THE METHOD OF TYPES AND LOSSY DATA-EMBEDDING†

Mark G. Kokes, Jerry D. Gibson, Geoffrey C. Orsak, and Victor J. Stolpman

Department of Electrical Engineering
Southern Methodist University
Dallas, TX 75275-0338
{kokes, gibson, gorsak, stolpman}@seas.smu.edu

ABSTRACT

Many digital signals possess statistical properties that can be exploited to increase the amount of usable information either stored with them in a fixed file size or transmitted with them over a fixed capacity channel. In this paper, we develop a new technique for embedding digital information into such signals. Using the *method of types* as a tool to analyze the statistical nature of digital signals, we implement a novel scheme which is capable of embedding significantly greater amounts of information than has been possible with traditional steganographic approaches. We are able to achieve these increased rates by not requiring the data to be either *hidden* or decoded *error free*. As an example, in wideband speech processing trials using ITU G.722 at 48, 56, and 64 kbps, we are able to embed up to 800 bits/second of additional information with practically no impact on the decoded speech signal. This additional bandwidth can be used for various low data rate applications such as control channel information, text transfer, or security and watermarking.

1. INTRODUCTION

Over the last decade, and concurrent with the growth of the Internet, digital media has sprung to the forefront of consumer interest. Already offering several distinct advantages over its analog counterpart, digital media has presented itself more recently as a candidate for yet another new technology, data-embedding. Data-embedding, as its name implies, suggests that digital information (i.e. data, text, audio, or video) can be inserted into the content of another digital signal (i.e. data, text, audio, or video).

To date, there are numerous applications for data-embedding. One of the most important applications is copyright protection of digital information. In the business sector, there is growing interest in a reliable, transparent mechanism to identify ownership and distribution channels for particular digital data sequences. In addition, many distributors of digital content are also looking for a cost effective solution for the transport of various control, reference, and descriptive signals which in turn can be used to differentiate as well as track access to their products and services. Many believe that data-embedding is the answer to these proposed problems. One application, which is synonymous with data-embedding, is the communication of secondary data sources through so-called *covert* channels. In this scenario, data-embedding algorithms are used to securely *hide* relatively small amounts of potentially encrypted (i.e. *secret*) information within a host digital signal. In this paper, we relax the transparency and secrecy constraints in the data-embedding problem for the purpose of simply increasing channel throughput. By approaching the problem in this manner, we hope to embed significantly greater amounts of information into digital signals.

The goal of this paper is to present a new technique for embedding information into digital signals. For the purpose of implementation, we use the ITU wideband speech codec, G.722 [1], at various bitrates to demonstrate our algorithm's performance and compare its embedding capacity. It is important to note, that though we simulate our algorithm using a speech codec, this new methodology is capable of embedding information into many types of digital signals and is not restricted to digitally compressed audio. In the sections that follow, we begin with brief descriptions of state-of-the-art techniques for embedding information into digital audio signals. We focus on digital audio data-embedding techniques in this paper simply for the sake of comparison. We follow these descriptions with background information regarding universal receiver design [2] via the method of types [3]. By re-examining some principle theories of universal receivers, we establish the mathematical groundwork for our data-embedding approach. We conclude this paper with an examination of our latest research efforts on data-embedding.

2. PREVIOUS WORK

In this section, we discuss some of the previous work done in audio data-embedding. As motivation for our solution, we present only the background information that is necessary to substantiate the novelty of our approach to embedding information within digital signals.

2.1. Audio Data-Embedding

Most of the published work in the area of data-embedding has concentrated on image and video applications. Unfortunately, a majority of the published work on audio data-embedding must be ascertained from patent literature.

†This research was supported, in part, by NSF Grant NCR-9796255.

This often makes it difficult to keep up with the state-of-the-art in audio data-embedding schemes. In this section, we summarize some of the available audio data-embedding literature and comment on it.

