# HELICAL INTERLEAVERS

Danny T. Chi
Member of Technical Staff
Kodak Berkeley Research

**Abstract**

Interleaving is a simple and effective way to improve the performance of an error correction scheme on a bursty channel. The interleaving of codewords will spread the effects of a long burst error into short bursts over several encoded sequences instead of a single codeword, and thus the chosen error correction scheme can correct them. This paper addresses a recently developed method, called helical interleaving ([1]) and presents some of its applications. The advantages of helical interleavers as compared with traditional interleavers are discussed. A relationship between helical interleavers and convolutional interleavers is also presented.

**Introduction**

A practical technique to cope with burst errors is that of using random-error-correcting codes in connection with a suitable interleaver/deinterleaver pair. An interleaver is a device that rearranges the ordering of a sequence of symbols in a deterministic manner. The deinterleaver applies the inverse operation to restore the sequence to its original ordering.

Two types of interleavers ([2]) are commonly used. Block interleavers accept symbols in blocks and perform identical permutations over each block of symbols. Convolutional interleavers have no fixed block structure, but they perform a periodic permutation over a sequence of coded symbols. More recently, Berlekamp and Tong developed a new class of interleavers, the helical interleavers which have certain advantages over block interleavers. We will briefly describe block interleavers and convolutional interleavers and then concentrate on helical interleavers and burst-forecasting strategies. Finally, we will present a relationship between the helical interleavers and the convolutional interleavers as the main result of this paper.

**1. Block Interleavers.**

The typical case of an (N,I) block interleaver involves taking the coded symbols and writing them by columns into a matrix with N rows and I columns. Their permutation consists of reading these symbols out of the matrix by rows prior to transmission. The deinterleaver simply performs the inverse operation. Symbols are written into the deinterleaver by rows and read out by columns. Block interleavers are best suited for block codes (especially for Reed-Solomon codes, see [3].). For example, a (n, k) cyclic code, interleaved to depth I, can be obtained by arranging I code words of the original code into I rows of a rectangular array that will be transmitted by columns. The parameter I is called the interleaving depth of the code. If the original code corrects up to t random errors, the interleaved code will have the same random-error-correction capability, but in addition it will be able to correct all bursts of length about I x t.

In general, the depth of an interleaver is defined as one less than the shortest burst length which can hit any codeword twice. The total interleaving delay, including transmitter's interleaver delay and receiver's interleaving delay, is 2(N-1)(I-1) ≈ 2NI and the memory requirement is NI.

The use of a block interleaver introduces essentially the same synchronization problem that the use of a block code presents. Synchronization of the interleaver/deinterleaver can be accomplished by using standard frame synchronization techniques. In this case, a sync word (or more than one) with good correlation properties is inserted periodically at the interleaver. These sync words could be either outside the codeword or inside the codeword. An alternate approach (which requires no additional overhead symbols) is to watch the performance of the decoder. For block codes, consecutive cases of undecodable codeblocks could be a good reason to start a synchronization search process. The synchronization ambiguity of block interleavers is of degree NI.

**2. Convolutional Interleavers.**

Figure 1 depicts an example of a (M , MJ) convolutional interleaver with M=4 and J=l, proposed by Ramsey ([4]) and Forney ([5]). The code symbols are shifted sequentially into a bank of M-1 delay lines where the i-th delay line delays the input symbol i x J time units (Figure 1). With each new code symbol coming, the wiper switches to the next delay line. The input and output wipers should operate synchronously. The deinterleaver performs the inverse operation.

M is chosen to be larger than the length of the burst errors. The total interleaving delay is M(M-1)J and the memory required is M(M-1)J/2. The delay and memory requirement are only half of that of a (M , MJ) block interleaver which can correct bursty errors of the same length.

