

Huffman Code Based Error Screening and Channel Code Optimization for Error Concealment in Perceptual Audio Coding (PAC) Algorithms

J. Nicholas Laneman, *Member, IEEE*, Carl-Erik W. Sundberg, *Fellow, IEEE*, and Christof Faller

Abstract—The class of Perceptual Audio Coding (PAC) algorithms yields efficient and high-quality stereo digital audio bitstreams at bit rates from 16 kb/sec to 128 kb/sec (and even higher bit rates). To avoid “pops and clicks” in the decoded audio signals due to erasure or undetected errors from transmission over unreliable channels, e.g., in the context of digital audio broadcasting (DAB), channel error detection combined with source error concealment, or source error mitigation, techniques are preferred to pure channel error correction. One simple and efficient way to perform channel error detection is to employ a high-rate block code; for example, the preferred solution for hybrid in-band on-channel (HIBOC) DAB in the FM band employs a cyclic redundancy check (CRC) code. Several joint source-channel coding issues arise in this framework because PAC contains a fixed-to-variable source coding component in the form of Huffman codes, so that the output audio packets are of varying length. We explore two such issues in this paper.

First, we develop methods for screening for undetected channel errors in the audio decoder by looking for inconsistencies between the number of bits decoded by the Huffman decoder and the number of bits in the packet as specified by control information within the bitstream. We evaluate this scheme by means of simulations of Bernoulli sources and real audio data encoded by PAC, both exposed to random bit errors as well as errors that pass undetected through a CRC decoder. Considerable reduction in undetected errors is obtained with little extra processing in the receiver and with little or no increase in the transmitted bit rate.

Second, we consider several configurations for the channel error detection codes, in particular CRC codes, by means of representative simulations and informal listening tests, for several audio coder bit rates of interest in DAB. One configuration employs a fixed-rate, fixed-blocklength code of optimized length outside the PAC algorithm. Another preferable set of formats employs variable-blocklength, variable-rate outer codes matched to the individual audio packets, with one or more codewords used per audio packet. In this case, better performance is obtained; however, to maintain a constant bit rate into the channel, PAC and CRC encoding must be performed jointly, e.g., by incorporating the CRC into the bit allocation loop in the audio coder.

Index Terms—Conformance testing, digital audio broadcasting, error detection coding, Huffman codes, source coding.

I. INTRODUCTION

TECHNICAL work is under way for establishing candidate standard schemes for digital audio broadcasting (DAB) in the US and around the world, both for terrestrial and satellite broadcasting. Digital audio broadcasting methods compatible with existing terrestrial analog FM and AM radio broadcasting are preferred by U.S. Broadcasters [1]. These potential applications have stimulated significant development of low bit rate audio coding algorithms, such as the perceptual audio coding (PAC) algorithm [2], [3], in parallel with, and often in conjunction with, robust and bandwidth-efficient transmission methods. Multistream transmission [4], [5] is a particularly appealing example of these developments.

PAC can achieve stereo, CD-quality audio at bit rates of 56–128 kb/sec. Bit rates of 64–96 kb/sec are suitable for digital audio broadcasting applications in the FM band. Daytime AM broadcasting may require bit rates in the range of 32–48 kb/sec, and versions of the PAC algorithm exist for bit rates as low as 16 kb/sec. There is remaining redundancy in the audio data because of limited delay and complexity of the practical source coders. As is the case with digital speech transmission in cellular systems, this redundancy can be leveraged in the audio decoder by error concealment, or error mitigation, algorithms. These algorithms essentially fill in lost frames by interpolating neighboring frames. Such algorithms are triggered by a signal, referred to as a flag, generated by, e.g., the channel decoder or various consistency checks within the audio decoder, indicating that a channel error has likely occurred. If the error concealment algorithm is activated infrequently, the smoothed output from the channel error detection and source error concealment approach is preferred to a solution with channel error correction only, because undetected channel errors may cause audible “pops and clicks” in the signal generated by the audio decoder.

The channel error detection mechanism can be efficiently implemented by employing a high-rate error detecting block code; for example, hybrid in-band on-channel (HIBOC) systems in the FM band utilize outer cyclic-redundancy check (CRC) codes for error detection, along with inner convolutional codes for channel robustness. Details of some of the digital audio broadcasting systems are given in [4]–[8], while background on channel coding methods is provided in [9]–[19]. In almost

Manuscript received October 19, 2000; revised August 1, 2002.

J. N. Laneman was with the Multimedia Communications Research Laboratory, Bell Labs, Lucent Technologies, Murray Hill, NJ 07974 USA. He is now with the Electrical Engineering Department, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: jnl@nd.edu).

C.-E. W. Sundberg was with the Multimedia Communications Research Laboratory, Bell Labs, Lucent Technologies, Murray Hill, NJ 07974 USA. He is now with iBiquity Digital Corp., Warren, NJ 07059 USA (e-mail: sundberg@ibiquity.com).

C. Faller was with the Multimedia Communications Research Laboratory, Bell Labs, Lucent Technologies, Murray Hill, NJ 07974 USA. He is now with Agere Systems, Murray Hill, NJ 07974 USA (e-mail: cfaller@agere.com).

Publisher Item Identifier 10.1109/TBC.2002.803705.

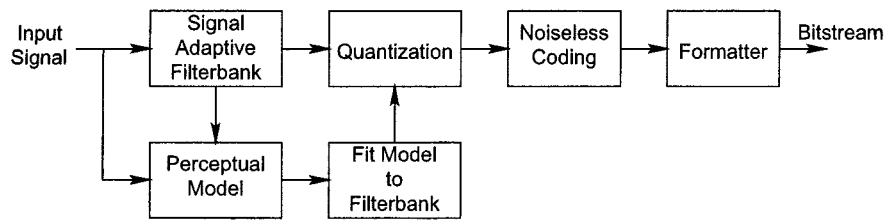


Fig. 1. Overview of the Perceptual Audio Coder (PAC).

all these DAB transmission system proposals, concatenated channel coding is used. The outer channel code (the one closest to the audio coder) can be a CRC block code for error mitigation flagging through error detection, or a more powerful Reed–Solomon code for error flagging either through error correction only or through limited error correction with error detection. When error correction only is employed, most Reed–Solomon decoders produce an error flag when the “failure to decode” state has occurred.

PAC performs lossless Huffman coding of the quantized transform coefficients of an audio frame, so that the resulting bitstream consists of a sequence of variable-length packets, where, throughout this paper, we measure the length of a packet in bits. This property raises many joint source-channel coding design issues, several of which we explore in this paper. For example, we exploit any inconsistency between Huffman code data and PAC control information to detect errors that pass undetected through the channel decoder(s). This processing takes place in the PAC decoder after CRC decoding. The result is a screening mechanism through which a considerable fraction of the undetected channel errors are converted to error concealment flags. As another example, we consider the influence of the outer channel code format on the performance of the audio decoder error concealment algorithm. Since the most convenient way of employing the outer code is to use a fixed-blocklength code of a fixed rate, a matching problem results. In this case, shorter blocklengths yield weaker codes that allow too many channel errors to pass undetected into the audio decoder, while longer blocklengths lead to more powerful codes that detect more errors than necessary and, in particular, are more likely to overlap and erase two consecutive audio packets. We observe that the preferred outer CRC code length depends on the bit rate of the audio decoder. As an alternative to the fixed-blocklength approach, we introduce a number of ways to apply the outer CRC code inside the PAC bit allocation scheme. This leads to an integrated PAC and CRC encoding unit with a better match with the error concealment algorithm in terms of frequency of activation. In addition, incorporating the CRC encoder inside the PAC rate loop leads to increases in the effective source coding bit rate for the same total channel input bit rate.

We note that the ideas on screening algorithms based on Huffman code consistency checks were created during discussions between Deepen Sinha and the second author. In this paper we present quantitative analysis of such algorithms. In parallel with this study, practical screening systems were developed and incorporated into real hybrid digital and all

digital audio broadcasting systems for the AM and FM bands [20].

An outline of this paper is as follows. Section II provides a more detailed overview of PAC and describes efficient error concealment algorithms for PAC. Section III shows how error detection and concealment can be improved by using redundant Huffman parsing information in the PAC coder bitstream. We also refer to this as a screener for undetected errors. Section IV describes several options for employing an outer channel code. Section V provides results of outer CRC code optimization for various audio coder bit rates from 16 kb/sec up to 96 kb/sec. Both code length and code rate are considered. The results of informal, subjective listening tests are included. The simulation setup is a Gaussian channel as well as Rayleigh fading channels with an inner convolutional code and an outer CRC code. This data gives guidance for other coding setups as well at different audio coder bit rates. Section VI closes the paper with discussion and conclusions.

II. BRIEF SUMMARY OF PAC

The PAC algorithm [2], [3] is a transform coding algorithm that incorporates advanced signal processing and psychoacoustic modeling techniques to achieve a high level of signal compression. Fig. 1 provides a block diagram of the PAC encoder. In brief, PAC uses a perceptually-designed, signal adaptive filterbank that switches between a modified discrete cosine transform (MDCT) and a wavelet transform to obtain a compact description of the signal [21]. The filterbank output is quantized using nonuniform vector quantizers, and the quantized coefficients are further compressed using an adaptive Huffman coding scheme. For the purposes of quantization, the filterbank outputs are grouped into so-called *codebands* so that quantizer parameters, e.g., stepsizes generated by a psychoacoustic model, are independently chosen for each codeband. A total of fifteen different Huffman codebooks are employed, with the best codebook chosen independently for each codeband. For stereo and multichannel audio material, either left/right, sum/difference, or other forms of channel combinations may be encoded.

