TURBO CODE PERFORMANCE AND DESIGN TRADE-OFFS

Raffi Achiba
Mehrnaz Mortazavi

Booz·Allen & Hamilton, Inc
McLean, Virginia

William Fizell

MILSATCOM JTEO
Falls Church, Virginia

ABSTRACT

Turbo codes were first proposed by Berrou and Glavieux in 1993, and shown to have a near Shannon limit error correction capability. Since then, turbo codes have become the focus of research and study among the coding community. Turbo codes are particularly attractive to higher data rate applications where the additional coding gain is necessary to maintain the link performance level with limited power. For instance, the Advanced EHF satellite system is a candidate for implementing turbo codes, which offer a superior performance compared to convolutional codes currently used in the Milstar system.

This paper presents a survey of turbo coding designs based on existing research journals and publications. It investigates the key design parameters for each coding scheme, such as choice of component codes, memory size, interleaver size, and the number of decoding iterations. In addition, it examines the trade-offs between improvement in code performance, and the overall delay and the computational complexity of the coding algorithm. This paper also presents Bit Error Rate (BER) performance comparisons between different turbo code designs, both in additive white Gaussian and Rayleigh fading environments.

1. INTRODUCTION

Turbo codes consist of concatenation of two or more component codes separated by interleavers. These component codes can be either block codes or convolutional codes. The initially introduced turbo codes are parallel concatenated convolutional codes (PCCC) whose encoder is formed by parallel concatenation of two recursive systematic convolutional codes joined by an interleaver. Later, a new class of turbo codes were introduced, the serial concatenated convolutional codes (SCCC) and the hybrid concatenated convolutional codes (HCCC).

There are a number of design parameters involved in determining the performance of turbo codes. In order to select a turbo coding scheme that offers the best performance for a specific system, a thorough study of the key design parameters and their effects on code performance is essential.

The comparisons and analysis presented in this paper introduce the reader to the most common types of turbo codes, as well as the basic turbo code design parameters.

2. PARALLEL CONCATENATED CONVOLUTIONAL CODES (PCCC)

Although the term "turbo codes" is used to refer to a wide variety of concatenated coding schemes, it once referred solely to a parallel concatenation of two constituent codes separated by an interleaver as shown below.

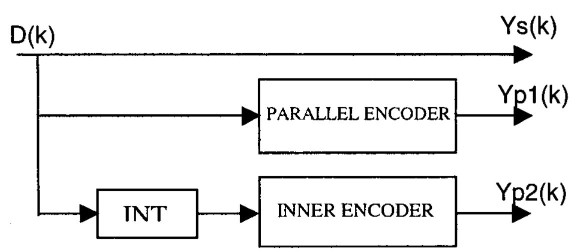

Figure 1. Block Diagram of a Rate 1/3 Parallel Concatenated Convolutional Code (PCCC)

In Figure 1, a rate 1/3 turbo code is obtained by parallel concatenation of two rate 1/2 recursive systematic convolutional (RSC) codes separated by a pseudo-random interleaver and is referred to as a parallel concatenated convolutional code (PCCC). For every information bit, D(k), the PCCC outputs a three-bit code word that consists of the systematic bit, Ys(k), followed by the parity bits, Yp1(k) and Yp2(k), generated by the two RSC encoders. The systematic bit of the inner code is not transmitted. Figure 2 shows the BER performance of a rate 1/3 turbo code and illustrates its superior performance compared to a rate 1/3 convolutional code with constraint length K=9.

