# Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding

Sergio Benedetto, *Fellow, IEEE*, Dariush Divsalar, *Fellow, IEEE*,
Guido Montorsi, *Member, IEEE*, and Fabrizio Pollara, *Member, IEEE*

***Abstract*—A serially concatenated code with interleaver consists of the cascade of an *outer* encoder, an interleaver permuting the outer codewords bits, and an *inner* encoder whose input words are the permuted outer codewords. The construction can be generalized to $h$ cascaded encoders separated by $h-1$ interleavers. We obtain upper bounds to the *average* maximum-likelihood bit error probability of serially concatenated block and convolutional coding schemes. Then, we derive design guidelines for the outer and inner encoders that maximize the *interleaver* gain and the asymptotic slope of the error probability curves. Finally, we propose a new, low-complexity iterative decoding algorithm. Throughout the paper, extensive comparisons with parallel concatenated convolutional codes known as "turbo codes" are performed, showing that the new scheme can offer superior performance.**

***Index Terms*— Concatenated codes, iterative decoding, serial concatenation, turbo codes.**

## NOMENCLATURE

| | |
|---|---|
| CC | Constituent Code. |
| PCCC | Parallel Concatenated Convolutional Code. |
| PCBC | Parallel Concatenated Block Code. |
| SCC | Serially Concatenated Code. |
| SCBC | Serially Concatenated Block Code. |
| SCCC | Serially Concatenated Convolutional Code. |
| ML | Maximum Likelihood. |
| IOWEF | Input–Output Weight-Enumerating Function. |
| CWEF | Conditional Weight-Enumerating Function. |
| SISO | Soft Input Soft Output module. |
| SW-SISO | Sliding Window—Soft Input Soft Output module. |
| LLR | Log-Likelihood Ratio. |
| MAP | Maximum *a posteriori*. |

## I. INTRODUCTION

IN his goal to find a class of codes whose probability of error decreased exponentially at rates less than capacity, while decoding complexity increased only algebraically, Forney [1] arrived at a solution consisting of the multilevel coding structure known as *concatenated code*. It consists of the cascade of an *inner* code and an *outer* code, which, in Forney's approach, would be a relatively short inner code (typically, a convolutional code) admitting simple maximum-likelihood (ML) decoding, and a long high-rate algebraic nonbinary Reed–Solomon outer code equipped with a powerful algebraic error-correction algorithm, possibly using reliability information from the inner decoder.

Initially motivated only by theoretical research interests, concatenated codes have since then evolved as a standard for those applications where very high coding gains are needed, such as (deep-) space applications and many others. Alternative solutions for concatenation have also been studied, such as using a trellis-coded modulation scheme as inner code [2], or concatenating two convolutional codes [3]. In the latter case, the inner Viterbi decoder employs a soft-output decoding algorithm to provide soft-input decisions to the outer Viterbi decoder. An interleaver was also proposed between the two encoders to separate bursts of errors produced by the inner decoder.

We find then, in a "classical" concatenated coding scheme, the main ingredients that formed the basis for the invention of "turbo codes" [4], namely two, or more, *constituent* codes (CC's) and an *interleaver*. The novelty of turbo codes, however, consists of the way they use the interleaver, which is embedded into the code structure to form an overall concatenated code with very large block length, and in the proposal of a parallel concatenation to achieve a higher rate for given rates of CC's. The latter advantage is obtained using systematic CC's and not transmitting the information bits entering the second encoder. In the following, we will refer to turbo codes as *parallel concatenated convolutional codes* (PCCC's). The so-obtained codes have been shown to yield very high coding gains at bit error probabilities in the range $10^{-5}-10^{-7}$; in particular, low bit error probabilities can be obtained at rates well beyond the channel cutoff rate, which had been regarded for long time as the "practical" capacity. Quite remarkably, this performance can be achieved by a relatively simple

Manuscript received July 26, 1996; revised November 1, 1997. This work was supported in part by NATO under Research Grant CRG 951208. The work of S. Benedetto and G. Montorsi was supported by Italian National Research Council (CNR) under "Progetto Finalizzato Trasporti (Prometheus)," by MURST (Progetto 40% "Comunicazioni con Mezzi Mobili"), and by Qualcomm. The research described in this paper was performed in part at the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, under Contract with the National Aeronautics and Space Administration (NASA). The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Ulm, Germany, June 29–July 4, 1997.

