All the quantities $\prod_{j\in A(i)} x_j$ for $i \in [m]$ can be calculated with time complexity $O(n)$ because of the assumption that $\boldsymbol{H}$ is sparse. This means that the time complexity for evaluating the gradient of $h(\boldsymbol{x})$ is $O(n)$, which is practical time complexity, because $O(n)$ is the same as the complexity of belief propagation (BP) decoding for LDPC codes. Note that the computation of the gradient of $h(\boldsymbol{x})$ requires only multiplication and addition. There is no need to compute a nonlinear function such as tanh required for BP.

4. Variations of Proximal Decoding

4.1 Proximal Decoding for AWGN Channel

As one of the simplest instances of proximal decoding, a variant for additive white Gaussian noise (AWGN) channels is discussed first. The AWGN channel is simply described as $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{w}$ where $\boldsymbol{w}$ is the additive white Gaussian noise term. The noise vector $\boldsymbol{w}$ follows the Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I})$. In this case, the approximate MAP rule can be written as $\hat{\boldsymbol{x}} = \mathrm{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 + \gamma h(\boldsymbol{x})$ based on the assumption of i.i.d. Gaussian noise. Because $\nabla\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 = \boldsymbol{x}-\boldsymbol{y}$, the core of proximal decoding for the AWGN channel can be summarized by

$$\boldsymbol{r}^{(k+1)} = \boldsymbol{s}^{(k)} - \omega(\boldsymbol{s}^{(k)} - \boldsymbol{y}) \tag{17}$$

$$\boldsymbol{s}^{(k+1)} = \boldsymbol{r}^{(k+1)} - \gamma\nabla h(\boldsymbol{r}^{(k+1)}). \tag{18}$$

The initial point can be set to $\boldsymbol{s}^{(0)} = \boldsymbol{0}$.

The time complexity for evaluating the above-mentioned recursive equations per iteration is $O(n)$ if $\boldsymbol{H}$ defies an LDPC code. Note that the computation of the recursive equations requires only addition and multiplication of real numbers, which is a desirable property for an implementation requiring high-speed software processing.

4.2 Proximal Decoding for Massive MIMO Channels

In this subsection, we focus on a massive MIMO channel as a target channel because decoding and detection problems for LDPC-coded massive MIMO channels are nontrivial problems that are practically important [16] for wireless cellular networks referred to as fifth-generation (5G) systems, as well as for future systems such as beyond-5G/6G systems. The authors recently proposed a detection algorithm for overloaded massive MIMO channels [17]. The architecture of the detection algorithm proposed in the previous study [17] is another trigger for the development of proximal decoding for massive MIMO channels.

Recall that the principle of proximal decoding is applicable to any channel model if we precisely know the negative log-likelihood function $L(\boldsymbol{x};\boldsymbol{y})$, and its gradient $\nabla L(\boldsymbol{x};\boldsymbol{y})$ can be evaluated efficiently.

Let $\boldsymbol{A} \in \mathbb{R}^{\mu\times n}$ be a channel matrix. Suppose that a received word $\boldsymbol{y} \in \mathbb{R}^{\mu}$ is given by $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$, where $\boldsymbol{w} \in \mathbb{R}^{\mu}$ is a Gaussian noise vector, the components of which follow an i.i.d. Gaussian distribution. The channel input vector $\boldsymbol{x}$ is assumed to be a codeword of $C(\boldsymbol{H})$, which implies that we assume BPSK modulation. In this problem setting, the PDF representing the channel is given by

$$p(\boldsymbol{y}|\boldsymbol{x}) = a\exp\left(-b\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2\right),$$

where a and b are positive constants. Hence, we have the following approximate MAP decoding problem for an LDPC-coded massive MIMO channel:

$$\hat{\boldsymbol{x}} \equiv \mathrm{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2 + \gamma h(\boldsymbol{x}). \tag{19}$$

Since $\nabla\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2 = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y})$, an iteration of proximal decoding for LDPC-coded massive MIMO channels can be summarized as

$$\boldsymbol{r}^{(k+1)} = \boldsymbol{s}^{(k)} - \omega\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{s}^{(k)} - \boldsymbol{y}) \tag{20}$$

$$\boldsymbol{s}^{(k+1)} = \boldsymbol{r}^{(k+1)} - \gamma\nabla h(\boldsymbol{r}^{(k+1)}). \tag{21}$$

In the following experiments, we set the initial value $\boldsymbol{s}^{(0)} = \boldsymbol{0}$. However, there are alternative choices for the initial point, i.e., an estimate of the zero-forcing detector or the MMSE detector can be used as an initial point.