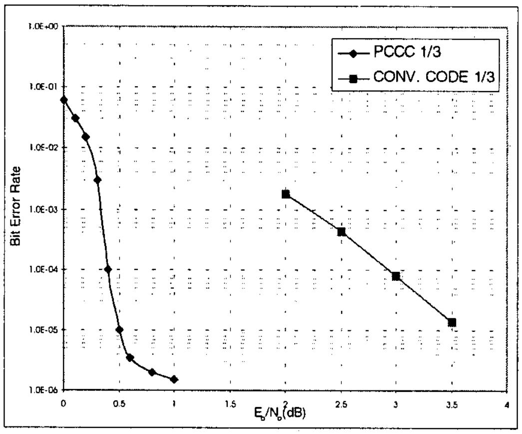

Figure 2. BER Performance of a Rate 1/3 Convolutional Code with K=9 and a Rate 1/3 PCCC with Two Sixteen-State RC Encoders and Interleaver Delay of 16,384 after 9 Iterations

The performance of a turbo code depends on several parameters, including the choice of component codes and interleaver type.

When convolutional codes are used alone the choice of code is almost exclusively non-recursive (NRC). When they are a part of a turbo code design, however, the use of RSC codes can be crucial.

2.1 RECURSIVE SYSTEMATIC CONVOLUTIONAL (RSC) CODES

An RSC code has a feedback path that adds the content of the shift register to the input bit. In addition, the first output bit out of the encoder is called the systematic bit because it is simply the input bit. An RSC code has an infinite impulse response, while the NRC code has a finite impulse response. As a result, recursive systematic and non-recursive convolutional codes have different minimum weight input, Wm, values. Wm is the minimum weight of a finite input sequence that could generate an error sequence of length 2(M+1) in the resulting codeword ([3], p.8). The weight of a sequence is the number of non-zero bits, and M is the number of shift registers in the encoder. The minimum weight input, Wm, is equal to 1 for NRC codes and 2 for RSC codes. The value of the minimum weight input is very important because the upper bound of the probability of error for a turbo code with interleaver length N is proportional to the term $N^{(1-Wm)}$ ([3], p.8). So, for NRC codes the upper bound of the probability of error will be proportional to 1, while for RSC codes it is proportional to N^{-1}.

Since turbo codes are block codes, it is necessary to flush the encoder at the beginning of each block, to prevent errors from carrying from one block to the next. RSC codes, however, are more complicated to flush than NRC codes. For NRC codes a sequence of M zeros is used to bring the encoder to an all-zero state, whereas, for RSC codes, a sequence of bits, called the tail sequence, which depends on the state of the encoder, is generated each time a block of data is coded.

2.2 INTERLEAVER

The interleaver in a turbo code scrambles the bits in each block of data before it enters the second encoder, so that the inputs to the two encoders are not correlated. The decoder also assumes that the inputs to the two component encoders are not correlated. By de-coupling the inputs to the two encoders, the interleaver provides a good codeword weight distribution, which improves the decoder performance.

Turbo codes use an iterative decoding algorithm, where the BER performance improves after each iteration. The performance of the iterative decoder depends on both the size and the strength of the interleaver. For a given set of component codes, the turbo code with a longer interleaver has a better performance. Longer interleavers are used for higher data rates where the resulting latency is tolerable. For lower data rates, however, long interleavers introduce long delays which are not desirable. Similarly, for the same set of component codes and interleaver size, the stronger the interleaver the better the performance.

The S-random interleaver is a random interleaver that maps any two bits, separated by a distance less or equal to S, to new locations with a separation distance greater than S, where S is an integer. Moreover, the spreading pattern is randomly generated, and thus, the permutation pattern is stored in a table, both in the encoder and the decoder sides. For a large block size, however, the search for an S-random interleaver can take a significant amount of time.

The turbo interleaver is another strong interleaver that yields good results and does not use a look up table. In stead, the turbo interleaver uses a mapping algorithm that is easy to implement, as shown below [6]. The turbo interleaver algorithm works for large block size, N, such that

$$N = 2^n \times 2^n, \qquad n \geq 3.$$

Each bit with row and column indices i and j is mapped to a new location of coordinates i' and j' derived as follows:

$$i' = (2^{n-1} + 1) \cdot (i + j) \quad \bmod 2^n$$
$$\zeta = i + j \quad \bmod 8$$
$$j = (j + 1) \cdot P(\zeta) - 1 \quad \bmod 2^n$$
$$P(\zeta) = \{17, 37, 19, 29, 41, 23, 13, 7\} \qquad 0 \leq \zeta \leq 7$$

