MULTI-RATE DETECTION FOR THE IS-95A CDMA FORWARD TRAFFIC CHANNELS USING THE 13KBPS SPEECH CODER

Edith Cohen and Hui-Ling Lou

AT&T Laboratories
Lucent Technologies Bell Laboratories
600 Mountain Avenue, Murray Hill, NJ 07974.

ABSTRACT

The IS-95A Direct Sequence Code Division Multiple Access (DS-CDMA) system is a U.S. digital cellular standard. In the forward traffic channels (from the base station to the mobiles), the speech encoder rate can be reduced to four different sub-rates, depending on the speech activity in the transmitter. At the receiver, the decoder does not know which one of the four speech encoder sub-rates the transmitter used. This paper focuses on the IS-95A system that uses the 13kbps speech encoder and describes an algorithm to determine the encoder rate used for the transmitted sequence, with high confidence, so that the appropriate Viterbi decoder can be applied only once and not four times as is done by conventional decoders. This can reduce the overall decoder complexity by about a factor of two to fourteen, depending on the speech encoder sub-rate used. It can also reduce the power consumption in the decoder.

1. INTRODUCTION

The IS-95A Direct Sequence Code Division Multiple Access (DS-CDMS) system is a U.S. digital cellular standard [1]. The system uses an 8kbps speech encoder and allows the encoder rate to be reduced when the speech activity in the transmitter is low. To improve the voice quality of the system, a similar system that uses a 13kbps speech encoder is also included in the Standard (Telecommunication Service Bulletin 74). Similar to the forward traffic channels in the 8kbps speech encoder system, four different speech encoder rates can be used depending on the speech activity. The four rates are 13.35kbps, 6.25kbps, 2.75kbps and 1.05kbps as shown in Figure 1. The speech encoder rate can vary from frame to frame. After adding one reserve bit, the frame quality indicators and the encoder tail bits, each frame can have an information bit rate of 14.4kbps (defined as **sub-rate** R_1), 7.2kbps (defined as **sub-rate** R_2), 3.6kbps (defined as **sub-rate** R_3) or 1.8kbps (defined as **sub-rate** R_4) as shown in Figure 2. These frames of information bits are encoded by a powerful rate 1/2, 256-state convolutional code [3]-[7] for error control in the forward traffic channels. After the channel encoder, each frame of data can be 28.8kbps, 14.4kpbs, 7.2kbps or 3.6kbps, depending on the the speech encoder used. Equivalently, for each frame that has a 20 msec duration, there are 576 symbols per frame for R_1 frames, 288 symbols per frame for R_2 frames, 144 symbols per frame for R_3 frames and 72 symbols per frame for R_4 frames. To ensure that all frames have 576 symbols, each coded symbols in each frame are repeated twice, four times and eight times for the R_2, R_3 and R_4 frames respectively (Figure 1). In order to ensure a constant transmission symbol rate of 19.2 ksps with 384 symbols per frame, two out of every six symbols in each frame are punctured (i.e. not transmitted) as shown in Figure 1. An interleaver is applied to the coded, repeated and punctured

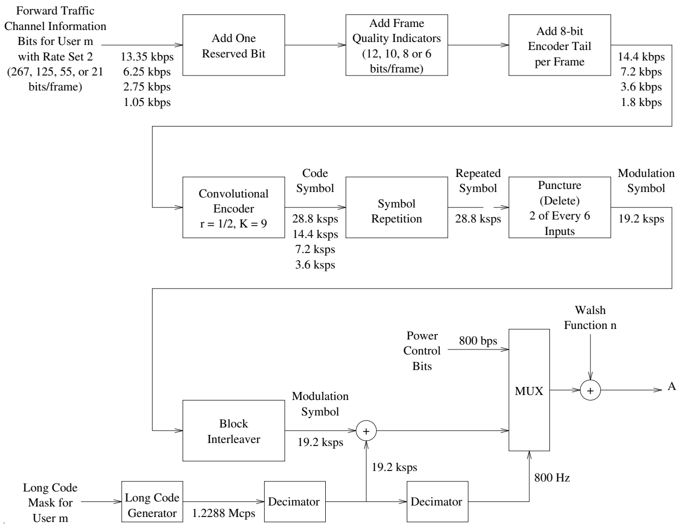

Figure 1. Forward Traffic Channel Structure

frame of symbols. After that, 16 symbols in each frame are erased to transmit power control information.