4.3 Proximal Decoding for Additive Correlated Gaussian Noise

In this subsection, we will study the case in which the additive noise vector follows a correlated Gaussian PDF. The idea of proximal decoding can be naturally applied to this case as well. In this case, the negative log-likelihood function takes on a quadratic form involving the inverse of the covariance matrix of the additive noise. The channel model is given by $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{w}$, where $\boldsymbol{x} \in C(\boldsymbol{H})$ and $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$. The covariance matrix $\boldsymbol{\Sigma}$ is assumed to be a symmetric positive definite matrix. The precision matrix $\boldsymbol{G}$ is defined by $\boldsymbol{G} = \boldsymbol{\Sigma}^{-1}$. In this case, the approximate MAP rule can be represented as

$$\hat{\boldsymbol{x}} \equiv \mathrm{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} \frac{1}{2}(\boldsymbol{y}-\boldsymbol{x})^T\boldsymbol{G}(\boldsymbol{y}-\boldsymbol{x}) + \gamma h(\boldsymbol{x}) \tag{22}$$

owing to the definition of the multivariate Gaussian PDF. The gradient corresponding to the negative log-likelihood is given by $\nabla\frac{1}{2}(\boldsymbol{y}-\boldsymbol{x})^T\boldsymbol{G}(\boldsymbol{y}-\boldsymbol{x}) = \boldsymbol{G}(\boldsymbol{x}-\boldsymbol{y})$.

Proximal decoding for additive correlated Gaussian noise channels is defined by the following recursive equations:

$$\boldsymbol{r}^{(k+1)} = \boldsymbol{s}^{(k)} - \omega\boldsymbol{G}(\boldsymbol{s}^{(k)} - \boldsymbol{y}) \tag{23}$$

$$\boldsymbol{s}^{(k+1)} = \boldsymbol{r}^{(k+1)} - \gamma\nabla h(\boldsymbol{r}^{(k+1)}). \tag{24}$$

The time complexity for evaluating the above-mentioned recursive equation is $O(n^2)$ per iteration because the multiplication $\boldsymbol{G}(\boldsymbol{x}-\boldsymbol{y})$ is the dominant computation in the case of LDPC codes.

4.4 Proximal Decoding for Nonlinear Vector Channels

In this subsection, we will study a fairly general channel model described by $\boldsymbol{y} = f(\boldsymbol{x}) + \boldsymbol{w}$, where $f : \mathbb{R}^n \to \mathbb{R}^{\nu}$ is a nonlinear vector function. We also assume that f is differentiable because we need to calculate a gradient vector involving f. The noise term $\boldsymbol{w} \in \mathbb{R}^{\nu} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$ represents additive white Gaussian noise. In this case, the approximate MAP decoding associated with the nonlinear vector channel model is given by $\hat{\boldsymbol{x}} \equiv \mathrm{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{y} - f(\boldsymbol{x})\|^2 + \gamma h(\boldsymbol{x})$. From this approximate MAP rule, we immediately have the core recursive formula of proximal decoding as

$$\boldsymbol{r}^{(k+1)} = \boldsymbol{s}^{(k)} - \omega\nabla\|\boldsymbol{y} - f(\boldsymbol{s}^{(k)})\|^2 \tag{25}$$

$$\boldsymbol{s}^{(k+1)} = \boldsymbol{r}^{(k+1)} - \gamma\nabla h(\boldsymbol{r}^{(k+1)}). \tag{26}$$

A nonlinear distortion of a transmitted signal may occur in a real channel. Nonlinearity of a power amplifier in a wireless transmitter is an evident example. A nonlinear clipping operation for reducing the peak-to-average power ratio (PAPR) is another source of nonlinear distortion in a wireless system. In such a case, the channel can be modeled by a nonlinear vector channel. It seems interesting to devise a decoding algorithm suitable for such a channel. Another motivation comes from the possibility of using a parametric function f_Θ to model the channel characteristics, where Θ is a set of controllable parameters. A neural network [18] is an evident example of such a parametric function. We often encounter an inevitable nonlinear effect in fields such as optical fiber communications, magnetic recording, and flash memories. Parametric modeling via f_Θ seems to be a promising approach for such a channel with nonlinear distortion.

For practical implementation, evaluation of the gradient $\nabla\|\boldsymbol{y} - f(\boldsymbol{x})\|^2$ should be efficient. In the following, we will discuss two scenarios in which the evaluation of the gradient is computationally tractable.

4.4.1 Component-Wise Nonlinearity

Here, we consider the case in which f is given by $f(\boldsymbol{x}) \equiv g(\boldsymbol{A}\boldsymbol{x})$, where $g : \mathbb{R} \to \mathbb{R}$ is a differential function. The matrix $\boldsymbol{A} \in \mathbb{R}^{\nu \times n}$ is assumed to be a full-rank matrix. The function g is applied component-wise to each component of $\boldsymbol{A}\boldsymbol{x}$.

For example, soft peak clipping by tanh for OFDM signals can be represented as $f(\boldsymbol{x}) = \tanh(\boldsymbol{D}\boldsymbol{x})$, where $\boldsymbol{D}$ is the inverse discrete Fourier matrix (DFT)†.

The gradient of $g(\boldsymbol{A}\boldsymbol{x})$ has a concise form:

$$\frac{1}{2}\nabla\|\boldsymbol{y} - g(\boldsymbol{A}\boldsymbol{x})\|^2 = -\boldsymbol{A}^T((\boldsymbol{y} - g(\boldsymbol{A}\boldsymbol{x})) \odot g'(\boldsymbol{A}\boldsymbol{x})), \tag{27}$$

where $\odot$ is the Hadamard operator representing the component-wise product of two vectors. If the evaluation of g and g' requires constant time, the evaluation of the above-mentioned gradient requires $O(n^2)$ computation, which is the allowable time complexity in a proximal decoding process.