2.3 CONSTRAINT LENGTH K

One important measure in designing convolutional codes is the constraint length, K, which is the maximum number of bits in a single output stream that can be affected by any input bit. In general, the constraint length is taken to be the length of the longest input shift register plus one, K = M+1. Component codes with different constraint lengths (K) produce different results. For example, given the same set of parameters, a 16-state (K=5) turbo code has a better performance than a 4-state (K=3) turbo code. Figure 3 demonstrates how the BER performance curves change for different number of states ranging from 2 to 32. Moreover, increasing the memory size does not affect the decoding delay. However, the computational complexity increases, and thus, the implementation becomes more expensive.

2.4 NUMBER OF ITERATIONS

The superior performance of turbo codes can be achieved by a relatively simple iterative decoding algorithm. The number of iterations used in the decoding algorithm affects the turbo code performance. As the number of iterations increases the decoder performance improves.

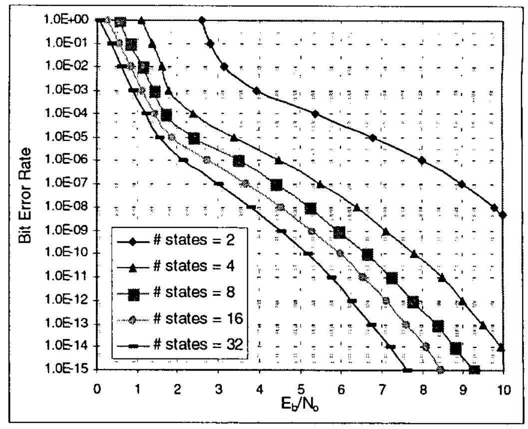

Figure 3. BER Performance of Rate 1/3 PCCCs with Different Memory Sizes and Interleaver Delay of 100 (Courtesy of [2])

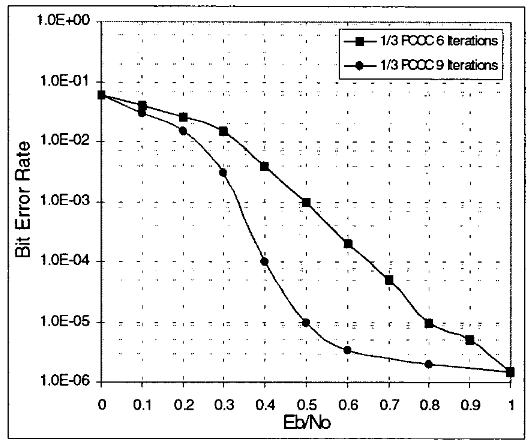

Figure 4. BER Performance of a Rate 1/3 PCCC for 6 and 9 Iterations with two 4-State RC Encoders and Interleaver Delay of 16,384 (Courtesy of [1])

However, this performance improvement is limited by the interleaver length and strength. For a given interleaver length, the performance gain becomes negligible after a certain number of iterations. Figure 4 shows the BER performance of a rate 1/3 PCCC for six and nine iterations.

As shown in figure 2, the slope of the bit error probability curve for the PCCC is relatively high for low E_b/N_o values but decreases considerably at higher E_b/N_o values where increasing the signal to noise ratio does not impact the bit error rate significantly. This phenomenon is referred to as the "floor" behavior, which is typical of PCCCs. The bit error rate value at which the "floor" behavior appears varies among different PCCC designs. For example, longer interleavers or lower code rates cause the "floor" behavior to appear at a lower bit error rate.

As was mentioned earlier, increasing the interleaver size improves the performance of a PCCC. The improvement in the BER performance as the interleaver size increases is referred to as the interleaver gain. In order to maximize the interleaver gain, a PCCC must use recursive convolutional encoders as component codes ([2], p.1) ([5], p.14). An RSC code is described by its generator polynomial. For a given number of shift registers in an RSC, the PCCC code performance can be optimized when the feed forward polynomial is of degree equal to the number of shift registers ([2], p.9). The PCCC, whose BER performance curve is shown in Figure 2, follows these two design guidelines; it consists of RSC component codes with four shift registers whose feed forward polynomial is of degree four.

