SISO Decoding of U-UV Codes

Changyu Wu †, Li Chen ‡
† School of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou, China
‡ School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China
Email: wuchy28@mail2.sysu.edu.cn, chenli55@mail.sysu.edu.cn

2021 IEEE Globecom Workshops (GC Wkshps) | 978-1-6654-2390-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/GCWkshps52748.2021.9682136

***Abstract*—U-UV structural coding with algebraic component codes can provide competent error-correction performance in the short-to-medium length regime. Constituted by BCH component codes and its ordered statistics decoding (OSD), the successive cancellation list (SCL) decoding of U-UV codes can outperform that of polar codes. However, the current SCL decoding is a soft-in hard-out (SIHO) process. Exploiting its list decoding feature, this paper proposes a soft-in soft-out (SISO) decoding for U-UV codes, providing the key technique for the codes to be further engaged in an iterative system. The proposal is designed based on the recursive structure of U-UV codes and the list decoding feature for both the component and the structural codes. Both the decoding complexity and its soft information transfer characteristics are also shown.**

***Index Terms*—U-UV codes, soft-in soft-out decoding, successive cancellation list decoding**

I. Introduction

Future communications require ultra low-latency for which the short-to-medium length channel codes will play an important role. However, most modern codes such as turbo codes [1], low-density parity-check (LDPC) codes [2] and polar codes [3] realize their capacity-approaching performance with a large codeword length. In the short-to-medium length regime, BCH codes, tail-biting convolutional codes, non-binary LDPC codes, polar codes, and the more recent polarization adjusted convolutional (PAC) codes [4] are known to be good candidates [5]. Their error-correction competency is often realized by a particular decoding mechanism, e.g., the successive cancellation list (SCL) decoding for polar codes and the sequential decoding for the PAC codes. U-UV structural codes were recently introduced as another good performing short-to-medium length code [6]. It is constructed by a number of small component codes under the U-UV structure which was also known as the Plotkin structure [7]. This structural coding results in polarized subchannel capacities, so that rates of the component codes can be designed accordingly. It has been shown that employing BCH component codes and its ordered statistics decoding (OSD) [8], the SCL decoding of U-UV codes outperforms that of polar codes [6].

The SCL decoding is a soft-in hard-out (SIHO) process, without providing the *a posteriori* probabilities (APPs) for the codeword (or message) symbols. If the U-UV codes are further concatenated, e.g., in coded modulations or code concatenations, iterative decoding would be desired. Therefore, it is also important to design the soft-in soft-out (SISO) decoding technique for the codes. SISO decoding of block codes can be categorized into the trellis based approach [9], [10], the belief propagation (BP) based approach [11]–[13] and the list based approach [14]–[16], respectively. The trellis based SISO decoding provides the optimal APPs, but its complexity depends on the number of trellis states. For block codes, this is exponential with codeword length, prohibiting its practice. The BP based SISO decoding will be effective only if the parity-check matrix of the code exhibits sparsity. The list based SISO decoding lifts these dependencies on the codes. The codeword symbol APPs can be generated by a list of decoding estimations with their likelihood metrics.

Exploiting the nature of the SCL decoding, this paper proposes the list based SISO decoding for the U-UV codes. The symbol-wise *a posteriori* log-likelihood ratio (LLR) can be determined by having a sufficiently large list of decoding estimations that also exhibit symbol plurality. It will be shown that this can be ensured by the list decoding of component codes and the U-UV code structure. We will show that the *a posteriori* symbol LLR can be straightforwardly calculated using the SCL decoding metrics. Decoding complexity will be analyzed. The soft information transfer characteristics of the SISO decoding will also be studied, demonstrating how will reliability of the soft output be affected by the decoding parameters. This work provides the key technique for U-UV codes to be further engaged in a coded modulation or a code concatenation system.

II. Preliminaries

A. U-UV Codes

In this paper, we consider binary U-UV codes. Therefore, it is assumed that the U code and V code are two binary block codes of length n with dimensions k_{U} and k_{V}, respectively. A U-UV code of length $2n$ and dimension $k_{\mathrm{U}} + k_{\mathrm{V}}$ can be constructed by [7]

$$\{(\boldsymbol{c}_{\mathrm{U}}|\boldsymbol{c}_{\mathrm{U}} + \boldsymbol{c}_{\mathrm{V}}) : \boldsymbol{c}_{\mathrm{U}} \in \mathcal{C}_{\mathrm{U}}, \boldsymbol{c}_{\mathrm{V}} \in \mathcal{C}_{\mathrm{V}}\}. \quad (1)$$

where $\mathcal{C}_{\mathrm{U}}$ and $\mathcal{C}_{\mathrm{V}}$ denote the codebooks of the U code and the V code, respectively, and $\boldsymbol{c}_{\mathrm{U}}$ and $\boldsymbol{c}_{\mathrm{V}}$ are their codewords. The above construction can be extended by involving more component codes. Fig. 1 illustrates the construction of an h levels U-UV code. It contains $\gamma = 2^h$ component codes with an overall length of $N = \gamma n$. This construction results in γ subchannels that convey the component codes. They have polarized capacities, based on which the component code rates can be designed [3], [17]. Considering the finite length transmission limit, the component code rates can be more precisely tuned by calculating the so called finite length rate [18].