Another advantage of convolutional interleavers over block interleavers is the deinterleaving synchronization ambiguity. The synchronization ambiguity for convolutional interleavers is only of degree M while it is of degree M(M-1)J for the corresponding block interleavers. Convolutional interleavers are best suited for convolutional codes. A (30,120) convolutional interleaver is used by the Tracking and Data Relay satellite system (TDRSS) which uses the followed Viterbi decoder to achieve deinterleaving synchronization. The basic idea is that if the deinterleaver is out of sync, then all path metrics generated by the Viterbi decoder will tend to remain relatively close together and this condition could easily be detected([6]).

## 3. Helical Interleaving

We explain helical interleavers through the following example of interleaving a code with a block length of four symbols. The numbering reflects the time ordering of the symbols at the interleaver. The codewords are written down the columns of a helical array as shown in Figure 2. When complete, the rows of the array are read out and transmitted. This is an example of a code of length 4 with helical interleaving of depth 3. The interleaving depth 3 means that a burst must have length at least 4 in order to strike any code block more than once. Note that consecutive symbols of a codeword are separated by precisely four symbols. Thus the receiver only has to know synchronization modulo 4. The deinterleaving process is the inverse operation of the interleaving process. In fact, the entire interleaver and deinterleaver process is symmetric. The above helical interleaver construction method can be generalized to any block length n with interleaving depth n-1.

The total interleaving delay is $(N-2)(N-1)+2 \approx (N-2)(N-1)$ and the memory requirement is $N(N-1)/2$; both are only about one half of the corresponding parameters for a block interleaver of equal depth.

The conventional depth (n-1) block interleaver for a code with length n requires the receiver to know the synchronization modulo n(n-1). One major advantage of the helical interleaver is that, due to helical symmetry, it requires the receiver only to know synchronization modulo n. Also, the helical interleaver has RAM size (n*(n-l)/2) which is only about half the size (n*(n- 1)) of the corresponding block interleaver of the same depth ([7]). We will refer to the above interleaving strategy as the N-order helical interleaving of the first type.

### 3.1 Variable-depth Helical Interleaving

We start by giving a simple example ([8]). Figure 3 depicts the interleaver timing for a code of length 8 helically interleaved to depth 5. (5 is not a devisor of 8 - 1 = 7.) Each of the 8 x 5 = 40 squares in Figure 4 represents a location in the interleaver's memory. The lower number inside a square specifies the time during which data is read from the location. Note that the upper numbers are just sequential across the rows. This means that data is read from the memory row by row. The ordering of the lower set of numbers is more involved. From time 0 to time 7, the first codeword is written into the 8 locations specified by the numbers in column 0, starting at row 0. From time 8 to time 15, the second codeword is written into the 8 locations specified in column 3, starting at row 1. From time 15 to time 23, the third codeword is written into the 8 locations specified in column 1, starting at row 3, and so on. The general rule is that the first character of a codeword should be written in and read from the same location at the same time, thus experiencing essentially zero delay. The second character of a codeword is delayed by 4 time units, the third one is delayed by 8 time units and so on. Note that any one of the 5 codewords in the 5 columns in Figure 3 has the same sequence of delay values, namely 0,4,8,...,24,28. This means that as far as the deinterleaver is concerned, all codewords are symmetric. This in turn implies that it is only necessary to acquire synchronization modulo 8 (the length of the code). This property is extremely useful for deinterleaving synchronization if we replace, for example, the last symbol in every code by a fixed synchronization symbol ([9]).

### 3.2 Restriction on Depth of Interleaving

In the above example, the first codeword is written in column 0, the second one in column 8 modulo 5 = 3, the third in column 2 x 8 (modulo 5) = 1, and so on. In general, for a codeword of length n helically interleaved to depth d, the k-th codeword is written in column (k-1)n modulo d. In order to go through all the columns in the memory before repetitions occur, d has to be relatively prime to n. This is the only restriction on the depth of helical interleaving. We also note that the helical structure is no longer preserved under this method. We will refer to this interleaving strategy as the helical interleaving of the second type.