A rate 1/2 PCCC can be generated with the same constituent codes used to construct a rate 1/3 PCCC. For instance, in the PCCC example of Figure 1, a rate 1/2 turbo code can be obtained by alternating between the parity bits of the first and second encoders and hence transmitting only one parity bit per code word. The process is referred to as puncturing.

3. SERIAL CONCATENATED CONVOLUTIONAL CODES (SCCC)

After the breakthrough discovery of turbo codes in 1993, several attempts were made to apply the iterative decoding to serial concatenation of convolutional codes. In July 1996, analytical and simulation results on performance of serial concatenated convolutional codes (SCCC) were finally published [3].

An SCCC consists of two cascaded component codes, an outer code and an inner code that are linked together by an interleaver that de-couples the output of the outer encoder from the input of the inner encoder.

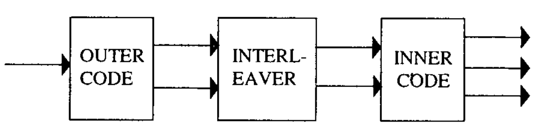

Figure 5. Block Diagram of a Rate 1/3 Serial Concatenated Convolutional Code (SCCC)

Figure 5 shows a rate 1/3 SCCC that consists of an outer code that is a rate 1/2 non-recursive convolutional (NRC) code, an inner code that is a rate 2/3 recursive convolutional (RC) code, and an interleaver of size N=16,384 that connects the two encoders. Unlike PCCCs, the interleaver of an SCCC operates on coded bits of the outer encoder. Therefore, for the same size interleaver, the interleaver delay in terms of input bits for an SCCC is reduced by R^{o}_{c} = outer code rate. In the above SCCC example, for an interleaver size of N and outer code rate of 1/2 , the interleaver delay in terms of input bits is reduced to N/2. Likewise, for the same interleaver delay, an SCCC can benefit from a larger size interleaver compared to a PCCC .

As in a PCCC, increasing the interleaver size improves the performance of an SCCC. In order for an SCCC to yield an interleaver gain, a recursive inner encoder must be used ([3], p.12). In addition, an NRC outer code is necessary to maximize this interleaver gain, since the interleaver gain of an SCCC depends on the outer code free distance, df_o. The interleaver gain (IG) as a function of the interleaver size and df_o can be computed as follows ([3], p.14):

$IG = R^{-(dfo/2)}$, when dfo is even,
$IG = R^{-[(dfo+1)/2]}$, when dfo is odd.
R = N2/N1 is the factor by which the interleaver size is increased.

Figure 6 shows the BER performance of a rate 1/3 SCCC and compares it to the performance of a rate 1/3 PCCC. The PCCC uses two rate 1/2 RSC codes, and the SCCC uses a rate 1/2 RSC outer code and a rate 2/3 RSC inner code. The interleaver delay, in terms of input bits, is 16,384 for both turbo codes. Both use 4-state component codes. The iterative decoding algorithm for both turbo codes uses nine iterations.

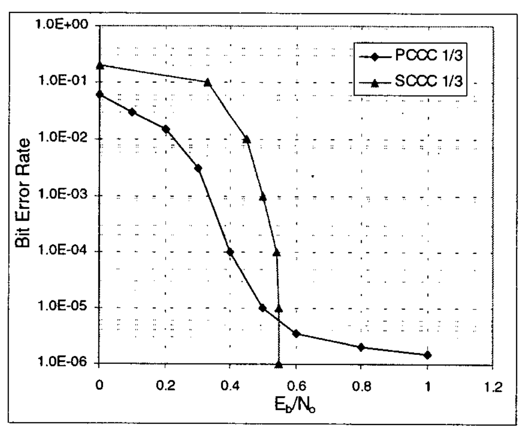

Figure 6. BER Performance of a Rate 1/3 PCCC and a Rate 1/3 SCCC with two 4-State RC Encoders and Interleaver Delay of 16,384 after 9 Iterations (Courtesy of [1])

As the BER performance curves show, the PCCC performs better that the SCCC for BER values above 10^{-5}. Their performance at 10^{-5} is comparable. For BER values below 10^{-5}, the SCCC behaves significantly better and does not manifest the "floor" behavior typical of PCCCs. At 10^{-6}, for instance, the SCCC has an advantage of at least 0.5 dB over the PCCC.

