DESIGN OF TURBO DECODER BASED ON SUM-PRODUCT ALGORITHM OF LDPC CODE

RAJKUMAR JAIN, LOKESH KUMAR

1Assistant Professor, Department of Electronics & Communication, AIET, Jaipur, Rajasthan, INDIA
2Assistant Professor, Department of Electronics & Communication, CITM, Jaipur, Rajasthan, INDIA

Email: 1rkjain1983@gmail.com, 2lokesh4in@yahoo.co.in

ABSTRACT

Turbo code has been used by multiple communication standards due to its performance near Shannon limit. However, its decoding algorithm is complicated. Through the research on Turbo code, a novel design of Turbo decoder is presented in this paper. Turbo code is decoded by the Sum-Product decoding algorithm of LDPC code, which is a low complicated decoding algorithm. Moreover, messages are stored compressively, and the check-to-variable messages and the variable-to-check messages are stored alternately.

Keywords— Turbo Code, Decoder, Low Density Parity Check code, Min Sum.

1. INTRODUCTION

In 1993, C. Berrou presented a novel coding scheme Turbo Code, which is based on convolutional code and concatenated code [2]. Turbo code is a kind of error control coding techniques which has strong error-correcting capability and attracts much attention due to its performance near Shannon limit. So Turbo code is widely used in mobile satellite communication, deep space communication, digital video broadcasting, etc. Furthermore, it has been become the error-correcting code standard of the third-generation mobile communication systems.

Nowadays, the most sophisticated and efficient decoding algorithms of Turbo code are Bahl Cocke Jelinek Raviv (BCJR) algorithm, Maximum a posteriori (MAP) algorithm [2], Log_MAP [3], Max_Log_MAP [4], Soft-Output Viterbi Algorithm (SOVA) [5], etc. Although they all have excellent decoding performance, they are not suitable to decode by hardware because of their high computational complexity, severe time delay and low throughput. Through the research on Turbo code, it is found that the check matrix is very sparse [6] and Turbo code is a special case of Low Density Parity Check [7] (LDPC) code. Therefore, a novel design of Turbo decoder is presented in this paper with low-density feature.

Turbo code is decoded by Sum-product decoding algorithm of LDPC code, which is a low complicated decoding algorithm. Moreover, messages are stored compressively, and the check-to-variable messages and the variable-to-check messages are stored alternately. So the required memories are reduced greatly. The simulation results show that the decoder has high decoding speed.

2. DECODING ALGORITHM

Figure 1: The parity check matrix of LDPC code
3. SUM-PRODUCT ALGORITHM OF LDPC CODE

Sum-Product decoding algorithm is the least complicated algorithm among the decoding algorithm of LDPC code. Its operation steps are as follows:

**Step 1**: Initialization of variable nodes:

\[ L(q_{ij}) \oplus L(c_i) \]

\( L(q_{ij}) \) represents the variable-to-check message, \( L(c_i) \) represents the initial message of variable nodes.

**Step 2**: Check Nodes Update (CNU):

\[ L(r_{ji}) = \prod_{r \in V_{f_{ji}}} \text{sign}(L(q_{r_{ij}})) \cdot \max_{r \in V_{f_{ji}}} |L(q_{r_{ij}})| \]

\( L(r_{ji}) \) represents the check-to-variable message.

**Step 3**: Variable Nodes Update (VNU):

\[ L(q_{ij}) = L(c_i) + \sum_{j \in C_{ij}} L(r_{ji}) \]

**Step 4**: Decision decoding:

\[ y_i' = \begin{cases} 1 & L(Q_i) < 0 \\ 0 & L(Q_i) \geq 0 \end{cases} \quad L(Q_i) = L(c_i) + \sum_{j \in C_i} L(r_{ji}) \]

\( y_i' \) represents the \( i \)th decoded bit, \( L(Q_i) \) represents the output LLR (Log-Likelihood Ratio) of decoded bit.

**Step 5**: Tentative decoding: Stop if all parity-check equations are satisfied \( y_i' \cdot HT = 0 \) or the maximum number of iterations is reached. Otherwise, go to the Step 2.

In order to facilitate the implementation by hardware, the CNU is replaced by

\[ L(r_{ji}) = (\text{sign}(L(q_{ij}))) \sum_{r \in V_{f_{ji}}} \text{sign}(L(q_{r_{ij}})) \]

The first step of CNU: get the 1st minimal, the 2nd minimal, the position of the 1st minimal and the sign computed by XOR the sign bits of all inputs. The second step of CNU: compare the current processed position with the position of 1st minimal; if the current processed position is just the position of 1st minimal, the absolute value of new \( L(r_{ji}) \) is replaced by the 2nd minimal; otherwise, by the 1st minimal. And the sign of new \( L(r_{ji}) \) is computed by

\[ \text{sign}(L(q_{ij}))) \cdot \text{sign} \]

The VNU is replaced by

\[ L(q_{ij}) = \sum_{j \in C_j} (L(r_{ji}) + (L(c_i) - L(r_{ji})) \]

4. SIMULATION AND ANALYSIS

Turbo code is decoded by Sum-product decoding algorithm of LDPC code in Additive White Gaussian Noise (AWGN) channel, its rate is 1/3 and the block size is 32400. The simulation results are shown in Fig. 3 and Fig. 4 shows that the performance of Sum product decoding algorithm is worse than the performance of Turbo LDPC decoding algorithms, because of the presence of short cycle in check matrix of Turbo code and short block length.