At the receiver, the decoder does not know which one of the four sub-rates was used by the transmitter for the current received frame of data. One way to determine the correct data rate at the receiver is to use the Viterbi channel decoder to decode the packet according to each of the four possible encoding methods, and then to select the decoded data packet that has the least number of errors as shown in Figure 3.

This paper proposes a multi-rate detection algorithm to predetermine, with high confidence, the sub-rate used for the current received frame of data before the Viterbi decoder is applied. Hence, an appropriate Viterbi decoder is applied only once (Figure 3). The multi-rate detection algorithm uses symbol repetition properties in the data frames to determine the transmitted symbol pattern in Gaussian and Rayleigh Fading channels using statistical methods. The algorithm introduces **Frame Error Rates (FER)** that are well below the maximum FER recommended in [9]. (The minimum standard requirement for the IS-95A system using the 13kbps speech coder is currently in the industry ballot process). The algorithm can reduce the complexity in the decoder and result in a speed up of about a factor of 2 to 14 in the decoder, depending on the speech encoder sub-rate used. It can also reduce the amount of power consumption in the decoder since the decoder can be turned off ones the decoding of the R_2, R_3 or R_4 frames are done. A soft-decision multi-rate detection algorithm for the IS-95A system using the 8kbps speech coder is given in [8]. This paper presents a soft-decision algorithm for the 13kbps speech

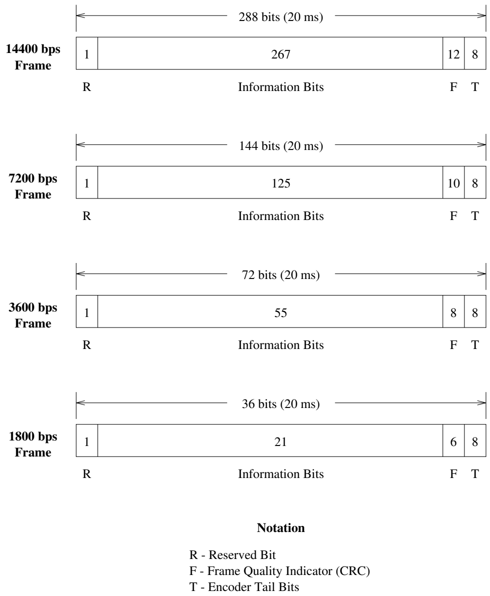

Figure 2. Forward Traffic Channel Frame Structure

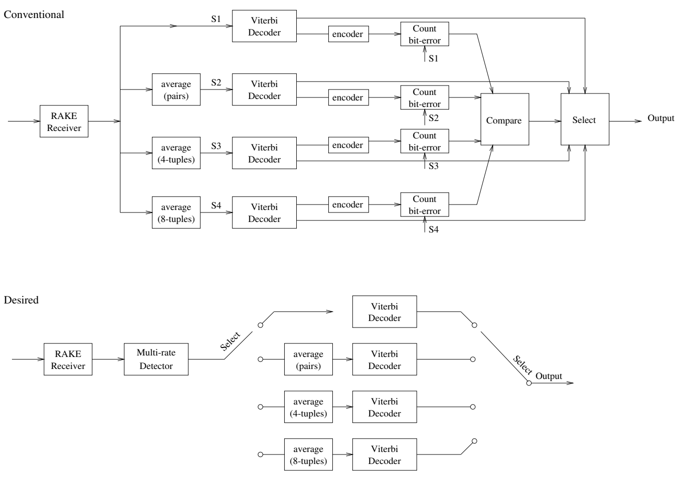

Figure 3. Objectives for the Multi-rate detector design

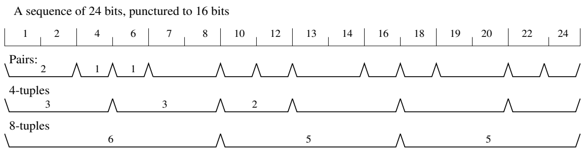

Figure 4. Punctured sequence of 24 bits

coder system. Multi-rate detection for the 13kbps speech coder system is more challenging than that of the 8kbps system since the coded symbols are punctured before transmission.

