Iterative Decoding of GSM Signals *

David A. Garr†, Jan Bajcsy, and Hisashi Kobayashi
Department of Electrical Engineering
Princeton University

Abstract

We propose a new receiver structure for GSM, where the received signal is decoded by the repetitive and iterative use of channel equalization and channel decoding. This structure is derived by treating the GSM system as a concatenated system, in which the convolutional encoder is its outer coder, and the GMSK modulator together with the multipath channel constitute its inner coder. We then apply the iterative scheme that we have recently proposed for a class of concatenated systems [1, 2]. In our scheme both inner and outer decoders adopt a soft-output Viterbi algorithm (SOVA) with reliability index, which Hagenauer and Hoeher [8] originally introduced for inner decoding in a concatenated system. We define distance metrics that are suitable to the iterative decoding.

We have constructed a simulator of the proposed receiver, using the COSSAP simulation language. Preliminary results of our simulation experiment will be discussed.

1 Introduction

GSM (Groupe Speciale Mobile, or the Global System for Mobile Communication) system is a Pan-European digital cellular [11] standard. It makes use of a convolutional code for channel error correction, and GMSK (Gaussian-filtered minimum shift keying), a bandwidth efficient modulation scheme. GMSK is similar to traditional MSK (minimum shift keying) [10], except that prior to modulation the signal is passed through a Gaussian lowpass filter. The signal spectrum is shaped to make it more bandwidth efficient. In the time domain, its effect is to spread the signal over multiple bit periods and introduce a *controlled* amount of ISI (inter-symbol interference). Thus, the GMSK signal can be viewed as mathematically equivalent to a correlative-level or partial-response coded signal [9].

Further ISI is introduced to the received signal by multipaths in the radio channel. We may model the combined effects of the Gaussian filtering and the multipath channel as a single impulse response $h(t)$ of finite duration given by

$$h(t) = h_{GMSK}(t) * h_{chan}(t), \tag{1}$$

where $*$ denotes convolution. By sampling $h(t)$ at $t = kT$ (where T is the symbol interval), we obtain the discrete-time impulse response $\{h_n\}$.

Figure 1: A typical GSM receiver

In typical coherent reception of the GSM signal, the GMSK signal is treated as a quadrature modulated signal [11] In a typical GSM receiver, shown in Figure 1 inner decoder, often refered to as a Viterbi equalizer, attempts to remove the ISI caused by the Gaussian filtering and multipath channel. The soft equalizer output is then de-interleaved and fed to the outer decoder, a Viterbi decoder. The Viterbi decoder produces a maximum-likelihood sequence estimate of the transmitted data [5]. The traditional receiver adopts a one-path decoding algorithm, in the sense that the output of the outer decoder is delivered immediately to "data sink".

In our proposed iterative decoding scheme, we adopt a soft decision in the outer decoder, which we return to the inner decoder. The inner decoder can use the additional information associated with the outer decoder's soft decision, which we call side information, to correct errors and ambiguities that may exist in the previous output of the inner decoder. This possibly improved input is then fed to the outer decoder, which will then presumably produce an improved version of the decoder output. We iterate this alternate decoding by the inner decoder and the outer decoder until we resolve all errors and ambiguities, or until no further enhancement in the decoder output is obtained. The power of iterative decoding was first noted by Berrou et al. [3] in their work on Turbo codes involving parallel concatenation of two (systematic recursive) convolutional encoders. Independently of our work [1, 2], iterative decoding for serially concatenated systems has been proposed by Picart et al. [13].

*This work has been supported, in part, by grants from the National Science Foundation, the New Jersey Commission on Science and Technology, Asahi Chemical Industry, Co., and the Ogasawara Foundation for the Promotion of Science and Engineering. The first author was supported by the 1997 Princeton Summer Institute Program, sponsored by the National Science Foundation and Princeton University

†Email: davegarr@ee.princeton.edu.