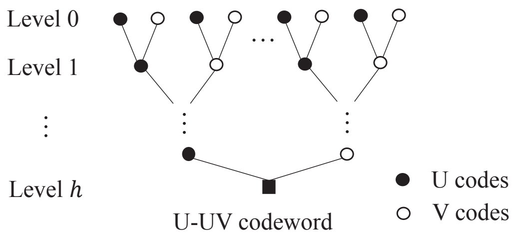

Fig. 1. Construction of an h levels U-UV code.

The U-UV codes can also be viewed as generalized concatenated codes (GCCs) [19]. Fig. 2 shows its GCC interpretation. In particular, let $\mathcal{C}^{(i)}$ and $\boldsymbol{c}^{(i)} = (c_0^{(i)}, c_1^{(i)}, ..., c_{n-1}^{(i)})$ denote the ith component code and its codeword, respectively, where $i = 0, 1, ..., \gamma - 1$. There are n polar codes [3] of length γ as the inner codes. Input of the jth polar encoder will be $(c_j^{(0)}, c_j^{(1)}, ..., c_j^{(\gamma-1)})$, i.e., the jth codeword symbol of all component codes. The U-UV codeword $\boldsymbol{v}$ is obtained by cascading the output of the n polar encoders, i.e., $\boldsymbol{v} = (v_0, v_1, ..., v_{\gamma-1}, ..., v_{N-\gamma}, v_{N-\gamma+1}, ..., v_{N-1})$.

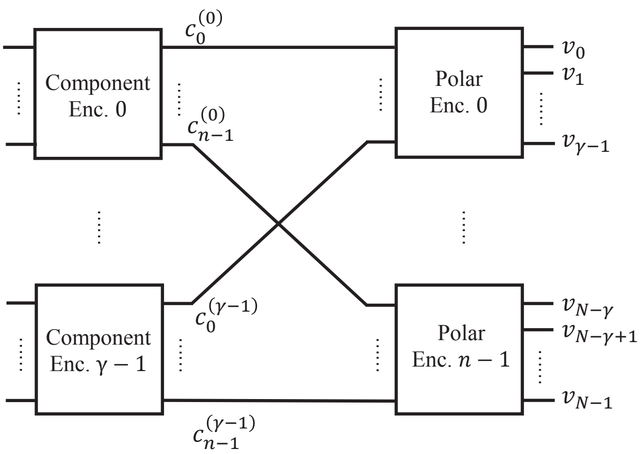

Fig. 2. The GCC interpretation of U-UV codes.

B. List Based SISO Decoding

This is demonstrated under the paradigm of a length n dimension k binary code and the bipolar modulation. Let $\boldsymbol{c} = (c_0, c_1, ..., c_{n-1})$ denote the transmitted codeword, and $\boldsymbol{x}(\boldsymbol{c})$ denote the bipolar modulated symbol vector that is generated by $c_j \mapsto x(c_j) : \{0, 1\} \mapsto \{1, -1\}$. It is assumed that $\boldsymbol{x}(\boldsymbol{c})$ is transmitted through the additive white Gaussian noise (AWGN) channel with a noise variance of σ^2. With the received symbol vector $\boldsymbol{y} = (y_0, y_1, ..., y_{n-1})$, an LLR vector $\boldsymbol{L}$ can be obtained, whose entries are defined as

$$L_j = \ln \frac{P(y_j|c_j = 0)}{P(y_j|c_j = 1)}, \tag{2}$$

where

$$P(y_j|c_j) = \left(2\pi\sigma^2\right)^{-\frac{1}{2}} e^{-\frac{|y_j - x(c_j)|^2}{2\sigma^2}} \tag{3}$$

is the symbol-wise channel transition probability. Based on (2) and (3), $L_j = \frac{2y_j}{\sigma^2}$. Hence, $P(y_j|c_j)$ can also be expressed as

$$P(L_j|c_j) = \left(2\pi\sigma^2\right)^{-\frac{1}{2}} e^{-\frac{|\frac{\sigma^2}{2} L_j - x(c_j)|^2}{2\sigma^2}}. \tag{4}$$

The vector-wise channel transition probability can be further defined as

$$P(\boldsymbol{L}|\boldsymbol{c}) = \prod_{j=0}^{n-1} P(L_j|c_j) = \left(2\pi\sigma^2\right)^{-\frac{n}{2}} e^{-\frac{\sum_{j=0}^{n-1} |\frac{\sigma^2}{2} L_j - x(c_j)|^2}{2\sigma^2}}. \tag{5}$$