The following section describes the multi-rate detection algorithm. The performance of the algorithm in **Additive White Gaussian Noise (AWGN)** and Rayleigh Fading channels is presented in Section 3.. Finally, we compare the implementational complexity of the multi-rate detection algorithm to that of the conventional method in Section 4..

2. MULTI-RATE DETECTION ALGORITHMS

The transmitter system with the 13kbps speech encoder is given in Figure 1. Prior to transmission, the frames are interleaved using the IS-95 standard interleaver [1]. After interleaving, 16 symbols in each frame are replaced by power control bits in the following manner: In each of the 16 sets of 24 consecutive bits, one of the first 16 bit positions is selected according to the content of four bit positions in the Long code. In effect, the position is selected randomly. After that, one power control bit is inserted in the selected position, replacing the original content. The power control bit positions can be reconstructed from the Long code at the receiver. At the receiver, data symbols in each frame are also de-interleaved.

In each frame of data, 384 symbols are transmitted (Figure 1). Our multi-rate detection algorithms utilize symbol repetition properties to identify the transmitted symbol pattern. Prior to the power control bit insertion, each frame of data are subject to puncturing of the 3rd and 5th bits in every 6-tuple. The tuple structure is depicted in Figure 4. Our multi-rate detection algorithm evaluates tuples of symbols according to the tuple structure. The encoded frames are "random looking" in the following sense. The probability that the encoder outputs each pair among 00,01,10,11 is about the same. In particular, the pairs $\{01, 10\}$ are about as likely as the pairs $\{00, 11\}$. Hence, it is very unlikely that a packet of sub-rate R_1 (respectively, R_2, R_3) will consist of a sequence of pairs (respectively, 4-tuples, 8-tuples) of identical symbols and be mistaken for a lower-rate packet. Puncturing and power control bit insertions destroy some of the 2-tuple, 4-tuple and 8-tuple structures. This coupled with the presence of noise in the communication channel make the rate detection a much more challenging problem. If the symbols in the tuple are not identical, it is either a higher rate packet or a low rate packet that was subject to noise. We use statistical methods to determine which is the case.

We considered both hard-decision detection and soft-decision detection methods. However, we found that the hard-decision multi-rate detector will not be able to match the performance of the conventional method since the Viterbi decoder, used in the conventional method, takes full advantage of the soft-decision of the received symbols to detect the sub-rates. Thus, we present a multi-rate detection algorithm based on soft-decision of the received symbols. This soft-decision detection algorithm can handle a much wider range of **Signal-to-Noise Ratio (SNR)**. The following section describes the soft-decision multi-rate detection algorithm.

2.1. Soft-decision multi-rate detection algorithm

The input to the soft-decision detector is a sequence of 384 symbols. The detector outputs a rate in $\{R_1, R_2, R_3, R_4\}$, where R_1 (corresponds to 14.4kbps of information bits) is the highest rate and R_4 (corresponds to 1.8kbps of information bits) is the lowest.

The algorithm inputs frames as arrays of size 384

$$(b_0, b_1, \ldots, b_{383}) \; ,$$

where each symbol, b_0 is quantized to an n-bit number since soft-decision is used. In our simulations, we used 4-bit quantization for each symbol although the algorithm is not limited to using only 4-bit symbols. To discard the power control bits, locations that are used to transmit the power control bits are also set to the value 0 before the multi-rate algorithm is applied. We note that a more complex algorithm that takes into account the power control bit locations can be devised. However, the success of our current approach indicates that it is not necessary.

The soft-decision multi-rate detection algorithm is as follows:

MULTI-RATE DETECTION ALGORITHM:
The algorithm first computes the quantities LE_i, E_i, UE_i, and r_i (for $i = 1, 2, 3$). These values are used later in the classification rule.