4.4.2 Use of Back-Propagation

Suppose that the gradient of the squared norm $\nabla\|\boldsymbol{y} - f(\boldsymbol{x})\|^2$ can be efficiently evaluated by back-propagation [18]. In such a case, we can embed back-propagation in a proximal decoding process. Computation of the gradient $\nabla\|\boldsymbol{y} - f(\boldsymbol{x})\|^2$ by back-propagation is beneficial for implementing a proximal decoder because we only need to specify the forward model of f without preparing a backward model for computing $\nabla\|\boldsymbol{y} - f(\boldsymbol{x})\|^2$ such as (27). For example, in the case of multi-layer neural networks, expressing the gradient becomes quite cumbersome.

Another advantage of the use of back-propagation is its suitability for feed-forward neural networks. When the function f is implemented by a feed-forward neural network, we can train f based on the dataset, which leads to a data-driven design of a decoder.

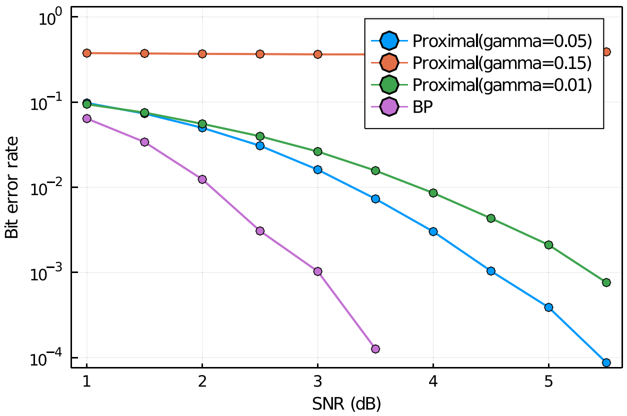

Fig. 1 Bit error rates of proximal decoding for AWGN channel. The BER performance of BP decoding is also included as a baseline. The code is the regular LDPC code (3, 6) with $n = 204, m = 102$.

5. Numerical Experiments

5.1 Proximal Decoding for AWGN Channel

Section 4.1 discussed proximal decoding for AWGN channels. Here, we present several experimental results of proximal decoding for AWGN channels. We conducted numerical experiments for measuring the bit error rate (BER) of proximal decoding. A (3,6)-regular LDPC code with $n = 204$ and $m = 102$ [19] was used as the target code. Figure 1 shows BER obtained by a numerical experiment. The horizontal axis represents SNR, i.e., $E_b/N_0 = 2/(R\sigma^2)$ in dB where R is the design code rate, $1 - m/n$. As a baseline, the BER performance of the belief propagation (BP) decoder is also included in the figure. We tested the three cases $\gamma \in \{0.01, 0.05, 0.15\}$. It is observed that the BER performance crucially depends on the choice of γ. The BER curve of proximal decoding with $\gamma = 0.05$ shows moderate decoding performance but there are gaps in the BP performance.

Although proximal decoding provides larger BER values compared with BP decoding over AWGN channels, one can consider a proximal decoder as a candidate of a reduced complexity decoder for applications requiring extremely high throughput. This is because proximal decoding for the AWGN channel requires no nonlinear function computation, such as tanh required for BP, or the minimum operation necessary for the min-sum algorithm. Only addition and multiplication operations are required for implementing a proximal decoder for the AWGN channel.

5.2 Proximal Decoding for LDPC-Coded MIMO Channel

5.2.1 Problem Setup

In this subsection, we follow the real-valued MIMO model discussed in [17]. Let $\boldsymbol{A}' \equiv \{a'_{i,j}\} \in \mathbb{C}^{M \times N}$ be a channel matrix, where $a'_{i,j}$ is the fading coefficient corresponding to the path between the jth transmit antenna and the ith

---

[†] In the case of a complex matrix such as the DFT matrix $\boldsymbol{D}$, a gradient descent step must follow the Wirtinger derivative of the negative log-likelihood function.

receive antenna. Here, we assume that each component of $\boldsymbol{A}'$ follows the *Kronecker model* [20], which is a simple channel model representing the spatial correlation between antenna elements. Let $\rho(0 \le \rho < 1)$ be the spatial correlation factor. The correlation matrix for the receiver side is given by $\boldsymbol{R}_r \equiv \{r_{i,j}\}_{1\le i,j\le M}, r_{i,j} = \rho^{|i-j|}$ and the correlation matrix for the transmitter side is given by $\boldsymbol{R}_t \equiv \{t_{i,j}\}_{1\le i,j\le N}, t_{i,j} \equiv \rho^{|i-j|}$. In the Kronecker model, a channel matrix $\boldsymbol{A}'$ is represented by $\boldsymbol{A}' \equiv \boldsymbol{R}_r^{1/2}\boldsymbol{G}(\boldsymbol{R}_t^{1/2})^T$, where each element of the matrix $\boldsymbol{G} \in \mathbb{C}^{M\times N}$ follows a complex circular Gaussian PDF with zero mean and unit variance. Note that $\boldsymbol{A}' = \boldsymbol{G}$ holds when there is no spatial correlation, i.e., $\rho = 0$.