S. Benedetto and G. Montorsi are with Dipartimento di Elettronica, Politecnico di Torino, Italy.

D. Divsalar and F. Pollara are with Jet Propulsion Laboratory, Pasadena, CA 91109 USA.

Publisher Item Identifier S 0018-9448(98)02352-9.

iterative decoding technique whose computational complexity is comparable to that needed to decode the two CC's.

In this paper, we consider the serial concatenation of interleaved codes or *serially concatenated codes* (SCC's), called SCBC or SCCC according to the nature of CC's, that can be block (SCBC) or convolutional (SCCC) codes. For this class of codes, we obtain analytical upper bounds to the performance of an ML decoder, propose design guidelines leading to the optimal choice of CC's that maximize the *interleaver gain* and the asymptotic code performance, and present a new iterative decoding algorithm yielding results close to capacity limits with limited decoding complexity. Preliminary results have appeared in [5] and [6]. Extensive comparisons with turbo codes of the same complexity and decoding delay are performed.

With this embodiment of results, we believe that SCCC can be considered as a valid, in some cases superior, alternative to turbo codes.

In Section II, we derive analytical upper bounds to the bit error probability of both SCBC's and SCCC's, using the concept of "uniform interleaver" that decouples the output of the outer encoder from the input of the inner encoder. In Section III, we propose design rules for SCCC's through an asymptotic approximation of the bit error probability bound assuming long interleavers or large signal-to-noise ratios. In Section III, we compare serial and parallel concatenations of block and convolutional codes in terms of maximum-likelihood analytical upper bounds. Section V is devoted to the presentation of a new iterative decoding algorithm and to its application to some significant codes. Performance comparison between SCCC's and PCCC's under suboptimum iterative decoding algorithms are presented in Section IV.

## II. Analytical Bounds to the Performance of Serially Concatenated Codes

### A. A Union Bound to the Bit Error Probability and Some General Warnings

Consider an $(n, k)$ linear block code and its *Input–Output Weight Enumerating Function* (IOWEF), defined as

$$A(W, H) \triangleq \sum_{w=0}^{k} \sum_{h=0}^{n} A_{w,h} W^w H^h = \sum_{w=0}^{k} W^w A(w, H) \tag{1}$$

where $A_{w,h}$ represents the number of codewords with weight $h$ generated by information words of weight $w$. In (1), we have also implicitly defined the *conditional* weight enumerating function (CWEF)

$$A(w, H) \triangleq \sum_{h=0}^{n} A_{w,h} H^h \tag{2}$$

as the function that enumerates the weight distribution of codewords generated by information words of a given weight $w$.

A linear block (or convolutional) code possesses the uniform

performance can be evaluated under the assumption that the all-zero codeword has been transmitted. Assume then that the all-zero codeword $\boldsymbol{x}_0$ has been transmitted, and define the *pairwise* error event $e_{0h}(w)$ as the event in which the likelihood of a codeword with weight $h$ and generated by an information word of weight $w$ is higher than that of the all-zero codeword $\boldsymbol{x}_0$.

