

Article

Hybrid Window Decoding for Joint Source Channel Anytime Coding System

Li Deng ^{1,2} , Zhiping Shi ^{1,2,*}, Yixin Wang ³, Xiaoxi Yu ⁴, Yong Liang Guan ⁴  and Zhiping Xu ⁵ 

¹ National Key Laboratory on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China; dengli@uestc.edu.cn

² Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Huzhou 313000, China

³ The Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore 138632, Singapore; wang_yixin@i2r.a-star.edu.sg

⁴ The School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore; xiaoxi001@e.ntu.edu.sg (X.Y.); eyllguan@ntu.edu.sg (Y.L.G.)

⁵ The School of Ocean Information Engineering, Jimei University, Xiamen 361021, China; zhipingxu@jmu.edu.cn

* Correspondence: szp@uestc.edu.cn

Abstract: Joint source channel anytime coding (JSCAC) is a kind of joint source channel coding (JSCC) systems based on the causal spatially coupled coding and joint expanding window decoding (JEWD) techniques. JSCAC demonstrates greatly improved error correction performance, as well as higher decoding complexity. This work proposes a joint hybrid window decoding (JHWD) algorithm for JSCAC systems, aiming to reduce the decoding complexity while maintaining comparable error correction performance with the state of the art. Unlike the traditional JEWD technique and its variants, the proposed JHWD algorithm utilizes a hybrid window structure. It achieves this by implementing adaptive local expanding window decoding in the sliding window region, guided by syndrome-check-based detection. The hybrid window decoding characteristics of the proposed JHWD algorithm can both effectively reduce the decoding complexity caused by window expanding, and also mitigate the error propagation caused by window sliding. An improved density evolution algorithm is proposed for the asymptotic performance analysis of the proposed JHWD algorithm. Both the analytical and simulation results indicate that the proposed JHWD decoding scheme is a good low-complexity implementation option for JSCAC systems, which is also suitable for other JSCC systems with the spatially coupled coding structure.

Keywords: joint source channel coding (JSCC); anytime coding; adaptive local expanding window decoding



Citation: Deng, L.; Shi, Z.; Wang, Y.; Yu, X.; Guan, Y.L.; Xu, Z. Hybrid Window Decoding for Joint Source Channel Anytime Coding System. *Entropy* **2024**, *26*, 940. <https://doi.org/10.3390/e26110940>

Academic Editors: Eduard Jorswieck and Lu Wei

Received: 3 September 2024

Revised: 16 October 2024

Accepted: 30 October 2024

Published: 1 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The separate source-channel coding technique is based on the Shannon separation coding theorem, which assumes that the source entropy is less than the channel capacity under the conditions of unrestricted delay, complexity and code length. That is, the separation coding is asymptotically optimal. Joint source channel coding (JSCC) is a kind of coding technology that designs source/channel encoding and decoding in an interdependent way, so as to achieve the overall optimal communication system [1]. JSCC technology can be used in any system that violates the condition of the separation coding theorem and needs to consider both the source side and the channel side, such as communication systems with limitations of power, bandwidth, complexity or delay, multi-user system with shared channel, and different types of communication environments composed of heterogeneous sources or channels, etc. In recent years, the increasingly mature deep learning technology has brought new design ideas for JSCC technology, and set off a new wave of research [2–4].

The benefits of JSCC are reflected in the following aspects: (a) Improve the transmission efficiency: with comprehensive consideration of the characteristics of the source and channel, JSCC can optimize the coding scheme, reduce the size of the transmitted data, so as to improve the transmission efficiency. This means that more information can be transmitted with the same bandwidth, or less transmission time with the same amount of information. (b) Reduce the error rate: JSCC can adjust the coding mode according to the specific situation of the channel to reduce the bit error rate during transmission. This means that data are less error-prone during transmission, improving the reliability and accuracy of data. (c) Improve the system robustness: JSCC is designed with channel variation and interference in mind, making the system more stable and reliable in the face of different channel conditions. This allows the system to adapt to a wider range of environments, reducing transmission failures due to channel changes.

In terms of JSCC technologies that are not based on deep learning, there are several typical schemes. At the sending side, the channel encoder can provide the channel state information (CSI) to the source encoder for the bit allocation [5,6] and the source code optimization [7]. On the other hand, the source encoder can also provide the source significance information (SSI) for the unequal error protection (UEP) during the channel coding or modulation [8,9]. At the receiving side, the source *a priori* information and *a posteriori* information can be used for iterative decoding between the source and channel decoders, which is one of the most popular JSCC solutions and also the focus of this paper. Fresia first proposed a JSCC system that completes the source compression and error correction using two block LDPC codes (D-LDPC), then implements the joint belief propagation (BP) decoding within a global factor graph model framework, including both source and channel coding structures [10]. After that, various codes were introduced into the JSCC system, such as the block double protograph LDPC (DP-LDPC) code [11,12], the block double Polar (D-Polar) code [13], the spatially coupled LDPC (SC-LDPC) code [14–16], and the spatially coupled repeat-accumulate (SC-RA) code [17,18], etc.

Among the above solutions, we focus on the scheme termed SC-RA based joint source channel anytime coding (JSCAC). It is a specific JSCC scheme combined with anytime coding techniques, i.e., causal spatially coupled coding and expanding window decoding [19,20]. The anytime properties of the JSCAC scheme account for both delay and reliability, making it particularly suitable for transmission scenarios demanding both low delay and high reliability. However, the prior-art expanding window decoding strategies have noticeable decoding complexities and memory space requirements [17,18], which affect the practical applications of JSCAC systems.

In order to reduce the decoding complexity while maintaining comparable error correction performance, a hybrid window decoding scheme is proposed. The novelty and contributions of this paper can be summarized as follows.

- A joint hybrid window decoding (JHWD) algorithm is proposed, which adaptively implements the local expanding window decoding in the sliding window region with the constraint of syndrome-check based detection.
- The density evolution algorithm of the JSCAC system in [18] is improved for the proposed JHWD scheme to analyze its asymptotic decoding complexity and anytime properties.
- Some numerical analyses and simulation results are provided to confirm the advantages of the proposed JHWD algorithm over other prior-art decoding schemes for JSCAC systems.

The remainder of this paper is organized as follows. The background of the SC-RA-based JSCAC system is briefly introduced in Section 2. The specific descriptions of the proposed JHWD scheme and improved density evolution algorithm are provided in Section 3. The numerical analyses and simulation results are presented in Section 4, followed by conclusions in Section 5.

2. Preliminaries

In this section, we briefly introduce the coding structure, the prior-art decoding strategies, and the density evolution analysis method of SC-RA-based JSCAC system.

2.1. Source Model

Let $\mathbf{s} = (s_1, \dots, s_n) \in \{0, 1\}^n$ be a binary sequence of length n generated by an independent identically distributed (i.i.d) nonuniform memory-less source with probability distribution $(p, 1 - p)$, where p is the probability of the binary bit “1” in a source sequence (named as source probability), and $p = P_r(s_i = 1) \neq 0.5$. The initial log-likelihood-ratio (LLR) of each source bit is denoted by $z^{sc} = \log(\frac{1-p}{p})$; and the initial LLRs of the received code word $\mathbf{y}$ after the binary-phase-shift-keying modulation (BPSK) and over the additive white Gaussian noise (AWGN) channel can be represented as $\mathbf{z}^{cc} = \frac{2\mathbf{y}}{\sigma^2}$, where σ^2 is the variance of channel noise.