## 4. Burst forecasting:

Burst forecasting ([1],[7])becomes easy if we restrict the forecasting to the rather near future and maintain a good record of recent past and the present. For example, predicting the weather with rather high accuracy is not difficult if we only attempt to predict the weather ten minutes from now. One simple prediction algorithm is to predict that it will be equal to the current weather. Similarly, one burst forecasting algorithm would only predict the channel behavior one symbol into the future.

We now describe a sophisticated helical interleaving-forecasting scheme for Reed-solomon codes. The decoding scheme is recursive. Assuming all words to the i-th word $C_i$ have been processed, the decoder first attempts to decode $C_i$ without reference to the previously decoded codewords. If the decoder succeeds in this “first pass,” it declares $C_i$ to be successfully decoded, and goes on to $C_{i+1}$. Otherwise, it goes back to $C_{i-1}$. If $C_{i-1}$

was not decoded successfully, the decoder gives up, declares $C_i$ to have been unsuccessfully decoded, and goes on to $C_{i+1}$. However, if $C_{i-1}$ was decoded successfully, it tries again to decode $C_i$, as follows. For each symbol error corrected in $C_{i-1}$, the decoder assigns a numerical "reliability" to the corresponding symbol, i.e., the symbol next transmitted over the channel, in $C_i$. The decoder makes several more attempts to decode $C_i$, by erasing various subsets of the potential erasure set[10]. If any one of these attempts succeeds, the decoder accepts it, and declares $C_i$ to have been successfully decoded. Otherwise, it declares $C_i$ to have been unsuccessfully decoded.

Kodak's model 888 decoder implements an 8-error correcting Reed-Solomon code RS(254,238) over 8-bit symbols, helically interleaved to depth 8 and with or without burst-forecasting strategies. When the bursts are relatively long (length=8), it shows that the performance improves by a factor nearly 18,000 if the burst-forecasting strategies are used.

Burst-forecasting algorithm can also be applied to block codes. Analysis results for the above burst-forecasting strategies show that helical interleaving does not enjoy a significant coding gain over block interleaving ([11]). Therefore, synchronization and memory advantages seem to be the two areas for which the helical interleaved clearly outperforms block interleavers.

## 5. A relationship between helical interleavers and convolutional interleavers.

We first present an input-output function for helical interleavers. Suppose the helical interleaver works on N-bit and the k-th output of the helical interleaver is expressed as H(k). The H(k) can be calculated as follows (see Figure 2):

step 1. We first count the index k as multiples of N(N-1):

Let $K = qN(N-1) + m$, where $0 < m \leq N(N+1)$.

step 2. We next count the index m as multiples of N-1:
Let $m = r(N-1) + s$, where $0 < s \leq N-1$.

Then

if $0 < s < r+2$,
$H(k) = k+(s-1)N+r-m-s+2.$

else,
$H(k) = (q-1)N+(1+s-q)N-s+r+2.$

For example let's compute H(7) for the last example:

7 = 0 x 12+7, thus q=0 and m=7. Next, the index m, as found in step 1, can be expressed as 7 = 2 x 3+1, thus r=2 and s=1. Now, s is less than r+2 and therefore, H(7) = 7+(1-1)*4+2-7-1+2 = 3.

We next present an input-output function for a M,MJ convolutional interleavers.The k-th output C(k) can be calculated as follows:

step 1. Divide k by J:

Let $k = qJ+m$, where $0 < rn \leq J$.

Then $C(k) = k-(m-1)MJ$.

If M=N-1 and J=1, then $C(k) = k - (m-1)(N-1)$. (2)