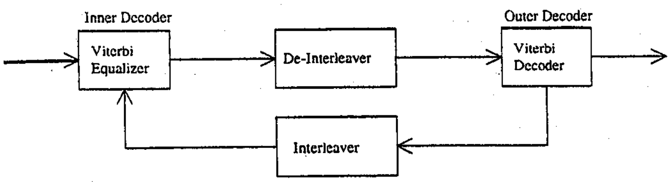

Figure 2: A new receiver structure with iterative decoding

2 Soft Decision and Iterative Decoding

In its basic form, the Viterbi algorithm produces only a maximum likelihood (ML) estimate of the transmitted sequence given the received signal. It provides no information regarding the degree of correctness (or reliability) of each symbol in the estimated sequence. The Viterbi algorithm can be modified, however, as suggested in [8] to also produce a reliability value indicating the decoder's confidence associated with each output symbol.

By modifying the typical GSM receiver of Figure 1 we now allow the Viterbi decoder to correct some of the errors the Viterbi equalizer may make. Should the equalizer be given some of this additional information one might receive from such a powerful error-correcting code, it might be able to improve on its estimation. We therefore wish to pipe the decoder's output values back to the equalizer as shown in Figure 2. The equalizer can then use these "side information" values, along with the original received values, to decode the signal once again. With the additional information obtained from the powerful GSM convolutional code, the first decoder can provide an improved signal estimate at its output. The decoder subsequently decodes the convolutionally-encoded data, and once again we can attempt to improve upon the values at its output by piping them back to the equalizer. These iterations can be performed any number of times, though the incremental reductions in the BER (bit error rate) become smaller as the number of iterations increases.

As the values from the Viterbi decoder will be fed back to the Viterbi equalizer, it becomes desirable for both the Viterbi blocks to output soft values. Though in the end we are still only interested in the most likely transmitted symbols, the Viterbi equalizer can make better use of the side information if it consists of soft values.

2.1 The inner decoder for iterative decoding

As detailed in Forney [5] and Kobayashi [9], the Viterbi algorithm for a channel with ISI models the transmitted sequence $\{s_n\}$ as a path in a *trellis diagram* with 2^{L-1} Markov states, where L is the duration of the impulse response g_n. The algorithm examines all possible paths on the trellis diagram, but by keeping track of only 2^{L-1} viable candidate paths as an ML estimate. For each of the 2^{L-1} states at any time in the trellis diagram, it chooses the path with the lowest cumulative metric. These cumulative distance metrics are called "path length metrics."

2.1.1 Path length metric for the inner decoder

As the conventional Viterbi algorithm [5] and Soft-Output Viterbi algorithm (SOVA) [8] assume only the received data sequence to be available at the inputs, they are not able to make use of side information values. We therefore face the problem of how to weight this additional information in the path length metric.

We abstract ourselves from the MSK-modulated sequence $\{x_n\}$ and its noisy replica $\{y_n\}$, and instead deal with derotated sequences $\{\alpha_n\}$ and $\{z_n\}$.

As is well known, an optimal decision rule is the maximum *a posteriori* probability (MAP) decision rule that, in our context, will choose the most likely N-bit sequence vector

$$\underline{\alpha}_i = (\alpha_{i1}, \alpha_{i2} \ldots \alpha_{iN}), \quad \underline{a}_i \in \mathbf{C}, \tag{2}$$

among $\mathbf{C}$, the set of all possible transmitted sequences, given the received sequence

$$\underline{z} = (z_1, z_2 \ldots, z_N). \tag{3}$$

by maximizing the *a posteriori* probability:

$$\max_{\underline{\alpha}_i \in \mathbf{C}} P\{\underline{\alpha}_i | \underline{z}\}, \tag{4}$$

Using Bayes' formula, the above can be rewritten as

$$\max_{\underline{\alpha}_i \in \mathbf{C}} P\{\underline{\alpha}_i | \underline{z}\} = \max_{\underline{\alpha}_i \in \mathbf{C}} \frac{P\{\underline{z}|\underline{\alpha}_i\} P\{\underline{\alpha}_i\}}{P\{\underline{z}\}}. \tag{5}$$

When either the *a priori* probability $P\{\underline{\alpha}_i\}$ is unknown or all possible sequences are assumed equally likely, the above MAP decision rule reduces to the maximum likelihood (ML) decision rule, which maximizes the conditional probability or likelihood function:

$$\max_{\underline{\alpha}_i \in \mathbf{C}} P\{\underline{z}|\underline{\alpha}_i\}. \tag{6}$$