2.2. Sketch of the SC-RA-Based JSCAC System

Figure 1 shows the encoding and decoding structures of the SC-RA-based JSCAC system [17], which is composed of the upper half part (source code) and the lower half part (channel code). Therein, the white circles represent the parity bit nodes (PNs) with a fixed degree of 2; the black and gray rectangles indicate the source syndrome nodes (SNs) and channel check nodes (CNs) with degrees of A^{sc} and A^{cc} , respectively; while the black and gray circles denote the source and channel information bit nodes (INs) with degrees of Q^{sc} and Q^{cc} , respectively. Each SN or CN is causally and exponentially connected to the former INs with an exponential rate of λ and a coupling length of γ . The exponentially distributed edge connections of source code and channel code are depicted by line thickness in Figure 1, which can also be written as [21]:

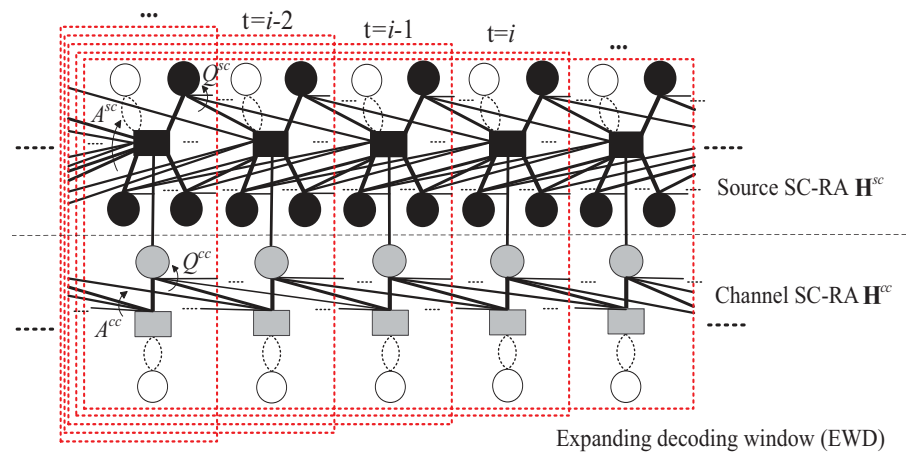


Figure 1. Encoding and decoding structures of the SC-RA-based JSCAC system [17]. Reproduced with permission of [17], Copyright ©2020, IEEE.

$$P_r(k) = \frac{e^{-k\lambda}(1 - e^{-\lambda})}{1 - e^{-\gamma\lambda}}, \quad (1)$$

where $k = j - i$ is the distance between the j -th SN (CN) with the connected i -th IN ($j \geq i$). The exponential rate of $\lambda = \lambda^{sc}$ (or λ^{cc}) and the coupling length of $\gamma = \gamma^{sc}$ (or γ^{cc}). The larger the parameter λ , the faster the error probability decays with increasing decoding delay. When $\lambda^{sc} = \lambda^{cc} = 0$, then $P_r(k) = 1$, which is equal to the uniform distribution. The uniform distribution has a slow decline rate over the decoding delay, which is undesired in

the anytime transmission. The source code rate and channel code rate can be calculated with the following equation [22]:

$$R = \frac{L}{L + \frac{Q}{A}(L - \gamma + 1 + 2(\gamma - \sum_{i=0}^{\gamma} (\frac{i}{\gamma})^A))}, \quad (2)$$

where L is the chain length. For the source code rate R^{sc} , $Q = Q^{sc}$, $A = A^{sc}$, $\gamma = \gamma^{sc}$; and for the channel code rate R^{cc} , $Q = Q^{cc}$, $A = A^{cc}$, $\gamma = \gamma^{cc}$. As $L \rightarrow \infty$, the asymptotic code rates of the SC-RA-based JSCAC system are $R^{cc} \rightarrow \frac{A^{cc}}{A^{cc} + Q^{cc}}$, $R^{sc} \rightarrow \frac{Q^{sc}}{A^{sc} + Q^{sc}}$.

2.3. Encoding Procedure of SC-RA-Based JSCAC System

The source parity-check matrix H^{sc} and the channel parity-check matrix H^{cc} have the same structure as (3)

$$\mathbf{H}_{[0,L-1]} = \begin{bmatrix} \mathbf{H}_{00} & & & & \\ \vdots & \ddots & & & \\ \mathbf{H}_{0(\gamma-1)} & \ddots & & & \\ & \ddots & \mathbf{H}_{(L-1)0} & & \\ & & \vdots & & \\ & & & \mathbf{H}_{(L-1)(\gamma-1)} \end{bmatrix}, \quad (3)$$

where L is the chain length, the involved sub-matrices for the encoding procedure of the i -th message block are from $\mathbf{H}_{i0}$ to $\mathbf{H}_{(i-\gamma+1)(\gamma-1)}$, $\gamma = \gamma^{sc}$ or (γ^{cc}) . The proposed JSCAC follows the concatenated source and channel encoding of block LDPC based JSCC systems.

(1) Syndrome source encoding: Assume the causal source streaming $\mathbf{s}_{[0,L-1]} = \{\mathbf{s}_i | 0 \leq i \leq L-1\}$ produces a N^{sc} -bit message block $\mathbf{s}_i$ at a time instant t and with a chain length of L . The message block $\mathbf{s}_i$ is compressed with the syndrome source coding as $\mathbf{u}_i = \mathbf{s}_i \mathbf{H}_{i0}^{scT} + \mathbf{s}_{(i-1)} \mathbf{H}_{(i-1)1}^{scT} + \dots + \mathbf{s}_{(i-\gamma+1)} \mathbf{H}_{(i-\gamma+1)(\gamma-1)}^{scT}$ with the block-length of M^{sc} . If $i < \gamma - 1$, then $\mathbf{u}_i = \mathbf{s}_i \mathbf{H}_{i0}^{scT} + \mathbf{s}_{(i-1)} \mathbf{H}_{(i-1)1}^{scT} + \dots + \mathbf{s}_0 \mathbf{H}_{0(i-1)}^{scT}$.

(2) Channel encoding: The compressed source sequence $\mathbf{u}_{[0,L-1]} = \{\mathbf{u}_i | 0 \leq i \leq L-1\}$ is encoded with the channel encoder to generate the channel codeword $\mathbf{x}_{[0,L-1]} = \{\mathbf{x}_i | 0 \leq i \leq L-1\}$. For each block $\mathbf{x}_i = [\mathbf{u}_i, \mathbf{c}_i]$ with the block-length of N^{cc} , the channel PNs $\mathbf{c}_i$ have a fixed degree of 2, checking both the coupled INs in the early transmitted blocks and local INs, which are denoted as $\mathbf{v}_i = [\mathbf{x}_{i-\gamma+1}, \dots, \mathbf{x}_{i-1}, \mathbf{u}_i]$ ($\mathbf{v}_i = [\mathbf{x}_0, \dots, \mathbf{x}_{i-1}, \mathbf{u}_i]$, if $i < \gamma - 1$). The generation of channel PNs $\mathbf{c}_i = \{c_{ij} | 1 \leq j \leq N^{cc} - M^{sc}\}$ is described as

$$c_{ij} = \begin{cases} \mathbf{v}_i \tilde{\mathbf{H}}^{ccT}(1, :), & j = 1 \\ \mathbf{v}_i \tilde{\mathbf{H}}^{ccT}(j, :) + c_{i(j-1)}, & \text{otherwise} \end{cases} \quad (4)$$

where $\tilde{\mathbf{H}}^{cc} = [\mathbf{H}_{(i-\gamma+1)(\gamma-1)}^{cc}, \dots, \mathbf{H}_{(i-1)1}^{cc}, \mathbf{H}_{i0}^{cc}(:, 1 : M^{sc})]$.