With an estimated codeword $\hat{\boldsymbol{c}} = (\hat{c}_0, \hat{c}_1, ..., \hat{c}_{n-1})$, its likelihood can be indicated by $P(\boldsymbol{L}|\hat{\boldsymbol{c}})$. We define

$$\lambda(\hat{\boldsymbol{c}}, \boldsymbol{L}) = \sum_{j: L_j \cdot x(\hat{c}_j) < 0} |L_j| \tag{6}$$

as the correlation distance between $\hat{\boldsymbol{c}}$ and $\boldsymbol{L}$. A smaller $\lambda(\hat{\boldsymbol{c}}, \boldsymbol{L})$ indicates $\hat{\boldsymbol{c}}$ is more likely to be transmitted. Let $\mathcal{L} = \{\hat{\boldsymbol{c}}\}$ denote the list of decoding estimations and $b \in \{0, 1\}$, we can further define

$$\mathcal{L}(j, b) = \{\hat{\boldsymbol{c}} : \hat{c}_j = b, \hat{\boldsymbol{c}} \in \mathcal{L}\} \tag{7}$$

as a subset of $\mathcal{L}$. It collects the decoding estimations with their jth symbol being b. Hence, the symbol-wise *a posteriori* LLR can be determined by

$$L'_j = \ln \frac{\max_{\hat{\boldsymbol{c}} \in \mathcal{L}(j,0)} P(\hat{\boldsymbol{c}}|\boldsymbol{L})}{\max_{\hat{\boldsymbol{c}} \in \mathcal{L}(j,1)} P(\hat{\boldsymbol{c}}|\boldsymbol{L})}. \tag{8}$$

Assuming all codewords are equally likely to be transmitted,

$$L'_j = \ln \frac{\max_{\hat{\boldsymbol{c}} \in \mathcal{L}(j,0)} P(\boldsymbol{L}|\hat{\boldsymbol{c}})}{\max_{\hat{\boldsymbol{c}} \in \mathcal{L}(j,1)} P(\boldsymbol{L}|\hat{\boldsymbol{c}})}. \tag{9}$$

Therefore, in order to obtain the *a posteriori* LLR for each codeword symbol, the list $\mathcal{L}$ shall be sufficiently large such that $\mathcal{L}(j, b)$ is not empty for all j and b. This SISO decoding approximates the outcome of the max-log-MAP decoding [20].

To generate a sufficiently large codeword list, the OSD [8] remains the best known approach. It is characterized by an order τ, where $0 \leq \tau \leq k$. Given $\boldsymbol{L}$, the most reliable basis (MRB) $\boldsymbol{\mathcal{I}}$ that contains the k most reliable independent positions is formed and $|\boldsymbol{\mathcal{I}}| = k$. The initial message $\boldsymbol{m}_{\boldsymbol{\mathcal{I}}}$ can be obtained by making hard-decisions on positions of $\boldsymbol{\mathcal{I}}$. An order-τ reprocessing is performed based on $\boldsymbol{m}_{\boldsymbol{\mathcal{I}}}$. It flips at most τ positions of $\boldsymbol{m}_{\boldsymbol{\mathcal{I}}}$ at a time, producing a new message and its corresponding codeword $\hat{\boldsymbol{c}}$. Hence, the decoding produces $\sum_{t=0}^{\tau} \binom{k}{t}$ codeword candidates, each of which is assigned with a likelihood metric of (6). The most likely candidate

$$\boldsymbol{w} = \arg\min_{\hat{\boldsymbol{c}}} \lambda(\hat{\boldsymbol{c}}, \boldsymbol{L}) \tag{10}$$

will be chosen as the final codeword estimation.

The SISO-OSD [15] produces the *a posteriori* LLR of (9) based on a list $\mathcal{L}$. In order to ensure $\mathcal{L}(j, b)$ is not empty for all j and b so that (9) can be computed for each codeword symbol, the SISO-OSD requires more reprocessing. $\mathcal{L}$ is initialized with $\boldsymbol{w}$ obtained by the OSD. Let $\boldsymbol{\mathcal{I}}_j = \boldsymbol{\mathcal{I}} \backslash j$ so that $|\boldsymbol{\mathcal{I}}_j| = k - 1$, and $\boldsymbol{w}_{\boldsymbol{\mathcal{I}}_j}$ denote the message vector with decisions on $\boldsymbol{\mathcal{I}}$ being the same as those in $\boldsymbol{w}$ but only w_j flipped. For each $j \in \boldsymbol{\mathcal{I}}$, an order-$\tau$ reprocessing will be performed based on $\boldsymbol{\mathcal{I}}_j$ and with $\boldsymbol{w}_{\boldsymbol{\mathcal{I}}_j}$ as the initial message. All new codeword candidates generated in these extra reprocessing are included in $\mathcal{L}$. This mechanism guarantees that $\mathcal{L}(j, b)$ is not empty for any j and b. That says both $\hat{c}_j = 0$ and $\hat{c}_j = 1$ will appear in at least one candidate of $\mathcal{L}$. It also enlarges the cardinality of $\mathcal{L}$ so that the reliability of the *a posteriori* LLRs is improved. For notation convenience, let $\hat{\boldsymbol{c}}(j, b)$ denote the most likely candidate in $\mathcal{L}(j, b)$ so that L'_j of (9) can be determined as

$$L'_j = \ln \frac{P(\boldsymbol{L}|\hat{\boldsymbol{c}}(j, 0))}{P(\boldsymbol{L}|\hat{\boldsymbol{c}}(j, 1))}. \tag{11}$$

The correlation distance between $\hat{\boldsymbol{c}}(j, b)$ and $\boldsymbol{L}$ is denoted as $\lambda(j, b)$. At the end of the decoding, the *a posteriori* LLR vector $\boldsymbol{L}' = (L'_0, L'_1, ..., L'_{n-1})$ can be obtained. The SISO-OSD algorithm with parameter τ is summarized as in **Algorithm 1**.