We assume QPSK modulation for transmitted signals. An equivalent real-valued MIMO model with BPSK modulation can be defined as $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$, where $\boldsymbol{A}$ is given by

$$\boldsymbol{A} \equiv \begin{bmatrix} \mathrm{Re}(\boldsymbol{A}') & -\mathrm{Im}(\boldsymbol{A}') \\ \mathrm{Im}(\boldsymbol{A}') & \mathrm{Re}(\boldsymbol{A}') \end{bmatrix} \in \mathbb{R}^{\mu\times n}.$$

Note that $\mu = 2M$ and $n = 2N$ hold. The transmitted word $\boldsymbol{x}$ is randomly chosen from $C(\boldsymbol{H})$ according to the uniform distribution. Each component of the noise vector $\boldsymbol{w} \in \mathbb{R}^\mu$ is an i.i.d. Gaussian PDF with zero mean and variance $\sigma_w^2/2$. In this model, σ_w^2 is related to the signal-to-noise ratio SNR by $\sigma_w^2 = (2N)/\mathsf{SNR}$. The details of the equivalence of the complex-valued model and real-valued model can be found in [17]. In the following experiment, we used the regular (3,6)-LDPC code with $n = 204$ and $m = 102$. The step-size parameter ω used in the gradient descent step was set to $\omega \equiv 2/(\lambda_{min}+\lambda_{max})$, where λ_{min} and λ_{max} are the minimum and maximum eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$, respectively. In the following experiments, we used the box projection (14) with $\eta = 1.5$ in the proximal step.

5.2.2 Baseline Schemes

For comparison, we exploited a proximal-based detection algorithm, referred to as the *tanh detector*, given by the following recursion [17], [21]:

$$\boldsymbol{r}^{(k+1)} = \boldsymbol{s}^{(k)} - \omega\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{s}^{(k)} - \boldsymbol{y}), \tag{28}$$

$$\boldsymbol{s}^{(k+1)} = \tanh(\alpha\boldsymbol{r}^{(k+1)}), \tag{29}$$

where α is a positive real value. Furthermore, the MMSE detector defined as $\hat{\boldsymbol{x}} \equiv \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T + (\sigma_w^2/2)\boldsymbol{I})^{-1}\boldsymbol{y}$ was also examined as a baseline scheme. Furthermore, as a baseline for joint detection and decoding, we employed the combination of the MMSE detector and BP decoding, denoted by MMSE + BP. Although the orthogonal AMP [22] is a state-of-the-art algorithm for joint detection and decoding, we here use MMSE+BP as a baseline because it is a practically common joint detection and decoding algorithm.

5.2.3 Convergence Behavior

Let $\hat{\boldsymbol{x}}$ be the estimated word obtained from these detection algorithms. The performance measure used herein is the averaged error value of $\|\boldsymbol{x} - \mathrm{sign}(\hat{\boldsymbol{x}})\|$, where $\boldsymbol{x}$ is the transmitted word, and $\hat{\boldsymbol{x}}$ indicates an estimate obtained from the detector.

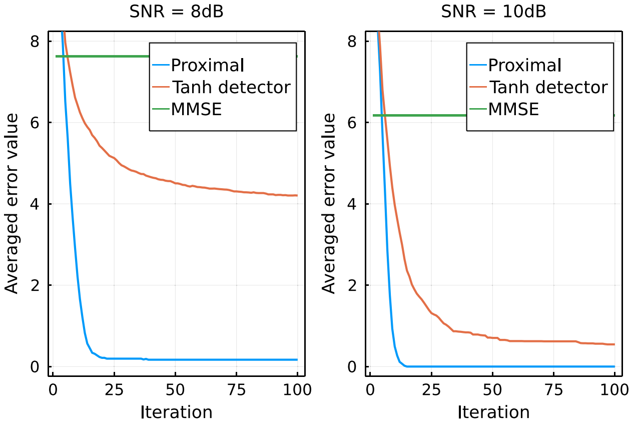

Fig. 2 Comparison of the averaged error value $\|\boldsymbol{x}-\mathrm{sign}(\hat{\boldsymbol{x}})\|$ under $\rho = 0$ (no spatial correlation). The numbers of received and transmitted antennas were $N = M = 102$. (Left): SNR = 8 (dB); right: SNR = 10 (dB). The error values were averaged for 100 trials. The choice of the parameter γ is critical for achieving appropriate performance of proximal decoding. We set $\gamma = 0.05$ for these experiments. The parameter α used in the tanh detector was set to 2.0. In all the schemes, the step-size parameter was set to $\omega = 2.0/(\lambda_{min} + \lambda_{max})$.

Figure 2 shows the averaged error as a function of the number of iterations when there is no spatial correlation, i.e., $\rho = 0.0$. Proximal decoding provides much smaller averaged error values and faster convergence compared with the tanh detector and MMSE detector. Moreover, the saturated value of proximal decoding is much smaller than that of the tanh detector. These results imply that the parity constraint included in the code-constraint polynomial is fairly beneficial in terms of obtaining a reasonable solution. We also observed that the convergence speed of proximal decoding is sensitive to the choice of γ. A pre-experiment was conducted to find an appropriate setting of γ. In this experiment, $\gamma = 0.05$ provided the best result.