1. **Computing LE_1, UE_1, E_1:**
   $E_1 \leftarrow 0$, $LE_1 \leftarrow 0$, $UE_1 \leftarrow 0$.
   For $i = 0, \ldots, 95$:
   (a) $E_1 \leftarrow E_1 + |b_{4i} + b_{4i+1}|$
   (b) $LE_1 \leftarrow LE_1 + ||b_{4i}| - |b_{4i+1}||$
   (c) $UE_1 \leftarrow UE_1 + |b_{4i}| + |b_{4i+1}|$
2. **Computing LE_2, UE_2, E_2:**
   $E_2 \leftarrow 0$, $LE_2 \leftarrow 0$, $UE_2 \leftarrow 0$.
   For $i = 0, \ldots, 47$:
   (a) $E_2 \leftarrow E_2 + |b_{8i} + b_{8i+1} + b_{8i+2}| + |b_{8i+3} + b_{8i+4} + b_{8i+5}| + |b_{8i+6} + b_{8i+7}|$
   (b) $LE_2 \leftarrow LE_2 + ||b_{8i} + b_{8i+1}| - |b_{8i+2}|| + ||b_{8i+3}| - |b_{8i+4} + b_{8i+5}|| + ||b_{8i+6}| - |b_{8i+7}||$
   (c) $UE_2 \leftarrow UE_2 + |b_{8i} + b_{8i+1}| + |b_{8i+2}| + |b_{8i+3}| + |b_{8i+4} + b_{8i+5}| + |b_{8i+6}| + |b_{8i+7}|$
3. **Computing LE_3, UE_3, E_3:**
   $E_3 \leftarrow 0$, $LE_3 \leftarrow 0$, $UE_3 \leftarrow 0$.
   For $i = 0, \ldots, 23$:
   (a) $E_3 \leftarrow E_3 + \left|\sum_{j=16i}^{16i+5} b_j\right| + \left|\sum_{j=16i+6}^{16i+10} b_j\right| + \left|\sum_{j=16i+11}^{16i+15} b_j\right|$
   (b) $LE_3 \leftarrow LE_3 + \left||\sum_{j=16i}^{j=16i+2} b_j| - |\sum_{j=16i+3}^{j=16i+5} b_j|\right| + \left||\sum_{j=16i+6}^{j=16i+7} b_j| - |\sum_{j=16i+8}^{j=16i+10} b_j|\right| + \left||\sum_{j=16i+11}^{j=16i+13} b_j| - |\sum_{j=16i+14}^{j=16i+15} b_j|\right|$
   (c) $UE_3 \leftarrow UE_3 + |\sum_{j=16i}^{j=16i+2} b_j| + |\sum_{j=16i+3}^{j=16i+5} b_j| + |\sum_{j=16i+6}^{j=16i+7} b_j| + |\sum_{j=16i+8}^{j=16i+10} b_j| + |\sum_{j=16i+11}^{j=16i+13} b_j| + |\sum_{j=16i+14}^{j=16i+15} b_j|$
4. **Computing the ratios r_1, r_2, r_3 :**

$$r_1 \leftarrow \frac{E_1 - LE_1}{UE_1 - LE_1},\ r_2 \leftarrow \frac{E_2 - LE_2}{UE_2 - LE_2},\ r_3 \leftarrow \frac{E_3 - LE_3}{UE_3 - LE_3}$$

The following claims are easy to establish. They help motivate our classification algorithm.

1. For $i = 1, 2, 3$, we have $LE_i \leq E_i \leq UE_1$, that is, LE_i is a lower bound and UE_i an upper bound on the quantity E_i. Hence, $r_i \in [0, 1]$.
2. If the signs of the received bits $(b_0, \ldots, b_{383})$ are selected randomly (as is approximately the case for sub-rate R_1), the expected values of E_i and r_i (for $1 \leq i \leq 3$) are
$$< E_i >= \frac{UE_i + LE_i}{2} \text{ and } < r_i >= 0.5\,.$$
3. For R_2 frames, the signs of every *pair* of bits (prior to puncturing) are independent and random, but the bits in each pair had the same sign when transmitted, and hence, are much more likely to have the same sign. If the channel is noiseless (that is, the signs of the received bits is the same as the transmitted bits) then $E_1 = UE_1$ and $r_1 = 1$. The value of r_1 decreases with noise but we still expect it to be high. The expected values of r_2 and r_3 are 0.5.
4. For R_3 frames, we have $< r_3 >= 0.5$ but r_1 and r_2 are high ($r_1 = r_2 = 1$ when the channel is noiseless.)
5. For R_4 frames, we expect r_1, r_2 and r_3 to be high.