For each possible data sequence $\underline{a}_i$, we have the corresponding noiseless received signal $\underline{s}_i$ given as

$$s_{i,n} = \alpha_{i,n} * g_n, \tag{7}$$

where $*$ is the convolutional sum with respect to the time index n. Given the impulse response $\{g_n\}$, there is a one-to-one correspondence between vectors $\underline{\alpha}_i$ and $\underline{s}_i$, thus we can write

$$\max_{\underline{\alpha}_i \in \mathbf{C}} P\{\underline{z}|\underline{\alpha}_i\} = \max_{\underline{\alpha}_i \in \mathbf{C}} P\{\underline{z}|\underline{s}\} \tag{8}$$

where the maximization on the right hand side is subject to (7).

The complex-valued (rotated) white Gaussian noise vector $\underline{v}$ has its real and imaginary components $\Re\{v_n\}$

and $\Im\{v_n\}$ independently and identically distributed with mean 0 and variance σ^2. Hence, we can write

$$P\{\underline{z}|\underline{s}_i\} = \left(\frac{1}{2\pi\sigma^2}\right)^N e^{-\|\underline{z}-\underline{s}_i\|^2/2\sigma^2}. \qquad (9)$$

Thus, as is well known, the ML decision rule reduces to a minimum distance decision rule for a channel with additive white Gaussian noise. The Viterbi algorithm is a computationally efficient method that performs this ML decision rule or corresponding minimum decision rule in a recursive manner.

For the iterative decoding procedure that we propose here, however, the side information from the previous iteration provides us with an estimate of the *a priori* probability $P\{\underline{\alpha}_i\}$ which is not equal for all possible data sequences $\underline{\alpha}_i$. We can no longer ignore it as described in (6), thus the likelihood function is not applicable as decision criteria for the iterative decoding procedure. We should instead estimate the most likely data sequence by returning to the original MAP (4). Using (5), the MAP estimate for the iterative scheme should now be defined as

$$\max_{\underline{\alpha}_i \in \mathbf{C}} P\{\underline{z}|\underline{\alpha}_i\}\hat{P}\{\underline{\alpha}_i\}. \qquad (10)$$

In our case, this maximization problem reduces to

$$\max_{\underline{\alpha}_i \in \mathbf{C}} \left(\frac{1}{2\pi\sigma^2}\right)^N \exp\left\{-\frac{1}{2\sigma^2}\|\underline{z}-\underline{s}_i\|^2 - \ln \hat{P}\{\underline{\alpha}_i\}\right\}. \qquad (11)$$

Thus, the minimum distance decision rule is the one that minimizes the following path metric:

$$\min_{\underline{\alpha}_i \in \mathbf{C}} \frac{1}{2\sigma^2}\|\underline{z}-\underline{s}_i\|^2 - \ln \hat{P}\{\underline{\alpha}_i\} =$$

$$= \min_{\underline{\alpha}_i \in \mathbf{C}} \sum_{n=1}^{N} \left[\frac{|z_n - s_{i,n}|^2}{2\sigma^2} - \ln \hat{P}\{\alpha_{1,n}\}\right]. \qquad (12)$$

To properly evaluate the last expression (12), we need an estimate of the noise variance σ^2.

To obtain an estimate of the noise variance, we take advantage of the known training sequence $\{c_n\}$ transmitted in the middle of each packet. During the training sequence, we replace the data bits $\{a_n\}$ by $\{c_n\}$, and using the channel response g_n we can model the received bits z_n as

$$z_n = \gamma_n * g_n + v_n. \qquad (13)$$

If we replace the impulse response $\{g_n\}$ with our estimate $\{\hat{g}_n\}$, we can estimate the channel noise as

$$\hat{v}_n = z_n - \gamma_n * \hat{g}_n. \qquad (14)$$

We take the averages of

$$\Re\{\hat{v}_n\}^2 = \Re\{z_n - \gamma_n * g_n\}^2$$

and

$$\Im\{\hat{v}_v\}^2 = \Im\{z_n - \gamma_n * g_n\}^2$$

over an appropriate length of the training sequence output.