5.2.4 Bit Error Rate Performance

The BER is the primary performance measure for detection algorithms for massive MIMO systems. Here, we investigated the BER performance of proximal decoding and several benchmark schemes, such as the tanh detector and MMSE detector (with/without BP decoding). The input of the BP decoder after MMSE detection was set to $\xi\hat{\boldsymbol{x}}$, where ξ is a positive constant, and $\hat{\boldsymbol{x}}$ is an MMSE estimation vector without binary quantization. The value of the scaling parameter ξ is crucial for deriving the full performance of BP decoding. In the following experiments, we set $\xi = 5$, which was adjusted heuristically. The channel model was the Kronecker model described in Sect. 5.2.1.

Figure 3 shows the BER performances of the proposed and benchmark schemes. The left-hand panel in Fig. 3 represents the case of no spatial correlation ($\rho = 0$), and the

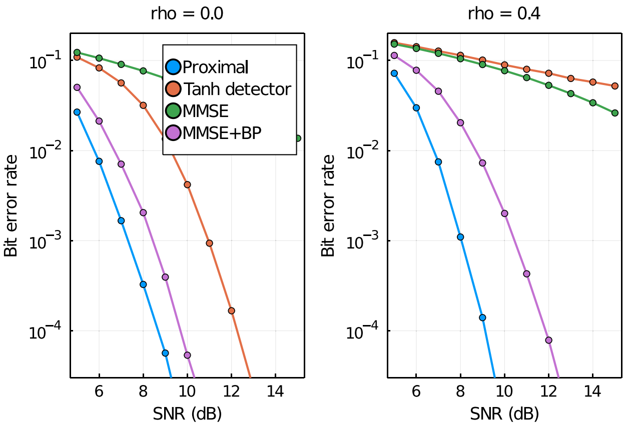

Fig. 3 Bit error rate performances of proximal decoding and baseline schemes. Left: without spatial correlation ($\rho = 0$); right with spatial correlation ($\rho = 0.4$). The number of received and transmitted antennas are $N = M = 102$. To obtain a BER point, 10,000 codewords were used for BER estimation. The parameter γ used in the code proximal operator was set to 0.05. The step-size parameter was set to $\omega = 2.0/(\lambda_{min} + \lambda_{max})$. The number of iterations for BP was 20, and the number of iterations for proximal decoding and the tanh detector was set to 50. The scaling factor $\xi = 5$ was used for MMSE+BP.

right-hand panel represents the results under the spatial correlation ($\rho = 0.4$). Here, although the MMSE detector is the simplest detector, the error curve is not very steep in either Fig. 3 (Left) or Fig. 3 (Right). Furthermore, the MMSE detector involves the inversion of a matrix requiring time complexity $O(n^3)$, which is not negligible in terms of the complexity in a massive MIMO scenario. The combination of the MMSE detector followed by the BP decoder (MMSE+BP) is a standard and practical configuration of a receiver for LDPC-coded massive MIMO channels.

The BER performance of MMSE + BP provides a much steeper error curve compared to the plain MMSE error curve. The tanh detector also achieves much smaller BERs compared with the naive MMSE detector when $\rho = 0$. Compared with the tanh detector and MMSE detector (with/without BP decoding), the BERs of proximal decoding are the smallest. In particular, the margin between the proposed method and MMSE + BP is approximately 3 dB at BER = 10^{-4} in Fig. 3 (right). Comparing Fig. 3 (Left) and 3 (Right) shows that the performance of MMSE+BP is degraded as ρ increases. The proposed method provides similar BER performances in both cases.

Although several studies have discussed joint detection and decoding for LDPC-coded MIMO channels, such as [23], [24], their time and circuit complexities are fairly higher than those of proximal decoding. The complexity of proximal decoding is $O(\ell n^2)$, where ℓ represents the number of iterations, which is lower than the complexity of the MMSE detector if ℓ is constant.

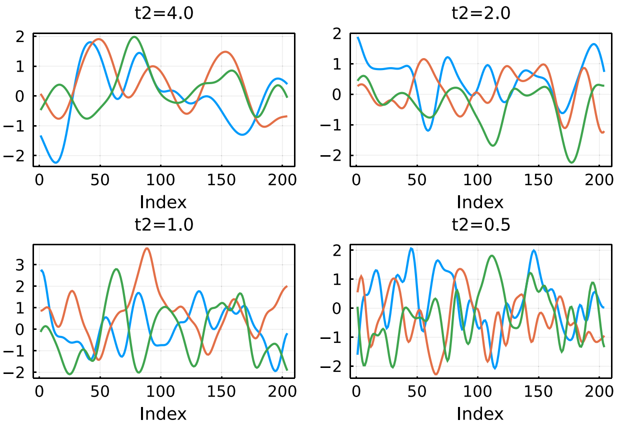

Fig. 4 Sample of correlated Gaussian noise for $t_2 = 0.5, 1.0, 2.0, 4.0$: $t_1 = 1.0, \Delta = 0.1$.