One of the first data-embedding techniques used was least significant bit replacement [4, 5]. Such techniques lead to problems as the precision of the host signal decreases toward 1 bit/sample. Other techniques have been devised based on a phase coding approach [6]. In these algorithms, the phase of the Fourier transform coefficients of a frame of the host signal is altered in a meaningful way. Echo coding has also been proposed for audio data-embedding [6]. In this method, multiple decaying echoes are placed in the spectrum of the host signal such that by using cepstral analysis, one can locate and decode the nature of the embedded symbol. Many spread-spectrum approaches have also been proposed for audio data-embedding applications [6]–[11]. Some authors propose embedding information as spread-spectrum (i.e. "colored") noise. Several other methods [12, 13] use spectral component replacement to embed data transparently into digital audio signals. Even simpler techniques have been attempted where one modifies signal peaks within a segment of host audio in order to force the signal to fall within embedded data-specified quantization ranges [14]. In this way, the embedded information is surmised by observing trends in the quantization patterns of the host signal. Many of these techniques are already present in commercial products. The common factor among most of these techniques, is that they are *limited* in their ability to achieve significant embedded throughput. Common techniques achieve embedded bitrates of 8-50 bps with corresponding error rates in the embedded bitstream between 10^{-3} and 10^{-2} [6]–[14].

2.2. Data-Embedding: One Viewpoint

The underlying principles of our new data-embedding scheme lie in work based on universal receiver and classifier design. Similar to the wireless communications problem, embedding data into digital signals can be thought of as transmitting information over a communication channel which is corrupted by *strong* interference and channel effects. Such a model for the case of a binary communication system is given as,

$$\begin{aligned} H_0 :&\quad s_0 + \eta(t), \quad \text{Symbol 0 Transmitted} \\ H_1 :&\quad s_1 + \eta(t), \quad \text{Symbol 1 Transmitted.} \end{aligned} \tag{1}$$

In this model, a data symbol is hypothesized (i.e. H_x) to be transmitted from one of two sources. The binary data symbol to be transmitted, s_x, corresponds to the data symbol that is to be embedded into the host signal, $\eta(t)$. The strong interference is representative of the host signal. Channel effects correspond to any pre- or post-processing done to the combined signal (i.e. $s_x + \eta(t)$).

Our idea is to apply universal classification techniques toward embedded bit detection using an M-ary hypothesis testing procedure. We note that the statistical properties of the host signal vary significantly from frame-to-frame and thus in order to achieve reliable detection of embedded content, we must have a detector which is robust to the changing characteristics of both the host and embedded signals. By using the method of types and an information theoretic distance measure, we look for the minimum distance between observed, empirical data distributions and distributions based on training data sequences of sufficient length. In our algorithm, the minimum distance between the empirical distribution of the test data sequence and that of the M empirical distributions derived from the training data sequences indicate the presence of one particular type of embedded symbol.

In this approach, some observed empirical distributions of host signal frames can be quite different from any empirical distribution derived by observing the host signal over a long period of time. This could lead to a false detection of embedded symbols at the decoder. If this happens, it is sometimes possible to adapt the embedding scheme to counteract such events. At other times, it is impossible to alter the content of a segment in a way that can be surmised by the decoder and produce the correct embedded symbol at the decoder without affecting the original content of the data frame. In these cases, our algorithm suffers embedded bit detection errors. In any event, the decoder is intelligent enough to adapt to the changing characteristics of both the host and embedded signals while working with only limited knowledge (i.e. with knowledge of only the combined signal, $s_x + \eta(t)$). Using this algorithm, the probability of getting an unworkable frame of data decreases as the framesize of the data segment increases. Consequently, as the size of the data frame increases, the *rate* at which data can be embedded into the host signal decreases. Thus, there is a tradeoff that must be balanced in order to achieve the desired data-embedding goal (i.e. maximized embedded throughput with minimal error probability).

With the general principles of our data-embedding approach stated, we now reference relevant background information and examine the mathematical building blocks of our new solution for embedding information into digital audio signals.

2.3. Type-Based Detection

We transform the data-embedding problem into a signal classification problem that can be cast as a M-ary hypothesis testing problem in which each hypothesis represents a different random source from which it is assumed any one embedded symbol is derived. In this research, we note that our channel model (i.e. $\eta(t)$) is rarely ever stationary and thus it varies with time depending on the characteristics of the *host signal*. Thus, there is an inherent need to use an adaptive detector to extract the embedded information. If we could somehow parameterize the channel model, then we could use the General Likelihood Ratio Test to detect the embedded content. Unfortunately, such a solution produces mediocre results at best [15].