The training sequence $\{c_n\}$ and the effective channel response $\{\hat{g}_n\}$ are 26 bits and 5-bits long, respectively. The convolved sequence is 30 bits long (i.e., $0 \leq n \leq 29$), but we must discard the first four bits ($0 \leq n \leq 3$) and the last four bits ($26 \leq n \leq 29$) because these are *transient* periods. Since the real and imaginary component of $\hat{v}_n$ should be statistically independent, we have a total of 44 independent samples. Accepting that the noise has zero mean[1], we have the the following expression for the sample variance:

$$\widehat{\sigma^2} = \frac{1}{44}\sum_{n=4}^{25} |z_n - \gamma_n * \hat{g}_n|^2. \qquad (15)$$

We substitute this estimate $\widehat{\sigma^2}$ of the noise variance into our path metric equation (12).

2.1.2 Reliability index for the inner decoder

We now require an algorithm for calculating the probability that each of our estimated bits $\hat{a}_n$ is estimated correctly. We turn to the Soft Output Viterbi algorithm (SOVA) of Hagenauer and Hoeher [8], which provides a basis for calculating such a reliability measure.

Whenever two paths on the trellis diagram merge in a common state, the Viterbi algorithm chooses the path with the smaller cumulative path metric as the *survivor* and discards the other. The SOVA calls for calculating the probability that a possible mistake is committed in this decision. Let us consider two paths $\underline{s}_1$ and $\underline{s}_2$ that merge at time k. From (12), and using our estimate of the noise variance, the cumulative metric $M_{i,k}$ of path $\underline{s}_i$ up to time k, which we denote $\underline{s}_{i,k}$ may be given as

$$M_{i,k} = \sum_{n=1}^{k} \frac{\|z_n - s_{i,n}\|^2}{2\widehat{\sigma^2}} - \ln(P\{\hat{\alpha_{i,n}}\}). \qquad (16)$$

Using the equations developed in Section 2.1.1, and following the arguments of Hagenauer and Hoeher [8], the probability that path $\underline{s}_{i,k}$ is correct can be approximated as

$$P\{\underline{s}_{i,k}\} \propto e^{-M_{i,k}}. \qquad (17)$$

Given two merging paths $\underline{s}_{1,k}$ and $\underline{s}_{2,k}$, and assuming that the former has a smaller metric (i.e. $M_{1,k} < M_{2,k}$ and hence $P\{\underline{s}_{1,k}\} > P\{\underline{s}_{2,k}\}$), the probability that the inner decoder's (i.e., Viterbi equalizer's) selection of $\underline{s}_{1,k}$ as the survivor is, in fact, an incorrect decision is given by

$$\psi_{1,k} = \frac{P\{\underline{s}_{2,k}\}}{P\{\underline{s}_{1,k}\} + P\{\underline{s}_{2,k}\}} \qquad (18)$$

[1] If the assumption that the noise has zero mean needs to be examined, then we must make a slight modification in the above path metric and the expression for the sample variance, by including the sample mean of the noise.

$$= \frac{e^{-M_{2,k}}}{e^{-M_{1,k}} + e^{-M_{2,k}}} \quad (19)$$

$$= \frac{1}{1 + e^{M_{2,k} - M_{1,k}}}. \quad (20)$$

If $\underline{s}_{2,k}$ has a smaller metric (i.e. $M_{1,k} > M_{2,k}$), the probability of an incorrect decision by the Viterbi equalizer is

$$\psi_{2,k} = \frac{1}{1 + e^{M_{1,k} - M_{2,k}}}. \quad (21)$$

We can obtain a general expression

$$\psi_{i \cap j,k} = \frac{1}{1 + e^{|M_{j,k} - M_{i,k}|}} \quad (22)$$

for any pair of paths $\underline{s}_i$ and $\underline{s}_j$ that merge at time k. Here the notation $i \cap j$ in the subscript signifies path index i or j whose metric is smaller than the other.

