Module 6

Channel Coding

Lesson 37

Turbo Coding

After reading this lesson, you will learn about

- ***Turbo Encoding and Turbo Code Structures;***
- ***SISO Decoders;***
- ***MAP Algorithms;***
- ***Applications;***
- ***Turbo Product Codes (TCP);***

Turbo codes represent a class of parallel concatenation of two convolutional codes. The parallel-concatenated codes have several advantages over the serial concatenated ones. The parallel decoder facilitates the idea of feedback in decoding to improve the performance of the system. There are some differences between conventional convolutional code and turbo codes. Several parameters affect the performance of turbo codes such as: a) component decoding algorithms, b) number of decoding iterations, c) generator polynomials and constraint lengths of the component encoders and d) interleaver type.

Turbo Encoding and Turbo Code Structures

A turbo encoder is sometimes built using two identical convolutional codes of special type, such as, recursive systematic (RSC) type with parallel concatenation. An individual encoder is termed a component encoder. An interleaver separates the two component encoders. The interleaver is a device that permutes the data sequence in some predetermined manner. Only one of the systematic outputs from the two component encoders is used to form a codeword, as the systematic output from the other component encoder is only a permuted version of the chosen systematic output.

Fig. 6.37.1 shows the block diagram of a turbo encoder using two identical encoders. The first encoder outputs the systematic V_0 and recursive convolutional V_1 sequences while the second encoder discards its systematic sequence and only outputs the recursive convolutional V_2 sequence. There are several types of interleavers such as,

a) Block interleaver,
b) Diagonal interleaver,
c) Odd-even block interleaver,
d) Pseudo-random interleaver,
e) Convolutional interleaver,
f) Helical interleaver,
g) Uniform interleaver,
h) Cyclic shift interleaver and
i) Code matched interleaver.

Depending on the number of input bits to a component encoder it may be binary or m-binary encoder. Encoders are also categorized as systematic or non-systematic. If the component encoders are not identical then it is called an asymmetric turbo code.

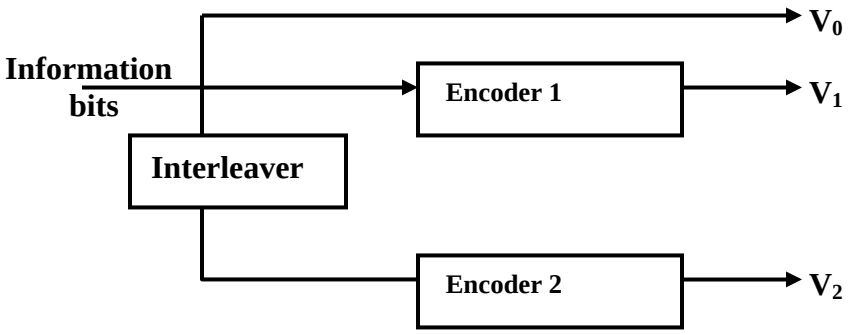

Fig. 6.37.1 *Generic block diagram of a turbo encoder*

Fig. 6.37.2 shows a schematic diagram for the iterative decoding procedure using two 'Soft-in-Soft-out' (SISO) component decoders. The first SISO decoder generates the soft output and subsequently an *extrinsic information* (EI). The extrinsic information is interleaved and used by the second SISO decoder as the estimate of the *a priori* probability (APP). The second SISO decoder also produces the extrinsic information and passes it after de-interleaving to the first SISO decoder to be used during the subsequent decoding operation.

Some of the major decoding approaches, developed for turbo decoding are:

a) Maxium Aposteriori Probability (MAP),
b) Log-MAP,
c) Max-Log-MAP and
d) Soft Output Viterbi Algorithm (SOVA).