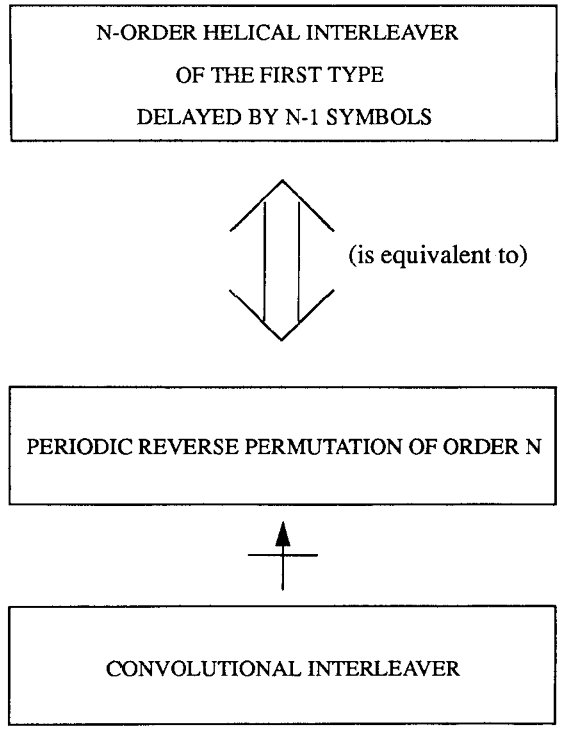
With some algebraic manipulation, we can show that (1) can be simplified to

$H(k) = k+N-m - (N-m)(N-1)$. (3)

Compare (3) with (2), if we replace k with k+N-1 in equation (3) and replace m with N-m+1 in (4); then H(k)=C(k) for all k. The replacement of k by k+N-1 means a simple delay and the replacement of k by k+N-1 could be realized by a so-called periodic reverse permutation circuit. For example, a periodic reverse permutation of order 4 will reverse the ordering of a data stream in groups of 4 elements as follows:

. . ., a,b,c,d, e,f,g,h, . . . ------------> . . . ,d,c,b,a, h,g,f,e . . .

Therefore, The helical interleaver can be implemented by convolutional interleaver and a periodic reverse permutation. We summarize this relationship as follows:.




As an example, the sequence coming out of the convolutional interleaver in Figure 2 is

1 * * * 5 2 * * 9 6 3 * 13 10 7 4 ...

If we apply the periodic reverse permutation of order 4 to the above sequence starting from the beginning, we would have:

* * * 1 * * 2 5 * 3 6 9 4 7 10 13 ...

which is exactly the same sequence as the output from the helical interleaver (of the first type) in Figure 2 delayed by 3 symbols.

**6. Conclusion**

We have discussed two different types of helical interleavers and their advantages over traditional block interleavers. We have also shown that the first one of the two types of helical interleavers is equivalent to a convolutional interleaver followed by a periodic reverse permutation circuit. Unfortunately, this relationship does not exist for helical interleaving of the second type. Although helical interleaving of the second type aud convolutional interleaving have about the same interleaving delay and memory requirement, the helical interleaving of the second type has better synchronization structure than that of the convolutional interleavers.

## References


[1] Elwyn R. Berlekamp, Po Tong, “Improved Interleavers for Digital Communications”, US Patent Number 4,559,625, Dec. 17, 1985.

[2] George C. Clark, Jr., J. Bibb Cain, “Error-Correction Coding for Digital Communications”, Plenum Press, 1981.

[3] E. R. Berlekamp, R. E. Peile, S. P. Pope,. “The Application of Error Control to Communications”, IEEE Com. Magazine, April 1987.

[4] John L. Ramsey, “Realization of Optimum Interleavers”, IEEE Trans. Inform., p. 338--345, May 1970.

[5] G. D. Forney, “Burst-correcting codes for the classic bursty channel”, IEEE trans. Com. Tech., 772-781, October 1971

[6] A. J. Viterbi, J.K. Omura, “Principles of Digital Communication and Coding”, McGraw-Hill book Company, 1979.