Using the union bound, the bit error probability under ML soft decoding for binary phase-shift keying (PSK) (or binary pulse amplitude modulation (PAM)) transmission over an additive white Gaussian noise channel with two-sided noise power spectral density $N_0/2$ can be upper-bounded as

$$\begin{aligned} P_b(e) &\le \sum_{h=1}^{n} \sum_{w=1}^{k} P_b[e|\boldsymbol{x}_0, e_{0h}(w)] P[e_{0h}(w)] \\ &= \frac{1}{2} \sum_{h=1}^{n} \sum_{w=1}^{k} \frac{w}{k} A_{w,h} \operatorname{erfc}\left(\sqrt{\frac{hR_cE_b}{N_0}}\right) \\ &= \frac{1}{2} \sum_{h=1}^{n} B_h^{(b)} \operatorname{erfc}\left(\sqrt{\frac{hR_cE_b}{N_0}}\right) \end{aligned} \tag{3}$$

where $R_c$ is the code rate, $E_b$ is the energy per *information* bit,[1] and where we have defined the *bit error multiplicity*

$$B_h^{(b)} \triangleq \sum_{w=1}^{k} \frac{w}{k} A_{w,h}. \tag{4}$$

Expressions (3) and (4) suggest that two ways can be followed to improve the bit error probability performance: the first, leading to the more traditional concept of good (and asymptotically good) codes, tries to increase the first, more significant weights $h$ in (3); the second, forming the basis of turbo codes and also of serially concatenated codes, aims at reducing the bit error multiplicities (4). To quote Forney's 1995 Shannon lecture:

> *Rather than attacking error exponents, turbo codes attack multiplicities, turning conventional wisdom on its head.*

A more compact, but looser, upper bound, can be obtained from (3) using the inequality

$$\tfrac{1}{2}\operatorname{erfc}(x) < \mathrm{e}^{-x^2} \tag{5}$$

which yields

$$P_b(e) < \sum_{w=1}^{k} \frac{w}{k} [A(w, H)]\big|_{H=\mathrm{e}^{-R_cE_b/N_0}}. \tag{6}$$

From (3) and (6), we conclude that, in order to upper-bound the bit error probability for any linear block code, we need to evaluate its CWEF. As a consequence, also for concatenated codes with interleavers we can use (3) and (6), provided that we are able to compute the CWEF of the overall code assuming that the CWEF's of the constituent codes (CC's) are known. This has been done already for "turbo codes," i.e., parallel concatenated codes, in [8]. In the following, we will show how to extend those results to the case of serial concatenation.

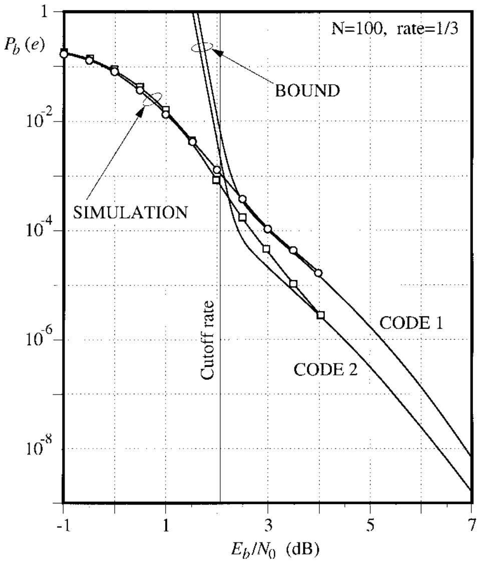

Fig. 1. Comparison of bit error probability curves obtained through union bounds and simulations for two parallel concatenated convolutional codes over an additive Gaussian noise channel. Also indicated is the $E_b/N_0$ value corresponding to the channel cutoff rate.

Before starting the analysis leading to the evaluation of the CWEF of a serially concatenated code (SCC) with interleaver, a warning to the readers is necessary. Both (3) and (6) stem from the union bound, stating that the probability of a union of events is less than or equal to the sum of the probabilities of the individual events. The union bound is used extensively as an upper limit to the error probabilities for digital transmission systems. The sums of the individual probabilities in the right-hand sides of (3) and (6), however, are not probabilities themselves, and can thus assume large values much greater than one. In fact, it is common knowledge in the field that union bounds are very close to the true probability in the case of maximum-likelihood decoding for medium-high signal-to-noise ratios, whereas they tend to diverge for low signal-to-noise ratios. A widely accepted rule of thumb is that the signal-to-noise ratio where they start to become unreliable is the one yielding the cutoff rate of the channel. The behavior of the bounds is illustrated as a typical example in Fig. 1, where we plot the bounds for two different rate $1/3$ parallel concatenated codes and compare them with simulation results obtained using the suboptimum, iterative decoding algorithm proposed to decode turbo codes. Also drawn in the figure is the $E_b/N_0$ corresponding to the channel cutoff rate.