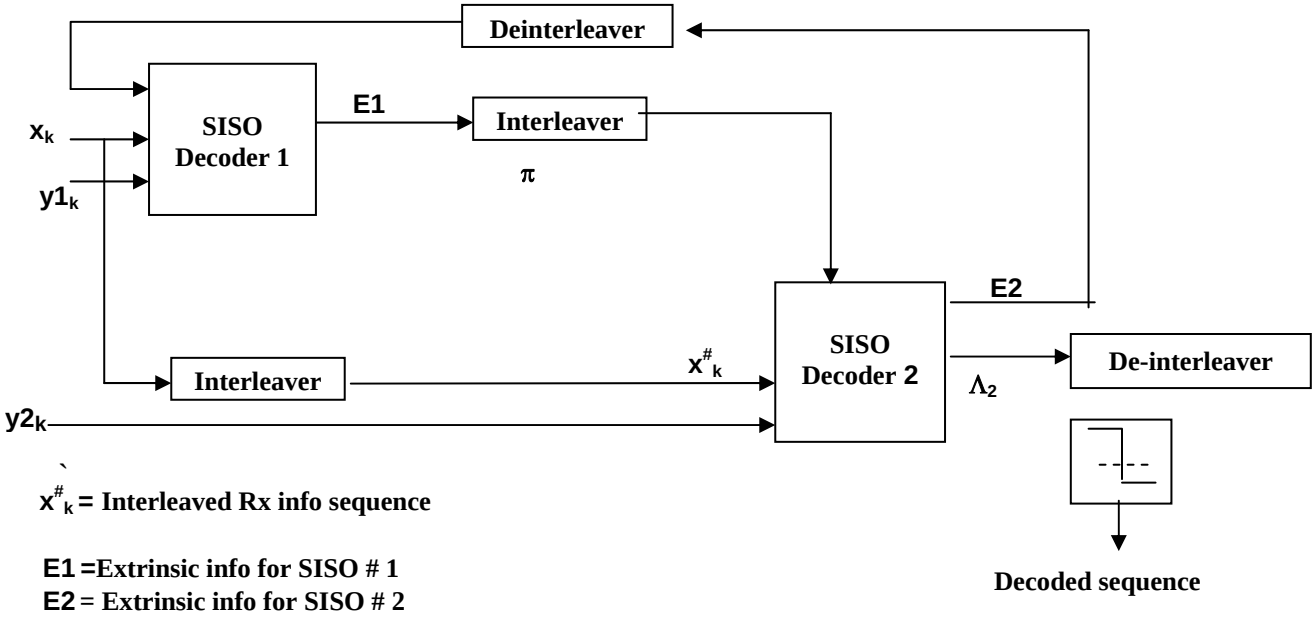

Fig. 6.37.2. *Block diagram of iterative turbo decoder*

The MAP algorithm is a Maximum Likelihood (ML) algorithm and the SOVA is asymptotically an ML algorithm at moderate and high SNR. The MAP algorithm finds the most probable information bit that was transmitted, while the SOVA finds the most probable information sequence to have been transmitted given the code sequence. That means the MAP algorithm minimizes the bit or symbol error probability, where as SOVA minimizes the word error probability. Information bits returned by the MAP algorithm need not form a connected path through the trellis while for SOVA it will be a connected path.

However the MAP algorithm is not easily implement able due to its complexities. Several approximations on the MAP algorithm are now available, such as the Max-Log-MAP algorithm where computations are largely in the logarithmic domain and hence values and operations are easier to implement. The Log-MAP algorithm avoids the approximations in the Max-Log-MAP algorithm through the use of a simple correction function at each maximization operation and thus its performance is close to that of the MAP algorithm.

A complexity comparison of different decoding methods per unit time for (n,k) convolutional code with memory order v is given in **Table 6.37.1**. Assuming that one table-look-up operation is equivalent to one addition, one may see that the Log-MAP algorithm is about three times complex than the SOVA algorithm and the Max-Log-MAP algorithm is about twice as complex as the SOVA algorithm.