2.4. Decoding Algorithms of SC-RA-Based JSCAC System

The red dashed rectangles in Figure 1 indicate the decoding windows expanding in size as decoding delay increases. The joint BP decoding for D-LDPC based JSCC [10] is implemented in each decoding window, which performs the source BP decoder and channel BP decoder in parallel, exchanging the mutual information through the public edges (red dashed lines in Figure 1). See [10] for more details of joint BP. This classical decoding method for JSCAC systems is the joint expanding window decoding (JEWD) [17], which directly implements the joint BP decoding in the expanding window with high complexities. Authors in [17] proposed a partial joint expanding window decoding (PJEWD) based on

the observation of convergence inconsistency between the source and channel decoders, i.e., stop message updating for all channel nodes once the channel decoding convergence is detected. Compared with the JEWD algorithm, the PJEWD algorithm can improve the error performance. However, the reduction ratio of decoding complexity is not significant, as it is affected by the channel conditions.

2.5. Density Evolution (DE) Algorithms for SC-RA-Based JSCAC System

In [18], the authors proposed a density evolution (DE) algorithm for an SC-RA-based JSCAC system with PJEWD scheme. Based on the DE analysis, the anytime properties of the SC-RA-based JSCAC system are provided, i.e., the error probability of any message block can exponentially decay to zero as the decoding delay approaches to infinity. Moreover, the lower bound of the delay exponent α is also provided, which is proportional to the product of λ^{sc} and Q^{sc} . The DE algorithm can also analyze the BP decoding threshold and the decoding complexity of the SC-RA-based JSCAC with PJEWD scheme. See more details of the encoding process, the decoding algorithms, and the DE algorithm of the SC-RA-based JSCAC system in [17,18].

3. Hybrid Window Decoding for JSCAC System

3.1. Hybrid Window Decoding Structure

In this section, we first describe the proposed JHWD algorithm for JSCAC systems in detail, then we provide the improved DE analysis method for the proposed JHWD algorithm.

The proposed hybrid window decoding structure is shown in Figure 2, which is composed of two main parts, i.e., the initial expanding window decoding phase and the sliding window decoding phase. Each part is numerically marked. In order to simplify the description, the source and channel couple lengths are set the same as $\gamma^{sc} = \gamma^{cc} = \gamma$. In the example shown in Figure 2, γ is set as 5. We explain the details of the proposed JHWD scheme as follows.

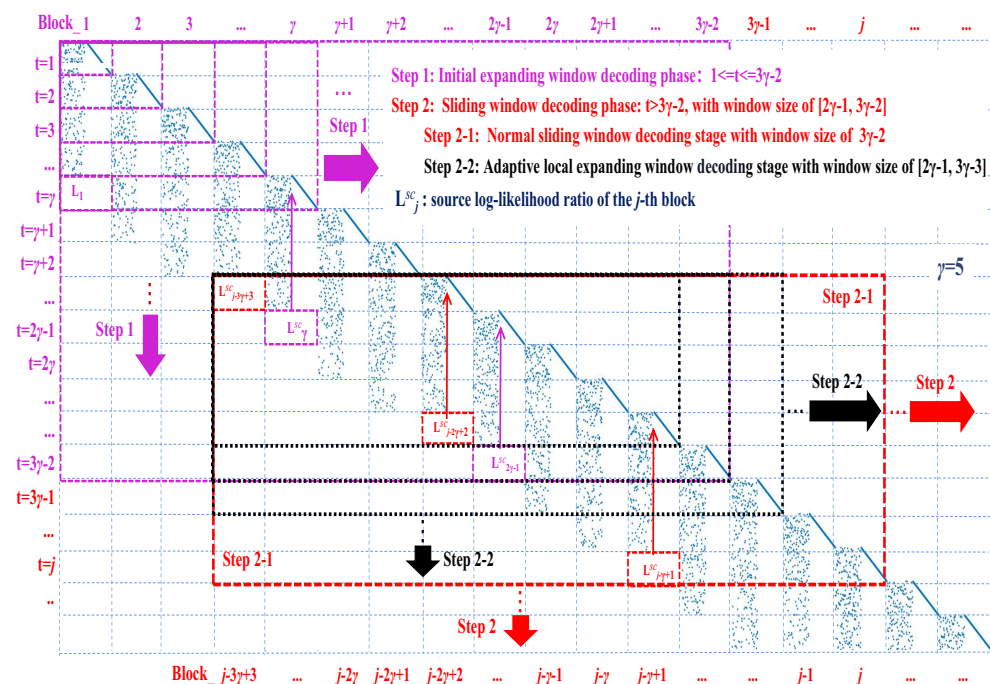


Figure 2. Hybrid window decoding structure of the proposed JHWD algorithm.

Application scenarios: The proposed JHWD scheme is designed for the source streaming with short block length, and with a chain length L far longer than the coupling length γ , i.e., $L \gg \gamma$. Accordingly, as shown in Figure 2, there is not any special decoding process

for blocks at the end of the data chain. Moreover, this work focuses on the decoding performances for the source streaming with short block lengths in low-latency and high-reliability communications.

Selection of window size: In the JSCAC system, the coupling length γ is supposed to be finite, and the encoding structure of any block is directly related to its $\gamma - 1$ coupled blocks. However, all blocks in the data chain are causally coupled and connected one by one. Therefore, the decoding performance of any block is still related to the backward-connected blocks beyond the coupling length. However, the mutual influence will gradually weaken as distance increases. In order to trade off the error correction performance and the decoding complexity, we adopt the sliding decoding window size as $3\gamma - 2$ block lengths, covering the coupling range where the object block is located and the two nearest adjacent coupling ranges with the greatest influences. These are called the first, the second and the third coupling interval, respectively.

Emendatory decoding strategies: Combined with the encoding structure of JSCAC system, some emendatory decoding strategies are adopted in the proposed JHWD scheme.

(1) Update the source *a posteriori* log-likelihood ratios (LLRs) at each time slot t after decoding as follows

$$\mathbf{L}_t^{sc} = (\mathbf{L}_{t-1}^{sc} + \mathbf{L}_t^{sc})/2. \quad (5)$$

The reason for updating the source LLRs instead of the channel LLRs is that the source sequence is not actually transmitted through the channel; rather, it is represented by its compressed syndrome. It is found that decoding the object block with the previously well-decoded source information can effectively improve the error correction performance.