Based on that we propose a simple classification algorithm that uses 3 preset threshold values $\{T_1, T_2, T_3\} \in [0.5, 1]$.

Simple Classification Rule:

1. **if** $r_3 > T_3$, **return** R_4. Otherwise,
2. **if** $r_2 > T_2$, **return** R_3. Otherwise,
3. **if** $r_1 > T_1$, **return** R_2. Otherwise,
4. **return** R_1.

Values for the thresholds T_1, T_2, T_3 can be determined using the histograms of values of r_1, r_2 and r_3 for each sub-rate. The thresholds can be set according to tradeoffs between the noise level we wish to handle and the misclassification rate we can tolerate. Optimally, the rate detector would track the performance of the Viterbi decoder and introduce a much lower frame error rate than the Viterbi decoder. That is, the rate detector would suffer small error rate on the range of noise levels and sub-rates where the Viterbi decoder performs well. The rate detector can afford to misclassify frames that are too noisy for the Viterbi decoder to decode, since these can not be decoded correctly regardless. Note that the noise level that the Viterbi decoder can handle increases for lower sub-rates since the R_2, R_3 and R_4 frames use a repetition, 4-times repetition and 8-times repetition code on top of the convolutional code (Figure 1). One of the difficulties we confronted was that for R_4 frames, with the 8-times repetition code, the Viterbi decoder tolerates a very noisy channel where the symbols in the frame look almost random to the rate detector. Hence, the frame can easily be confused with the symbols in the R_1 frames.

We tested the performance of the simple classification algorithm with careful setting of the thresholds. The misclassification rate can be significantly reduced if we use more complex conditions, involving more than one of the r_i's at a time. To illustrate the weakness of the simple rule, consider the case where a high value of r_3 is used to differentiate R_4 from $\{R_1, R_2, R_3\}$. When the sub-rate is R_1, the quantities LE_3, E_3, and UE_3 have relatively small values as a result of the independence of the signs of consecutive symbols. Hence, r_3 has a large standard deviation and high values are more likely (the expected value is still 0.5). Therefore, by considering r_3 alone, there is a wider range where R_1 frames can be confused with noisy R_4 frames. To compensate for this situation, when the value of r_3 is in the ambiguous range, the enhanced method classifies the transmission rate based on the values of r_1 and r_2 as well. Similar behavior and correction occur for the parameter r_2 and sub-rates $\{R_1, R_2, R_3\}$. Below we present one variant of a finer-tuned classification algorithm. This is the one we used to obtain the results given in Section 3.. Obviously, many variants are possible yielding different tradeoffs and results. The finer-tuned algorithm also considers the parameter UE_1. The magnitude of UE_1 is correlated with the noise level and is smaller for low noise levels.

Example of finer-tuned Classification Rule:

- **if** $(r_3 \geq .78)$ **return** R_4
- **if** $(r_3 \geq .73)$
  - **if** $(r_2 - r_3 \geq .1)$ **return** R_3;
  - **if** $(r_1 + r_2 + r_3 < 1.88)$ **return** R_1;
  - **if** $(r_3 + r_2 - r_1 \leq .5)$ **return** R_2;
  - **return** R_4;
- **if** $(r_3 \geq .68)$

- if $(r_2 - r_3 > .1)$ **return** R_3;
- if $(r_1 + r_2 + r_3 \leq 1.88)$ **return** R_1;
- if $(r_3 + r_2 - r_1 \leq .46)$ **return** R_2;
- **return** R_4;