Some general comments, partly based on Fig. 1, are appropriate:

- As previously anticipated, the upper bounds based on the union bound diverge at a signal-to-noise ratio close to the channel cutoff rate. Obtaining tighter upper bounds capable of extending the validity interval of the union bounds for concatenated codes is an important, and still widely open, topic for research. The new bounds could be [10]. A successful application of the Gallager bound to parallel concatenated codes with interleavers has been described in [11], where it is shown that the new bound extends the validity of the union bound for some range of signal-to-noise ratios below the channel cutoff rate, typically 0.5 dB. On the other hand, those attempts would still be based on the hypothesis of maximum-likelihood decoding. Thinking of applying them to the suboptimum iterative decoding seems not realistic.

- To obtain the divergence of the union bound one needs to compute a very large number of terms for the summation in the right-hand side of (3), or (6), and this was indeed the case for the example to which the curves of Fig. 1 refer. In that case, however, the interleaver length $N$ was limited to 100. When $N$ becomes very large, as it is required to approach the channel capacity, only a limited number of terms in the summations (3) and (6) can be obtained with a reasonable computational complexity. As a consequence, the obtained upper bounds are still very accurate above the channel cutoff rate, but may not present the divergence at cutoff rate. In those cases, the reader should only consider as reliable the bit error probability values above the cutoff rate, or perhaps half a decibel below it, according to the results of [11]. A way to overcome this drawback, and indeed to always show the divergence of the union bound, has been proposed in [12]. It has, however, to rely on the looser bound (6).

- The main tool used in this paper to analytically predict the performance and to find design rules about the main CC's parameters is a union bound. We have seen, on the other hand, that the union bound is tight only for medium-high signal-to-noise ratios. One could then question the validity of the approach, which suffers the paradox of using a bound not applicable to very low signal-to-noise ratios in order to design coding schemes intended to work near channel capacity.

  We are conscious of this inconsistency, yet, for one hand, have simply nothing better to propose, and, on the other hand, we had widely verified by simulation that the parallel concatenated codes designed on the basis of our rules are indeed very good also at very low signal-to-noise ratios (see [13] and [14]). In the remainder of this paper, we will show that this heuristic validation of the design rules also holds for serially concatenated codes with interleavers.

- The last observation concerns still another inconsistency, resulting from the fact that we are using bounds based on maximum-likelihood decoding to design codes that are decoded according to a different, suboptimum algorithm. Also in this case we invoke the heuristic validation stemming from a large number of simulations, which show the convergence of the simulated performance toward the analytical bounds. In Fig. 1, where the simulated points have been obtained with the suboptimum, itera-

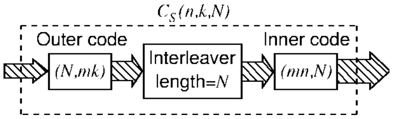

Fig. 2. Serially concatenated $(n,\, k,\, N = mp)$ block code.

### B. Evaluating the Bit Error Probability Upper Bound for Serially Concatenated Block Codes

We will now show how to apply the union bounds (3) and (6) to the case of serially concatenated codes with interleavers. For simplicity of the presentation, we begin considering serially concatenated block codes (SCBC's).