Algorithm 1 SISO-OSD

Input: $\boldsymbol{L}$, τ;
Output: $\mathcal{L}$, $\boldsymbol{L}'$;

```
Form the MRB 𝓘 and the initial message m_𝓘;
Perform OSD with order τ;
Pick up w in creating 𝓛;
for each j ∈ 𝓘 do
    Determine 𝓘_j and w_{𝓘_j};
    Perform order-τ reprocessing based on 𝓘_j and w_{𝓘_j},
    enlarging 𝓛;
end for
for each j ∈ {0, 1, ..., n − 1} do
    for each b ∈ {0, 1} do
        Find ĉ(j, b) from 𝓛(j, b);
    end for
    Determine L'_j as in (11);
end for
```

III. SCL Decoding of U-UV Codes

This section revisits the SCL decoding of U-UV codes, which underpins the proposed SISO decoding. Parameterized by a decoding output list size l, the SCL decoding generates a list of U-UV codeword estimations. It can be illustrated under the GCC paradigm of Fig. 2. Assume that a U-UV codeword $\boldsymbol{v}$ of length N is transmitted and the received LLR vector is $\boldsymbol{L}$. It will be equally partitioned into n LLR subvectors, which are the input of the n inner SC decoders [3]. The inner decoders can function in parallel and estimate their message symbol LLRs successively. Once the ith message symbol LLRs of all inner codes have been produced, they are collected in forming an LLR vector $\boldsymbol{L}^{(i)}$, which is the input of the ith component decoder. Correspondingly, the γ component codes are decoded successively. After component code $\mathcal{C}^{(i)}$ is decoded, codeword estimation $\hat{\boldsymbol{c}}^{(i)} = (\hat{c}_0^{(i)}, \hat{c}_1^{(i)}, ..., \hat{c}_{n-1}^{(i)})$ will be fed back to the n inner decoders for further LLR update. The inner decoders and component decoders exchange their decoding output in such a manner until all component codes are decoded. At the end, a U-UV codeword estimation $\hat{\boldsymbol{v}}$ is obtained.

If the component codes are decoded by a list decoding algorithm, e.g. the OSD, the SCL decoding of U-UV codes can be designed. Let $\mathcal{L}_i$ denote the decoding output list of the ith component code, i.e.,

$$\mathcal{L}_i = \left\{ \hat{\boldsymbol{c}}_0^{(i)}, \hat{\boldsymbol{c}}_1^{(i)}, ..., \hat{\boldsymbol{c}}_{l-1}^{(i)} \right\}, \tag{12}$$

where $|\mathcal{L}_i| = l$. In the SCL decoding process, multiple decoding paths are maintained. With one initial path at the beginning, path expansion will be performed each time after a component code is decoded. An existing path will branch into l separate paths, each of which is emancipated from a candidate of $\mathcal{L}_i$. This will result in an exponentially growing number of decoding paths, triggering a prohibitive complexity. In order to rationalize the complexity, path pruning following each path expansion will be necessary. That says after a component code is decoded, only the l most reliable expanded paths will be preserved. For this, we define the accumulated correlation distance as

$$\Lambda^{(i)} = \lambda(\hat{\boldsymbol{c}}^{(0)}, \boldsymbol{L}^{(0)}) + \lambda(\hat{\boldsymbol{c}}^{(1)}, \boldsymbol{L}^{(1)}) + \cdots + \lambda(\hat{\boldsymbol{c}}^{(i)}, \boldsymbol{L}^{(i)}). \tag{13}$$

It indicates the reliability of a decoding path that reaches component code $\mathcal{C}^{(i)}$. A smaller $\Lambda^{(i)}$ indicates the decoding path is more reliable. After the last component code is decoded, each of the l decoding paths corresponds to a U-UV codeword estimation with a likelihood metric $\Lambda^{(\gamma-1)}$. These codeword estimations are kept in the SCL decoding output list $\mathcal{L}$, and the most likely one within is normally chosen as the decoding output.

IV. SISO Decoding of U-UV Codes

Armed with the above mentioned SCL decoding of U-UV codes, its SISO decoding can be further derived. Given a list of decoding estimations $\mathcal{L} = \{\hat{\boldsymbol{v}}\}$, we again define

$$\mathcal{L}(j,b) = \{\hat{\boldsymbol{v}} : \hat{v}_j = b, \hat{\boldsymbol{v}} \in \mathcal{L}\} \tag{14}$$

as a subset of $\mathcal{L}$, where $b \in \{0,1\}$ and $j \in \{0,1,...,N-1\}$. It collects all decoding estimations with the jth symbol being b. Based on (9), the *a posteriori* LLR of v_j can be generated by

$$L'_j = \ln \frac{\max\limits_{\hat{\boldsymbol{v}} \in \mathcal{L}(j,0)} P(\boldsymbol{L}|\hat{\boldsymbol{v}})}{\max\limits_{\hat{\boldsymbol{v}} \in \mathcal{L}(j,1)} P(\boldsymbol{L}|\hat{\boldsymbol{v}})}. \tag{15}$$

Again, it should be ensured that $\mathcal{L}(j,b)$ is not empty for all j and b so that (15) can be computed for each codeword symbol. This can be guaranteed by utilizing the U-UV code structure and the SCL decoding.