- **if** $(r_3 \geq .63)$
  - **if** $(r_1 + r_2 + r_3 \leq 1.86)$ **return** R_1;
  - **if** $(r_2 - r_3 \geq .08)$ **return** R_3;
  - **if** $(r_2 + r_3 - r_1 \leq .44)$ **return** R_2;
  - **return** R_4;
- **if** $(r_2 \geq .71)$ **return** R_3;
- **if** $(r_2 \geq .64)$
  - **if** $(r_1 \geq r_2 + .15)$ **return** R_2;
  - **if** $(r_1 + r_2 \leq 1.28)$ **return** R_1;
  - **return** R_3;
- **if** $(r_1 \geq .74)$ **return** R_2;
- **if** $((r_1 \geq .7)\&\&(UE_1 < 410))$ **return** R_2;
- **return** R_1;

An alternate way of improving the classification algorithm is using the fact that if sub-rate R_4 is transmitted, only $\frac{1}{8}$ of the power is transmitted, compared to the power transmitted for sub-rate R_1 [1]. Thus, if the received power is very low, it is either because the signal has gone through a deep fade or sub-rate R_4 is transmitted. In this case, we can bias the detector to classify as R_4 since even if another sub-rate was transmitted, the signal that has gone through a deep fade will be so noisy that the Viterbi decoder will not be able to decode the sequence correctly anyway for the higher sub-rate.

3. PERFORMANCE EVALUATION

We simulated the performance of our algorithm in Gaussian and Rayleigh fading environments and compared this performance to that of the recommended minimum performance given in [9]. (The minimum standard requirement for the IS-95A system using the 13kbps speech coder is currently in the industry ballot process). The threshold settings and the classification rules are heavily dependent on the range of the SNR ratios used in the channel. In our simulations, we assumed that the information bits are random and we appended eight zero tail bits before sending the information packet into the convolutional encoder (Figure 1). The interleaver and de-interleaver are used according to the IS-95 Standard. We also erased 16 symbols per frame for power control bits before sending the frames into the channel (Section 2.). We used the Rayleigh fading model given in [11] for the Rayleigh fading channels. We also used multi-path fading models according to [9]. At the receiver, we uniformly quantized each received symbol to 4 bits before sending the symbols into the multi-rate detector.

We generated 10^4 to 10^6 frames of random information bits for our simulations. We studied the misclassification rates with different SNR using our algorithm for each sub-rate. We assumed that if the multi-rate detector misclassifies the frame to a wrong sub-rate, we lose the frame and thus increase the FER. After rate detection, the received frame of data is sent into the Viterbi decoder. If one single bit is in error at the output of the Viterbi decoder, the frame is considered to be in error and thus also constitutes to the FER. This FER using the multi-rate detector algorithm is then compared to that of the ideal FER at the Viterbi decoder output, where the Viterbi decoder knows exactly which sub-rate is used in the transmitter. Notice that this is the best possible Viterbi decoder FER performance. This performance will always be better or the same as that of the conventional method where the Viterbi decoder is used four times to detect the sub-rate. This is because there are sequences that can be a valid coded sequence of more than one sub-rate. For example, the all zero coded bit sequence can be a valid coded sequence for all four sub-rates. Thus, unless the transmitted sub-rate is known at the the decoder, the decoder has no way of differentiating the sub-rates.

Rate (bps)	$E_b/N_o(dB)$	Max FER allowed	Detector FER
14400	5.2	0.05	0.0004
	5.8	0.01	0.0002
	6.2	0.003	0.0002
7200	3.7	0.03	0.0004
	4.1	0.01	0.0002
	4.4	0.005	0.0002
3600	3.1	0.03	0.0004
	3.6	0.01	0.0004
	3.9	0.005	< 0.0001
1800	2.5	0.03	0.0016
	3.0	0.01	0.0002
	3.4	0.005	0.0006

Table 1. Performance of the multi-rate detection algorithm in AWGN

Rate (bps)	$E_b/N_o(dB)$	Max FER allowed	Detector FER
14400	7.3	0.04	0.0003
	9.4	0.01	0.0002
	10.8	0.004	0.0001

Table 2. Performance of the multi-rate detection algorithm in Rayleigh Fading Channels: Case 4