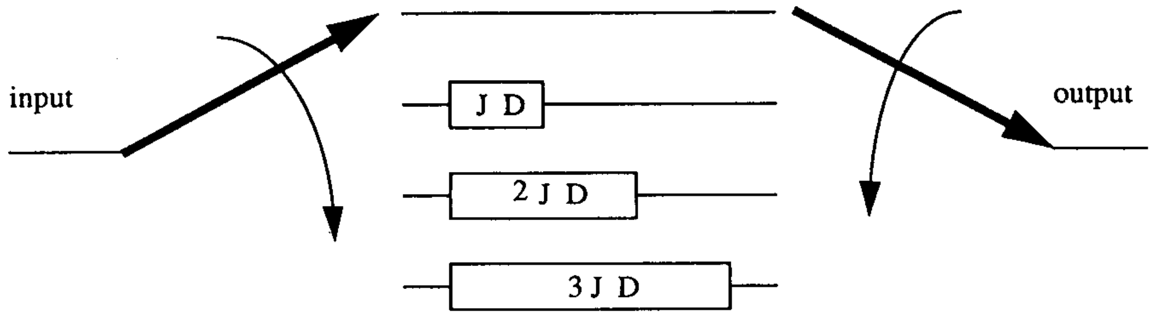
[7] “Interleaved Coding for Bursty Channels”, Cyclotomics’ (now Kodak Berkeley Research) Final Report on Phase I, Small Business Innovation Research, NSF Grant No. ECS-8260180, April 12, 1983.

[8] “Final Report on Variable-Depth Helical Interleaving”, Prepared by Cyclotomics (now Kodak Berkeley Research) for Office of Naval Research under Contract Number N00014-84-0407.

[9] W. W. Wu, D. Haccoun, R. Peile, Y. Hirata, “Coding for Satellite Communication”, IEEE, J-SAC, p.724-748, May 1987.

[10] G. D. Forney, “Concatenated Codes”, Combridge, MIT Press, 1966.

[11] “Report on Erasure Declaration with Helical Interleaving”, Robert J. McEliece, unpublished, May 1, 1989.




input sequence : 1 2 3 4 5 6 7 8 9 10 11 12 13...
output sequence : 1 * * 2 5 * 3 6 9 4 7 10 13...

**FIGURE 1. COVOLUTIONAL INTERLEAVER WITH M=4, J=1.**

| | | |
|---|---|---|
| **1** | * | * |
| **2** | **5** | * |
| **3** | **6** | **9** |
| **4** | **7** | **10** |
| 13 | **8** | **11** |
| 14 | 17 | **12** |
| 15 | 18 | 21 |
| 16 | 19 | 22 |
| **25** | 20 | 23 |
| **26** | **29** | 24 |
| **27** | **30** | **33** |
| **28** | **31** | **34** |
| | **32** | **35** |
| | | **36** |

into helical interleaver 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 ....
interleaved data 1 * * 2 5 * 3 6 9 4 7 10 13 8 11 14 17 12 15 18 21 ...

**FIGURE 2. HELICAL INTERLEAVER OF DEPTH 3.**

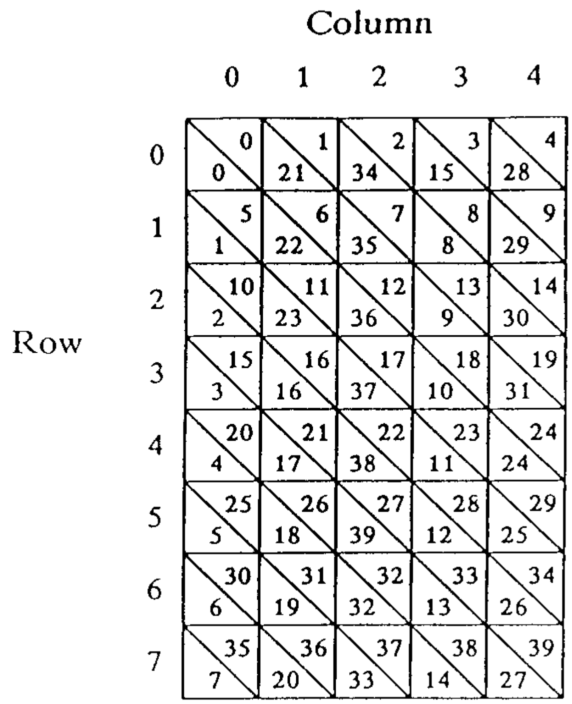



**FIGURE 3. HELICAL INTERLEAVING OF CODE LENGTH 8 WITH DEPTH 5.**