A. PAC Bitstream Description

The format of the PAC bitstream for DAB applications is depicted in Figs. 2 and 3. PAC is a blockwise algorithm that formats compressed audio information into a packetized bitstream. For example, at audio sampling bit rates of 44.1 kHz, each packet contains compressed data for 1024 input samples

		Bits
PAC SYNC		23
Bit Rate		5
Sampling Rate (Input)		4
Sampling Rate (Output)		4
Block Length		12
Half Block Length		12
Next Block Length		12
Next Half Block Length		12
Bit Buffer State		4
Sequence Number		5
Stream ID		2
Ancillary Data (Cons) Length		1/11
Auxiliary Data (Var) Length		10
Number of Channels		3
Audio Proc. Mode		2
Enhancement Modes		var
AUDIO DATA	Filterbank State	var
	PER CHANNEL DATA (see below)	
AUXILIARY DATA		var
ANCILLARY DATA		var

Fig. 2. PAC stereo bitstream (packet) description—I.

from each channel, regardless of the number of channels. Additionally, each packet contains control information including, e.g., Huffman codebook selection, quantizers, and channel combination information. Although a long-term average bit rate can be maintained, packet lengths are variable because the 1024 transform samples are compressed in lossless fashion with fixed-to-variable Huffman codes.

The PAC coder has to quantize the frequency bands in such a way that the quantization noise remains below the masking threshold. The complex interaction between the choice of quantizer step sizes (scale factors) and Huffman coding on the resulting bit rate requires an iterative process commonly referred to as a rate loop. Usually several iterations are required until the bit demand for a given frame is within the range needed to maintain the average bit rate. As we will see in Section IV, one way to introduce an outer error detecting code, for transmission over unreliable channels, is inside this rate loop. In this case, the overhead bits required for the cyclic redundancy check code are taken into account in the overall bit budget for the rate loop.

Depending upon the intended application, additional information may be added to the first packet, or to each group of several packets. For unreliable transmission channels, such as DAB over radio, a header containing, e.g., PAC synchronization information, sampling rate, transmission bit rate, audio coding modes, and so forth, may be added. Critical control information can be further protected from channel errors by repeating it in two consecutive packets. This is an example of a very simple method of unequal error protection (UEP) in the PAC framework. More advanced UEP schemes are described in [8].

B. Error Mitigation Techniques

When PAC operates over unreliable transmission channels, errors inevitably occur in the bitstream. Given some indication

of an error, PAC decoder error mitigation, or error concealment, techniques attempt to reduce the impact of these errors on the output audio. Examples of error concealment techniques include:

- Inter-packet interpolation
- Heuristic rules for interpolation based upon characteristics of the MDCT
- Use of partial packets

As we will see in Section V, these techniques preserve audio quality without severe artifacts for packet loss rates (packet flag rates) of up to 10–12%. A reliable method of flagging packets with errors is required. Furthermore, a much lower rate of undetected packets in error is assumed. In the following section we will describe a method of screening undetected packet errors using Huffman code information. This effectively converts packet errors into flagged packets.

III. SCREENER BASED ON HUFFMAN CODE AND CONTROL INFORMATION INCONSISTENCY

Each packet in the PAC bitstream contains a sequence of Huffman codewords describing a fixed block of audio samples, e.g., 1024 samples from a given channel. Well-known Huffman codes are efficient fixed-to-variable lossless data compression codes [9], that are self-parsing, i.e., even in the presence of transmission errors, Huffman decoding of received bits continues in a sequential fashion. When errors occur in a Huffman encoded stream of data, not only the particular codewords in a frame change, but also the *number* of codewords in a frame may change. More specifically, the number of bits needed to decode a fixed block of audio, e.g., 1024 samples or transform coefficients, is likely to be different when transmission errors occur; hence, control information indicating the number of codewords or bits that should be present in the packet can be used to screen for transmission errors. Fig. 4 shows simple examples of how errors can be detected and also can pass through undetected. Source sequences of length 2 bits in blocks of length 16 bits are Huffman coded in this example using the code table in Fig. 4. The coded sequence in the example is of length 11 bits. Two alternative error patterns are shown. The received sequence of 11 bits marked “Detected” has a single error in bit position two (encircled). When this sequence is Huffman decoded, 18 source bits are produced, inconsistent with 16 bits. Thus, this error is detected by the Huffman screener, since the receiver is expecting 16 source bits for every transmitted block. The received sequence marked “Undetected” has a double error, but it is Huffman decoded to the correct number 16 of source bits. Thus, in this case the screener is unable to detect the error.

Conveniently, the PAC bitstream contains some of the required control information in a robust, i.e., highly channel protected, format, and additional information can easily be added as necessary. This control information has already been used, e.g., for reliable synchronization and buffering at the receiver, and we may leverage it as a consistency check against the number of bits demanded by the Huffman decoder for decoding a packet corresponding to a block of audio. Assuming that the control information is correctly received, any inconsistency between the

SECTIONING INFO (number of sections + section boundaries + Huffman codebooks)	SCALE-FACTORS	ENHANCEMENT DATA (e.g., predictors, model parameters, etc.)	COEFFICIENTS
-------------------------------------------------------------------------------------	---------------	-------------------------------------------------------------------	--------------

Fig. 3. PAC stereo bitstream (packet) description—II.

Seq.	Prob.	Codeword	Source: 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1
00	0.01	000	Coded: 1 0 0 1 1 0 1 1 1 1 1
01	0.09	001	Detected: 1 ① 0 1 1 0 1 1 1 1 1
10	0.09	01	Undetected: 1 0 ① 1 1 0 ① 1 1 1 1
11	0.81	1	

Fig. 4. Examples of Huffman consistency checking on 8 consecutive realizations of a 4-ary source. Codeword parsings are underlined. The received sequence marked “Detected” has a single transmission error and the Huffman decoder produces an incorrect number of source symbols. The sequence marked “Undetected” has a double error but the Huffman decoder produces a correct number of source symbols.

packet length and bits required by the Huffman decoder can be used to flag a transmission error.

To examine the screening efficiency of this so-called Huffman consistency check in a controlled setting, we consider a white binary source with probability p of being a 1. We construct a Huffman code for this source using vectors of 8 bits. Routines for Huffman table construction, encoding, and decoding are taken from [10]. We prescribe the framelength n in uncoded source bits, e.g., 128, 256, 512, and 1024 bits. We apply the Huffman code to a sequence of $n/8$ randomly generated source vectors, and record the number of bits B_e used to encode the source. We then introduce channel errors, as described next. The receiver decodes $n/8$, 8-bit vectors from the received sequence, and records the number of bits B_d used to decode the source. If $B_d \neq B_e$, then a channel error has occurred, and the screener can indicate a flag to the audio decoder error mitigation unit; otherwise, the error has gone undetected by the screener.

In principle, the screener based on the Huffman code and control information consistency can be used both with and without an outer code flag generated by, e.g., a CRC code. As a result, we consider two forms of channel errors: random bit errors generated by a binary symmetric channel (BSC), and errors that would pass undetected through a linear CRC error detecting block code. We characterize the efficiency of the Huffman consistency check using conditional flag and undetected error rates, i.e., given that an error occurs in the channel, we compute the conditional flag rate $P_{f|e}$ and the conditional undetected error rate $P_{u|e}$ of the Huffman decoder. In this sense, the Huffman decoder, along with side information about the length of the frame, can be viewed as an additional error detecting code. We note that in either case, without the Huffman screener, the conditional flag rate $P_{f|e} = 0$ while the conditional undetected error rate $P_{u|e} = 1$.

In this paper, we are only considering one of the simplest methods of using properties of Huffman codes for error screening. Other methods that may yield further improvements—although potentially at the expense of additional complexity and transmitted bits—are described in [22]–[24] and references therein. As we will see, the simple Huffman screener that we develop provides considerable improvement;

more general and involved approaches may warrant further study.

A. Random Errors

Random bit errors can be generated from a BSC with error probability P_e . We intentionally bias the frame error generator to guarantee that at least one bit error occurs in each block; this simple form of “importance sampling” reduces wasteful computation, especially for small P_e .

We may rewrite the conditional undetected error rate as

$$P_{u|e} = \sum_w p(w|e) P_{u|e,w}, \quad (1)$$

where w is the weight of an error event, with $p(w|e)$ a suitably conditioned Binomial distribution with the BSC crossover probability P_e as its parameter, and $P_{u|e,w}$ the conditional undetected error rate of the Huffman consistency check for an error of weight w . A similar expression may be written for the conditional flag rate. From (1), we immediately see that the efficiency of the Huffman consistency check depends upon the efficiency for a particular error weight along with the relative frequency with which errors of that weight occur.

Fig. 5 shows the conditional undetected error rate $P_{u|e,w}$ versus the weight w of random errors introduced by a BSC with crossover probability $P_e = 0.1$. The simulation uncertainty for large error weights is due to the fact that fewer of the large error weight sequences are generated in the simulation, thereby increasing the variance in the estimator; however, the general trends are apparent from these results, indicating that higher-weight errors are much more likely to be detected by the Huffman consistency check. Thus, we see that the Huffman consistency check will perform poorly on a BSC with low crossover probability P_e , because the average in (1) will be dominated by low-weight errors, for which the conditional undetected error rate is high. From the results in Fig. 5, we also conclude that the relative efficiency of the Huffman screener improves with increasing p values. Thus, in an environment of a skewed source with large gains due to the Huffman code, the Huffman screener performs better.

