FUZZY LOGIC BASED CONVOLUTIONAL DECODER FOR USE IN MOBILE TELEPHONE SYSTEMS

A. Falkenberg
A. Kunz
Siemens AG
Frankenstrasse 2, D-46393 Bocholt, Germany
Tel.: +49 2871 91 3697, Fax: +49 2871 91 3387
email: Andreas.Falkenberg@bch.siemens.de

ABSTRACT

Efficient convolutional coding and decoding algorithms are most crucial to successful operation of wireless communication systems in order to achieve high quality of service by reducing the overall bit error rate performance. A widely applied and well evaluated scheme for error correction purposes is well known as Viterbi algorithm [7]. Although the Viterbi algorithm has very good error correcting characteristics, computational effort required remains high. In this paper a novel approach is discussed introducing a convolutional decoder design based on fuzzy logic [8]. A simplified version of this fuzzy based decoder is examined with respect to bit error rate (BER) performance. It can be shown that the fuzzy based convolutional decoder here proposed considerably reduces computational effort with only minor BER performance degradation when compared to the classical Viterbi approach.

1. INTRODUCTION

Error correcting codes like convolutional codes are commonly used to reduce bit errors in CDMA Terminals. A widely applied decoding algorithm is the trellis based decoder introduced first by Viterby [7]. Further innovations of this decoding concept have been reported using a cascaded structure of Viterby decoders like turbo codes [4] including interleaving as well. The decoder design here proposed has reduced complexity requiring less computational ressources compared to the Viterby concept. The slightly lower performance in terms of BER may be neglected as computational effort is reduced in order to obtain current economisation in the microporcessor design yielding to a longer stand by time of a hand-held set. So, the scheme to be discussed in the following might be interesting for implementation in both CDMA terminals and reference systems.

2. SIMULATION ENVIRONMENT

Prior to the description of the convolutional coder in the next section, an overview over the simulated system will be given. Fig. 1 show the main parts of the simulated system.

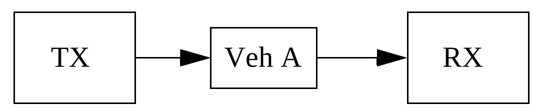

Fig. 1: System Overview

The Vehicular A channel model [1][6] was used for the simulation results given for example in this paper. The receiver part shown in Fig. 2 consists of a RAKE receiver, deinterleaving and decoding units. Thus the model is only used as testbench for the decoder, it does not perfectly reflect the requirements given by the current 3GPP standart.

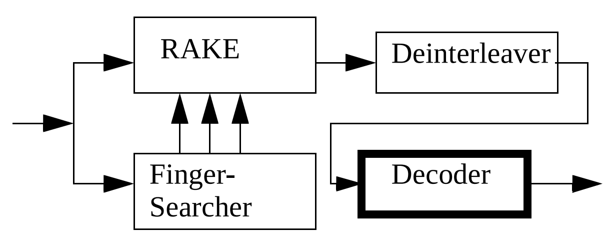

Fig. 2: The RX part

The RAKE receiver detects the CDMA signal [2], which is spread with a factor of 128. After deinterleaving the proposed decoder or in comparision the Viterby decoder are used to decode the sent data. A comparision between the sent data and the decoded data leads to the bit error rate (BER) measurement. Simulations are made with COSSAP, which is the system simulator sold by SYNOPSYS.

3. CONVOLUTIONAL CODES

A simple example of a convolutional coder is shown in Fig.1. More complex versions of convolutional encoders are used for example in the upcoming 3GPP standard [9], which will play a major role in 3rd generation mobile communication systems.

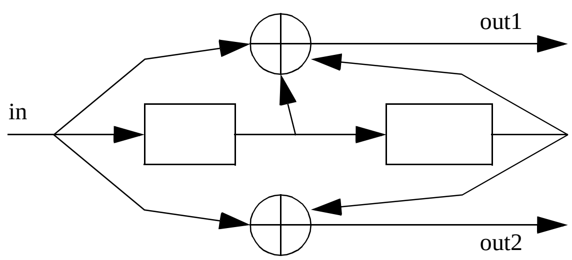

Fig. 3: Convolutional Coder

As can be seen from this Fig. 1, the input data ***in*** is shifted into the register consisting of bit0 and bit1. Two generator polynomials g1 and g2 are used, which each define one output depending on the actual input and the content of the shift register. These generator polynomials produce two output bits ***out1*** and ***out2*** for each input bit by performing modulo 2 additions (see X). After converting the parallel 2 bit outputs ***out1*** and ***out2*** into a serial bitstream the coded data of the encoder of rate r=1/2 is ready. To decode the coded sequence after reception, a Viterby decoder, also named trellis decoder, can be used. The functionality of this decoder is not further investigated in this paper. For a detailed description, please refer to [5][7].