How can we solve the problem of robust detection of the embedded content? It has been shown that under general circumstances, type-based detectors have asymptotic performance measures comparable to those of the clairvoyant detector [16, 17]. The type characterizing the various hypotheses can be estimated from only the sample (i.e. observed) data. Because the optimal clairvoyant detector depends only on the true probability distributions, it is apparent that empirical histograms (i.e. *types*) must be calculated from training data and compared to the empirical histogram of the observed test data in order to differentiate between hypotheses. When faced with classifying observed data frames based on the training data types, optimal performance is not guaranteed by merely calculating the empirical likelihood ratio [16]. Rather, it has been shown that better performance can be obtained by concatenating the training data for each hypothesis with the observed test data [15]. In this way, one can assess how different the type derived from these longer sequences are from the types of (a) the training data and (b) the observation, by utilizing the Kullback-Leibler distance measure. The rather surprising form of the hypothesis test leads to an exponential increase in the probability of detection with increased numbers of observations (i.e. samples of test data). Moreover, with the definition of a rejection region, the decay rate can be controlled by the user. Furthermore, it has been shown that no detector based solely on the test and training data sequences has a larger asymptotic rejection probability decay rate for the same exponential error decay rate [17]. This result implies that with increasing numbers of test data samples, we are less likely *not* to be able to differentiate the type (i.e. empirical histogram) of the observed test data frame from any one of the M types derived from the training data.

Consider the following M-ary hypothesis testing problem:

$$\begin{array}{rccl} H_1: & X^n \sim P_1 & \text{Source 1} \\ H_2: & X^n \sim P_2 & \text{Source 2} \\ \vdots & \vdots & \vdots \\ H_M: & X^n \sim P_M & \text{Source M} \\ H_{M+1}: & \text{Rejection Region} & \end{array} \quad (2)$$

where the test vector X^n is of length n. We assume that under hypothesis H_i, the test vector, X^n, is generated by a source with probability measure P_i (unknown to the detector). In addition, due to the absence of an accurate statistical model for the M sources, we assume that there exist training vectors T_i^N, $i = 1, 2, ...M$ of length N from each of the M possible data sources. Therefore, the classification between source types is made on the basis of the test vector, X^n, and the training vectors, T_i^N, $i = 1, 2, ...M$.

We now expand on the mathematical quantities used to differentiate the correct source density from those which lead to false detections of embedded data symbols. It has been shown that the asymptotically optimal Generalized Likelihood Ratio Test (GLRT) for determining if a finite alphabet test sequence, X^n, arose from the same source as a finite alphabet training sequence, T_i^N, is:

$$h_i(X, T_i) = \frac{1}{n} \log \left\{ \frac{\sup_{Q_1, Q_2} Q_1(X^n) Q_2(T_i^N)}{\sup_Q Q(X^n, T_i^N)} \right\}, \quad (3)$$

where Q_1, Q_2, and Q denote source densities [18].

From an intuitive point of view, one can see that if the data sequences X^n and T_i^N arise from the same source, then h_i will converge to zero in the limit. Alternatively, if the data originated from different sources, then h_i will converge to some constant greater than zero which will allow for discrimination between the proposed M hypotheses. It was originally shown by Gutman [17] that this test offers asymptotically optimal performance over a very wide range of source statistics.

Unfortunately, due to the requirement of the supremum calculations in 3, the detector is *not practical* to implement. However, through the use of the method of types, the log-likelihood ratio is reduced to

$$\begin{aligned} h_i(X, T_i, \lambda) &= d_{KL}(Q_{X^n}, Q_{(X^n, T_i^N)}) \\ &+ \frac{N}{n} d_{KL}(Q_{(T_i^N)}, Q_{(X^n, T_i^N)}) - \lambda. \end{aligned} \quad (4)$$

The quantities $Q_{(T_i^N)}$, Q_{X^n}, and $Q_{(X^n, T_i^N)}$ represent the types of the data vectors, T_i^N, X^n, and the concatenated vectors (X^n, T_i^N). These types represent the empirical (*histogram*) estimates of the joint statistics of the data vectors. The distance metric is the functional d_{KL}, the well known divergence or *relative entropy* between the probability mass functions in its argument. λ is a positive constant chosen to satisfy some design criterion (i.e. rejection region). In addition to the above, we offer an alternative interpretation for $h_i(X, T_i, \lambda)$ in terms of the entropies of the types,

$$\begin{aligned} h_i(X, T_i, \lambda) &= \frac{N+n}{n} H(Q_{(X^n, T_i^N)}) - H(Q_{X^n}) \\ &- \frac{N}{n} H(Q_{T_i^N}) - \lambda. \end{aligned} \quad (5)$$