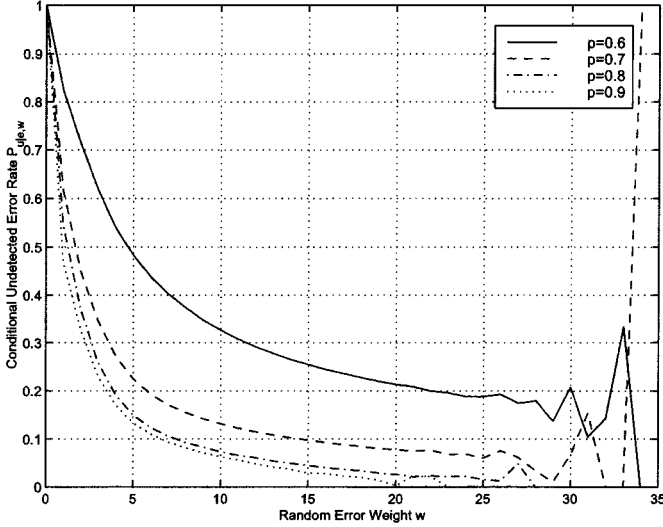


Fig. 5. Conditional undetected error rate $P_{u|e,w}$ efficiency of the Huffman consistency check versus the weight w of random errors introduced by a BSC with crossover probability $P_e = 0.1$. The uncoded blocklength of the data sequences were 128, and several values of the source skewness were simulated.

B. Undetected CRC Errors

The results of the previous section suggest that the Huffman consistency check can be much more efficient on the average if proportionally more of the errors have higher weight than on a BSC; therefore, channel coding, which increases the minimum Hamming error weight, should help make the Huffman consistency check more efficient.

As a simple but pertinent example of this effect, we generate low-weight errors that would pass undetected through a CRC code. These can be conveniently generated by adding, modulo 2, an appropriately shifted version of the generator polynomial of the CRC code to the transmitted codeword. To eliminate the unnecessary computation of CRC encoding and decoding, the check bits can be artificially added to the bitstream, the error sequence added, and then the check bits are removed before Huffman decoding.

Table I gives the results for simulations of the Huffman screener with undetected errors from the 16-bit redundancy ITU CRC code with generator polynomial $x^{16} + x^{12} + x^5 + 1$ [10], and a 31-bit redundancy CRC code with generator polynomial $x^{31} + x^{27} + x^{20} + x^{16} + x^{11} + x^9 + x^4 + 1$ arbitrarily chosen to have roughly twice the number of check bits and twice the generator weight of the ITU CRC code.

From these results it seems clear that the Huffman consistency check is much more effective when proportionally more of the errors have larger weight and more structure than observed on a BSC. Of the three issues examined in Table I, namely, uncoded framelenh, source skewness, and CRC code redundancy, the order of their impact on the efficiency of the Huffman consistency check appears to be source skewness, uncoded framelenh, and CRC code redundancy.

C. PAC With Undetected CRC Errors

We applied the Huffman screener to real audio data encoded with PAC at 64 kb/sec. In this experiment we wanted to find out

TABLE I
HUFFMAN SCREENER RESULTS FOR UNDETECTED CRC-ITU ERRORS

Framelength	p	16-Bit CRC (CCITT)		31-Bit CRC	
		P_{fle}	P_{ule}	P_{fle}	P_{ule}
128	0.6	89.637%	10.363%	90.437%	9.563%
128	0.7	90.625%	9.375%	91.745%	8.255%
128	0.8	92.590%	7.410%	93.582%	6.418%
128	0.9	95.865%	4.135%	96.191%	3.809%
256	0.6	89.354%	10.646%	90.198%	9.802%
256	0.7	91.359%	8.641%	92.550%	7.450%
256	0.8	93.498%	6.502%	94.678%	5.322%
256	0.9	96.471%	3.529%	96.916%	3.084%
512	0.6	88.915%	11.085%	89.761%	10.239%
512	0.7	91.469%	8.531%	92.854%	7.146%
512	0.8	93.968%	6.032%	95.203%	4.797%
512	0.9	96.766%	3.234%	97.241%	2.759%
1024	0.6	88.180%	11.820%	89.147%	10.853%
1024	0.7	91.575%	8.425%	92.790%	7.210%
1024	0.8	94.158%	5.842%	95.484%	4.516%
1024	0.9	96.981%	3.019%	97.474%	2.526%

how reliably a Huffman screener within the PAC audio coder would detect bit-errors within the bits of the Huffman codewords. PAC data represents a more complex source than the one we examined in the preliminary experiments, since it uses various multidimensional Huffman codebooks for each frame.

For the experimental setup we partitioned the bits of each PAC frame into four regions. Two Huffman data regions (left and right stereo channels) and two side information data regions (left and right). The Huffman data regions contain only pure Huffman codeword data.

Fig. 6 illustrates our Huffman screener implementation. The Huffman encoder encodes 1024 quantized spectral coefficients at once. The number of bits used for the Huffman codewords that encode the 1024 coefficients is additionally transmitted to the decoder (this number is not necessary for the Huffman decoding process). In the decoder, bits are decoded by the Huffman decoder until 1024 spectral coefficients are decoded or the number of Huffman bits transmitted to the decoder are used. There are three scenarios for the decoding process. If 1024 coefficients are decoded and all bits are used, then the Huffman data is valid. If 1024 coefficients are decoded but not all bits are used, the Huffman data is invalid. The last case is when the bits are used before 1024 coefficients are decoded. In this case the Huffman data is also invalid.

The experimental Huffman screening process works as follows for each of the left and right audio channels in a frame. We assume that the data in the side information data regions arrives at the decoder without errors. In addition to the standard PAC side information we transmit the number of bits contained in the Huffman data regions. The decoder then uses the appropriate Huffman codebooks to decode the bits in the Huffman data region. Huffman codewords are decoded until the number of required spectral coefficients are decoded. The PAC bit-buffer is modified such that it supplies zero bits if the end of the frame is reached to prevent wrongly decoded frame from using bits of the next frame. If the number of bits used for decoding 1024 spectral coefficients is not the same as the number of bits contained in the Huffman data region, it is assumed that there were bit errors and the frame is flagged as lost (in error).

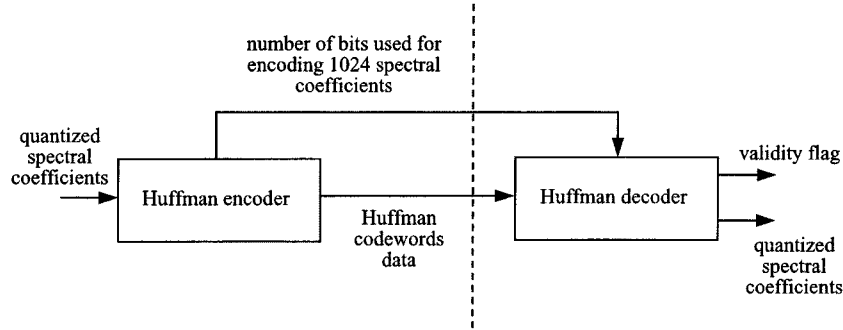


Fig. 6. Experimental setup for the PAC Huffman error screener.

TABLE II
HUFFMAN SCREENER RESULTS FOR PAC FILES
WITH UNDETECTED CRC ERRORS

Audio File	Number of Frames	16-Bit CRC (CCITT)	31-Bit CRC
		$P_{u e}$	$P_{u e}$
sqam55	835	30.76%	23.25%
sqam69	877	30.96%	23.24%
sqam70	504	30.24%	22.80%

We insert lowest Hamming weight CRC errors as in Section III-B into each PAC frame. This is done both for the 16 bit CRC (CCITT) scheme and the 31 bit CRC with the generators given previously. The resulting $P_{u|e}$ are given in Table II. To obtain significance, we ran each file through the simulation 1000 times, with a different random seed for the channel simulator each time. The simulator uniformly chooses a random location in the data frame to insert the CRC error of lowest Hamming weight as above. These results suggest that the current version of the screener allows (on average) only 2 to 3 out of 10 undetected CRC errors to pass through undetected. Note that the best results we obtained for our simple source model in Section III-B were 2 to 3 out of 100.

We also applied random errors, generated as in Section III-A, to the three files above compressed with the same PAC coder. Due to the longer average frame length with real data compared to that considered in Section III-A, we employ a BSC with crossover probability $P_e = 0.01$. The results, in terms of conditional undetected errors, are given in Fig. 7. As before, the screening efficiency is very low for low Hamming weight errors. For increasing error weight, the screener achieves at least an order of magnitude improvement in undetected errors. These results are consistent with the results for the worst case CRC error events in Table II. The trends are also consistent with the theoretical results in Section III-B. Again, we note that the large w results in Fig. 7 are somewhat noisy due to our limited number of trials.

The results in this section assume that the PAC decoder has access to control information indicating the exact number of bits in both channels for the stereo system, and this information is shown as side information in Fig. 6; however, adding two, 16-bit words to the PAC bitstream for these purposes creates significant overhead. For a practical implementation the Huffman screener scheme is modified as follows. Since the PAC bitstream already includes as control information the number of bytes in

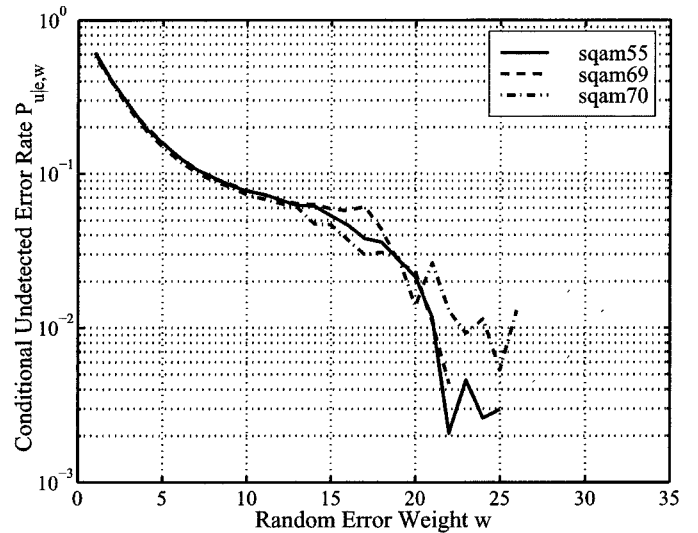


Fig. 7. Conditional undetected error rate $P_{u|e,w}$ efficiency of the PAC Huffman consistency check versus the weight w of random errors introduced by a binary symmetric channel with crossover probability $P_e = 0.01$. Three different audio files were encoded by the PAC audio encoder, corrupted by bit errors, and screened for consistency in the decoder.

a packet, we may add three more bits of control information to specify the total number of bits in the packet. At the end of the decoding process of the whole frame, we check the number of bits used by the Huffman decoder. If more or less bits were used, the Huffman data of the left or right audio channel was invalid and the whole frame is flagged as invalid. This scheme is not as flexible because it can not be detected whether an error was in the left or in the right channel, but it is a practical implementation requiring very little overhead.