For the SCL decoding, the reliability of a decoding path is evaluated by the accumulated correlation distance $\Lambda^{(\gamma-1)}$ as in (13). Given an estimated U-UV codeword $\hat{\boldsymbol{v}}$, we use $\hat{\boldsymbol{c}}^{(i)}(\hat{\boldsymbol{v}})$ to denote its ith component codeword. Assume all component codeword estimations are independent such that

$$P(\boldsymbol{L}|\hat{\boldsymbol{v}}) = \prod_{i=0}^{\gamma-1} P(\boldsymbol{L}^{(i)}|\hat{\boldsymbol{c}}^{(i)}(\hat{\boldsymbol{v}})). \tag{16}$$

Subsequently, L'_j of (15) can be approximated as

$$L'_j \approx \ln \frac{\max\limits_{\hat{\boldsymbol{v}} \in \mathcal{L}(j,0)} \prod_{i=0}^{\gamma-1} P(\boldsymbol{L}^{(i)}|\hat{\boldsymbol{c}}^{(i)}(\hat{\boldsymbol{v}}))}{\max\limits_{\hat{\boldsymbol{v}} \in \mathcal{L}(j,1)} \prod_{i=0}^{\gamma-1} P(\boldsymbol{L}^{(i)}|\hat{\boldsymbol{c}}^{(i)}(\hat{\boldsymbol{v}}))}. \tag{17}$$

Note that the successive decoding mechanism implies that $\boldsymbol{L}^{(i)}$ will be computed based on the previous component codeword estimations $\hat{\boldsymbol{c}}^{(0)}, \hat{\boldsymbol{c}}^{(1)}, ..., \hat{\boldsymbol{c}}^{(i-1)}$. However, we apply the assumption of (16) and yield an approximation for L'_j as in (17). This is consistent with the SCL decoding metric, since intermediate reliability metric for decoding path selection is necessary. Based on (13), let us further define

$$\Lambda(\hat{\boldsymbol{v}}) = \sum_{i=0}^{\gamma-1} \lambda(\hat{\boldsymbol{c}}^{(i)}(\hat{\boldsymbol{v}}), \boldsymbol{L}^{(i)}) \tag{18}$$

as the likelihood metric for an estimated U-UV codeword $\hat{\boldsymbol{v}}$. Based on (5), (6) and (17), the *a posteriori* LLR approximation can be determined by

$$L'_j \approx \min_{\hat{\boldsymbol{v}} \in \mathcal{L}(j,1)} \Lambda(\hat{\boldsymbol{v}}) - \min_{\hat{\boldsymbol{v}} \in \mathcal{L}(j,0)} \Lambda(\hat{\boldsymbol{v}}). \tag{19}$$

The above equation shows that the *a posteriori* LLR can be straightforwardly determined by calculating the discrepancy between two accumulated correlation distances, i.e., the SCL decoding metrics. However, we should still guarantee that $\mathcal{L}(j,b)$ is not empty for all j and b. The following shows that with the U-UV structure, this can be substantiated by performing the SISO-OSD on the last component code $\mathcal{C}^{(\gamma-1)}$.

In estimating a U-UV codeword using the SCL decoding, let us consider that component codeword estimations $\hat{\boldsymbol{c}}^{(0)}, \hat{\boldsymbol{c}}^{(1)}, ..., \hat{\boldsymbol{c}}^{(\gamma-2)}$ have been produced with the accumulated correlation distance $\Lambda^{(\gamma-2)}$. Based on the U-UV structure, the last component estimation $\hat{\boldsymbol{c}}^{(\gamma-1)}$ will uniquely determine the U-UV codeword estimation $\hat{\boldsymbol{v}}$. That says they exhibit a one-to-one correspondence. Therefore, we can guarantee that $\mathcal{L}(j,b)$ is not empty for all j and b by producing $\hat{\boldsymbol{c}}^{(\gamma-1)}(\hat{\boldsymbol{v}})$ that enables $\hat{v}_j = b$.