4. FUZZY LOGIC

In the following section the main ideas of the fuzzy logic are described as first introduced by Lukasiewicz [3] in 1915. Starting from these ideas a convolutional decoder will be build up in the following chapter and a result for a decoder of rate r= 1/3 with constraint length of 9 will be shown as it is used in the 3GPP standard. The main idea of fuzzy logic is to use not only 1 and 0 as inputs and output of the logic functions but the whole range of real numbers between 0 and 1. The logic functions AND and OR can be defined as $[0,1] \times [0,1] \rightarrow [0,1]$ functions as shown here:

$$x \wedge y = min(x, y) \tag{1}$$

$$x \vee y = max(x, y) \tag{2}$$

If only the values 0 and 1 are used for the variables x and y, the logic functions are given by this definition. A second possibility of defining the logic functions AND and OR is given by the following definition:

$$x \wedge y = xy \tag{3}$$

$$x \vee y = x + y - xy \tag{4}$$

For both possibilities, the logical NOT can be defined by the following $[0,1] \rightarrow [0,1]$ function:

$$\neg x = 1 - x \tag{5}$$

Further definitions are as well possible to define fuzzy-logic functions. Here, always the binary logical functions are given if only 0 and 1 is used. The definitions can be proven according to the laws which hold for binary logic [8].

An important function used in the decoding algorithm is the average operator, which is not existant in the classical binary logic.

$$av(x, y) = \frac{x + y}{2} \tag{6}$$

In this paper no theoretical results about fuzzy logic are shown. For a more detailed description, please refer to [8]. The most usual fuzzy operators are the so called Lukasiewicz operators [3] which are the first defined minimum and maximum operators.

5. FUZZY CONVOLUTIONAL DECODER

The first step to build up a fuzzy convolutional decoder is to look at the truth table of the logical function given by the encoder. The following Table 1 displays the output according to the input and the actual state for the example given in Fig. 3. A new state is given by a right shift of the actual state with the input shifted into the register. The table shown represents an uncomplete decoder table. The undefined values are names „don’t care values“ in the following. These „don’t care values“ can be different dependend on the design of the decoder. Using a fixed 0,1 logic the don’t cares have to be choosen by design constraints, e.g. they have to be optimized and finally internally defined by the designer of the decoder circuit. Using fuzzy logic the don’t cares might be represented by a value of 0.5 and have to be included in the design of the decoder. Table 2 shows the truth table of the decoder using the value 0.5 for the don’t care values.

Table 1: Encoder Table

INPUT	STATE s	OUTPUT	STATE’
0	00	00	00
1	00	11	10
0	01	11	00
1	01	00	10
0	10	10	01
1	10	01	11
0	11	01	01
1	11	10	11

The following Table 2 represents the decoder. The output defined in Table 1 represents the input signal. As mentioned above the table is not complete and the missing values, the so-called dont‘cares are filled with a value of 0.5.

Table 2: Decoder Table

Table 2: Decoder Table

D_STATE $s_0 s_1$	Dec_IN $i_0 i_1$	D_STATE’ $o\ s_0$	Dec_OUT o
00	00	00	0
00	01	0.5 0	0.5
00	10	0.5 0	0.5
00	11	10	1
01	00	10	1
01	01	0.5 0	0.5
01	10	0.5 0	0.5
01	11	00	0
10	00	0.5	0.5
10	01	11	1
10	10	01	0
10	11	0.5 1	0.5
11	00	0.5 1	0.5
11	01	01	0
11	10	11	1
11	11	0.5 1	0.5

The only function which has to be calculated is the generated one bit decoder output signal, which is the one bit input of the coder. The calculation of the new states is done by a shift of the old state with the calculated output shifted into the register. With this additional assumptions only one output bit has to be calculated using a four bit input. All of the inputs are representations of fuzzy logic, e.g. the soft values are used. According to Table 2 the following fuzzy logic function can be defined which is a modified fuzzy normal form including the don’t care elements of the table. As an example here a modified disjunctive normal form is shown.