IV. OUTER CODE OPTIMIZATION

In most audio transmission applications an “outer” forward error correction (FEC) unit is required to flag transmission errors so that an error mitigation algorithm can be invoked to fill in lost packets by interpolating between neighboring packets. The Huffman code based screening algorithm described in Section III may also be used for this purpose. However, as we have seen, it is more effective when used in conjunction with an outer error detecting block code.

The outer code in its simplest form is often a CRC block code used for error detection only. The outer code can alternatively be

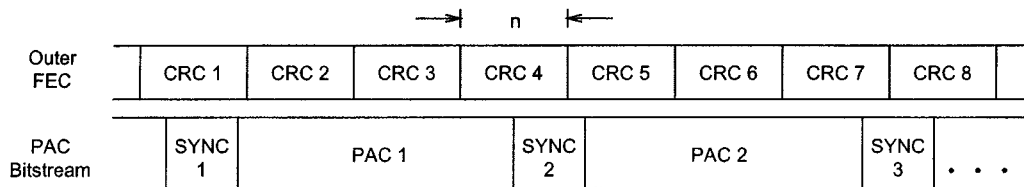


Fig. 8. Illustration of fixed CRC encoding outside of the PAC rate loop.

a Reed–Solomon code [12], used either for error detection only, error correction only, or combinations of error detection and correction. In the case of error correction only, the “failure to decode” signal from a bounded-distance or other Reed–Solomon decoder can provide an error detection signal. The outer code can also be a BCH code [12], or any other binary block code which is used with a decoder that performs both error correction and error detection. However, in this paper we deal only with CRC codes. Interfacing problems between the outer code and the PAC encoder are universal and the proposed solutions to follow also apply in principle to the other outer codes.

Given the variable packet lengths in the PAC bitstream, several different configurations may be utilized as described below. Out of these configurations, those described in Sections IV-A and IV-B have been utilized in the prior art.

In the rest of this section we will use the following terminology. A CRC code is a systematic cyclic block code with block length n bits, k information bits and $n - k$ check bits. The latter bits are appended at the end of a block of information bits and denoted “check CRC” in the figures. A fixed CRC code is a unique code with given length n and rate k/n . There are fixed length codes with different rates and fixed rate codes with different lengths. Any block code may be of full length or a shortened version of the same code [12], [13].

A. Fixed CRC Code

Given the importance of partial frame error flags for error concealment it may be necessary to generate several flags for each PAC packet, especially long packets. One way to achieve such partial flagging is to make the outer FEC asynchronous with the PAC packet and of a fixed, and suitably optimized, code word length at a selected code rate [7]. In this case, the PAC and CRC are not aligned, as indicated by Fig. 8.

Note that this scheme requires separate synchronization frames for the outer FEC and PAC decoder.

Because the PAC and CRC blocks are asynchronous, the CRC encoding can be performed outside the rate loop, as illustrated in Fig. 9.

B. Variable CRC, Single Code

The fixed CRC scheme above, although desirable in that it offers partial flagging, suffers from the problem that a particular FEC block may overlap two adjacent PAC packets and may trigger double packet losses. Furthermore, it requires separate synchronization for the FEC as noted above. As an alternative, a fixed redundancy CRC, i.e., having a fixed number of check bits in the error detecting code, may be added to each audio packet

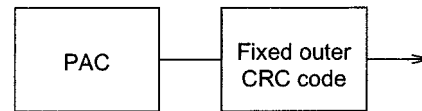


Fig. 9. Fixed bit rate PAC with fixed (both block length and code rate) outer code.

irrespective of its length, as depicted in Fig. 10. The number of bits allocated for CRC are taken from the bit allocation before executing the rate loop. Here we use shortened CRC codewords that match the PAC frame length. The “check CRC” fields in Fig. 10 are all of the same length while the information “packet bits” field vary in length. Thus, in general the actual rate of the CRC code varies from codeword to codeword, but an effective rate can be computed by summing the total number of source bits in a long interval and dividing by the total number of source and parity bits in that interval. Note that shorter audio frames may be over protected by this method, and longer audio frames may be under protected. In addition, there is no partial frame flagging for long frames.

C. Variable CRC, Multiple Codes

We may readily adapt the variable CRC approach to incorporate the benefits of partial flagging observed for the fixed CRC approach, by breaking each PAC frame into multiple CRC blocks. Additionally, the CRC redundancy may be adapted to individual PAC packets, e.g., less redundancy for very short packets and more redundancy for more critical long packets. In this manner, CRC bits can be better matched to the criticality of the audio information. There are a number of schemes which may be utilized in this configuration, as illustrated in Figs. 12–14.

In any of the variable CRC configurations, the number of CRC redundancy in bits are a function of the PAC packet length, and needs to be accounted for in the PAC encoder bit allocation and rate loop. Since the rate loop modifies packet length at each iteration, the corresponding CRC redundancy must be recalculated at each iteration. Consequently, the variable CRC schemes require joint PAC and CRC encoding as shown in Fig. 11, in contrast to the separate encoding employed for the fixed CRC configurations as shown in Fig. 9.

In the scheme of Fig. 11, the CRC choice is given by the length of the final PAC frame after rate loop iterations. This in effect requires a lookup table which is also known to the PAC decoder.

Fig. 12 shows an example with three CRC codewords in sequence in one PAC frame. They are all shortened from the same full length CRC code, thus the check CRC fields A, B, C are of

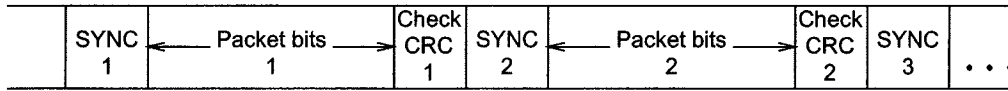


Fig. 10. Illustration of variable length, fixed redundancy CRC inside the PAC rate loop.

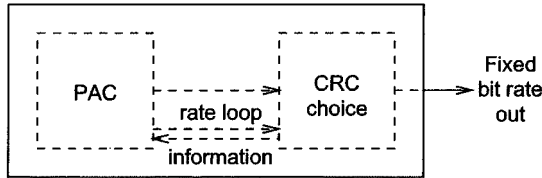


Fig. 11. Variable length outer code with CRC inside the PAC bit allocation. Multiple outer codewords on each PAC frame is also possible.

equal length. Other PAC frames can in principle have a higher or a lower number of CRC codewords depending on frame length.

Fig. 13 shows a different approach with two nested CRC codes, where code a also covers code b. The CRC fields can be of equal or different length.

Finally, Fig. 14 illustrates the case in Fig. 12 with an example with 2 codewords in a frame with different lengths of the check CRC field, i.e., different CRC codes.

V. PERFORMANCE EVALUATION AND COMPARISONS

A. Fixed CRC Systems

1) *Fixed CRC, Singlestream System:* A number of computer simulations and informal listening tests were carried out with the objective of finding out the “Point of Failure” (POF) and “Threshold of Audibility” (TOA) levels for PAC audio coders at a variety of rates such as 16, 32, 48 kb/sec, 64 kb/sec, and 96 kb/sec. These coders use error mitigation algorithms described in Section II.

There are two key issues. At what levels do POF and TOA occur and which block length is preferable for the CRC code or alternatively the Reed–Solomon code. Error mitigation is triggered by a so-called flag signal, i.e., a block is deemed to be in error. This can happen when a CRC is not satisfied i.e., an error is detected) or when a Reed–Solomon decoder fails to decode a certain codeword. As in [7], we study the relationship between the variable PAC frame length and the fixed length CRC (RS) block length and select a preferred CRC block length. For 96 kb/sec, this preferred block length was of the order of 500 bits. However, the optimum length is not very distinct. For audio coders with lower bit rates, the PAC frames are shorter; thus, to maintain similar proportions, the CRC frames should also be shorter.

To evaluate the CRC outer codes we ran a number of software simulations using PAC encoding and decoding or real audio signals. When the CRC outer code detects an error, a flag is passed on to the audio decoder for error mitigation for the VA case. For the LVA, the flag is sent to the LVA instead up to list size L . If no alternative is found that satisfies the CRC, a flag is passed on to the error mitigation algorithm. For simplicity we assume a Gaussian channel. Other system simulation details are described in the tables. The details of the LVA simulations are given in [7], [25].

Several simulations and listening tests were performed with actual audio signals. We use the following performance measures:

- **PAC Flag Rate:** Fraction of PAC frames that are flagged as being at least partially in error, invoking the error mitigation routine.
- **Pair Flag Rate:** Fraction of consecutive pairs of PAC frames that are flagged as being at least partially in error.
- **PAC Frame Erasure Rate:** Fraction of PAC frames that are flagged as being completely in error (erased). No partial information from these frames is used by the error mitigation routine.
- **Pair Erasure Rate:** Fraction of consecutive pairs of PAC frames that are erased.
- **Undetected CRC Errors:** Number of CRC blocks which are declared error free, but in fact contain errors.
- **CRC Block Erasure Rate:** Fraction of CRC blocks which are declared as containing errors.
- **Decoded BER:** Decoded bit error rate at output of VA or LVA.