The simulation performance results using the multi-rate detection algorithm for each case according to [9] are shown in Tables 1- 3. In each Table, the information bit rate, the SNR (E_b/N_o), the maximum FER recommended and the FER due to the multi-rate detector are given. We used the finer-tuned classification rule described in Section 2.1.. The minimum performance requirement standard contains two parts, with part one dealing with AWGN channels and part two dealing with Rayleigh fading channels. For Rayleigh fading channels, there are two different cases according to the two different channel configurations relevant to the 13kbps speech coder system. Case 4 has two multi-paths of equal strength and the vehicular velocity is 8km/hr. Case 5 has 3 multi-paths with the third path having half the power compared to the first two paths. The vehicle speed for Case 5 is 100 km/hr. Detailed configuration for each case is given in [9].

For sub-rate R_4, the Viterbi decoder can decode sequences even at very low SNR because an average value of the 8-tuples is computed and sent into the Viterbi decoder as one input value. Thus, the effective SNR of the averaged values at the Viterbi decoder input is much higher. To make sure that the multi-rate detector will have similar performance to that of the Viterbi decoder for sub-rate R_4, we have to set the thresholds and classification rules in a way that even if the sequence is quite random looking, we will still consider that as an 8-tuples sequence. That is, we will detect R_4 even if the ratio r_3 is relatively low. This increases the chance of confusing R_4 with R_3, R_2 and R_1. Thus, trading off the performance of detecting sub-rates R_3, R_2 and R_1. Nevertheless, we can see from the tables that the FER introduced by the multi-rate detection algorithm is well below the maximum FER recommended in [9].

4. IMPLEMENTATIONAL COMPLEXITY

This section considers the complexity of implementing the soft-decision multi-rate detector and compared it to that of the traditional method where the Viterbi decoder is used to decode the sequence four times to determine the transmitted sub-rate.

Rate (bps)	$E_b/N_o(dB)$	Max FER allowed	Detector FER
14400	7.2	0.03	0.0004
14400	8.1	0.005	0.0004
7200	4.8	0.03	0.0001
7200	5.6	0.005	< 0.0001
3600	3.9	0.03	0.0007
3600	4.8	0.005	0.0001
1800	3.1	0.03	0.004
1800	4.1	0.005	0.001

Table 3. Performance of the multi-rate detection algorithm in Rayleigh Fading Channels: Case 5

Referring to the soft-decision multi-rate detection algorithm described in Section 2.1.: 573 addition and 384 absolute value computations are required to compute E_1, LE_1 and UE_1. 909 addition and 576 absolute value operations are required to compute E_2, LE_2 and UE_2, considering that some of the computations, such as $|b_{8i} + b_{8i+1}|$, in E_2, LE_2 and UE_2 computations are common. Similarly, 669 addition and 288 absolute value operations are required to compute E_3, LE_3 and UE_3. To compute r_1, r_2 and r_3, 6 addition and 3 division operations are required. Thus, a total of 2154 addition, 1248 absolute value and 3 division operations are required for the most straightforward implementation of the algorithm. This count includes adding zeros (the power control bits) that may be avoided depending on the hardware implementation. Depending on the classification rules (Section 2.1.), a few comparison operations are also required.

The conventional method decodes the received sequence four times, using the Viterbi decoder, and then compares each bit in the decoded sequence for each sub-rate to the sign of the corresponding symbol in the received sequence. The sub-rate corresponds to the decoded sequence that has the least number of bit errors compared to the received sequence is selected. To decode one frame of R_1, the Viterbi decoder has to go through 288 iterations since there are 288 information bits in the frame. Similarly, the Viterbi decoder has to go through 144, 72 and 36 iterations to decode sequences of sub-rates R_2, R_3 and R_4 respectively. For each iteration, just to update the survivor path alone, the Viterbi decoder requires 512 addition and 256 comparison operations [12, 13]. This does not include branch metric computations, updating the optimum paths, path trace back and the comparison between the decoded bit sequence to the received sequence of symbols [12]. Thus, we can see that the number of computations required by the soft-decision multi-rate detector is very small compared to that of decoding one frame of data using the Viterbi algorithm. If sub-rate R_1 is selected by the multi-rate detector, the Viterbi algorithm is used once to decode the R_1 sequence. Thus, 288 iterations are required to decode the frame. If sub-rate R_4 is selected, only 36 iterations are required to decode the frame. Thus, we can see that the multi-rate detector can decrease the complexity by about a factor of two to fourteen depending on the speech encoder sub-rate used. Furthermore, for decoding the frames with fewer information bits (R_2, R_3 and R_4), the decoder can be turned off once the decoding is done. This reduces the power consumed by the decoder.