To each bit $s_{i,n}$ $(0 \leq n \leq k)$ of a candidate path $\underline{s}_i$ on the trellis diagram, we assign a corresponding *reliability* value $r_{i,n}$, which indicates the probability that the bit decision made is incorrect. Due to the fact that, in such a binary system, two paths which merge at time k will always have the same kth data bit bits [5], each bit is assigned an initial reliability value of 0 when it is decided upon. As time progresses and the Viterbi equalizer selects one path over the other each time it encounters two survived paths merge at a common state in the trellis. It makes 2^{L-1} such decisions at each time index k. For a bit position where the binary value of the discarded path (i.e., the nonsurvivor) is the equivalent to the corresponding bit of the survivor, a potential error would have no effect on the decoded bit. For those bit positions where the values of the non-survivor and the survivor differ, however, we must lower our confidence and update the reliability values. Following [8], we adopt the following updating rule at decision at time k:

$$r_{i,n} \longleftarrow r_{i,n}(1 - \psi_{i,k}) + (1 - r_{i,n})\psi_{i,k}, \quad (23)$$

where $r_{i,n}$ on the right hand side is the old reliability value assigned at some time before k.

2.2 The outer decoder for iterative decoding

The second major component of the iterative receiver is the Viterbi decoder. This block is considerably less complex than the Viterbi equalizer. The equalizer must estimate the complex valued channel impulse response for each transmitted data packet and generate the possible received data sequences on the fly. The decoder, on the other hand, works with a binary valued impulse response predefined by the generating polynomials of the convolutional code.

Nevertheless, the Viterbi decoder needed for our iterative scheme differs from a traditional Viterbi-based convolutional decoder in some significant areas. Our soft input data consists of probability values rather than the Gaussian distributed variables assumed in the traditional scheme. We must therefore compute our own path length metric. As we are interested in piping data back to the Viterbi equalizer, we must also make our decoder capable of outputting soft decision values. In order for the Viterbi equalizer to make use of this data, however, these soft output values must correspond to convolutionally encoded data. The decoder output must therefore output encoded values, rather than the decoded data the traditional Viterbi decoder outputs.

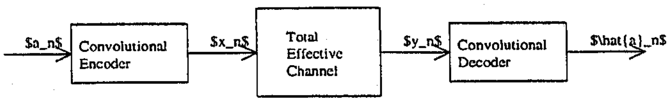

Figure 3: The outer decoder (i.e., the Viterbi decoder)

Throughout this section we work with the model of Figure 2.2. The total effective channel includes everything which comes between the encoder and decoder, including the interleaver, GMSK modulator, multipath channel, Viterbi equalizer, and de-interleaver. Note that we re-use certain alphabetical symbols. While some of the variables in this section may be analogous to those in Section 2.1 sharing the same alphabetical symbol, they represent distinct signals and should not be confused.

2.2.1 Path length metric for the outer decoder

If the input to the convolutional decoder can be modeled as the encoded signal corrupted by additive white Gaussian noise, as is assumed in the traditional analysis, the Euclidean distance metric provides an optimal path metric. The input to our outer decoder, however, is the output of the de-interleaver that follows the inner decoder (i.e, our Viterbi equalizer), which produces output in the form of probability values. We must therefore compute a new path length metric.

We denote the input to the outer decoder $\{y_n\}$. The goal of the outer decoder is to provide a best estimate of $\{a_n\}$, the transmitted data stream prior to convolutional, given this input. Let the length of this unencoded sequence $\{a_n\}$ be denoted by N. The GSM standard [7] adopts a convolutional code of rate 1/2, thus the length of the channel encoded sequence $\{x_n\}$ is of length $2N$, as is the decoder input $\{y_n\}$. Again we use vector notation to represent these sequences:

$$\underline{a} = (a_1, a_2 \ldots a_N), \quad \underline{x} = (x_1, x_2 \ldots x_{2N}), \quad \text{etc.} \quad (24)$$

The Viterbi algorithm adopted by the outer decoder should select the sequence $\underline{a}_i = (a_1, a_2 \ldots a_N)$ that maximizes *a posteriori* probability

$$\max_{\underline{a}_i \in \mathbf{C}} P\{\underline{a}_i | \underline{y}\},$$

where $\mathbf{C}$ now represents the set of all possible binary sequences of length N that can be generated by the source.