For simplicity, all simulations were run on an additive white Gaussian noise channel, characterized by the E_D/N_0 , energy per dimension over noise power spectral density. This figure is related to more conventional measures by

$$\frac{E_b}{N_0} = \frac{1}{R} \frac{E_D}{N_0},$$

$$\frac{E_s}{N_0} = N_D \frac{E_D}{N_0},$$

where R is the rate of the code in information bits per dimension (e.g., convolutional code rate for BPSK or QPSK signaling) and N_D is the number of dimensions per symbol (e.g., 1 for BPSK, 2 for QPSK).

Table III shows the various CRC block sizes that were used, along with the corresponding generator polynomials. Each CRC is guaranteed to detect any error pattern with $2t$ or less errors. However, most error patterns with more than $2t$ errors are also detectable. In fact, the only undetectable error patterns are those that are CRC codewords. The fraction of error patterns that are undetectable is therefore 2^{k-n} . The overhead is $n-k$, expressed as a percentage of k . The CRC codes in the top part of the table have roughly 6% overhead, while those in the bottom part have roughly 3–4% overhead. One can see that as a general rule CRC's with longer block sizes have better error detection capability for a given percent overhead.

Table IV shows the audio signals that were used in the simulations along with their lengths, expressed in terms of PAC frames and CRC blocks. These lengths are provided so that the reader may determine the statistical significance of the error, flagging, and erasure rates given in the subsequent tables. Note that the experiments are carried out for CRC codes, but they will also

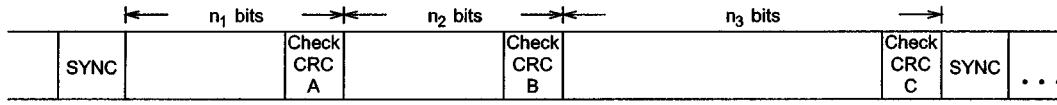


Fig. 12. Illustration of variable CRC with multiple codewords per packet, independent coding.

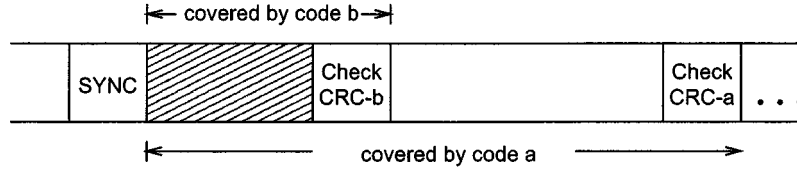


Fig. 13. Illustration of variable CRC with multiple codewords per packet, nested coding.

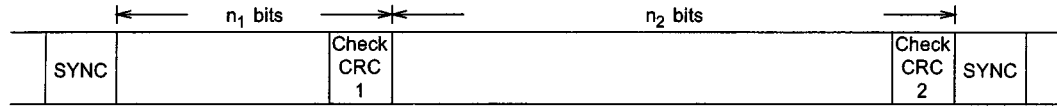


Fig. 14. Illustration of variable CRC with variable redundancy.

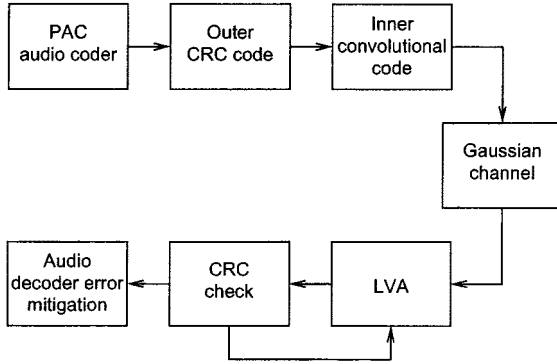


Fig. 15. System block diagram used for the simulations.

give a strong hint at results for RS codes with error detection only and similar flag rates. No LVA (List Viterbi Algorithm) was used in these experiments.

The first results are summarized in Tables V–VII. There we give the PAC flag rate (P_F), the pairwise PAC flag rate (P_{2F}) for 3 different CRC codes for the same channel. These results are given for different signal-to-noise ratios for an additive white Gaussian channel. The decoded bit error rate (BER) and the channel signal to noise ratio E_b/N_0 is also given for the memory 6, rate 2/5 convolutional code with QPSK, see [2], [12].

Informal listening tests were conducted to determine POF and TOA for 16 kb/sec, 32 kb/sec, 48 kb/sec, and 64 kb/sec PAC and also to determine the suitable block length for the CRC-code (RS code). Preliminary screening for suitable source material was performed by listening to the Olympic theme CD tracks (4 1/2 minute long) and two shorter audio samples (female vocal and pop). The Olympic CD track was found to be the most critical from the point of view of susceptibility to channel errors. Moreover, the long length of this CD track leads to fairly stable statistics. Therefore, the Olympic theme CD track was chosen for the purpose of listening tests. Informal listening experiments based on 2 listeners reveal that POF occurs at flag rates about 10% for 16 kb/sec, 32 kb/sec and 48 kb/sec PAC. Likewise, TOA occurs at flag rates about 1% for all three audio coders. These

TABLE III
CRC SIZES. OVERHEAD IS EXPRESSED AS A PERCENTAGE OF k

(n, k)	Generator (octal)	$2t$	2^{k-n}	Overhead (%)
(1018,968)	67 441 634 100 257 771	10	9×10^{-16}	5.2
(507,480)	1 530 225 571	6	7×10^{-9}	5.6
(248,232)	267 543	4	2×10^{-5}	6.9
(127,120)	211	2	8×10^{-3}	5.8
(1016,976)	30 135 372 217 233	8	9×10^{-13}	4.1
(506,488)	1 112 711	4	4×10^{-6}	3.7
(248,240)	435	2	4×10^{-3}	3.3

TABLE IV
AUDIO SIGNALS

Audio Signal	Type	PAC Frames	(248,240)-CRC Blocks	(506,488)-CRC Blocks
iso6	speech	355	2774	1364
iso13	instrumental	712	6111	3006
sqam55	instrumental	1170	11986	5895
sqam70	pop	1424	14210	6989
symp5	classical	10001	102493	50407
slip	pop	12764	132149	64992
diam	pop	14933	156608	77021

results are similar to the ones obtained previously for 96 kbps PAC [7], [25]. The listening tests also favor shorter CRC (or RS) blocks. We believe this happens primarily because P_{2F} is substantially lower for the shorter blocks. A good compromise between CRC-code (RS code) design and listening results is a (248, 240) CRC; i.e., a block length of about 240 bits with a very low frequency of undetected errors. As the audio coder bit rate is lowered, the “optimum” block length for the CRC code also decreases. In particular for the 16 kb/sec audio coder, the block length of the (248, 240) CRC should be considered as an upper limit.

From our informal listening tests we also conclude that the lower rate audio coders are more robust to bit errors at a given decoded bit error rate. This is evident from the relationship between P_F and BER in Tables V–VII. The results in Tables V and VI are statistically significant (in terms of enough error events) for 3.0 and 3.5 dB. For Table VII this is the case for 2.5 and

TABLE V

48 kb/sec PAC. COMPARISONS BETWEEN RATE OF FLAGGED PAC FRAMES (P_F), RATE OF FLAGGED PAC FRAME PAIRS P_{2F} FOR THREE DIFFERENT CRC BLOCK LENGTHS AT THREE DIFFERENT CHANNEL SIGNAL-TO-NOISE LEVELS E_b/N_0

CRC	P_F	P_{2F}	BER	E_b/N_0 (dB)
127, 120	15.3%	2.88%	$2.12 \cdot 10^{-4}$	3.0
248, 240	15.5%	3.24%	$2.18 \cdot 10^{-4}$	3.0
506, 488	17.1%	4.46%	$2.10 \cdot 10^{-4}$	3.0
127, 120	4.19%	0.26%	$4.63 \cdot 10^{-5}$	3.5
248, 240	3.93%	0.44%	$4.31 \cdot 10^{-5}$	3.5
506, 488	5.46%	1.09%	$5.38 \cdot 10^{-5}$	3.5
127, 120	1.00%	0.02%	$9.04 \cdot 10^{-6}$	4.0
248, 240	0.91%	0.07%	$9.77 \cdot 10^{-6}$	4.0
506, 488	1.13%	0.18%	$9.41 \cdot 10^{-6}$	4.0

TABLE VI

32 kb/sec PAC. COMPARISONS BETWEEN RATE OF FLAGGED PAC FRAMES (P_F), RATE OF FLAGGED PAC FRAME PAIRS P_{2F} FOR THREE DIFFERENT CRC BLOCK LENGTHS AT THREE DIFFERENT CHANNEL SIGNAL-TO-NOISE LEVELS E_b/N_0

CRC	P_F	P_{2F}	BER	E_b/N_0 (dB)
127, 120	9.19%	1.27%	$1.97 \cdot 10^{-4}$	3.0
248, 240	10.0%	1.91%	$1.91 \cdot 10^{-4}$	3.0
506, 488	11.5%	3.30%	$2.07 \cdot 10^{-4}$	3.0
127, 120	2.46%	0.18%	$4.36 \cdot 10^{-5}$	3.5
248, 240	2.73%	0.35%	$4.42 \cdot 10^{-5}$	3.5
506, 488	3.28%	0.75%	$5.04 \cdot 10^{-5}$	3.5
127, 120	0.66%	0.05%	$1.08 \cdot 10^{-5}$	4.0
248, 240	0.69%	0.07%	$1.00 \cdot 10^{-5}$	4.0
506, 488	0.91%	0.22%	$1.08 \cdot 10^{-5}$	4.0