(2) In order to alleviate the error propagation caused by sliding window decoding [23], a kind of syndrome-check based detection, shown in (6), is carried out before the local expanding window decoding, where $\hat{\mathbf{S}}_k^{sc}$ and $\hat{\mathbf{U}}_k^{cc}$ are the source and channel decoding results of k -th joint BP, and $\hat{\mathbf{H}}_k^{sc}$ is the involved source parity-check matrix. If the detection result is false, a local expanding window decoding is adaptively carried out within the decoding window until the detection result is true. This syndrome-check-based detection is implemented once, prior to the matrix multiplication in the joint BP decoding stage of the local expanding window decoding. However, it controls the adaptive execution of the local expanding window decoding, and thus can greatly reduce the decoding iterations and message updates.

$$S_{check0} = \begin{cases} 1 & \text{if } \hat{\mathbf{S}}_k^{sc} \cdot \hat{\mathbf{H}}_k^{sc} = \hat{\mathbf{U}}_k^{cc}, \\ 0 & \text{else.} \end{cases} \quad (6)$$

Decoding structure descriptions: The details of the JHWD algorithm are provided in Algorithm 1. The parameters $\mathbf{H}^{sc}$ and $\mathbf{H}^{cc}$ indicate the source and channel parity-check matrix. $I_{\max 1}$ and $I_{\max 2}$ denote the maximum iterations of different decoding stages; and $I_{\max 2}$ is far smaller than $I_{\max 1}$. p is the source probability, σ_{ch} is the variance of channel noise, and T is the maximum observed decoding delay.

There are two main steps in Algorithm 1. First, in the initial expanding window decoding phase of Step 1, the traditional JEWD in [17] is implemented within the time slots range of $t \in [1, 3\gamma - 2]$. Second, when $t > 3\gamma - 2$, the window is slid forward one block and the sliding window decoding is implemented at each time slot in Step 2.

Especially, in the sliding window decoding phase of Step 2, it also includes two sub-steps. (a) Step 2-1: In the normal sliding window decoding stage, the initial source LLR, i.e., $z^{sc} = \log \frac{1-p}{p}$, is used for decoding with $I_{\max 1}$, and the decoding range is the whole window length, i.e., $3\gamma - 2$. (b) Step 2-2: In the adaptive local expanding window decoding stage, the source updated *a posteriori* LLRs $\mathbf{L}_t^{sc}$ is used for decoding with $I_{\max 2}$. The decoding range expands with the window size region of $[2\gamma - 1, 3\gamma - 3]$, including the first $2\gamma + k - 2$ blocks ($1 \leq k < \gamma$) of the sliding window. The syndrome-check-based detection is carried out before the local expanding window decoding. As shown in line 16 to line 24 of Algorithm 1, if $S_{check0} = 0$, the local expanding window decoding is carried

out within the sliding window region. If $S_{check0} = 1$, the adaptive local expanding window decoding is terminated, and the *a posteriori* source LLRs are updated for the involved blocks with (5). After each local expanding window decoding, we update L_t^{sc} only for blocks in the first coupling interval. The motivation is that in any decoding window range, the LLRs of message blocks in the first coupling interval can converge earlier than the blocks in the other two coupling intervals.

Algorithm 1: Joint hybrid window decoding (JHWD) algorithm

Input: $H^{sc}, H^{cc}, I_{\max 1}, I_{\max 2}, T, z^{sc}, \sigma_{ch}$;
Output: the hard decision result $\hat{S}_{[1,T]}$;

- 1 **Step 1: Initial expanding window decoding phase**
- 2 **for** $t = j, 1 \leq j \leq 3\gamma - 2$ **do**
- 3 $\hat{H}_t^{sc} = H^{sc}[1 : j, 1 : j], \hat{H}_t^{cc} = H^{cc}[1 : j, 1 : j];$
- 4 implement joint BP with $\hat{H}_t^{sc}, \hat{H}_t^{cc}, I_{\max 1}, Z^{sc};$
- 5 Update L_t^{sc} of Block[1:j] with (5);
- 6 **Step 2: Sliding window decoding phase:**
- 7 **for** $t = j, 3\gamma - 2 < j \leq T$ **do**
- 8 **Step 2-1: Normal sliding window decoding stage**
- 9 $\hat{H}_t^{sc} = H^{sc}[j - 2\gamma + 2 : j, j - 3\gamma + 3 : j],$
- 10 $\hat{H}_t^{cc} = H^{cc}[j - 2\gamma + 2 : j, j - 3\gamma + 3 : j];$
- 11 Implement joint BP with $\hat{H}_t^{sc}, \hat{H}_t^{cc}, I_{\max 1}, Z^{sc};$
- 12 Update L_t^{sc} of Block $[j - 3\gamma + 3 : j]$ with (5);
- 13 **Step 2-2: Adaptive local expanding window decoding stage**
- 14 Initiate $S_{check0} = 0;$
- 15 **for** $1 \leq k < \gamma$ **do**
- 16 **if** $S_{check0} == 0$ **then**
- 17 $\hat{H}_k^{sc} = H^{sc}[j - 2\gamma + 2 : j - \gamma + k, j - 3\gamma + 3 : j - \gamma + k];$
- 18 $\hat{H}_k^{cc} = H^{cc}[j - 2\gamma + 2 : j - \gamma + k, j - 3\gamma + 3 : j - \gamma + k];$
- 19 $Z_k^{sc} = L_t^{sc}[j - 3\gamma + 3 : j - \gamma + k];$
- 20 Do joint BP with $\hat{H}_k^{sc}, \hat{H}_k^{cc}, I_{\max 2}$ and $Z_k^{sc};$
- 21 Syndrome-check based detection $\rightarrow S_{check0};$
- 22 **if** $S_{check0} == 1$ **then**
- 23 Update L_t^{sc} with (5) for Block $[j - 3\gamma + 3 : j - \gamma + k];$
- 24 break ;
- 25 **Output:** $\hat{S}_{[1,T]} = -(\text{sign}(L_T^{sc}) + 1)/2;$
- 26 **end**

3.2. Density Evolution Analysis for JHWD Algorithm

In order to analyze the anytime properties and decoding complexity for the JHWD algorithm, we revised the density evolution algorithm for JSCAC with PJEWD scheme in [18]. In the following density evolution analyses, the specific formulas for message updates are omitted, which are the same as (6) to (15) in [18], and are referred to as “*Message updates*” for simplicity of description.

Step 1: Initialize the parameters of $x_{cv}^0(i_c, t) = x_{cp}^0(j_c, t) = \mu_0 = \frac{2}{\sigma_n^2}, 0 < i_c, j_c \leq t$, and $x_{sv}^0(i_s, t) = x_{sp}^0(j_s, t) = \nu_0 = \log \frac{1-p}{p}, 0 < i_s, j_s \leq t$, where $x_{cv}^l(i_c, t)$ and $x_{cp}^l(i_c, t)$ are messages exchanged from channel IN_{i_c} and channel PN_{j_c} to connected channel CNs at the l -th iteration and time t ; $x_{sv}^l(i_s, t)$ ($x_{sp}^l(j_s, t)$) are messages exchanged from source IN_{i_s} (source PN_{j_s}) to connected source SNs at the l -th iteration and time t . Initialize the number of message updates as $Num_{me} = 0$, and the number of iterations as $l = 0$.

Step 2: Initial expanding window decoding ($1 < t \leq 3\gamma - 2$)

(1) Implement “*Message updates*” at the time slot t within the expanding window of $[1 : t]$ with $I_{\max 1}$.