Since the relation between the source sequence $\underline{a}$ and the channel encoded sequence $\underline{x}$ is a one-to-one mapping, we may write

$$\max_{\underline{a}_i \in \mathbf{C}} P\{\underline{a}_i | \underline{y}\} = \max_{\underline{a}_i \in \mathbf{C}} P\{\underline{x}_i | \underline{y}\}. \quad (25)$$

The input stream $\underline{y}$ to our outer decoder is the stream of soft outputs from the inner decoder, hence it is in the form of

$$y_n = \hat{P}\{x_n = 1\}. \tag{26}$$

We can rewrite the *a posteriori* probability of the symbol at bit n a possible transmitted sequence $\underline{x}_i$ as

$$P\{x_{i,n}|y_n\} = \begin{cases} 1 - y_n, & x_{i,n} = 0 \\ y_n, & x_{i,n} = 1 \end{cases} \tag{27}$$

$$= 1 - x_{i,n} - y_n + 2x_{i,n}y_n. \tag{28}$$

Assuming the de-interleaver shuffles bits sufficiently randomly, we treat the de-interleaved reliability probabilities as independent from symbol to symbol. We then have the following simple product form for the posterior probability of sequence $\underline{x}_i$, given the output from the de-interleaver $\underline{y}$:

$$P\{\underline{x}_i|\underline{y}\} = \prod_{n=1}^{2N} P\{x_{i,n}|y_n\}$$

$$= \prod_{n=1}^{2N}(1 - x_{i,n} - y_n + 2x_{i,n}y_n). \tag{29}$$

Thus the most likely transmitted data sequence $\underline{a}_i$ should be the one that attains the following maximization criterion:

$$\max_{\underline{a}_i \in \mathbf{C}} P\{\underline{x}_i|\underline{y}\} = \max_{\underline{a}_i \in \mathbf{C}} \prod_{n=1}^{2N}(1 - x_{i,n} - y_n + 2x_{i,n}y_n)$$

$$= \max_{\underline{x}_i \in \mathbf{C}} \exp\left(\sum_{n=1}^{2N} \ln\,(1 - x_{i,n} - y_n + 2x_{i,n}y_n)\right). \tag{30}$$

The above maximization problem is then equivalent to the following minimum distance decision rule:

$$\min_{\underline{x}_i \in \mathbf{C}} \sum_{n=1}^{N} -\ln\,(1 - x_{i,n} - y_n + 2x_{i,n}y_n). \tag{31}$$

Note that if we take the expectation of this metric with respect to $\{y_n\}$, we will have the *equivocation entropy* [14] associated with the channel encoded sequence, given the inner decoder output.

2.2.2 Reliability index for the outer decoder

To calculate the reliability of the decoded bits, we follow an argument analogous to Section 2.1.2. We turn again to the soft output Viterbi algorithm (SOVA) [8].

From (31) above, the cumulative path length metric at bit k for a given path $\underline{s}_i$ can be given by

$$M_{i,k} = \sum_{n=1}^{k} -\ln\,(x_{i,n} + y_n - 2x_{i,n}y_n). \tag{32}$$

Using the argument that led to (17), the probability that path $\underline{x}_{i,k}$ (the first k bits of the path $\underline{x}_i$) is correct can be expressed as

$$P\{\underline{x}_i\} \propto e^{-M_{i,k}}. \tag{33}$$

Let us consider two paths $\underline{x}_1$ and $\underline{x}_2$ that merge at bit k in the trellis diagram. The probability that a mistake is committed in choosing the survivor path using (22) is given as

$$\psi_{i \cap j,k} = \frac{1}{1 + e^{|M_{j,k} - M_{i,k}|}}, \tag{34}$$

where the notation $i \cap j$ is path index i or j whose metric is smaller than the other.

We again assign reliabilities $r_{i,n}$ to each decoded bit of each path $\underline{x}_i$ to indicate the probability that the bit is chosen incorrectly. We can again assign bits initial values of 0 (see Section 2.2.2). Again we have made an error with probability $\psi_{i \cap j,k}$ in each location where the estimated bit of the discarded path differs from the corresponding bit of the survivor path. We update the reliability values for these bits using (23):

$$r_{i,n} \longleftarrow r_{i,n}(1 - \psi_{i,k}) + (1 - r_{i,n})\psi_{i,k}.$$

2.2.3 Optimum use of the side information

The desired output of a convolutional decoder is almost always the decoded bits. However, in order for the inner decoder to make use of any information obtained from the outer decoder, the relevant information must be encoded with the GSM convolutional code.