TABLE VII

16 kb/sec PAC. COMPARISONS BETWEEN RATE OF FLAGGED PAC FRAMES (P_F), RATE OF FLAGGED PAC FRAME PAIRS P_{2F} FOR THREE DIFFERENT CRC BLOCK LENGTHS AT THREE DIFFERENT CHANNEL SIGNAL-TO-NOISE LEVELS E_b/N_0

CRC	P_F	P_{2F}	BER	E_b/N_0 (dB)
127, 120	14.7%	2.95%	$7.11 \cdot 10^{-4}$	2.5
248, 240	17.9%	5.59%	$8.39 \cdot 10^{-4}$	2.5
506, 488	21.6%	10.3%	$8.13 \cdot 10^{-4}$	2.5
127, 120	4.99%	0.51%	$1.96 \cdot 10^{-4}$	3.0
248, 240	5.70%	1.17%	$2.14 \cdot 10^{-4}$	3.0
506, 488	6.95%	2.57%	$2.06 \cdot 10^{-4}$	3.0
127, 120	1.27%	0.16%	$4.12 \cdot 10^{-5}$	3.5
248, 240	1.53%	0.31%	$5.28 \cdot 10^{-5}$	3.5
506, 488	2.11%	0.75%	$5.44 \cdot 10^{-5}$	3.5

3.0 dB. Given our previous experience, we do not expect big changes in P_F and P_{2F} with longer runs for 4.0 dB.

Tables VIII and IX show the results of simulations where rate-2/5 and rate-4/5 convolutional codes are pushed to the nominal point of failure (PAC Flag Rate $\sim 10^{-1}$), i.e., the E_D/N_0 is chosen so that error-mitigation induced artifacts are clearly audible. For further details on the channel codes, see [6]. These convolutional codes are used in certain proposed digital audio broadcasting (DAB) systems [6], [25].

2) *Fixed CRC, Multistream System:* The results of the previous subsection are for a 96 kb/sec so-called singlestream DAB system in the FM band. Simulations were run for a Gaussian channel. In this section, we examine via simulations the effects

of CRC length on the performance of a different kind of multistream DAB system in the FM band [5]. This system employs a PAC audio coder with a rate of 64 kbps and use $R = 1/2$ convolutional coding with Viterbi algorithm decoding and OFDM with DQPSK in frequency [5]. As the full multistream audio decoder was not available for these simulations, we examine a single sideband using a singlestream PAC audio coder operating at 64 kbps.

Tables VIII–X, with data from [25], show the effect of CRC block length using 96 kb/sec. In general, longer CRC blocks lead to higher PAC frame erasure rates. Indeed, in informal listening tests more error-mitigation induced artifacts can be heard when the (1016, 976)-CRC is used than when the (248, 240)-CRC or (506, 488)-CRC is used.

Table XI lists the audio signals employed in our experiments and provides the number of CRC blocks in each file for several blocklengths. Table XII shows our results from simulations over an AWGN channel, while Table XIII shows our results from simulations over the EIA Urban Fast fading channel described in [5].

B. Variable CRC Results

As we discussed earlier in Section IV, a variable-blocklength CRC encoder must be incorporated into the PAC audio coder in order to ensure a fixed bit rate into the channel. Thus, to fairly compare a variable-blocklength CRC system having total output bit rate R with a (n, k) fixed-blocklength CRC system having input audio bit rate R' , we must set $R' = (k/n)R$.

For the multistream system under consideration, the maximum bit rate in one sideband is $R = 62.5$ kbps, obtained from considering a 400 kHz FM channel, 512 OFDM subcarriers with 80 subcarriers per digital sideband, differential QPSK modulation, and rate 1/2 convolutional coding [5]. For the (n, k) fixed-blocklength CRC codes under consideration, the rates are around $k/n \approx 96\%$, so the PAC encoder bit rate should be set to $R' \approx 60$ kbps. [For the (248, 240) code upon which we focus, R' can be as high as 60.5 kbps in practice, though we will fix it at 60 kbps for our experiments.]

Table XIV lists the audio files employed in our experiments, and Table XV lists the variable-blocklength, fixed-overhead CRC codes [11] we examine, along with estimates of the effective source bit rates R_{eff} for the audio files from Table XIV coded with a PAC encoder containing variable-blocklength CRC's inside the rate loop and operating at total rate $R = 62.5$. We have restricted our attention to CRC codes with the number of parity bits being an integer number of bytes (8 bits) for ease of implementation. Note the rate improvement compared to $R_{\text{eff}} = 60.48$ kb/sec for the (248, 240) fixed CRC.

Table XVI shows the performance of fixed- and variable-blocklength CRC codes in a multistream DAB system in the FM band operating over a channel with AWGN interference. The files from Table XIV were each transmitted over the channel ten times, and the results were averaged.

From the above results, it seems clear that variable-blocklength CRC codes matched to the PAC frames reduce the PAC frame double flag rate P_{2F} by a factor of 2 or more. Furthermore, the PAC flag rates P_F of the matched codes appear to be as good or better than the fixed (248, 240) code, and this

TABLE VIII
FULL-BANDWIDTH CODE (1111, 1111, 1010) NEAR POINT OF FAILURE (PAC FLAG RATE $\sim 10^{-1}$).
 $E_D/N_0 = -1.0$ dB. DECODER IS CONVENTIONAL VITERBI ALGORITHM

CRC Size	Signal	Decoded BER	CRC Block Erasure Rate	Undet. CRC Errors	PAC Flag Rate	Pair Flag Rate
(248,240)	diam	2.0×10^{-4}	1.2×10^{-2}	0	1.3×10^{-1}	2.6×10^{-2}
(506,488)	diam	2.1×10^{-4}	2.6×10^{-2}	0	1.4×10^{-1}	4.0×10^{-2}
(248,240)	slip	2.1×10^{-4}	1.3×10^{-2}	1	1.3×10^{-1}	2.4×10^{-2}
(506,488)	slip	2.1×10^{-4}	2.6×10^{-2}	0	1.5×10^{-1}	4.0×10^{-2}
(248,240)	symph5	2.0×10^{-4}	1.2×10^{-2}	0	1.3×10^{-1}	2.6×10^{-2}
(506,488)	symph5	2.0×10^{-4}	2.4×10^{-2}	0	1.4×10^{-1}	3.5×10^{-2}

TABLE IX
HALF-BANDWIDTH CODE (0110, 1001, 0010) NEAR POINT OF FAILURE (PAC FLAG RATE $\sim 10^{-1}$).
 $E_D/N_0 = 3.4$ dB. DECODER IS CONVENTIONAL VITERBI ALGORITHM

CRC Size	Signal	Decoded BER	CRC Block Erasure Rate	Undet. CRC Errors	PAC Flag Rate	Pair Flag Rate
(248,240)	diam	3.4×10^{-4}	1.2×10^{-2}	2	1.2×10^{-1}	2.1×10^{-2}
(506,488)	diam	3.4×10^{-4}	2.3×10^{-2}	0	1.3×10^{-1}	3.5×10^{-2}
(248,240)	slip	3.3×10^{-4}	1.1×10^{-2}	0	1.2×10^{-1}	2.2×10^{-2}
(506,488)	slip	3.3×10^{-4}	2.2×10^{-2}	0	1.3×10^{-1}	3.2×10^{-2}
(248,240)	symph5	3.2×10^{-4}	1.1×10^{-2}	3	1.1×10^{-1}	1.8×10^{-2}
(506,488)	symph5	3.2×10^{-4}	2.2×10^{-2}	0	1.2×10^{-1}	3.1×10^{-2}

TABLE X
EFFECT OF CRC BLOCK LENGTH. THE CONVOLUTIONAL CODE RATE IS 2/5. $E_D/N_0 = -1.0$ dB. DECODER IS CONVENTIONAL VITERBI ALGORITHM

CRC Size	Audio Signal	Decoded BER	CRC Block Erasure Rate	PAC Frame Erasure Rate	Pair Erasure Rate
(248,240)	iso6	1.8×10^{-4}	1.0×10^{-2}	2.5×10^{-2}	0.0×10^0
(506,488)	iso6	1.7×10^{-4}	2.6×10^{-2}	3.4×10^{-2}	0.0×10^0
(1016,976)	iso6	2.9×10^{-4}	6.6×10^{-2}	7.9×10^{-2}	1.4×10^0
(248,240)	iso13	2.1×10^{-4}	1.2×10^{-2}	6.6×10^{-2}	8.4×10^{-3}
(506,488)	iso13	2.2×10^{-4}	2.5×10^{-2}	6.5×10^{-2}	2.8×10^{-3}
(1016,976)	iso13	1.8×10^{-4}	4.6×10^{-2}	7.6×10^{-2}	1.1×10^{-2}
(248,240)	sqam55	1.9×10^{-4}	1.2×10^{-2}	7.2×10^{-2}	1.0×10^{-2}
(506,488)	sqam55	2.0×10^{-4}	2.3×10^{-2}	7.6×10^{-2}	6.0×10^{-3}
(1016,976)	sqam55	1.9×10^{-4}	4.5×10^{-2}	9.3×10^{-2}	1.2×10^{-2}
(248,240)	sqam70	1.8×10^{-4}	1.2×10^{-2}	6.2×10^{-2}	6.3×10^{-3}
(506,488)	sqam70	2.0×10^{-4}	2.5×10^{-2}	7.2×10^{-2}	6.3×10^{-3}
(1016,976)	sqam70	2.1×10^{-4}	4.9×10^{-2}	9.5×10^{-2}	1.3×10^{-2}

TABLE XI
AUDIO SIGNALS USED FOR THE MULTISTREAM EXPERIMENTS

Audio Signal	Type	PAC Frames	(127, 120) Blocks	(248, 240) Blocks	(506, 488) Blocks	(1016, 976) Blocks
sqam55	instrumental	846	14550	7254	3572	1782
sqam69	electronic	904	15450	7722	3800	1908
sqam70	pop	655	11250	5616	2774	1386

is with better audio quality. We observed very few undetected errors for any of the codes of interest during our experiments, even though for the CRC-8 code we allowed the blocklength to range beyond $2^8 - 1$. These results suggest employing the variable-blocklength CRC-8 to maintain the highest effective audio

source coding rate, or CRC-CCITT codes to maintain the second highest effective audio source coding rate and reduce the undetected error rate of the CRC-8. The rate of undetected errors should be much lower with the long codes in Table XV with increased Hamming distance.