With the LLR vector $\boldsymbol{L}^{(\gamma-1)}$ and the estimated component codewords $\hat{\boldsymbol{c}}^{(0)}, \hat{\boldsymbol{c}}^{(1)}, ..., \hat{\boldsymbol{c}}^{(\gamma-2)}$, let

$$\hat{\boldsymbol{c}}^{(\gamma-1)}(j,b) = \mathop{\arg\min}_{\hat{\boldsymbol{c}}^{(\gamma-1)}(\hat{\boldsymbol{v}}) : \hat{\boldsymbol{v}} \in \mathcal{L}(j,b)} \lambda(\hat{\boldsymbol{c}}^{(\gamma-1)}(\hat{\boldsymbol{v}}), \boldsymbol{L}^{(\gamma-1)}) \tag{20}$$

denote the most likely estimation of the last component code that constitutes a candidate in $\mathcal{L}(j,b)$. Its correlation distance with $\boldsymbol{L}^{(\gamma-1)}$ is denoted by $\lambda^{(\gamma-1)}(j,b)$, i.e., $\lambda^{(\gamma-1)}(j,b) = \lambda(\hat{\boldsymbol{c}}^{(\gamma-1)}(j,b), \boldsymbol{L}^{(\gamma-1)})$. Based on II-B, the SISO-OSD is able to ensure the existence of both $\hat{\boldsymbol{c}}^{(\gamma-1)}(j,0)$ and $\hat{\boldsymbol{c}}^{(\gamma-1)}(j,1)$ for all U-UV codeword position j. Therefore, in the proposed SISO decoding, the component codes $\mathcal{C}^{(0)}, \mathcal{C}^{(1)}, ..., \mathcal{C}^{(\gamma-2)}$ can be decoded by the OSD with their respective orders, but the last component code $\mathcal{C}^{(\gamma-1)}$ will have to be decoded by the SISO-OSD.

Let us further define

$$\Lambda(j,b) = \Lambda^{(\gamma-2)} + \lambda^{(\gamma-1)}(j,b) \tag{21}$$

as the decoding metric of a complete path. With the SCL decoding mechanism, after component code $\mathcal{C}^{(\gamma-2)}$ is decoded, l most reliable decoding paths are preserved with their metrics $\Lambda^{(\gamma-2)}$. After the SISO-OSD of the last component code $\mathcal{C}^{(\gamma-1)}$, path metrics sets $\{\Lambda(j,b)\}$ are obtained for all j and b. Based on (19), the *a posteriori* LLR for a U-UV codeword symbol v_j can be determined by

$$L'_j = \min\{\Lambda(j,1)\} - \min\{\Lambda(j,0)\}. \tag{22}$$

Note that the most reliable decoding path must be involved in the calculation of L'_j for each j. Consequently, at most $N+1$

U-UV codeword estimations will be utilized in evaluating the *a posteriori* LLR vector $\boldsymbol{L}' = (L'_0, L'_1, ..., L'_{N-1})$. The extrinsic LLR for symbol v_j can be further determined by $L''_j = L'_j - L_j$.

The SISO decoding of U-UV codes is summarized as in **Algorithm 2**. Note that the OSD orders for different component codes may vary. We use τ_i to denote the OSD order for the ith component, and $\boldsymbol{\tau} = \{\tau_0, \tau_1, ..., \tau_{\gamma-1}\}$.

Algorithm 2 SISO Decoding of U-UV Codes

Input: $\boldsymbol{L}$, $\boldsymbol{\tau}$, l;
Output: $\boldsymbol{L}'$;
Initialize a decoding path;
for each $i \in \{0, 1, ..., \gamma-2\}$ **do**
  for each decoding path **do**
    Run the inner decoders in producing $\boldsymbol{L}^{(i)}$;
    Perform OSD-τ_i to obtain $\mathcal{L}_i$ as in (12);
    Perform path expansion based on $\mathcal{L}_i$;
    Compute $\Lambda^{(i)}$ for each expanded path as in (13);
  end for
  Keep the l most reliable paths;
end for
for each decoding path **do**
  Run the inner decoders in producing $\boldsymbol{L}^{(\gamma-1)}$;
  Perform the SISO-OSD with order $\tau_{\gamma-1}$ in obtaining $\hat{\boldsymbol{c}}^{(\gamma-1)}(j,b)$ and $\lambda^{(\gamma-1)}(j,b)$ for all j and b;
  for each $j \in \{0, 1, ..., N-1\}$ **do**
    Calculate $\Lambda(j,0)$ and $\Lambda(j,1)$ as in (21);
  end for
end for
Calculate $\boldsymbol{L}'$ as in (22);

V. Decoding Complexity

The above SISO decoding complexity can be characterized by considering the SC decoding of inner polar codes, the OSD of component codes $\mathcal{C}^{(0)}, \mathcal{C}^{(1)}, ..., \mathcal{C}^{(\gamma-2)}$ and the SISO-OSD of $\mathcal{C}^{(\gamma-1)}$. Since the n inner polar codes are of length γ and there are l decoding paths, the LLR update complexity of the inner decoders is $O(ln\gamma \log_2 \gamma) = O(lNh)$. The complexity of an order τ OSD is $O(n^\tau)$. Since it is used to decode component codes $\mathcal{C}^{(0)}, \mathcal{C}^{(1)}, ..., \mathcal{C}^{(\gamma-2)}$, its complexity is $O(ln^{\tau_m})$, where $\tau_m = \max\{\tau_0, \tau_1, ..., \tau_{\gamma-2}\}$. The SISO-OSD of the last component code $\mathcal{C}^{(\gamma-1)}$ can be considered as performing the order $\tau_{\gamma-1}$ OSD for $k'+1$ times, where k' is the dimension of $\mathcal{C}^{(\gamma-1)}$. Since the subchannel that conveys $\boldsymbol{c}^{(\gamma-1)}$ exhibits the largest capacity among all, it is usually true that $k' \gg 1$. Hence, the complexity of decoding the last component code will be $O(lk'n^{\tau_{\gamma-1}})$. Compared with the SCL decoding of U-UV codes [6], the SISO decoding costs a higher complexity due to its extra reprocessing in decoding $\mathcal{C}^{(\gamma-1)}$.