(2) Update the following messages as follows:

$$\begin{aligned}x_{cv}^t(i_c, t) &= (x_{cv}^{t-1}(i_c, t) + x_{cv}^t(i_c, t))/2, \\x_{cp}^t(j_c, t) &= (x_{cp}^{t-1}(j_c, t) + x_{cp}^t(j_c, t))/2, \\x_{sv}^t(i_s, t) &= (x_{sv}^{t-1}(i_s, t) + x_{sv}^t(i_s, t))/2, \\x_{sp}^t(j_s, t) &= (x_{sp}^{t-1}(j_s, t) + x_{sp}^t(j_s, t))/2.\end{aligned}\quad (7)$$

(3) Update the number of message updates Num_{me} .

Step 3: Sliding window decoding ($3\gamma - 2 < t \leq T$)

(1) Normal sliding window decoding stage: implement “*Message updates*” at the time slot t within the sliding window of $[t - 3\gamma + 3 : t]$ with $I_{\max 1}$; then update messages with (7), and update Num_{me} .

(2) Adaptive local expanding window decoding stage: The syndrome-check-based detection is adopted by checking $x_{sv}^t(i_s, t)$, as below:

$$S_{check1} = \begin{cases} 1 & \text{if } |x_{sv}^t(i_s, t) - x_{sv}^{t-1}(i_s, t)| < \epsilon, \\ 0 & \text{else.} \end{cases}\quad (8)$$

where ϵ is a threshold with a very small value. If $S_{check1} = 0$, implement “*Message updates*” at the time slot t within the local expanding window of $[t - 3\gamma + 3 : t - \gamma + k]$ ($k = 2 : \gamma$) with $I_{\max 2}$. If $S_{check1} = 1$, which means that $x_{sv}^t(i_s, t)$ could not update anymore compared with $x_{sv}^{t-1}(i_s, t)$, update messages with (7) and Num_{me} .

Step 4: Calculate the error probability with (17) in [18].

4. Simulation and Analysis

In this section, we explore the decoding complexity, anytime properties and error correction performance of the proposed JHWD algorithm with both numerical analyses and simulations. The benchmark algorithms are the joint expanding window decoding (JEWD) algorithm and the partial joint expanding window decoding (PJEWD) algorithm for JSCAC systems in [17,18]. We also compared the proposed algorithm with the traditional joint source channel decoding (JSCD) for block D-LDPC-based JSCC systems [10] and the SC-LDPC-based JSCC scheme with sliding variable-window decoding (SVWD) [16]. The additive white Gaussian noise (AWGN) channel and the binary phase-shift keying (BPSK) modulation are adopted. The simulation parameters are set as follows: $Q^{sc} = 4$, $A^{sc} = 12$, $Q^{cc} = 3$, $A^{cc} = 3$, $R^{sc} = 0.25$, $R^{cc} = 0.5$, $\lambda^{sc} = 0.5$, $\lambda^{cc} = 0.1$, $I_{\max 1} = 30$, $I_{\max 2} = 5$, $p = 0.02$, and $E_b/N_0 = 2$ dB for anytime properties analyses.

We first analyze the asymptotic average message updates of the SC-RA-based JSCAC system as the decoding delay using different decoding schemes. As shown in Figure 3, as the decoding delay increases, the average message updates of all the three decoding schemes show slow and nearly linear growth trends, and their differences become more significant. The right Y-axis of Figure 3 shows the average message updates reduction ratio of PJEWD and the proposed JHWD compared to the traditional JEWD algorithm. As t is greater than 600, the growth rates tend to flatten out. In the decoding delay region of $[100, 1000]$, the message update reduction ratio of PJEWD to JEWD increases from 26% to 48%; while the reduction rate of the proposed JHWD to JEWD increases from 58% to 94%. In other words, the message updates reduction ratio of the proposed JHWD is about two times higher than the PJEWD algorithm.

In Figure 4, the asymptotic anytime properties of SC-RA-based JSCAC with different decoding schemes are analyzed. The anytime properties of a single block in the first sliding window region is observed from $t = 14$. It can be seen that with the same coupling length of $\gamma = 5$, the bit error probability curves of the JEWD and PJEWD algorithm have similar descent rates, while the curve slope of the proposed JHWD algorithm is slightly slower.

Figure 4 also shows the error probabilities of the proposed JHWD with different γ . It can be seen that the smaller the coupling length, the larger the curve slope. However, in the finite length regime (as seen in Figure 5), a smaller coupling length tends to lead a higher error floor. Correspondingly, as shown in Figure 3, the greater the coupling length, the higher the decoding complexity. These results indicate that the anytime property and decoding complexity of the proposed JHWD are related to the coupling length γ , and a proper coupling length should be selected to trade off the error performance and decoding complexity.

In Figure 5, the simulation performances of a single message block is observed as a function of decoding delay within the first sliding window region, from $t = 14$, with a source block length of $N^{sc} = 64$ bits and E_b/N_0 is 2 dB. As shown in Figure 5a, the bit error rate (BER) curve of the proposed JHWD algorithm is located between the JEWD and PJEWD algorithms, which is consistent with the numerical results in Figure 4; and it lastly reaches the error floor at $t = 26$ (also within the sliding window size of $3\gamma - 2$). Moreover, the error floor of the proposed JHWD algorithm is closer to that of the PJEWD algorithm. When the coupling length is larger, i.e., $\gamma = 8$, the slopes of the BER curves are slightly smaller than those of $\gamma = 5$ before the time slot $t = 5$, but can achieve lower BER performance. when $t = 21$, the BER of the proposed JHWD algorithm is lower than 10^{-6} . Figure 5b shows the average message updates of a single message block as the decoding delay. The message updates of the JEWD and PJEWD algorithms approximately linearly increase within the observed time slot region. As to the hybrid window decoding of the proposed JHWD algorithm, the adaptive local expanding window decoding strategy with smaller iteration rounds of $I_{\max 2}$ does not result in a significant increase of message updates, which is about 10 times lower than those of the JEWD and PJEWD algorithms at $t = 26$. When the coupling length is $\gamma = 8$, the average message updates of the proposed JHWD algorithm are relatively higher than those of $\gamma = 5$. This experiment indicates that the anytime property of the proposed JHWD algorithm is between the JEWD algorithm and PJEWD algorithm but with lower decoding complexity.