5.3 Proximal Decoding for Correlated Gaussian Noise Channels

In this subsection, we present experimental results on proximal decoding for correlated Gaussian noise channels. Although such a decoding problem has been discussed previously, it still seems to be a non-trivial problem. For example, [25] proposed a joint decoding method based on a BP decoder and a Kalman filter for handling correlated Gaussian noise; however, its decoding process is fairly heavy in terms of the time complexity owing to Kalman filtering.

The received word was modeled as $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{w}$, where $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. In the following experiment, we assumed that the noise is a Gaussian process defined by the Gaussian kernel

$$k(i,j) = t_1 \exp\left(-\frac{(\Delta i - \Delta j)^2}{t_2}\right).$$

In other words, the (i,j)-element of the covariance matrix $\Sigma_{i,j}$ is given by the kernel function as $\Sigma_{i,j} \equiv k(i,j)$. The kernel parameters t_2 and Δ are related to the correlation length of a Gaussian process. For fixed Δ, if the value of t_2 increases, the correlation of the neighboring noise components becomes stronger. Figure 4 shows several noise samples following the multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. From Fig. 4, we can observe a relationship between the correlation length and t_2.

We conducted an experiment for evaluating the bit error rate of proximal decoding for additive correlated Gaussian noise. The precision matrix $\boldsymbol{G}$ used in the proximal decoding (23) is given by $\boldsymbol{G} = \boldsymbol{\Sigma}^{-1}$. In the experiments, the parameters t_1 and Δ were fixed at $t_1 = 1.0$ and $\Delta = 0.1$, respectively. The parameters used in the proximal decoder were $\omega = 10^{-12}, \gamma = 0.01$. The maximum number of iterations was set to 100. None of the experiments involved early stopping by the parity check. The same regular LDPC code (3, 6) with $n = 204, m = 102$ was used in the experiment. From the pre-experiments, we observed that the decoding performance depends on the correlation length. Hence, we plotted the BER curve as a function of t_2.

Figure 5 shows the BER performance of proximal decoding for correlated Gaussian noise channels. For reference, the BER curve of BP decoding is also included. The input to the BP decoder is given by $\boldsymbol{l} \equiv 2\boldsymbol{y}/t_2$. We can see

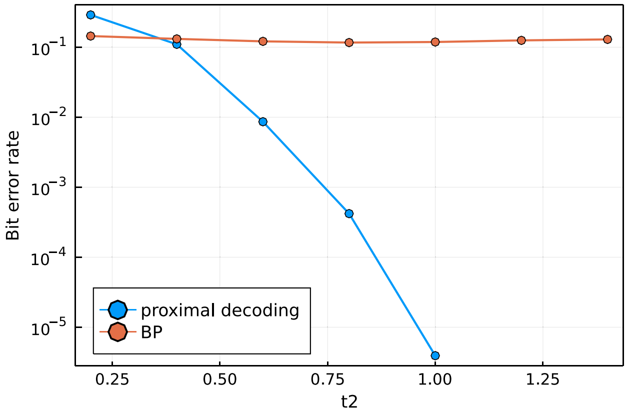

Fig. 5 Bit error rate of proximal decoding for additive correlated Gaussian noise channel: $t_1 = 1.0, \Delta = 0.1$. The regular LDPC code $(3, 6)$ with $n = 204$ and $m = 102$ were used.

that the proximal decoder achieves sufficiently small BER when t_2 becomes large. This indicates that the proximal decoder works appropriately for correlated Gaussian noise channels. The baseline curve of BP is nearly flat because a naive BP decoder cannot use the correlation between noise components.

5.4 Proximal Decoding for Nonlinear Vector Channels

In this subsection, we assume a nonlinear vector channel defined by $\boldsymbol{y} = \boldsymbol{W}_2 \operatorname{swish}(\boldsymbol{W}_1\boldsymbol{x}) + \boldsymbol{w}$, where swish defined by $\operatorname{swish}(x) \equiv x/(1 + \exp(-x))$ is component-wisely applied to the argument vector. The swish function [26] is known as a differentiable (at $x = 0$) counter part of ReLU function. In the following experiment, each element of $\boldsymbol{W}_1 \in \mathbb{R}^{500\times204}, \boldsymbol{W}_2 \in \mathbb{R}^{204\times500}$ was generated according to $\mathcal{N}(0, 0.1^2)$. The additive Gaussian noise vector $\boldsymbol{w}$ follows $\mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$. The channel model is somewhat artificial, but it is a simple example of a nonlinear function based on feed-forward neural network architecture. For the implementation of the decoder, we used the back-propagation mechanism provided by Flux.jl [27] on Julia Language [28].