VI. Performance Analysis

The *a priori-a posteriori* (and -extrinsic) information transfer characteristics of the proposed SISO decoding will first

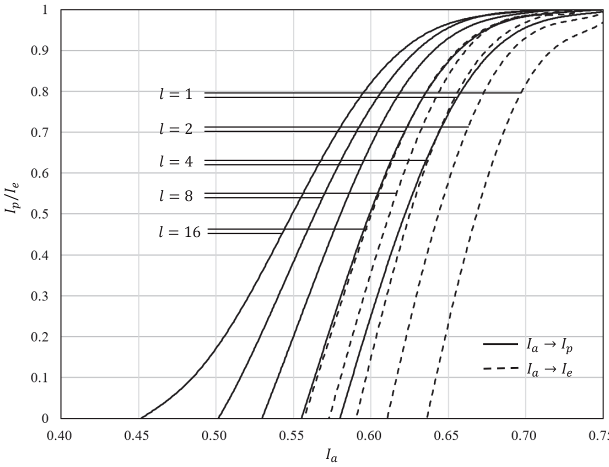

Fig. 3. Soft information transfer characteristics of SISO decoding of the (252,139) U-UV code, with OSD orders $\boldsymbol{\tau} = \{7, 2, 2, 0\}$.

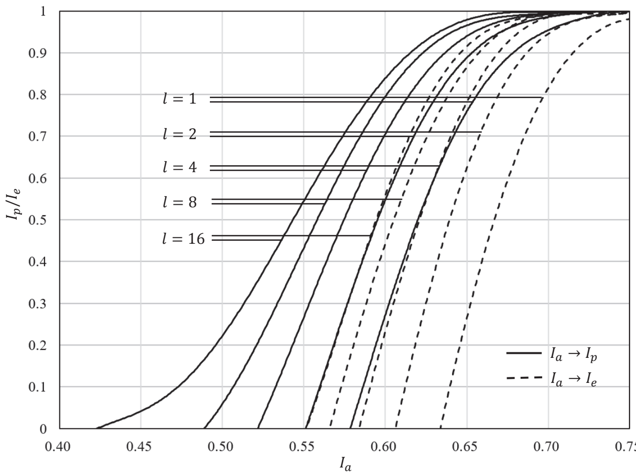

Fig. 4. Soft information transfer characteristics of SISO decoding of the (252,139) U-UV code, with OSD orders $\boldsymbol{\tau} = \{7, 3, 2, 1\}$.

be analyzed. Assuming that the U-UV codeword symbols of 0 and 1 are equally likely to be transmitted, the mutual information between the transmitted bipolar symbols $x(v)$ and the corresponding received LLRs L is defined as [21]

$$\begin{aligned} I(x(v), L) &= H(x(v)) - H(x(v)|L) \\ &= 1 - \int_{-\infty}^{+\infty} P(L|x(v)=1) \log_2\left(1 + e^{-L}\right) dL, \end{aligned} \tag{23}$$

where $H(\cdot)$ is the binary entropy function. After the SISO decoding, both the *a posteriori* LLR L' and the extrinsic LLR L'' can be obtained. Hence, $I(x(v), L')$ and $I(x(v), L'')$ can be evaluated. For simplicity, let $I_a = I(x(v), L)$, $I_p = I(x(v), L')$ and $I_e = I(x(v), L'')$, denote the *a priori*, the *a posteriori* and the extrinsic mutual information, respectively. The $I_a \to I_p$ and $I_a \to I_e$ transfer characteristics can be obtained via Monte-Carlo simulations. Note that the received LLRs can be generated based on a predefined I_a assuming an all-zero codeword is transmitted. As the SISO decoding yields L' and L'', I_p can be determined by

$$I_p = 1 - \mathbb{E}\left[\frac{1}{N}\sum_{j=0}^{N-1}\log_2(1+e^{-L'_j})\right], \qquad (24)$$

while I_e can also be determined similarly.

Figs. 3 and 4 show the *a priori-a posteriori* (and -extrinsic) transfer characteristics of SISO decoding of the 2 levels (252,139) U-UV code. Its component codes $\mathcal{C}^{(0)}, \mathcal{C}^{(1)}, \mathcal{C}^{(2)}$ and $\mathcal{C}^{(3)}$ are the (63, 7), (63, 36), (63, 39) and (63, 57) binary primitive BCH codes, respectively. They show the transfer characteristics obtained with different OSD orders, which are $\boldsymbol{\tau} = \{7, 2, 2, 0\}$ and $\boldsymbol{\tau} = \{7, 3, 2, 1\}$, respectively. It can be seen that by increasing the decoding parameters, e.g., the OSD orders for the component codes or the list size l, a better $I_a \to I_p$ or $I_a \to I_e$ transfer characteristics can be obtained.