5. CONCLUSION

The IS-95 DS-CDMA system uses four different sub-rates for the speech encoder. If the speech activity is low, the speech encoder rate can be reduced to $\frac{1}{2}$, $\frac{1}{4}$ or $\frac{1}{8}$ of the full-rate (13kbps). The frame of information bits at the output of the speech coder will go through a rate 1/2 256 state convolutional channel encoder. To ensure a constant transmission rate of 19.2 ksps, this coded bit sequence at the output of the convolutional encoder will be repeated twice, four times and eight times to form a 2-tuples sequence, 4-tuples sequence and 8-tuples sequence for the $\frac{1}{2}$ rate, $\frac{1}{4}$ rate and $\frac{1}{8}$ rate speech coders respectively. The sequence will also be subject to puncturing and power control bits erasure.

At the receiver, the Viterbi decoder that decodes the sequences coded by the convolutional coder does not know which speech encoder sub-rate is used. The conventional method uses the Viterbi decoder to decode the received sequence four times, based on the four possible coding schemes, in order to determine the most likely sub-rate used in the transmitter.

This paper takes advantage of the pairs and the tuple symbol patterns to determine the most likely speech encoder sub-rate used before the Viterbi decoder for the appropriate sub-rate is applied at the receiver. That is, the Viterbi decoder is used once and not four times. Thus, the algorithm can reduce the overall decoder complexity and increase the throughput by about a factor of two to fourteen, depending on the speech encoder sub-rate used. It can also reduce the power consumption in the decoder.

REFERENCES

[1] TIA/EIA Interim Standard 95: *Mobile Station-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular Standard*, July 1993.

[2] TIA/EIA Interim Standard 98: *Recommended Minimum Performance Standards for Dual-Mode Wideband Spread Spectrum Cellular Mobile Stations*, December 1994

[3] K. S. Gilhousen, I. M. Jacobs, R. Padovani, A. J. Viterbi, L. A. Weaver and C. E. Wheatley, "On the Capacity of a Cellular CDMA System," *IEEE Trans. on Vehicular Technology,* vol. 40, No. 2, May 1991, pp 303-312.

[4] G. C. Clark Jr. and J. B. Cain, *Error Control Coding for Digital Communications*, Plenum, New York, 1981.

[5] E. A. Lee and D. G. Messerschmitt, *Digital Communication*, Kluwer Academic Publishers, Boston, MA, 1988.

[6] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill Book Company, New York, NY, 1979.

[7] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, John Wiley and Sons, New York, NY, 1965.

[8] E. Cohen and H. Lou, "Multi-rate Detection for the IS-95 CDMA forward traffic channels", *IEEE GLOBECOM'95*, November 1995.

[9] Draft American National Standard for Telecommunications: *Recommended Minimum Performance Standards for 1.8 to 2.0 GHz Code Division Multiple Access (CDMA) Personal Stations*, August 95.

[10] ANSI J-STD-008: *Personal Station/Base Station Compatibility Requirements for 1.8 - 2.0 GHz CDMA personal communication system*, March 1995.

[11] W. C. Jakes, *Microwave Mobile Communications*, John Wiley & Sons, New York, 1974.

[12] H. Lou, "Implementing the Viterbi Algorithm", *IEEE Signal Processing Magazine*, Vol. 12, No. 5, September 1995, pp 42-52.

[13] G. D. Forney Jr., "The Viterbi Algorithm", *IEEE Proceedings*, vol. IT-61, No. 3, March 1973, pp 268–278.