TABLE XII
CRC AND PAC FLAG RATES FOR ALL THE AUDIO FILES OVER AN AWGN CHANNEL MODEL

E_s/N_0 (dB)	CRC Code	Bit-Error Rate	CRC Flag Rate	PAC Frame Erasure Rate	Double PAC Frame Erasure Rate
6.25	(127,120)	$2.72 \cdot 10^{-4}$	0.60%	13.37%	5.84%
	(248,240)	$2.55 \cdot 10^{-4}$	1.10%	14.63%	7.21%
	(506,488)	$2.32 \cdot 10^{-4}$	1.95%	14.11%	7.55%
	(1016,976)	$2.75 \cdot 10^{-4}$	4.58%	18.63%	11.56%
6.5	(127,120)	$1.21 \cdot 10^{-4}$	0.30%	8.51%	4.29%
	(248,240)	$1.15 \cdot 10^{-4}$	0.57%	8.91%	4.41%
	(506,488)	$1.16 \cdot 10^{-4}$	1.04%	7.94%	4.18%
	(1016,976)	$1.42 \cdot 10^{-4}$	2.30%	10.91%	6.64%
7.0	(127,120)	$2.75 \cdot 10^{-5}$	0.07%	4.06%	2.63%
	(248,240)	$2.20 \cdot 10^{-5}$	0.13%	3.89%	2.75%
	(506,488)	$1.64 \cdot 10^{-5}$	0.20%	3.60%	2.69%
	(1016,976)	$2.75 \cdot 10^{-5}$	0.41%	3.89%	2.80%

TABLE XIII
CRC AND PAC FLAG RATES FOR ALL THE AUDIO FILES OVER THE EIA URBAN FAST FADING CHANNEL MODEL

E_s/N_0 (dB)	CRC Code	Bit-Error Rate	CRC Flag Rate	PAC Frame Erasure Rate	Double PAC Frame Erasure Rate
12.0	(127,120)	$2.14 \cdot 10^{-3}$	2.37%	14.40%	11.16%
	(248,240)	$2.01 \cdot 10^{-3}$	3.65%	14.74%	11.33%
	(506,488)	$2.01 \cdot 10^{-3}$	5.82%	15.83%	12.13%
	(1016,976)	$1.92 \cdot 10^{-3}$	7.64%	17.66%	14.59%
12.5	(127,120)	$1.00 \cdot 10^{-3}$	1.29%	11.14%	8.35%
	(248,240)	$9.68 \cdot 10^{-4}$	2.14%	10.34%	7.44%
	(506,488)	$8.72 \cdot 10^{-4}$	3.53%	10.97%	8.47%
	(1016,976)	$1.01 \cdot 10^{-3}$	5.09%	12.11%	9.90%
13.0	(127,120)	$4.54 \cdot 10^{-4}$	0.70%	8.46%	5.84%
	(248,240)	$3.94 \cdot 10^{-4}$	1.03%	7.20%	5.38%
	(506,488)	$3.94 \cdot 10^{-4}$	2.01%	9.31%	7.27%
	(1016,976)	$4.71 \cdot 10^{-4}$	3.36%	9.31%	7.61%
14.0	(127,120)	$9.97 \cdot 10^{-5}$	0.18%	4.97%	3.66%
	(248,240)	$7.46 \cdot 10^{-5}$	0.29%	4.57%	3.43%
	(506,488)	$6.12 \cdot 10^{-5}$	0.14%	4.57%	3.15%
	(1016,976)	$1.06 \cdot 10^{-5}$	1.19%	5.60%	4.41%

TABLE XIV
AUDIO FILES

Audio Signal	Type	PAC Frames	(248, 240) Blocks
sqam55	instrumental	847	6786
sqam69	electronic	905	7254
sqam70	pop	656	4212

TABLE XV
VARIABLE-BLOCKLENGTH, FIXED-OVERHEAD CRC CODES

Common Name	Parity Bits	$g(x)$ (Octal)	R_{eff}
CRC – 8	8	0725	62.23 kbps
CRC – CCITT	16	0210041	61.97 kbps
CRC – 24	24	0140050401	61.71 kbps
CRC – 32 _B	32	00460216667	61.44 kbps

VI. DISCUSSION AND CONCLUSIONS

In this paper, we consider joint source-channel code design issues for audio transmission applications. We evaluate methods

of applying outer CRC codes to PAC, an audio coder that has variable frame length. Similar experiments to these should also be repeated for other outer codes such as Reed–Solomon codes, which are preferred for digital audio broadcasting systems in the AM band with multilevel modulation [4], [26], [27]. We conclude that the variable length outer CRC code integrated with the PAC audio coder is a more effective method of applying the outer CRC code. This method could be used, e.g., for digital audio broadcasting systems in the FM band [5].

We have also introduced a simple error screening method based on checking Huffman code and control information consistency. This is an efficient screener for undetected errors after the CRC decoder. This type of screener does not require any significant extra control information. It can be used, e.g., for digital audio broadcasting systems in the FM band [5]. It can be used both for fixed length and variable length CRC codes as well as with list Viterbi algorithm decoders [7], [18], [25]. The screening algorithm can also be used with Reed–Solomon codes, both in terrestrial digital audio broadcasting systems

TABLE XVI
FIXED- AND VARIABLE-BLOCKLENGTH CRC CODE STATISTICS ON AN AWGN CHANNEL

E_s/N_0	CRC	P_b	CRC Flag Rate	PAC P_F	PAC P_{2F}
6.25	(248, 240)	2.607×10^{-4}	1.099×10^{-2}	8.634×10^{-2}	1.792×10^{-2}
	CRC – 8	2.467×10^{-4}	7.467×10^{-2}	7.467×10^{-2}	6.320×10^{-3}
	CRC – CCITT	2.632×10^{-4}	7.670×10^{-2}	7.670×10^{-2}	7.318×10^{-3}
	CRC – 24	2.573×10^{-4}	7.907×10^{-2}	7.907×10^{-2}	8.067×10^{-3}
	CRC – 32_B	2.655×10^{-4}	8.227×10^{-2}	8.227×10^{-2}	7.484×10^{-3}
6.5	(248, 240)	1.212×10^{-4}	5.539×10^{-3}	4.481×10^{-2}	7.110×10^{-3}
	CRC – 8	1.203×10^{-4}	3.796×10^{-2}	3.796×10^{-2}	1.538×10^{-3}
	CRC – CCITT	1.052×10^{-4}	3.484×10^{-2}	3.484×10^{-2}	1.414×10^{-3}
	CRC – 24	1.213×10^{-4}	4.099×10^{-2}	4.099×10^{-2}	2.370×10^{-3}
	CRC – 32_B	1.206×10^{-4}	4.311×10^{-2}	4.311×10^{-2}	2.744×10^{-3}
7.0	(248, 240)	2.123×10^{-5}	1.118×10^{-3}	8.762×10^{-3}	5.405×10^{-4}
	CRC – 8	2.648×10^{-5}	8.596×10^{-3}	8.596×10^{-3}	1.663×10^{-4}
	CRC – CCITT	2.111×10^{-5}	8.098×10^{-3}	8.098×10^{-3}	0.000×10^{-4}
	CRC – 24	1.800×10^{-5}	7.683×10^{-3}	7.683×10^{-3}	4.158×10^{-5}
	CRC – 32_B	2.576×10^{-5}	9.884×10^{-3}	9.884×10^{-3}	0.000×10^{-5}

in the AM band and satellite based digital audio broadcasting systems using concatenated convolutional codes and outer Reed–Solomon codes.

ACKNOWLEDGMENT

The ideas on Huffman code based error screening resulted from discussions between D. Sinha and C.-E. W. Sundberg. Thanks are due to D. Sinha for providing PAC audio coding software, listening, and technical discussions, and to P. Kroon for technical discussions.