If we were to feed hard decisions from the outer decoder back to the inner decoder, we could obtain encoded data by simply passing the decoded data through an encoder. Since we are interested in making full use of the reliability values however, we should not perform such a standard operation. An encoded bit may depend on not only the present bit but on past bits as well. Were we to pass the soft decision information through an encoder, the reliability of each encoded bit could be no higher than the least reliable of the bits on which it depends. Considerable information would be lost.

The preferred method is to output codewords directly from the outer decoder. A normal soft-output decoder keeps track of the decoded bits and reliabilities corresponding to each survivor path We modify the Viterbi decoding algorithm so that it retains not only these decoded values but also the corresponding values for *coded* bits. For each path at bit k, sequences corresponding to the bit value and reliability index are kept for the k data bits and the $2k$ encoded bits.

To determine the encoded bits is trivial; it simply requires that we use a lookup table to map the state transitions a path takes to encoded outputs. To generate reliabilities, we modify the method of Section 2.2.2. We initially assign each channel bit of a given encoded path a reliability value of 0 or ψ, depending on whether the survivor bit differs from that of the nonsurvivor. Each time two paths merge, we use (2.2.2) to update the reliabilities of each encoded bit whenever the encoded bits of the two merging paths differ. Soft output values which correspond accurately to the encoded bits are obtained. We may now pass this data back to the inner decoder.

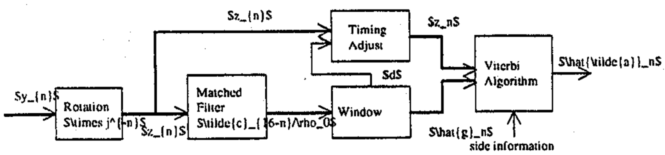

Figure 4: COSSAP implementation of the inner decoder

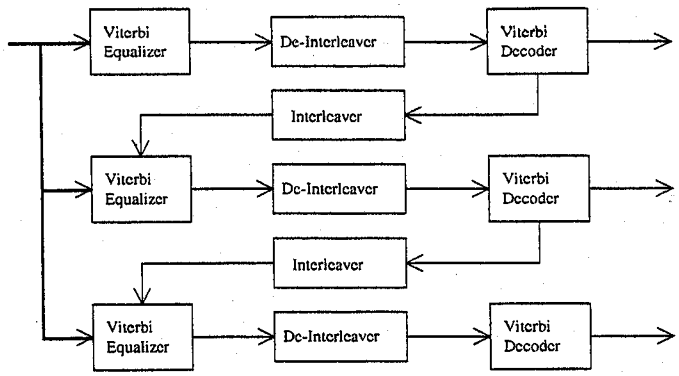

Figure 5: Implementation of the proposed iterative receiver

3 Simulation Experiments

3.1 A COSSAP simulator

We decided to adopt the COSSAP design environment [4], since it offers a rather comprehensive GSM library containing many GSM-specific signal processing blocks. We found its data stream-driven simulation approach easier to work with than a time-driven simulation approach adopted in other commercially available simulation packages. In addition, COSSAP provides tools which allow one to create new blocks in C simpler.

To implement our iterative decoding, we first created the inner decoder (the Viterbi equalizer) as a new hierarchical block in our simulator. Figure 3.1 shows the primitive blocks of which the hierarchical block consists. The block labeled "Rotation" forms $\{z_n\}$ by multiplying the incoming signal $\{y_n\}$ by j^{-n}. "Matched Filter" is a FIR (finite impulse response) filter which estimates the channel impulse response $\{g_n\}$ using the training sequence γ_{25-n}. "Window" is a windowing block which outputs a timing offset value d to help with synchronization, and "Timing Adjust" shifts the signal to account for this offset.

The block labeled "Viterbi Algorithm" then receives two sets of complex inputs, the synchronized $\{z_n\}$ and the windowed $\{g_n\}$. Making use of the channel estimate, it equalizes the received signal by using the algorithms described in the previous sections. It was created using COSSAP's "generic C" code [4], which makes generating code much simpler by abstracting the user from issues such as memory management and code optimization.