The scheme of two serially concatenated block codes is shown in Fig. 2. It is composed of two cascaded CC's, the *outer* $(p,\, k)$ code $C_o$ with rate $R_c^o = k/p$ and the *inner* $(n,\, p)$ code $C_i$ with rate $R_c^i = p/n$, linked by an interleaver of length $N = mp$ that is an integer multiple of the length $p$ of the outer codewords. The scheme works as follows: the $mp$ bits of a number $m$ of codewords of the outer code are written into the interleaver of length $N = mp$, and read in a different order according to the permutation performed by the interleaver. The sequence of $N$ bits at the output of the interleaver is then sent in blocks of length $p$ to the inner encoder.

The overall SCBC is then an $(n,\, k)$ code with rate

$$R_c^S = R_c^o \times R_c^i = k/n$$

and we will refer to it as the $(n,\, k,\, N = mp)$ code $C_S$.

In the following, we will derive an upper bound to the ML performance of the overall code $C_S$, assuming first that $m = 1$, and then extending the result to the general case. We assume that the outer and inner CC's are linear, so that also the SCBC is linear and the *uniform error property* applies, i.e., the bit error probability can be evaluated assuming that the all-zero codeword has been transmitted.

In order to apply the upper bounds (3) and (6) to the SCBC, we need to evaluate the CWEF of the code $C_S$, assuming that we know the CWEF's of the CC's.

If $p$ is low, we can compute the coefficients $A_{w,\,h}$ of the CWEF $A(w,\, H)$ (2) by letting each individual information word with weight $w$ be first encoded by the outer encoder $C_o$ and then, after the $p$ bits of the outer codeword have been permuted by the interleaver, be encoded by the inner encoder $C_i$ originating an inner codeword with a certain weight. After repeating this procedure for all the information words with weight $w$, we should count the inner codewords with weight $h$, and their number would be the value of $A_{w,\,h}$. When $k$ is large, or, in the case $N = mp$, when $m$ is large, the previous operation becomes too complex, and we must resort to a different approach.

The key point, here, is that we would like to obtain a simple relationship between the CWEF's of the two CC's, an operation that is prevented by the fact that the knowledge of the information word weight is not enough to obtain the weight of the inner codeword, which, instead, depends on the

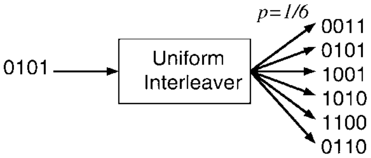

Fig. 3. The action of a uniform interleaver of length 4 on sequences of weight 2.

As in [8] and [15], a crucial step in the analysis consists in replacing the actual interleaver that performs a permutation of the $N$ input bits with an abstract interleaver called *uniform interleaver*, defined as a probabilistic device that maps a given input word of weight $l$ into all distinct $\binom{N}{l}$ permutations of it with equal probability $P = 1/\binom{N}{l}$ (see Fig. 3).

Use of the uniform interleaver permits the computation of the "average" performance of the SCBC, intended as the expectation of the performance of SCBC's using the same CC's, taken over the set of all interleavers of a given length. A theorem proved in [8] guarantees the meaningfulness of the average performance, in the sense that there will always be, for each value of the signal-to-noise ratio, at least one particular interleaver yielding performance better than or equal to that of the uniform interleaver.

Let us define the IOWEF and the CWEF of the SCBC $C_S$ as $A^{C_S}(W,\, H)$ and $A^{C_S}(w,\, H)$. Their definition and meaning are the same as in (1) and (2).

As seen, to apply the bounds (3) and (6) to the bit error probability we need to evaluate the CWEF of the SCBC from the knowledge of the CWEF's of the outer and inner codes, which we call $A^{C_o}(w,\, L)$ and $A^{C_i}(l,\, H)$, where the first enumerates the weight distributions of the outer codewords generated by information words of weight $w$, and the second enumerates the weight distributions of the inner codewords generated by outer codewords of weight $l$.