Finally, Fig. 5 shows the SISO decoding performance for the U-UV code with $\boldsymbol{\tau} = \{7, 3, 2, 1\}$. It maintains the same performance as the SCL decoding [6]. The U-UV codes can outperform a similar rate and length polar code, i.e., the (256, 140) polar code whose SCL decoding [22], [23] is assisted by a length 8 cyclic redundancy check (CRC) code.

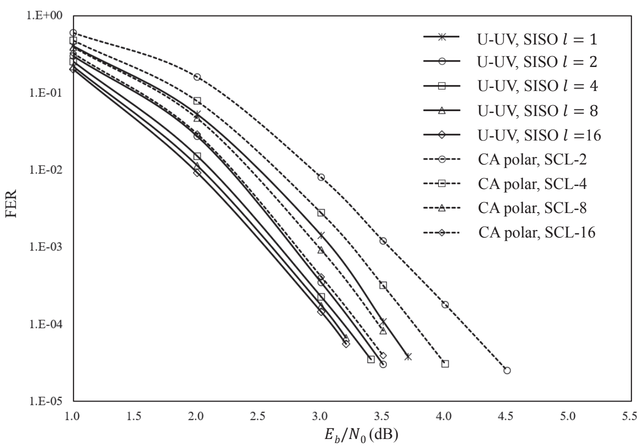

Fig. 5. SISO decoding performance of the (252, 139) U-UV code.

VII. Conclusion

This paper has proposed the SISO decoding for U-UV codes utilizing the list decoding feature of its original SCL decoding. It has been shown that the *a posteriori* and extrinsic LLRs of the codeword symbols can be generated by the likelihood functions of particular estimated codewords. For this, the SISO-OSD is recommended to decode the last component code, while other component codes can be decoded by the OSD. Therefore, the SISO decoding complexity is incremental to the SCL decoding in handling the last component code. Soft information transfer characteristics of the proposed SISO decoding has been analyzed. It has also been shown that the SISO decoding maintains the SCL decoding performance. This work paves the way for U-UV codes to be further engaged in iterative systems.

Acknowledgment

This work is sponsored by the National Natural Science Foundation of China (NSFC) with project ID 62071498.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun.*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[3] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-Input memoryless channels," in *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051-3073, Jul. 2009.

[4] E. Arıkan, "From sequential decoding to channel polarization and back again," arXiv:1908.09594, 2019.

[5] M. Coskun *et al.*, "Efficient error-correcting codes in the short block-length regime," *Phys. Commum.*, vol. 34, pp. 66–79, Mar. 2019.

[6] J. Cheng and L. Chen, "BCH Based U-UV Codes and Its Decoding," in *Proc. the IEEE Int. Symp. Inform. Theory (ISIT)*, Melbourne, Australia, 2021.

[7] M. Plotkin, "Binary codes with specified minimum distance," *IRE Trans. Inf. Theory*, vol. 6, no. 4, pp. 445–450, Sept. 1960.

[8] M. Fossorier and S. Lin, "Soft decision decoding of linear block codes based on ordered statistics," in *Proc. the IEEE Int. Symp. Inform. Theory (ISIT)*, Trondheim, Norway, 1994, pp. 395-.

[9] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (Corresp.)," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284-287, Mar. 1974.

[10] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inf. Theory*, vol. 24, no. 1, pp. 76-80, Jan. 1978.

[11] R. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, 1963.

[12] D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399-431, Mar. 1999.

[13] J. Jiang and K. R. Narayanan, "Iterative Soft-Input Soft-Output Decoding of Reed–Solomon Codes by Adapting the Parity-Check Matrix," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3746-3756, Aug. 2006.

[14] R. Pyndiah, "Near optimum decoding of product codes: block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003–1010, Aug. 1998.

[15] M. Fossorier and S. Lin, "Soft-input soft-output decoding of linear block codes based on ordered statistics," in *Proc. Globecom*, vol. 5, Nov. 1998, pp. 2828–2833.

[16] P. Martin, D. Taylor and M. Fossorier, "Soft-input soft-output list-based decoding algorithm," *IEEE Trans. Comm.*, vol. 52, no. 2, pp. 252-262, Feb. 2004.

[17] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Trans. Commum.*, vol. 60, no. 11, pp. 3221–3227, Nov. 2012.

[18] Y. Polyanskiy, V. Poor, and S. Verd'u, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[19] P. Trifonov and P. Semenov, "Generalized concatenated codes based on polar codes," in *Proc. IEEE Int. Symp. Wirel. Commun. Syst. (ISWCS)*, Aachen, Germany, 2011, pp. 442–446.

[20] J. Hagenauer, E. Offer and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 429-445, Mar. 1996.

[21] S. ten Brink, "Convergence behaviour of iteratively decoded parallel concatenated codes," *IEEE Trans. Comm.*, vol. 49, Oct. 2001.

[22] K. Niu and K. Chen, "CRC-Aided Decoding of Polar Codes," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668-1671, Oct. 2012.

[23] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.