	MAP	**Log-MAP**	**Max-Log-MAP**	**SOVA**
Additions	$2.2^k.2^v+6$	$6.2^k.2^v+6$	$4.2^k.2^v+8$	$2^k.2^v+9$
Multiplications	$5.2^k.2^v+8$	$2^k.2^v$	$2.2^k.2^v$	$2^k.2^v$
Max.operations		4.2^v-2	4.2^v-2	2.2^v-1
Look-ups		4.2^v-2		
Exponentiation	$2.2^k.2^v$			

Table 6.37.1 *Complexity comparison of various decoding algorithms*

Applications

Turbo codes have been proposed for various communication systems such as deep space, cellular mobile and satellite communication networks. Turbo code, due to its excellent error correcting capability, has been adopted by several standards as Consultative Committee for Space Data Systems (CCSDS) for space communication, CDMA2000, UMTS, 3GPP, W-CDMA for cellular mobile, DVB-RCS *(Digital Video Broadcasting – [with a] Return Channel [through] Satellite)* for Satellite communications.

Turbo Product Codes (TPC)

A Turbo Product Code is a concatenation of two block codes on which the principle of "turbo" or iterative soft-input/soft-output (SISO) decoding can be applied in a phased manner. The decoding process is iterated several times feeding the output of second component decoder back to the input of the first decoder. A product code may be viewed as a relatively large code built from smaller block codes. A two-dimensional product code is built from two component codes with parameters $C_1(n_1, k_1, d_1)$ and C_2 with parameter (n_2, k_2, d_2), where n_i, k_i and d_i stand for code word length, number of information bits, and minimum hamming distance respectively.

The product code $P = C_1 \text{ X } C_2$ is obtained by placing (k_1 X k_2) information bits in an array of K_1 rows and K_2 columns. The parameters of product code P are $n = n_1 X n_2$, $k = k_1 X k_2$, $d = d_1 X d_2$ and code rate is $R = R_1 X R_2$, where R_i is the code rate of C_i. Thus, very long block codes can be built with large minimum Hamming distance. **Fig. 6.37.3** shows the procedure for construction of a 2D product code using two block codes C1 and C2 .

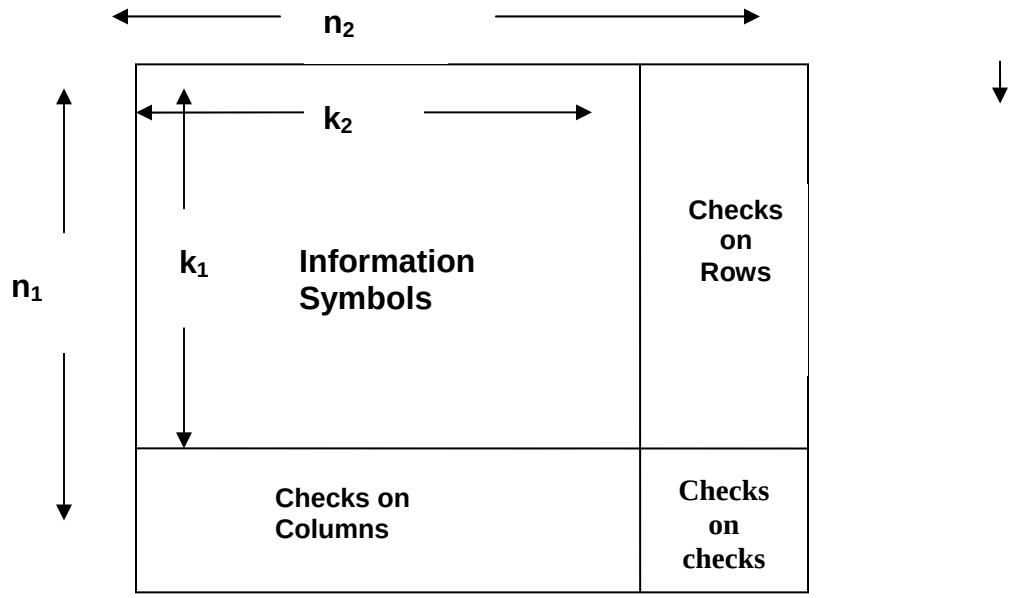

Fig. 6.37.3 *An example of a 2D product code constructed using two component codes*

Source:http://nptel.ac.in/courses/Webcourse-contents/IIT%20Kharagpur/Digi%20Comm/pdf-m-6/m6l37.pdf