To do this, we exploit the properties of the uniform interleaver, which transforms a codeword of weight $l$ at the output of the outer encoder into all its distinct $\binom{N}{l}$ permutations. As a consequence, each codeword of the outer code $C_o$ of weight $l$, through the action of the uniform interleaver, enters the inner encoder generating $\binom{N}{l}$ codewords of the inner code $C_i$. Thus the number $A_{w,\,h}^{C_S}$ of codewords of the SCBC of weight $h$ associated with an information word of weight $w$ is given by

$$A_{w,\,h}^{C_S} = \sum_{l=0}^{N} \frac{A_{w,l}^{C_o} \times A_{l,h}^{C_i}}{\binom{N}{l}}. \tag{7}$$

From (7) we derive the expressions of the CWEF and IOWEF of the SCBC as

$$A^{C_S}(w,\, H) = \sum_{l=0}^{N} \frac{A_{w,l}^{C_o} \times A^{C_i}(l,\, H)}{\binom{N}{l}} \tag{8}$$

$$A^{C_S}(W,\, H) = \sum \frac{A^{C_o}(W,\, l) \times A^{C_i}(l,\, H)}{(N} \tag{9}$$

where $A^{C_o}(W, l)$ enumerates the weight distributions of the information words that generate codewords of the outer code with a given weight $l$.

*Example 1:* Consider the $(7, 3)$ serially concatenated block code obtained by concatenating the $(4, 3)$ parity-check code to a $(7, 4)$ Hamming code through an interleaver of length $N = 4$. The IOWEF $A^{C_o}(W, L)$ and $A^{C_i}(L, H)$ of the outer and inner code are

$$\begin{aligned} A^{C_o}(W, L) &= 1 + W(3L^2) + W^2(3L^2) + W^3(L^4)\\ A^{C_i}(L, H) &= 1 + L(3H^3 + H^4) + L^2(3H^3 + 3H^4)\\ &\quad + L^3(H^3 + 3H^4) + L^4H^7 \end{aligned}$$

so that

$$\begin{array}{l|l} A^{C_o}(W, 0) = 1 & A^{C_i}(0, H) = 1\\ A^{C_o}(W, 1) = 0 & A^{C_i}(1, H) = 3H^3 + H^4\\ A^{C_o}(W, 2) = 3W + 3W^2 & A^{C_i}(2, H) = 3H^3 + 3H^4\\ A^{C_o}(W, 3) = 0 & A^{C_i}(3, H) = H^3 + 3H^4\\ A^{C_o}(W, 4) = W^3 & A^{C_i}(4, H) = H^7 \end{array}$$

Through (9) we then obtain

$$\begin{aligned} A^{C_S}(W, H) &= \sum_{l=0}^{4} \frac{A^{C_o}(W, l) \times A^{C_i}(l, H)}{\binom{N}{l}}\\ &= \frac{1 \cdot 1}{1} + \frac{0 \cdot (3H^3 + H^4)}{4}\\ &\quad + \frac{(3W + 3W^2) \cdot (3H^3 + 3H^4)}{6}\\ &\quad + \frac{0 \cdot (H^3 + 3H^4)}{4} + \frac{W^3 \cdot H^7}{1}\\ &= 1 + W(1.5H^3 + 1.5H^4)\\ &\quad + W^2(1.5H^3 + 1.5H^4) + W^3H^7. \qquad \square \end{aligned}$$

Previous results (8) and (9) can be easily generalized to the more interesting case of an interleaver with length $N$ being an integer multiple (by a factor $m > 1$) of the length of the outer codewords. Denoting by $A^{C_o^m}(W, L)$ the IOWEF of the new $(mp, mk)$ outer code, and similarly by $A^{C_i^m}(L, H)$ the IOWEF of the new $(mn, mp)$ inner code, it is straightforward to obtain

$$\begin{aligned} A^{C_o^m}(W, L) &= [A^{C_o}(W, L)]^m\\ A^{C_i^m}(L, H) &= [A^{C_i}(L, H)]^m. \end{aligned} \tag{10}$$