The above expression for the discriminant function in terms of the entropies is computationally preferable for on-line processing as the entropies of the training sequences can be pre-computed. Note that the joint type of X^n and T_i^N in terms of the marginals is defined as

$$Q_{(X^n, T_i^N)} = \frac{n Q_{X^n} + N Q_{T_i^N}}{n+N}. \quad (6)$$

With only a few general assumptions, the type based detector has been shown to have asymptotic performance measures comparable to those of the clairvoyant detector [15]. In addition, in [2], the behavior of the type-based detector relative to the amount of training data used has been explicitly shown. These demonstrations provide evidence that the type-based detector can in fact achieve globally optimum performance even with *limited* amounts of training data. These results are particularly applicable to the experiments conducted in this research.

3. CURRENT RESEARCH

In the following, we describe the fundamentals of our new data-embedding technique. A block diagram of our embedding codec pair is presented in Fig. 1. To begin,

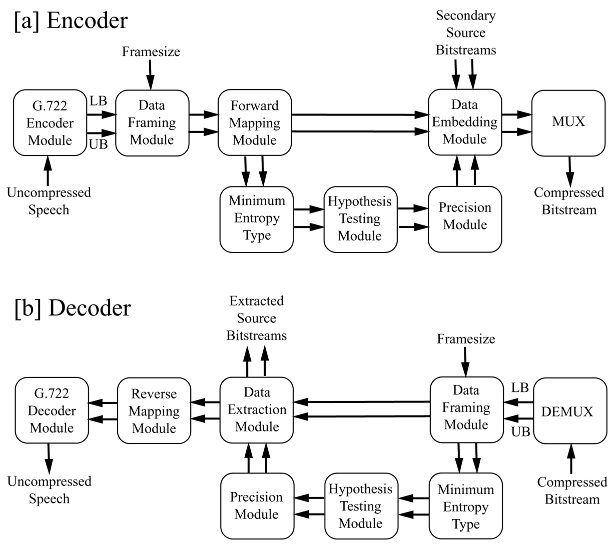

Fig. 1: Block diagrams for data-embedding encoder and decoder pair. Double lines are representative of dual bitstreams (i.e. LB: lower band, UB: upper band).

we examine the data-embedding encoder. In the encoder, Fig. 1 (a), a wideband speech signal (i.e. 16 kHz sampling, 256 kbps) acts as the input to the ITU G.722 module. The G.722 module compresses the digital input signal using one of three modes of operation (i.e. 48, 56, or 64 kbps output). Histograms for the lower band G.722 codewords from each of the three modes of operation can be seen in Fig. 2 (a,c,e). A histogram for the upper band G.722 codewords from all G.722 modes of operation can be seen in Fig. 3 (a). Following the compression stage, the data-embedding procedure begins. The compressed G.722 codewords are framed and mapped in a pre-defined manner. The mapping function is particular to the preceding compression scheme and is necessary in order to take full advantage of type-based detection at the decoder. In this case, the mapping function counters the folded binary coding scheme of G.722. The mapping is 1:1 and thus it is completely reversible. In Fig. 2 (b,d,f), histograms of the mapped G.722 lower band codewords are presented. Fig. 3 (b) shows a histogram of the mapped G.722 upper band codewords. We see that the resulting mapped histograms appear more natural in their structure.

After mapping, it is necessary to determine how many bits can be embedded into the lower band and upper band frames independently. This decision is made in the precision module. The number of bits embedded in each data frame changes on a frame-by-frame basis. This adaptation is done to counteract adverse statistical properties present in some data frames. It is important to note that the encoder and decoder must come to the same conclusion regarding the bit precision of the embedded symbols. Only that information available to the decoder is permitted in formulating the number of bits embedded in a frame of data.