REFERENCES

- [1] C.-E. W. Sundberg, D. Sinha, P. Kroon, and B.-H. Juang, "Technology advances enabling 'In-Band on Channel' DSB systems," in *Proc. Int. Conf. Broadcast Asia*, Singapore, June 1998, pp. 289–296.
- [2] N. S. Jayant and E. Y. Chen, "Audio compression: Technology and applications," *AT&T Tech. J.*, vol. 74, no. 2, pp. 23–34, Mar.–Apr. 1995.
- [3] D. Sinha, J. D. Johnston, S. M. Dorward, and S. R. Quackenbusch, "The Perceptual Audio Coder (PAC)," in *The Digital Signal Processing Handbook*, V. K. Madisetti and D. B. Williams, Eds: CRC/IEEE Press, 1998, pp. 42-1–42-17.
- [4] H.-L. Lou, D. Sinha, and C.-E. W. Sundberg, "Multistream transmission for hybrid IBOC-AM with embedded/multidescriptive audio coding," *IEEE Trans. Broadcast.*, Sept. 2002, to be published.
- [5] C.-E. W. Sundberg, D. Sinha, D. Mansour, M. Zarrabizadeh, and J. N. Laneman, "Multistream hybrid in band on channel systems for digital audio broadcasting in the FM band," *IEEE Trans. Broadcast.*, vol. 45, no. 4, pp. 410–417, Dec. 1999.
- [6] B. Chen and C.-E. W. Sundberg, "Complementary punctured pair convolutional codes for digital audio broadcasting," *IEEE Trans. Commun.*, vol. 48, no. 11, pp. 1829–1839, Nov. 2000.
- [7] —, "List Viterbi algorithm for continuous transmission," *IEEE Trans. Commun.*, vol. 49, no. 5, pp. 784–792, May 2001.
- [8] D. Sinha and C.-E. W. Sundberg, "Unequal Error Protection (UEP) for perceptual audio coders," in *Proc. IEEE Int. Conf. Acoustics, Speech, & Signal Proc. (ICASSP)*, vol. 5, Phoenix, AZ, April 1999, pp. 2423–2426.
- [9] T. Cover and J. Thomas, *Elements of Information Theory*: John Wiley & Sons, 1991.
- [10] W. H. Press *et al.*, *Numerical Recipes in C: The Art of Scientific Computing*: Cambridge University Press, 1993.
- [11] D. J. Costello, Jr., J. Hagenauer, H. Imai, and S. B. Wicker, "Applications of error-control coding," *IEEE Trans. Information Theory*, vol. 44, no. 6, pp. 2531–2560, Oct. 1998.
- [12] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1983.
- [13] G. C. Clark, Jr. and J. B. Cain, *Error-Correction Coding for Digital Communications*. New York: Plenum Press, 1981.
- [14] J. Hagenauer, "Rate compatible punctured convolutional codes (RCPCC-codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.
- [15] J. Hagenauer, N. Seshadri, and C.-E. W. Sundberg, "The performance of rate compatible punctured convolutional codes for digital mobile radio," *IEEE Trans. Commun.*, vol. 38, no. 7, pp. 966–980, July 1990.
- [16] R. V. Cox, J. Hagenauer, N. Seshadri, and C.-E. W. Sundberg, "Sub-band speech coding and matched convolutional channel coding for mobile radio channels," *IEEE Trans. Acoustics, Speech & Signal Processing*, vol. 39, no. 8, pp. 1717–1731, August 1991.
- [17] N. Seshadri and C.-E. W. Sundberg, "Generalized Viterbi algorithms for error detection with convolutional codes," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM)*, vol. 3, Dallas, TX, Nov. 1989, pp. 1534–1538.
- [18] —, "List Viterbi decoding algorithms with applications," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 313–323, Feb./Mar./Apr. 1994.
- [19] C. Nill and C.-E. W. Sundberg, "List and soft symbol output Viterbi algorithms: Extensions and comparisons," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 277–287, Feb./Mar./Apr. 1995.
- [20] See white papers under Technology/Regulatory. [Online]. Available: www.ibiquity.com.
- [21] D. Sinha and J. D. Johnston, "Audio compression at low bit rates using a signal adaptive switched filterbank," in *Proc. IEEE Int. Conf. Acoust. Speech & Signal Proc.*, vol. 2, May 1996, pp. 1053–1056.
- [22] R. Bauer and J. Hagenauer, "Turbo-FEC/VLC-Decoding and its application to text compression," in *2000 Conf. Information Sci. Syst.*, Princeton, NJ, Mar. 2000, pp. WA6–WA11.
- [23] —, "Iterative source/channel-decoding using reversible variable length codes," in *Proc. IEEE Data Compression Conf. (DCC)*, Snowbird, UT, Mar. 2000, pp. 93–102.
- [24] Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 158–162, Feb./Mar./Apr. 1995.
- [25] B. Chen and C.-E. W. Sundberg, "An integrated error correction and detection system for digital audio broadcasting," *IEEE Trans. Broadcast.*, vol. 46, no. 1, pp. 68–78, Mar. 2000.
- [26] S.-Y. Chung and H. L. Lou, "Multilevel RS/convolutional concatenated coded QAM for hybrid IBOC-AM broadcasting," *IEEE Trans. Broadcast.*, vol. 46, no. 1, pp. 49–59, Mar. 2000.
- [27] J. N. Laneman and C.-E. W. Sundberg, "Reed–Solomon coding algorithms for digital audio broadcasting in the AM band," *IEEE Trans. Broadcast.*, vol. 47, no. 2, pp. 115–122, June 2001.



J. Nicholas Laneman (S'93) was born in St. Charles, MO, USA. He received B.S. degrees in electrical engineering and in computer science from Washington University, St. Louis, MO, in 1995. At the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, he earned S.M. and Ph.D. degrees in electrical engineering in 1997 and 2002, respectively. Since 2002, he has been on the faculty of the Department of Electrical Engineering, University of Notre Dame, South Bend, IN, USA.

From 1995 to 2002, Dr. Laneman was affiliated with the Department of Electrical Engineering and Computer Science and the Research Laboratory of Electronics, MIT, where he held a National Science Foundation Graduate Research Fellowship and served as both a Research and Teaching Assistant. He received the MIT EECS Harold L. Hazen Teaching Award in 2001. During 1998 and 1999 he was also with Lucent Technologies, Bell Laboratories, Murray Hill, NJ, both as a Member of the Technical Staff and as a Consultant, developing robust source and channel coding methods for digital audio broadcasting. He has 5 patents pending. His current research interests lie in the broad areas of communications and signal processing, with particular emphasis on wireless communications and networking. He is a Member of Sigma Xi, Eta Kappa Nu, Tau Beta Pi, and the Institute of Electrical and Electronics Engineers (IEEE).



Carl-Erik W. Sundberg (S'69–M'75–SM'81–F'90) was born in Karlskrona, Sweden on July 7, 1943. He received the M.S.E.E. and the Dr. Techn. degrees from the Lund Institute of Technology, University of Lund, Lund, Sweden in 1966 and 1975, respectively.

Before 1976 he held various teaching and research positions at the University of Lund. During 1976 he was an ESA Research Fellow at the European Space Research and Technology Centre (ESTEC), Noordwijk, The Netherlands. From 1977 to 1984 he was a Research Professor (Docent) in the Department of

Telecommunication Theory, University of Lund, Lund, Sweden. During this time he also held positions as consulting scientist at Ericsson and Saab. From 1984 to 2000 he was a Distinguished Member of Technical Staff (DMTS) at Bell Laboratories, Murray Hill, NJ, USA and during 2001 he was a DMTS at Agere Systems, Murray Hill, NJ, USA. Currently, he is a Senior Scientist at iBiquity Digital Corp., Warren, NJ, USA. In 1978 he started his consulting company SundComm which has been involved in studies of error control and digital modulation techniques for Swedish Defense as well as a number of companies and international organizations. His research interests include source coding, channel coding, digital modulation, fault-tolerant systems, digital mobile radio, digital audio broadcasting, spread-spectrum, digital satellite systems and optical communications. He has published over 95 journal papers and contributed to over 140 conference papers. He has 67 patents, granted and pending. He is a coauthor of *Digital Phase Modulation*, (New York: Plenum, 1986), *Topics in Coding Theory*, (New York: Springer-Verlag, 1989) and *Source-Matched Digital Communications* (New York: IEEE Press, 1996).

Dr. Sundberg has been a member of the IEEE European–African–Middle East Committee (EAMEC) of Communications Society (ComSoc) from 1977 to 1984. He is a member of ComSoc Communication Theory Committee. He has also been a member of the Technical Program Committees for the International Symposium on Information Theory, St. Jovite, Canada, October 1983; for the International Conference on Communications, ICC'84, Amsterdam, The Netherlands, May 1984; for the 5th Tirrenia International Workshop on Digital Communications, Tirrenia, Italy, September 1991; for the International Telecommunications Symposium, ITS'94, Rio de Janeiro, Brazil, August 1994; and for the International Zurich Seminar on Broadband Communications, IZS2000, Zurich, Switzerland, February 2000. He has organized and chaired numerous sessions at a number of international meetings. He has been a member of the International Advisory Committee for the ICCS'88, '90, '92, and '94 in Singapore. He served as Guest Editor for the IEEE Journal on Selected Areas in Communications (JSAC) in 1988–1989. He is a member of the Swedish URSI Committee (Svenska Nationalkommittén för Radiovetenskap). In 1986 he and his coauthor received the IEEE Vehicular Technology Society's Paper of the Year Award and in 1989 he and his coauthors were awarded the Marconi Premium Proc. IEE Best Paper Award. Two of his papers were selected for inclusion in the IEEE Communications Society 50th Anniversary Journal Collection Volume 2002. He is a Fellow of the IEEE since 1990 and is listed in Marquis Who's Who in America.

Christof Faller received the M.S. (Ing.) degree in electrical engineering from ETH Zurich, Switzerland, in 2000. During his studies, he worked as an independent consultant for Swiss Federal Labs, applying neural networks to process parameter optimization of sputtering processes, and spent one year at the Czech Technical University (CVUT) in Prague. In 2000 he started to work as a consultant for the Speech and Acoustics Research Department, Bell Laboratories, Lucent Technologies. After one and a half years consulting, partially from Europe while pursuing other interests in music, he became a Member of Technical Staff, focusing on new techniques for audio compression applied to digital satellite radio broadcasting.

Recently, he moved with the majority of researchers from the Acoustics and Speech Research Department from Bell Labs to the newly formed Media Signal Processing Research Department of Agere Systems.

His research interests include generic signal processing, specifically audio coding, control engineering, and neural networks. He won the first prize in the ABB (Asea Brown Boveri) youth science contest organized Swiss nationwide in honor of the 100 year existence of ABB (formerly BBC) in 1991.