As was mentioned earlier, the "floor" behavior of PCCCs appears at a lower bit error rate for longer interleavers. The bit error rate at which the "floor" behavior of a PCCC starts affects the performance of a PCCC compared to an SCCC. Figure 7 shows a comparison of the performance of a rate 1/3 PCCC to a rate 1/3 SCCC for a smaller interleaver delay of 1024. Both turbo codes use the same component codes as in the previous example, but they experience a much smaller interleaver delay of 1024, compared to 16,384. The results are shown for seven decoding iterations.

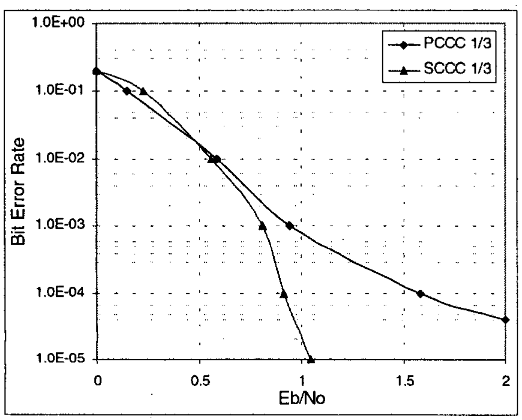

Figure 7. BER Performance of a Rate 1/3 PCCC and a Rate 1/3 SCCC with two 4-State RC Encoders and Interleaver Delay of 1024 after 7 Iterations (Courtesy of [3])

Figure 7 shows that for a smaller interleaver delay of 1024, the "floor" behavior of the PCCC starts at a much higher BER value of 10^{-3}, compared to 10^{-5} when the interleaver delay is 16,384. As a result the range of BER values where the PCCC outperforms the SCCC is reduced to values above 10^{-2}. For BER values lower than 10^{-2} the SCCC performs better than the PCCC and does not exhibit the "floor" behavior typical of PCCCs.

When comparing serial and parallel concatenated convolutional codes, the following general statement can be made for the specific encoder configurations examined: PCCCs perform better for high BER values while SCCCs perform extremely well at low BER values. The BER value at which both turbo codes have comparable BER performance varies depending on the interleaver size.

A higher rate SCCC can be constructed using a different set of component codes. Figure 8 shows the BER performance of a rate 1/2 SCCC compared to a rate 1/2 PCCC. The rate 1/2 SCCC is formed by using a rate 1/2 four-state NRC outer code and a rate 1 two-state RC inner code. The rate 1/2 PCCC uses two identical rate 2/3 RSC codes. The interleaver delay in terms of input bits is 256 for both turbo codes.

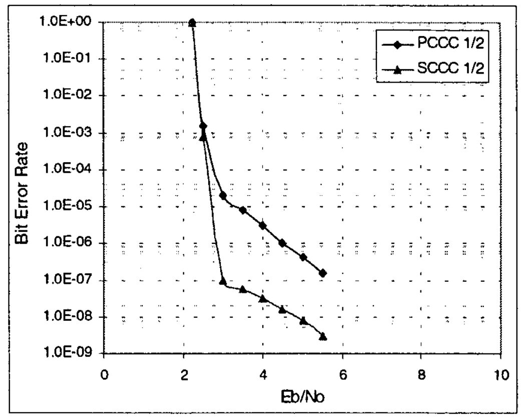

Figure 8. BER Performance of a Rate 1/2 PCCC and a Rate 1/2 SCCC in AWGN with Interleaver Size of 256 (Courtesy of [4])

Figure 8 shows that for an interleaver delay of 256 and higher code rate of 1/2, the PCCC does not outperform the SCCC, even at high BER values. In addition, for lower BER values, the SCCC provides a significant gain over the PCCC. For example, at 10^{-7} the SCCC has an advantage of at least 2 dB over the PCCC. Figure 8 reveals the "floor" behavior of both parallel and serial turbo codes, however, in the case of the SCCC, this phenomenon appears at a much lower bit error rate.