To adaptively determine the precision of the embedded symbol, we use a minimum entropy approach. The encoder

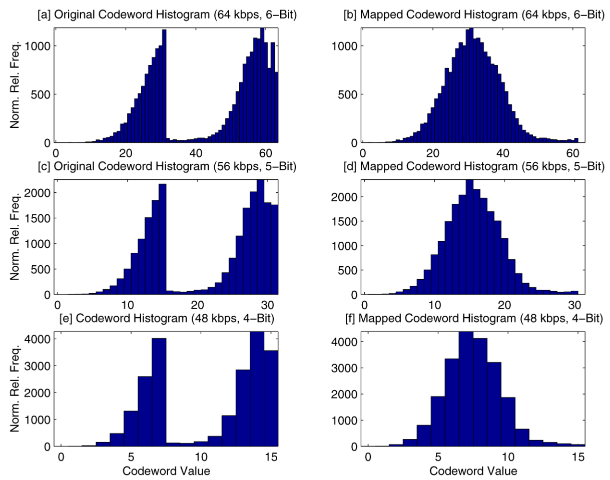

Fig. 2: Original and mapped G.722 lower band codewords for 48, 56, and 64 kbps modes. Note that G.722 uses folded binary codewords (i.e. [a], [c], and [e]). The histograms, [a]-[f], are a product of 60 G.722 compressed wideband speech signals (30 Male, 30 Female) taken from the TIMIT speech corpus. The mapped codeword histograms in [b], [d], and [f] are representative of the *base master types* of the lower band signals for each of G.722's three modes of operation.

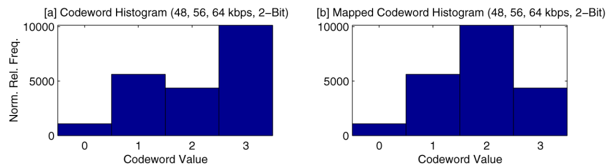

Fig. 3: Original and mapped G.722 upper band codewords for 48, 56, and 64 kbps modes. Note that G.722 uses folded binary codewords (i.e. [a]). The histogram in [b] is also representative of the *base master type* of the upper band signals for each of G.722's three modes of operation.

forms a minimum entropy type from the current data frame. The minimum entropy type is constructed in the following manner. The original test type is modulated to all possible bin locations and made symmetric. Each time the entropy of the new symmetric type is calculated. Based on the bin number of the minimum entropy symmetric type, the symmetric type is re-centered. The distance from each training type to the centered symmetric type is calculated. Based on the location of the training type closest to the re-centered symmetric type, a value for the offset of the re-centered symmetric type is determined. Using this offset, a penalty constant is derived, and this constant dictates the bit precision to be used for embedding information into the current data frame. In this way, if the penalty value calculated at the encoder is substantial, less bits are embedded into the current codeword frame. If the penalty term is small, more bits (i.e. up to $log(M)$) can be embedded into the current frame. Using such a scheme allows the precision module to be adaptive. This process is also reproducible at the decoder using the received (i.e. embedded) data frame.

Once the precision for the current lower band and up-

Table I: Results for embedding data into lower and upper bands of G.722 bitstream at 48, 56, and 64 Kbps. At 48, 56, and 64 Kbps, the lower band is coded using 4, 5, and 6 bits/sample. At 48, 56, and 64 Kbps, the upper band is coded using 2 bits/sample. Simulation results are averaged over 10 iterations per sequence using random binary secondary sources in both the lower and upper bands.

Speech Sequence (M)ale (F)emale	Lower Band Simulations Bit Errors / Embedded Bits						Upper Band Simulations Bit Errors / Embedded Bits		
	48 kbps		56 kbps		64 kbps		48/56/64 kbps		
	240 bps	315 bps	400 bps	530 bps	560 bps	740 bps	200 bps	300 bps	500 bps
01 (M)	0/640	0/825	1/1051	2/1471	2/1473	3/1961	0/531	1/785	18/1320
02 (M)	2/720	1/998	1/1252	3/1650	2/1750	5/2309	0/624	3/925	9/1560
03 (M)	0/673	1/919	0/1132	1/1515	1/1601	0/2052	0/570	2/845	27/1425
04 (M)	0/555	0/712	0/914	0/1203	0/1251	0/1611	0/456	1/675	17/1140
05 (M)	0/543	0/699	0/898	0/1200	0/1242	0/1633	0/453	0/670	9/1125
06 (F)	0/561	0/1073	0/1352	1/1775	1/1971	2/2479	0/678	7/1005	61/1695
07 (F)	0/541	0/705	1/900	0/1181	0/1211	0/1660	1/450	4/665	35/1125
08 (F)	1/697	2/894	1/1140	1/1513	0/1571	1/2118	1/570	9/845	58/1425
09 (F)	1/711	4/951	2/1211	5/1597	3/1697	11/2223	1/603	5/895	36/1500
10 (F)	1/601	3/793	3/1001	7/1297	6/1421	14/1895	4/507	3/750	46/1260
Totals	5/6242	11/8569	9/10857	20/14402	15/15188	36/19941	7/5442	35/8060	316/13575
% Error	0.08	0.13	0.08	0.14	0.10	0.18	0.13	0.43	2.33