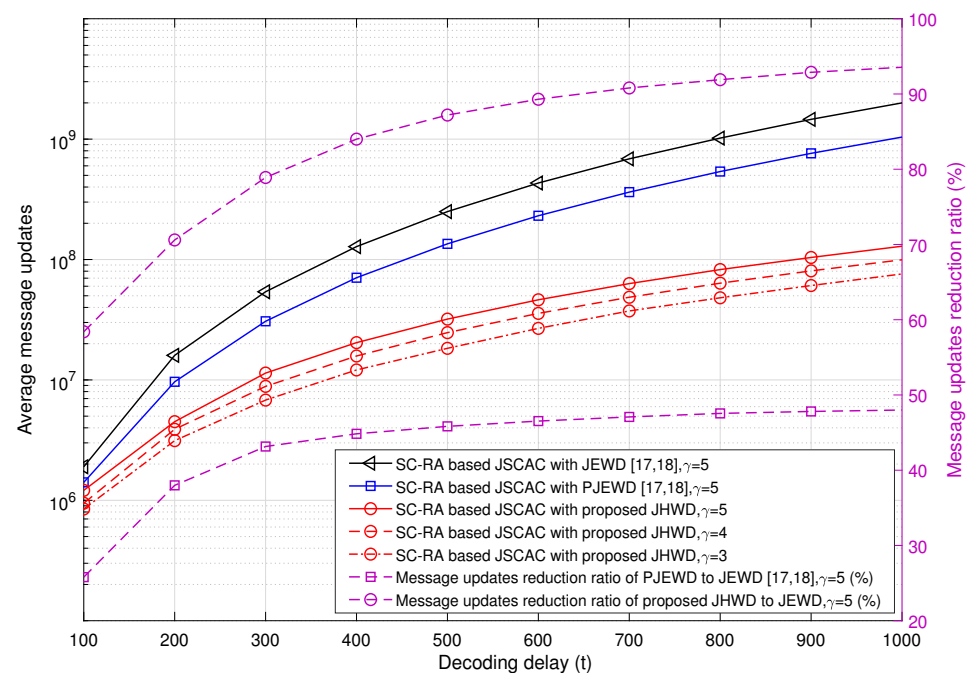


Figure 3. Asymptotic decoding complexity analysis based on the density evolution [17,18].

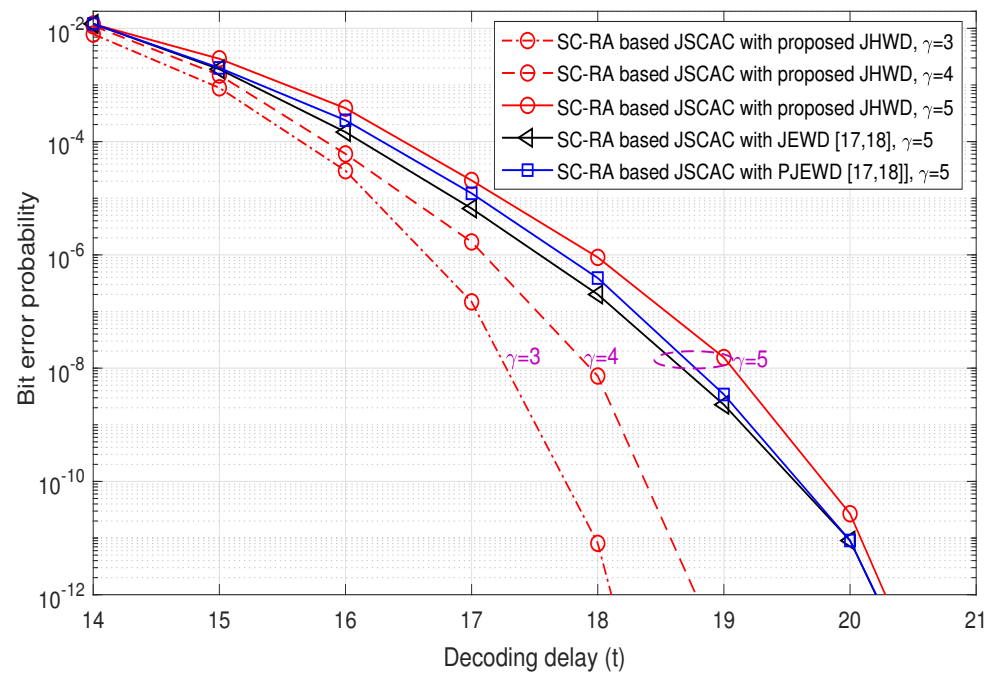


Figure 4. Asymptotic anytime property analysis based on the density evolution [17,18].

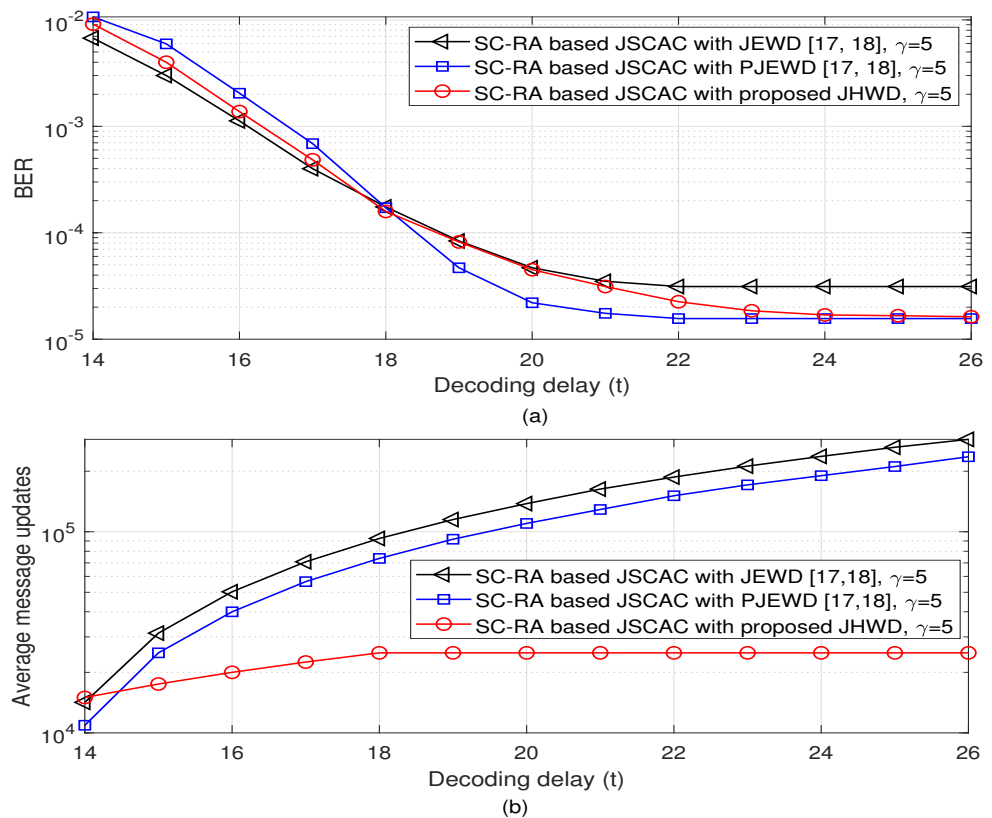


Figure 5. BER performance and decoding complexity of a single message block with $N^{sc} = 64$ bits as the decoding delay within a sliding window region [17,18].

In Figure 6, the simulation performance of the data chain at the time slot $t = 100$ versus E_b/N_0 is observed. In Figure 6a, the proposed JHWD algorithm has the smallest curve slope in the low signal to noise ratio (SNR) region. In the high SNR region, larger than 1.5 dB, the proposed JHWD algorithm can obtain a lower error floor than that of JEWD algorithm and

is closer to the BER performance of the PJEWD algorithm. Figure 6b shows the average message updates normalized by that of the JEWD scheme. The decoding complexity of traditional JEWD does not change with SNR. While the decoding complexities of the proposed JHWD and PJEWD scheme are lower than that of the JEWD scheme, especially in the high SNR region. When $E_b/N_0 = 3$ dB, the average message updates of PJEWD can be decreased about 26.47%, while the proposed JHWD can be decreased about 69.61% compared with the JEWD algorithm. These simulation results show that the proposed JHWD can reduce the decoding complexity to a great extent without sacrificing too much error correction performance.