From the IOWEF's (7)–(10), we obtain the CWEF's $A^{C_o^m}(W, l)$ and $A^{C_i^m}(l, H)$ of the new CC's, and, finally, through (8) and (9), the CWEF and IOWEF of the new $(n, k, N = mp)$ SCBC $C_S^m$

$$A^{C_S^m}(w, H) = \sum_{l=0}^{N} \frac{A^{C_o^m}_{w,l} \times A^{C_i^m}(l, H)}{\binom{N}{l}} \tag{11}$$

$$A^{C_S^m}(W, H) = \sum_{l=0}^{N} \frac{A^{C_o^m}(W, l) \times A^{C_i^m}(l, H)}{\binom{N}{l}}. \tag{12}$$

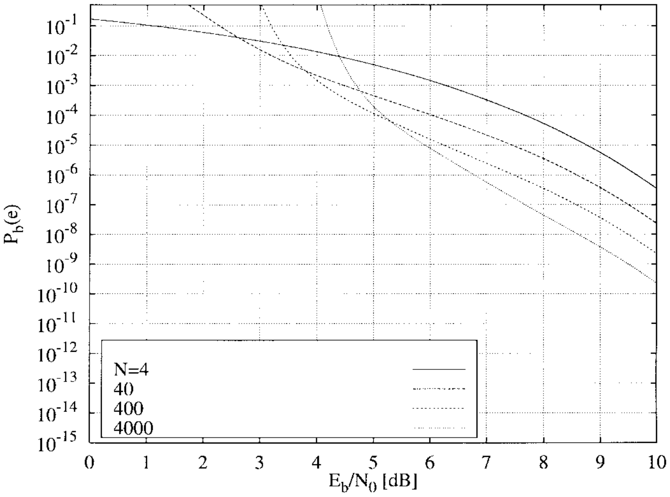

Fig. 4. Analytical bounds for serially concatenated block code of Example 2 (SCBC1 in Table I).

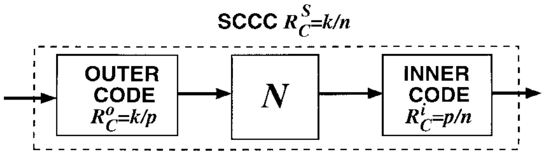

Fig. 5. Serially concatenated $(n, k, N)$ convolutional code.

*Example 2:* Consider again the CC's of Example 1, linked by an interleaver of length $N = 4m$, and use (2), (3), and (11). The so obtained upper bound to the bit error probability is plotted in Fig. 4 for various values of the integer $m$. The curves show the *interleaver gain*, defined as the factor by which the bit error probability is decreased with the interleaver length at a given signal-to-noise ratio. Contrary to parallel concatenated block codes [8], the curves do not exhibit the interleaver gain saturation. Rather, the bit error probability seems to decrease regularly with $m$ as $m^{-1}$. We will explain this behavior in Section III. □

### *C. Serially Concatenated Convolutional Codes*

The structure of a serially concatenated convolutional code is shown in Fig. 5. It refers to the case of two convolutional CC's, the outer code $C_o$ with rate $R_c^o = k/p$, and the inner code $C_i$ with rate $R_c^i = p/n$, joined by an interleaver of length $N$ bits, generating an SCCC $C_S$ with rate $R_c^S = k/n$. $N$ will be assumed to be an integer multiple[2] of $p$. We assume, as before, that the convolutional CC's are linear, so that the SCCC is linear as well, and the uniform error property applies.

The exact analysis of this scheme can be performed by appropriate modifications of that described in [8] for PCCC's. It requires the use of a *hypertrellis* having as *hyperstates* pairs of states of outer and inner codes. The hyperstates $S_{ij}$ and $S_{lm}$ are joined by a *hyperbranch* that consists of all pairs of paths

[2] Actually, this constraint is not necessary. We can choose in fact inner and outer codes of any rates $R_c^i = k_i/n_i$ and $R_c^o = k_o/n_o$, constraining the interleaver to be an integer multiple of the minimum common multiple of $n_o$

DOCKET ALARM

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

### LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

### FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

### E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase®
Smarter legal research.