per band frames is determined, the actual data-embedding step can occur. After obtaining the bits to be embedded from the secondary source module, the secondary bit sequences are formed into symbols. Based on these symbols and the locations of the current frame's types (i.e. both lower and upper bands), the data frames are modulated in a way corresponding to the embedded data symbols. Modulation in the frame domain corresponds to modulation in the type domain. The modulation is performed based on one of $log(M)$ gridded patterns which corresponds to the embedded precision chosen for each of the current data frames. Note this procedure occurs independently for both the lower and upper data frames. After embedding the secondary symbols, the framed sequences are multiplexed and transmitted over the channel to the decoder. In our simulations the media over which we transmit the encoded data frames is assumed to be *error free*.

The decoder can be seen in Fig. 1 (b). Similar to the encoder, the decoder buffers the lower band and upper band frames and uses the minimum entropy approach discussed previously to adaptively determine the number of bits embedded within the current frame. The decoder uses only the received data frame to determine the embedded bit precision. Because the procedure used to determine the embedded bit precision is shift (i.e. modulation) tolerant, the decoder comes to the same conclusion as the encoder. Up to the decision regarding the embedded bit precision, the decoder is *exactly* like the encoder. The difference between the two lie in the data extraction procedure. Using the embedded precisions surmised from the encoded frames and knowledge of the grid system in place for all possible embedded bit precisions, the data extraction module performs a hypothesis testing process utilizing equations (5) and (6) to determine the embedded symbol contained within the current data frame. With knowledge of the embedded symbol and the embedded bit precision, the decoder demodulates the received data frame to recover the contents of the mapped data frame. An example of the encoding/decoding process is summarized in Fig. 4. With the embedded symbol and mapped data frame secure, the decoder reverse maps the lower and upper codeword frames and buffers each until enough samples are present to send through the G.722 decoder.

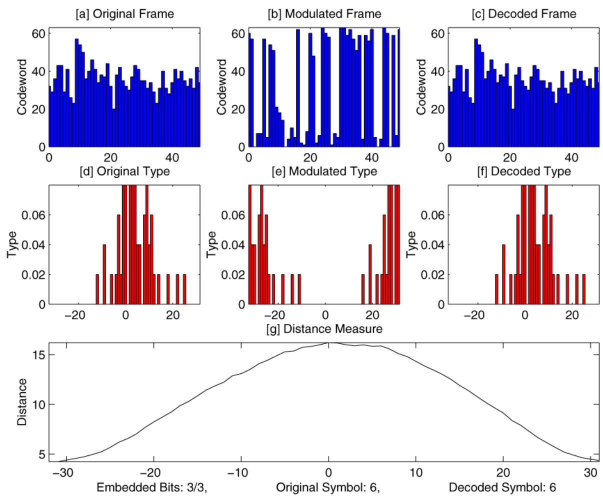

Fig. 4: Example encoding/decoding procedure (i.e. frame and type domains) for one lower band frame using 6 bits per sample, (i.e. G.722 in 64 kbps mode), a framesize of 50 samples, 3 embedded bits, and an embedding symbol of 6. [a], [b], [d], and [e] represent the encoding process. [b], [c], [e], [f], and [g] represent the decoding process.

4. SIMULATIONS AND RESULTS

Numerous results from our wideband speech processing trials are presented in Table I. Table I is split into two independent portions, lower band simulations and upper band simulations. This is done to demonstrate the independence of the embedding process between the two bands. In the lower band simulations, two embedded bitrates are examined within the confines of each operational mode of G.722 (i.e. 48, 56, and 64 kbps output). For each input sequence (i.e. 1-10), the average number of embedded bit errors incurred and the average number of bits embedded during 10 trials over that sequence are shown. From this information, the average embedded bit error rate is calculated and displayed in terms of percent error for each output bitrate and corresponding operational mode. In the upper band, three embedded bitrates are examined. These simulated results are valid for all of the operational modes of G.722 since in each mode two bits per sample is used to compress the upper band.