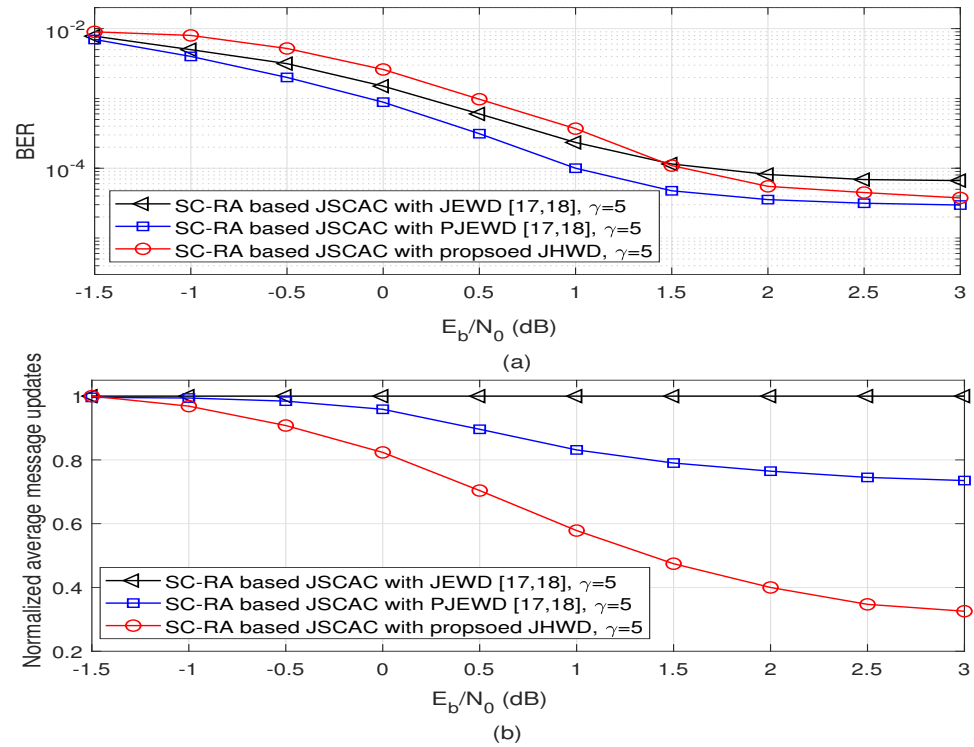


Figure 6. BER performance and decoding complexity of the data chain at the time slot $t = 100$ ($L = 100$) with $N^{sc} = 64$ bits [17,18].

Figure 7 provides the BER performance comparisons with the state of the art, especially the (3,12)- and (3,6)- SC-LDPC-based JSCC scheme with sliding variable-window decoding (SVWD) of comparable window size [16]. As seen in Figure 7, with the same code rates of $R^{sc} = 0.25$, $R^{cc} = 0.5$ and the code length of 6400 bits, the block D-LDPC-based JSCC (LDPC i-8-2-4) [10] has the highest error floor. When the coupling length γ is 3, which is comparable to the syndrome former memory of $m = 2$ in [16] ($\gamma = m + 1$), and the adaptive decoding window size is $2\gamma - 1 = 5$ to $3\gamma - 2 = 7$ (versus the sliding variable-window size of $W_f = 3$ to $W_{max} = 6$ in [16]), the SC-RA-based JSCAC with proposed JHWD has a lower error floor. However, as γ is increased to match the larger window sizes of [16], the preponderance of SC-RA-based JSCAC with the proposed JHWD is gradually reduced. When γ is 5, the JSCAC scheme with the proposed JHWD demonstrates a similar performance to the SC-LDPC JSCC scheme with SVWD of $W_f = 9$. The narrowing gap might be caused by the mismatch between γ and m in [16]. Moreover, Figure 7 also shows the BER performance of i.i.d uniform memory-less source with separate channel coding of $R^{cc} = 1/7$ without loss of the transmission rate. Compared with the separate source channel coding system, the JSCC schemes can achieve early error fall regions and better error performances in the low SNR region, but suffer from error floors in the high SNR region.

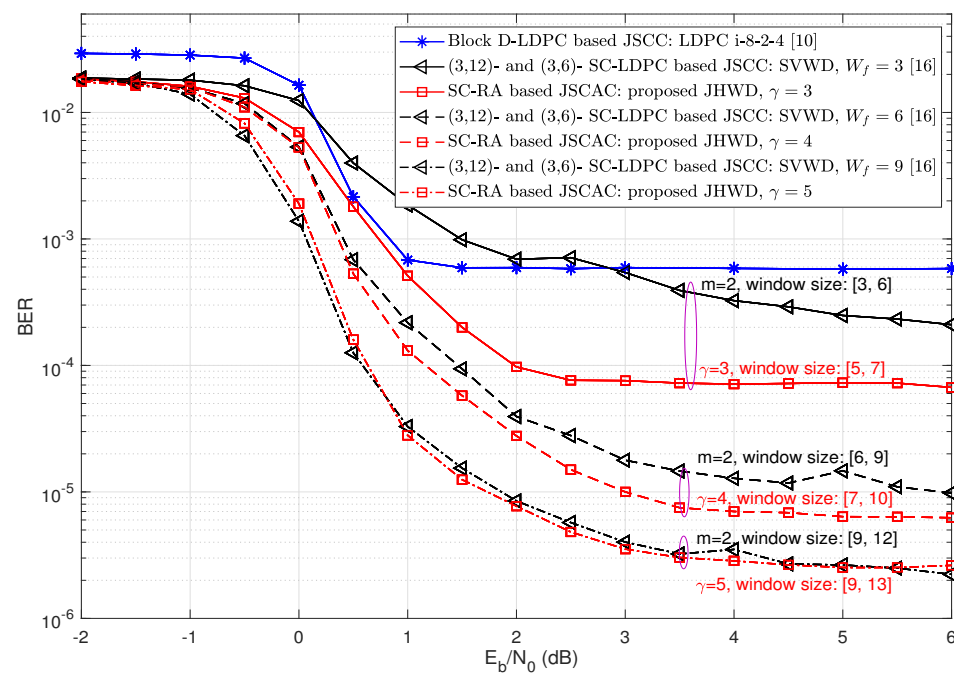


Figure 7. BER performance comparisons with other kinds of JSCC schemes, $R^{sc} = 0.25$, $R^{cc} = 0.5$, $N^{sc} = 640$ bits, $N^{cc} = 320$ bits, and $L = 20$ [10,16].

5. Conclusions

This paper considers a decoding scheme to reduce the decoding complexity without compromising the error correction performance for JSCAC system. We have observed that the improvement of post-decoding BER of the global expanding window decoding in a JSCAC system with finite coupling length gradually decreases with the decoding delay, while the decoding complexity increases approximately linearly. Our key idea, called joint hybrid window decoding (JHWD), is to implement an adaptive local expanding window decoding within the sliding window region. We have also proposed an improved density evolution algorithm to analyze the asymptotic decoding complexity of the JHWD algorithm. The numerical analyses and simulation results have shown that the proposed JHWD algorithm can greatly reduce the decoding complexity while maintaining BER performance similar to that of prior-art decoding schemes, which could be a good alternative for the application of JSCAC systems in low complexity scenarios.