The decoding algorithm used in this example is the maximum likelihood (ML) decoding algorithm. However, for larger interleaver sizes the ML decoding algorithm becomes very complex and physically unrealizable ([5], p.1). For long interleavers, a sub-optimum iterative decoding algorithm that is based on a maximum a posteriori (MAP) decoder is implemented. The iterative decoding technique approaches the ML performance bound as the number of iterations increases ([5], p.16).

As was illustrated by several simulation results [3][4], SCCCs outperform PCCCs at low BER values. At higher BER values PCCCs perform better than SCCCs by a few tenths of a dB. Since the "floor" behavior typical of turbo codes appears at a higher BER value for PCCCs than for SCCCs, the additional coding gain offered by SCCCs consistently improves for a longer range of BER values. Furthermore, the maximum interleaver gain PCCCs can provide is R^{-1}, where R is the factor by which the interleaver size is increased. On the other hand, the interleaver gain for SCCCs can be as high as R^{-3} or R^{-4} for higher E_b/N_0 values depending on the value of df_o.

4. HYBRID CONCATENATED CONVOLUTIONAL CODES (HCCC)

A Hybrid Concatenated Convolutional Code (HCCC) is a generalized system that combines PCCCs and SCCCs. An example of an HCCC is shown in Figure 9. Without the upper branch, the HCCC shown below becomes an SCCC. Similarly, without the outer code, it becomes a PCCC. If both the parallel and the outer codes are of rate 1/2, and the inner code of rate 2/3, then the overall turbo code rate is 1/4. The systematic bit of the parallel code is usually not transmitted.

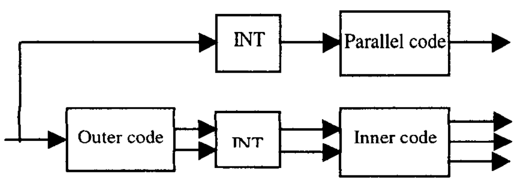

Figure 9. Block diagram of a rate 1/4 Hybrid Concatenated Convolutional Code (HCCC).

The design criteria for HCCCs, in terms of the type of component codes used, remains the same as for PCCCs and SCCCs. The HCCC BER performance is optimum when the parallel and inner codes are recursive to ensure that the minimum weight input, Wm, is equal to 2, and the outer code is non-recursive to guarantee that the outer code free distance, dfo, is larger than 1 ([4], p.7).

The drawback to HCCCs, however, is that they introduce a considerable amount of delay. First, HCCCs are known for their low coding rates, 1/4 at most. Second, because HCCCs use more than two encoders and one interleaver, the decoding delay is significant. Therefore, HCCCs are ideal for extremely high data rates, where the resulting delay is tolerable.

5. TURBO CODE PERFORMANCE IN A FADING ENVIRONMENT

In order to address the end-to-end link performance requirements in a hostile environment (i.e., jamming, scintillation), the performance of turbo codes in the presence of jamming and scintillation must be studied. Figure 10 shows the simulation results on the performance of a rate 1/2 PCCC and a rate 1/2 SCCC in a Rayleigh fading environment. This example uses the same turbo codes as shown in Figure 8.