Because the two bands are addressed independently, one can get a feel for the embedding capacity of our new data-embedding procedure by combining any one result from the lower band simulations with any one result from the upper band simulations. The tradeoff demonstrated by these results is that of embedded rate versus error probability in the embedded bitstream. One can choose the desired combined embedding rate and have a pre-determined error probability or vice versa.

5. CONCLUSIONS

In this paper, we have introduced and experimented with a new lossy data embedding scheme. We demonstrate that in speech processing trials, significantly greater amounts of information can be embedded compared to traditional steganographic approaches for digital audio. These increased rates are achieved by not requiring the data to be both *hidden* or decoded *error free*. We stress that this approach is not only appropriate to the speech processing trials presented in this paper but is also applicable to many types of digital data (fixed file size or streaming).

6. ACKNOWLEDGEMENTS

Some of the ideas presented here were clarified in discussions with Justin Ridge and Alison Campbell. The authors would like to take this opportunity to thank these individuals.

REFERENCES

[1] ITU, "Recommendation G.722: 7 KHz audio-coding within 64 kbit/sec," Nov., 1988.

[2] V. Stolpman and G. Orsak, "Type-Based Receiver for Wideband CDMA," *Proc. of IEEE Wireless Comm. and Networking Conf.*, Sept. 1999.

[3] I. Csiszar, "The Method of Types," *Trans. Info. Theory*, Vol. 44, No. 6, pp. 2505–2523, Oct. 1998.

[4] P. Bassia and I. Pitas, "Robust Audio Watermarking in the Time Domain," *Proc. IX Euro. Sig. Proc. Conf.*, Vol. I, pp. 25–28, Sept. 1998.

[5] M. Cooperman and S. Moskowitz, "Steganographic Method and Device," *U.S. Patent 5,613,004*, 1997.

[6] W. Bender, et al, "Techniques for Data Hiding," *IBM Sys. J.*, Vol. 35, No. 3&4, pp. 313–336.

[7] R. Pruess, et al, "Embedded Signaling," *U.S. Patent 5,319,735* , 1994.

[8] C. Lee, K. Moallemi, R. Warren, "Method and Apparatus for Transporting Auxillary Data in Audio Signals," *U.S. Patent 5,822,360*, 1998.

[9] D. Moses, "Simultaneous Transmission of Data and Audio Signals by Means of Perceptual Coding," *U.S. Patent 5,473,631*, 1995.

[10] M. Swanson, et al, "Robust Audio Watermarking Using Perceptual Masking," *Signal Processing*, Vol. 66, No. 3, pp. 337–355, May, 1998.

[11] M. Swanson, et al, "Current State of the Art and Future Directions for Audio Marking," *IEEE Int. Conf. on Multimedia Comp. and Sys.*, Vol. 1, Npp. 19–24, 1999.

[12] J. Tilki and A. Beex, "Encoding a Hidden Digital Signature onto an Audio Signal Using Psychoacoustic Masking," *Proc. Seventh Int. Conf. on Sig. Proc. Apps. and Tech.*, pp. 476–480, 1996.

[13] J. Wolosewicz, "Apparatus and Method for Encoding and Decoding Information in Audio Signals," *U.S. Patent 5,774,452*, 1998.

[14] J. Woloeswicz and K. Jemeli, "Apparatus and Method for Encoding and Decoding Information in Analog Signals," *U.S. Patent 5,828,325*, 1998.

[15] N. Warke, "Adaptive Signal Classification Using Information Theoretic Techniques," *Ph.D. Thesis*, George Mason University Fall, 1997.

[16] J. Ziv, "On Classification With Empirically Observed Statistics and Universal Data Compression," *IEEE Trans. Inform. Theory*, Vol. IT-34, pp. 278–286, Mar., 1988.

[17] M. Gutman, "Asymptotically Optimal Classification for Multiple Tests With Empirically Observed Statistics," *IEEE Trans. Inform. Theory*, Vol. IT-35, pp. 401–408, Mar., 1989.

[18] V. Poor, *An Introduction to Signal Detection and Estimation*, New-York: Springer-Verlag, 1988.