Author Contributions: Conceptualization, L.D. and Y.W.; methodology, L.D. and X.Y.; software, L.D. and Y.W.; validation, Z.S. and X.Y.; formal analysis, Y.L.G.; investigation, Y.W.; resources, Z.X.; data curation, X.Y.; writing—original draft preparation, L.D.; writing—review and editing, Z.S. and Y.L.G.; visualization, Z.X.; supervision, Y.L.G.; project administration, Z.S.; funding acquisition, L.D. and Z.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62371101 and 62461017; the National Key Laboratory of Wireless Communications Foundation under Grant 2023KP01609; the National Key Research and Development Program under Grant 2022YFC3005702; the ZTE Industry-University-Institute Cooperation Funds under Grant IA20240000001; the Natural Science Foundation of Guangxi under Grant 2022GXNSFBA035508 and 2023GXNSFAA026018.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author

Acknowledgments: The authors thank the anonymous reviewers and the Editor for their helpful comments and review of this article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AWGN	additive white Gaussian noise
BER	bit error rate
BP	belief propagation
BPSK	binary phase-shift keying
CNs	check nodes
CSI	channel state information
D-LDPC	double low-density parity check
DP-LDPC	double protograph LDPC
JEWD	joint expanding window decoding
JHWD	joint hybrid window decoding
JSCC	joint source-channel coding
JSCAC	joint source-channel anytime coding
LLRs	log-likelihood ratios
PJEWD	partial joint expanding window decoding
PNs	parity bit nodes
INs	information bit nodes
SNR	signal to noise ratio
SNs	syndrome nodes
SC-LDPC	spatially coupled LDPC code
SSI	source significance information
SC-RA	spatially coupled repeat-accumulate code
UEP	unequal error protection

References

1. Massey, J.L. Joint source and channel coding. *Commun. Sys. Rand. Process Theory*. **1978**, *11*, 279–293.
2. Lyu, S.; Dai, J.; Zhang, P. New paradigm of joint source and channel processing: Semantic communications. *ZTE Technol. J.* **2022**, *27*, 2–8. [\[CrossRef\]](#)
3. Xu, J.; Chen, W.; Ai, B. Deep joint source-channel coding based CSI Feedback. *ZTE Technol. J.* **2022**, *27*, 29–33. [\[CrossRef\]](#)
4. Dong, Y.; Dai, J.; Niu, K.; Wang, S.; Yuan, Y. Joint Source-Channel Coding for 6G Communications. *China Commun.* **2022**, *19*, 101–115. [\[CrossRef\]](#)
5. Cheung, G.; Zakhor, A. Bit allocation for joint/channel coding of scalable video. *IEEE Trans. Image Process.* **2000**, *9*, 340–356. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Bystrom, M.; Stockhammer, T. Dependent source and channel rate allocation for video transmission. *IEEE Trans. Commun.* **2004**, *3*, 258–268. [\[CrossRef\]](#)
7. Shirazinia, A.; Chatterjee, S.; Skoglund, M. Joint source-channel vector quantization for compressed sensing. *IEEE Trans. Signal Process.* **2014**, *62*, 3667–3681.
8. Sanchez, V.; Mandal, M. Efficient channel protection for JPEG2000 bitstream. *IEEE Trans. Circuits Syst. Video Technol.* **2004**, *14*, 554–558. [\[CrossRef\]](#)
9. Zhou, C.; Lin, C.; Zhang, X.; Guo, Z. A novel JSCC scheme for UEP-based scalable video transmission over MIMO systems. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1002–1015. [\[CrossRef\]](#)
10. Fresia, M.; Pérez-Cruz, F.; Poor, H.V.; Verdú, S. Joint source and channel coding. *IEEE Signal Process. Mag.* **2010**, *27*, 104–113. [\[CrossRef\]](#)
11. He, J.; Wang, L.; Chen, P. A joint source and channel coding scheme base on simple protograph structured codes. In Proceedings of the IEEE International Symposium on Communications Information Technologies (ISCIT), Gold Coast, Australia, 2–5 October 2012; pp. 65–69.
12. Jiang, X.; Zeng, G.; Chen, W.; Wen, M.; Li, J. Two-Layer LDPC Codes for Low Complexity ML Detection in GSM Systems. *IEEE Wireless Commun. Lett.* **2018**, *7*, 408–411. [\[CrossRef\]](#)
13. Dong, Y.; Niu, K.; Dai, J.; Wang, S.; Yuan, Y. Joint Source and Channel Coding Using Double Polar Codes. *IEEE Commun. Lett.* **2021**, *25*, 2810–2814. [\[CrossRef\]](#)
14. Golmohammadi, A.; Mitchell, D.G.M. Concatenated spatially coupled LDPC codes with sliding window decoding for joint source-channel coding. *IEEE Trans. Commun.* **2022**, *70*, 851–864. [\[CrossRef\]](#)
15. Jiang, X.; Yuang, S.; Huang, P.; Zeng, G. High-Speed Reconciliation for CVQKD Based on Spatially Coupled LDPC Codes. *IEEE Photon. J.* **2018**, *10*, 1–10. [\[CrossRef\]](#)

16. Lian, Q.; Chen, Q.; Zhou, L.; He, Y.; Xie, X. Adaptive decoding algorithm with variable sliding window for double SC-LDPC coding system. *IEEE Commun. Lett.* **2023**, *27*, 404–408. [[CrossRef](#)]
17. Deng, L.; Wang, Y.; Yu, X.; Noor-A-Rahim, M.; Guan, Y.; Shi, Z. Joint Source Channel Anytime Coding. In Proceedings of the 2020 IEEE Global Communications Conference (Globecom), Taipei, Taiwan, 7–11 December 2020.
18. Deng, L.; Yu, X.; Wang, Y.; Noor-A-Rahim, M.; Guan, Y.; Shi, Z.; Zhang, Z. Joint Source Channel Anytime Coding based on Spatially Coupled Repeat-Accumulate Codes. *IEEE Trans. Commun.* **2023**, *71*, 6215–6230. [[CrossRef](#)]
19. Sahai, A.; Mitter, S. The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link-part I: Scalar systems. *IEEE Trans. Inf. Theory* **2006**, *52*, 3369–3395. [[CrossRef](#)]
20. Grosjean, L. Practical Anytime Codes. Ph.D. Dissertation, KTH Royal Institute of Technology, Stockholm, Sweden, 2016.
21. Noor-A-Rahim, M.; Khyam, M.; Guan, Y.; Ali, G.; Nguyen, K.; Lechner, G. Delay-Universal Channel Coding with Feedback. *IEEE Access* **2018**, *6*, 37918–37931. [[CrossRef](#)]
22. Johnson, S.; Lechner, G. Spatially coupled repeat-accumulate codes. *IEEE Commun. Lett.* **2013**, *17*, 373–376. [[CrossRef](#)]
23. Zhu, M.; Mitchell, D.; Lentmaier, M.; Costello, D. Error Propagation Mitigation in Sliding Window Decoding of Spatially Coupled LDPC Codes. *IEEE J. Sel. Areas Inf. Theory* **2023**, *4*, 470–486. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.