Figure 6 depicts the error norms $\|\operatorname{sign}(\boldsymbol{s}^{(k)}) - \boldsymbol{x}\|$ as a function of iteration. The hyper parameters were set to $\gamma = 0.05$ and $\omega = 0.1$ according to the result of a pre-experiment. The left panel presents a case where the noise standard deviation σ is small, i.e., a high SNR case ($\sigma = 0.4$). From the left panel, we can see that the error norm curves (10 trials) decrease rapidly, and they eventually fall down to zero. On the other hand, when $\sigma = 0.9$ (a low SNR case), some of the error norm curves stay at high values (above 10). This result implies that the proximal decoding for this channel shows appropriate error correction behaviors when σ is sufficiently small. It can be remarked that the proximal decoding is applicable to fairly complex nonlinear functions, such as $f(\boldsymbol{x}) = \boldsymbol{W}_2 \operatorname{swish}(\boldsymbol{W}_1\boldsymbol{x})$. A neural network-based nonlinear vector function seems promising to model a nonlinear behavior of a channel. We may be able to expect that such a function can be handled in the framework of proximal decoding.

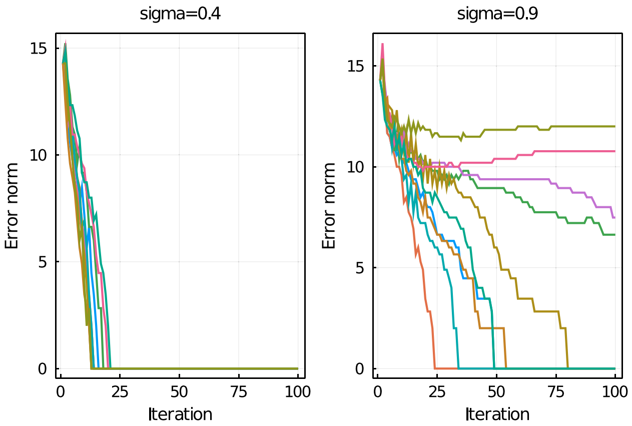

Fig. 6 Error norm of decoding processes proximal decoding for a nonlinear vector channel defined with swish function. The regular LDPC code $(3, 6)$ with $n = 204$ and $m = 102$ was used. Left: $\sigma = 0.4$; right: $\sigma = 0.9$. The SNR was set to 3 dB. The parameters $\gamma = 0.05, \omega = 0.1$.

6. Conclusions

We presented proximal decoding to approximate MAP decoding for LDPC codes. The key idea of the proposed algorithm is the use of the code-constraint polynomial, which acts as a regularizer corresponding to the codewords of an LDPC code. A codeword stationary point often attracts a point generated by the iterative process in a proximal gradient process. The pull-in property arises from the gradient of a code-constraint polynomial.

Further, the principle of proximal decoding was described, and it is fairly universal, i.e., it can be applied to many types of non-trivial channels if the gradient of the negative log-likelihood function can be evaluated efficiently. For presenting the derivation process of a recursive formula for proximal decoding, we discussed the derivation of *channel-specific* proximal decoding formulas for LDPC-coded massive MIMO channels, correlated Gaussian noise channels, and nonlinear vector channels. These channel-specific formulas were derived coherently according to the principle of proximal decoding.

In addition, we presented the results of numerical experiments for several variations of proximal decoding. For LDPC-coded massive MIMO channels, proximal decoding was shown to be comparable to known detection methods, such as MMSE + BP. In terms of a trade-off between decoding complexity and BER performance, proximal decoding appears to be promising for LDPC-coded MIMO channels. In the case of nonlinear vector channels, we proposed proximal decoding involving a back-propagation process for computing the gradient of a negative log-likelihood function. We also presented numerical results for a nonlinear vector channel defined on the basis of a nonlinear function with a feed-forward neural network architecture. We may be able to use a feed-forward neural network to capture nonlinear distortion of the channel, and such a neural network model can be directly used for decoding.

Acknowledgments

We appreciate constructive comments given by the reviewers for improving the quality of this paper. A part of the results of this study was presented at the IEEE International Symposium on Information Theory (ISIT) [29]. The study was supported by a JSPS Grant-in-Aid for Scientific Research (B)19H02138 and (A) 22H00514.