Path length metrics are assigned according to (12), using both the received signal $\{z_n\}$ and the side information values from the previous iteration. The reliabilities are assigned and updated according to (22) and (23). The estimated bits and corresponding reliability values are combined into the form of $P\{\text{bit} = 1\}$.

As the convolutional code is predefined by the GSM standard [7], a predefined lookup table for the codeword bits associated with all 32 possible state transitions can be included in the code. Path length metrics in its Viterbi decoder are assigned according to (31), and reliabilities assigned and updated according to (34) and (2.2.2). We keep track of not only decoded bits and their reliabilities but also of the channel bits and their reliabilities. The estimated bits and reliabilities (for both the decoded and coded data) are again combined in the form of $P\{\text{bit} = 1\}$.

Two implemented Viterbi blocks were connected into an iterative scheme. For simplicity, multiple pairs of Viterbi blocks were concatenated in a series configuration as shown in Figure 3.1 rather than piping the side information obtained from the decoder back to the original equalizer as shown in Figure 2.

For the first iteration, as we have no side information values, we give the Viterbi equalizer values of .5 for each bit (indicating a .5 probability of the bit being 1). In subsequent iterations as well, there are other bits for which we must use an input value of .5. The convolutional code does not protect all the data bits, but only the more important "Class I" bits [11, 7]. For the less important, unencoded "Class II" bits, we have no additional information from the Viterbi decoder and so we assign to them values of .5.

For the duration of the 26-bit training sequence, on the other hand, we know exactly what the transmitted bits are. We also know the values of the guard bits which are added to the beginning and end of the data packet, and the "Stealing bit," flags which are only set to indicate an emergency control signal [7]. Here, we input values of 0.0 or 1.0, depending on the data bit. When we take the logarithm of these values in the path length metric (12), their values outweigh any other contribution made to the cumulative metric by the received signal.

3.2 Preliminary results and efforts under way

The two Viterbi blocks have been completely implemented, and the full working iterative simulation described in Section 3.1 is under way. As the preliminary results show, there is incremental improvement from the iteration, but further fine-tuning of the receiver is needed.

References

[1] J. Bajcsy and H. Kobayashi, Invention Disclosures submitted to Princeton University, June 28 and August 22, 1996; filed to the US Patent Office, "Error Control of Generalized Concatenated Systems".

[2] J. Bajcsy and H. Kobayashi, "Error Control of Generalized Concatenated Systems", *Proc. of 1997 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'97)*, pp. 749-752, August 20–22, 1997, Victoria, B.C., Canada.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error Correcting Coding and Decoding: Turbo Codes (I)," *Proc. ICC'93*, pp. 1064-1070, Geneva, Switzerland, May 23-26, 1993.

[4] COSSAP documentation. From "Synopsys Online Documentation" CD-ROM.

[5] G. D. Forney, "The Viterbi Algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268-278, Mar. 1973.

[6] D. A. Garr, "Iterative Decoding for the GSM System," A Junior Independent Project Report, January, 1998.

[7] GSM Recommendation 05.03.

[8] J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications." *GLOBECOM 1989 Conference Record*, pp. 1680-1686, Dallas, TX.

[9] H. Kobayashi, "Correlative Level Coding and Maximum Likelihood Decoding," *IEEE Trans. Information Theory*, vol. IT-17, no. 5, pp. 586-594, September 1971.

[10] S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Comm. Mag.*, pp. 14-22, July, 1979.

[11] R. Steele (Ed.), *Mobile Radio Communications*, Pentech Press, London, and IEEE Press, New York 1992, 1994, 1995.

[12] M. Mouly and M. B. Pautet, *The GSM System for Mobile Communications*, Cell & Sys., France, 1992.

[13] A. Picart, P. Didier and A. Glavieux, "Turbo-detection: A New Approach to Combat Channel Frequency Selectivity", ICC 97, *Conference Record*, pp. 1498-1502, Montreal, Canada.

[14] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech. J., vol. 27, pp. 379-423, July 1948*, also in D. Slepian (ed.) *Key Papers in The Development of Information Theory*, pp. 5-28, IEEE Press, 1973.