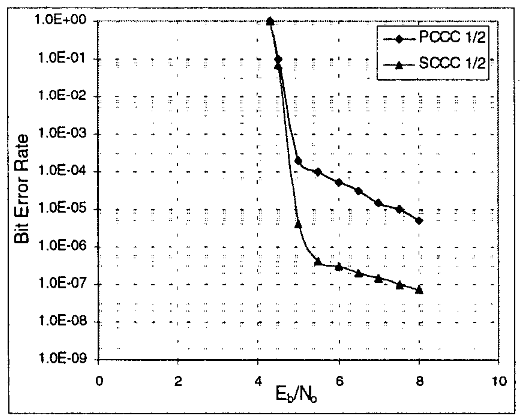

Figure 10. BER Performance of a Rate 1/2 PCCC and a Rate 1/2 SCCC in a Fading Environment with Interleaver Size of 256 (Courtesy of [4])

Figure 10 shows that the SCCC maintains its superior performance over the PCCC in a Rayleigh fading environment.

The bit error rate performance of turbo code degrades in a Rayleigh fading environment. Figure 11 demonstrates a degradation of at least 2 dB in the performance of a rate 1/2 SCCC when the environment is changed from AWGN to Rayleigh Fading. The channel interleaver can improve the turbo code performance in a fading environment.

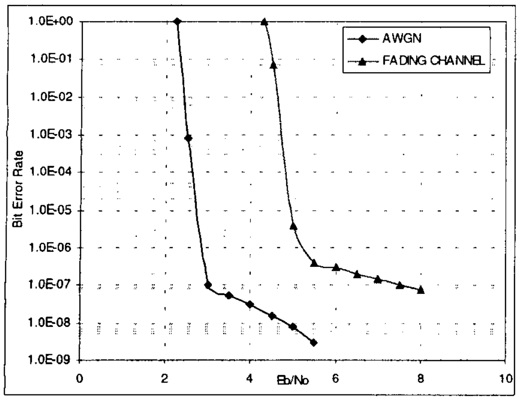

Figure 11. BER Performance of a Rate 1/2 SCCC in AWGN and Fading Channels with Interleaver Size of 256 (Courtesy of [4])

6. SUMMARY

Various independent studies and simulations have shown that turbo codes achieve near Shannon limit BER performances. In fact, turbo codes have a much stronger error correcting capability than convolutional codes or any other Block codes.

The turbo decoder employs an iterative decoding algorithm that is of relatively low complexity, and can easily be implemented.

The performance of a turbo code depends on the memory size, generating polynomials, number of decoding iterations, interleaver type, and interleaver size. Increasing the interleaver size improves the BER performance at the cost of a larger overall delay. On the other hand, increasing the memory size improves the BER at the cost of greater computational complexity. Moreover, the method of concatenating the component codes affects the turbo code performance. For example, for a given interleaver delay and depending on the desired BER value, serial concatenation could yield a better performance than parallel concatenation.

Generally, the choice of turbo code will depend on the data rate, desired range of BER values, and the expected noise environment.

Finally, although simulation results have shown that SCCCs are less susceptible to fading environment than PCCCs, it is necessary in both cases to use a channel interleaver to maintain the turbo code performance over a fading channel.

7. REFERENCES

[1] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-Input Soft-Output Maximum A Posteriori (MAP) Module to Decode Parallel and Serial Concatenated Codes," Proceedings of ICC'96, Dallas, Texas, June 1996.

[2] S. Benedetto, G. Montorsi, "Design of Parallel Concatenated Convolutional Codes," IEEE Transactions on Communications, Vol. 44, No. 5 May 1996.

[3] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial Concatenation of Interleaved Codes: Performance Analysis , Design, and Iterative Decoding," The TDA Progress Report 42-126, Jet Propulsion Laboratory, Pasadena, CA, Aug. 15, 1996.

[4] D. Divsalar, and F. Pollara, "Serial and Hybrid Concatenated Codes with Applications."

[5] S. Benedetto, and G. Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," IEEE Transactions on information Theory, Vol. 42, No. 2, March 1996.

[6] V. Kasemsri, "Simulation of Turbo Codes for the Advanced EHF SATCOM System," MITRE, Center for Air Force C2 Systems, Bedford, Massachusetts, September 1998.