References

[1] R.G. Gallager, Low Density Parity Check Codes, MIT Press, 1963.
[2] J. Feldman, "Decoding error-correcting codes via linear programming," Massachusetts Institute of Technology, Ph.D. thesis, 2003.
[3] J. Feldman, M.T. Wainwright, and D.R. Karger, "Using linear programming to decoding binary linear codes," IEEE Trans. Inf. Theory, vol.51, no.1, pp.954–972, 2005.
[4] T. Wadayama, K. Nakamura, M. Yagita, Y. Funahashi, S. Usami, and I. Takumi, "Gradient descent bit flipping algorithms for decoding LDPC codes," IEEE Trans. Commun., vol.58, no.6, pp.1610–1614, 2010.
[5] G. Sundararajan, C. Winstead, and E. Boutillon, "Noisy gradient descent bit-flip decoding for LDPC codes," IEEE Trans. Commun., vol.62, no.10, pp.3385–3400, 2014.
[6] P.O. Vontobel, "Interior-point algorithms for linear-programming decoding," IEEE Information Theory and Applications Workshop, 2008.
[7] T. Wadayama, "Interior point decoding for linear vector channels based on convex optimization," IEEE Trans. Inf. Theory, vol.56, no.10, pp.4905–4921, 2010.
[8] X. Zhang and P.H. Siegel, "Efficient iterative LP decoding of LDPC codes with alternating direction method of multipliers," IEEE International Symposium on Information Theory (ISIT), 2013.
[9] X. Liu and S.C. Draper, "The ADMM penalized decoder for LDPC codes," IEEE Trans. Inf. Theory, vol.62, no.6, pp.2966–2984, 2016.
[10] X. Liu and S.C. Draper, "ADMM LP decoding of non-binary LDPC codes in $\mathbb{F}_{2^m}$," IEEE Trans. Inf. Theory, vol.62, no.6, pp.2985–3010, 2016.
[11] Y. Wang and J. Bai, "Proximal-ADMM decoder for nonbinary LDPC codes," arXiv preprint arXiv:2010.09534, 2020.
[12] N. Parikh and S. Boyd, Proximal algorithms (Foundations and Trends in Optimization), Now Publisher, 2014.
[13] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[14] I. Daubechies, M. Defrise, and C.D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," Comm. Pure Appl. Math., vol.57, no.11, pp.1413–1457, 2004.
[15] R. Tibshirani, "Regression shrinkage and selection via the lasso," J. Royal Stat. Society, Series B, vol.58, pp.267–288, 1996.
[16] E.G. Larsson, O. Edfors, F. Tufvesson, and T.L. Marzetta, "Massive MIMO for next generation wireless systems," IEEE Commun. Mag., vol.52, no.2, pp.186–195, 2014.
[17] S. Takabe, M. Imanishi, T. Wadayama, R. Hayakawa, and K. Hayashi, "Trainable projected gradient detector for massive overloaded MIMO channels: Data-driven tuning approach," IEEE Access, vol.7, pp.93326–93338, 2019.
[18] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2017.
[19] D.J.C. MacKay, "Encyclopedia of sparse graph codes [online]," Available: http://www.inference.phy.cam.ac.uk/mackay/codes/data.html
[20] D.S. Shiu, G.J. Foschini, M.J. Gans, and J.M. Kahn, "Fading correlation and its effect on the capacity of multielement antenna systems," IEEE Trans. Commun., vol.48, no.3, pp.502–513, 2000.
[21] D. Divsalar, M.K. Simon, and D. Raphaeli, "Improved parallel interference cancellation for CDMA," IEEE Trans. Commun., vol.46, no.2, pp.258–268, 1998.
[22] J. Ma, L. Liu, X. Yuan, and L. Ping, "On orthogonal amp in coded linear vector systems," IEEE Trans. Wireless Commun., vol.18, no.12, pp.5658–5672, 2019.
[23] A. Sanderovich, M. Peleg, and S.S Shamai, "LDPC coded MIMO multiple access with iterative joint decoding," IEEE Trans. Inf. Theory, vol.51, no.4, pp.1437–1450, 2005.
[24] I. Hwang, H.J. Park, and J.W. Lee, "LDPC coded massive MIMO systems," Entropy, vol.21, no.3, 231, 2019.
[25] T. Wadayama, "An iterative decoding algorithm for channels with additive linear dynamical noise," IEICE Trans. Fundamentals, vol.E86-A, no.10, pp.2452–2460, Oct. 2003.
[26] P. Ramachandran, B.Z. Barret, and Q.V. Le, "Swish: A self-gated activation function," arXiv preprint arXiv:1710.05941, 2017.
[27] M. Innes, "Flux: Elegant machine learning with Julia," Journal of Open Source Software, vol.3, no.25, 602, 2018.
[28] J. Bezanson, S. Karpinski, B. Viral, and A. Edelman, "Julia: A fast dynamic language for technical computing," arXiv preprint arXiv:1209.5145, 2012.
[29] T. Wadayama and S. Takabe, "Proximal decoding for LDPC-coded massive MIMO channels," IEEE International Symposium on Information Theory, 2021.

Tadashi Wadayama was born in Kyoto, Japan, on May 9, 1968. He received the B.E., the M.E., and the D.E. degrees from Kyoto Institute of Technology in 1991, 1993, and 1997, respectively. On 1995, he started to work with Faculty of Computer Science and System Engineering, Okayama Prefectural University as a research associate. From April 1999 to March 2000, he stayed in Institute of Experimental Mathematics, Essen University (Germany) as a visiting researcher. On 2004, he moved to Nagoya Institute of Technology as an associate professor. Since 2010, he has been a full professor of Nagoya Institute of Technology. His research interests are in coding theory, information theory, and coding and signal processing for digital communication/storage systems. He is a member of IEEE.

Satoshi Takabe received the B.Sc., M.Sc., and Ph.D. degrees in multidisciplinary sciences from the University of Tokyo, Japan, in 2012, 2014, and 2017, respectively. In 2017, he started to work with the Department of Computer Science, Nagoya Institute of Technology as an assistant professor. Since 2021, he has been an associate professor at the Department of Mathematical and Computing Science, School of Computing, Tokyo Institute of Technology. He received IEEE Information Theory Society Japan Chapter Young Researcher Best Paper Award in 2018. His research interests include signal processing, information theory, and machine learning. He is a member of IEEE, ORSJ, and JPS.