The following normal form is the disjunctive normal form, given by the table:

$$o = (\bar{s}_0\bar{s}_1 i_0 i_1 \vee \bar{s}_0 s_1 \bar{i}_0\bar{i}_1 \vee s_0\bar{s}_1\bar{i}_0 i_1 \vee s_0 s_1 i_0 \bar{i}_1) \quad (7)$$

The implicit assumption while using the disjunctive normal form is that all don’t care values are set to a zero value. The konjunctive normal form would calculate the result as if the don’t care elements are set to one. These forms do not depend on wheather fuzzy or boolean logic is used. The element which makes the difference to a classical decoder is to include the don’t cares into the calculation and set them to a fuzzy value of 0.5. With the average operator defined earlier in this paper the following terms can be defined in addition to the above mentioned ones. For each state there are two input values which lead to a 0.5 output in the table. As an example the second and the third row of the Table 2 is considered. Here, the state is 00, where for the inputs 01 and 10 the result of the decoding unit is 0.5. This can be exactly calculated using the following 4 terms with the average operator av is defined by Eq. 6. The terms represent row 5 to 16 of Table 2:

$$t_1 = \bar{s}_0 \wedge \bar{s}_1 \wedge av(i_0, i_1) \quad (8)$$

But additional to the second and the third row of the table with this single term the first four rows can be exactly calculated. The following four terms can be defined in the same way for the other states:

$$t_2 = \bar{s}_0 \wedge s_1 \wedge av(\bar{i}_0, \bar{i}_1) \quad (9)$$

$$t_3 = s_0 \wedge \bar{s}_1 \wedge av(\bar{i}_0, i_1) \quad (10)$$

$$t_4 = s_0 \wedge s_1 \wedge av(i_0, \bar{i}_1) \quad (11)$$

As a result, a fuzzy normal form can be defined by the following formula:

$$o = t_1 \vee t_2 \vee t_3 \vee t_4 \quad (12)$$

The concept works for all convolutional codes. If the output is calculated the states, which now get fuzzy can be calculated by a shift operation as described above.

6. CONCLUSION

Fig. 4 depicts results using the decoder as described above in a CDMA environment where signals are to be transmitted between mobile stations and the base station of a wideband CDMA system. As stated above only a small degradation in BER performance is observed compared to the classical Viterby approach for bit error rates lower than $2 \cdot 10^{-2}$. This degradation is the price to be paied on the merit of economizing computational ressources leading as well to very good results in terms of raw BER performance which may be further increased by for example interleaving techniques.

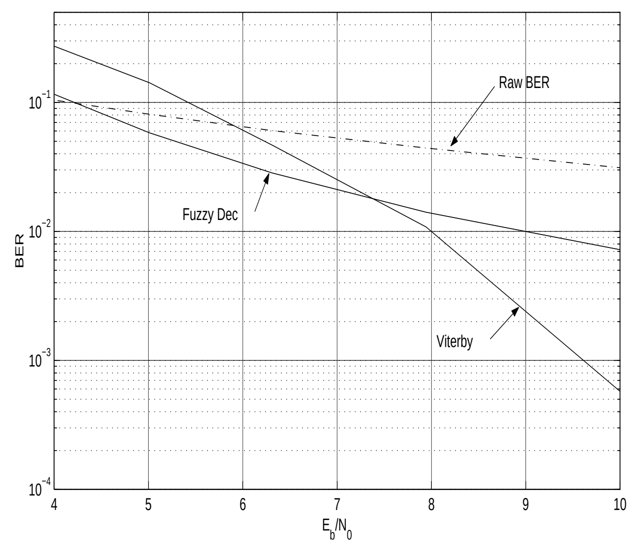

Fig. 4: BER vs. Raw BER for the Fuzzy Decoder

As shown in Fig. 4 the degradation can only be observed for a BER which is better than $2 \cdot 10^{-2}$. This means if lower BER values are expected the Decoder presented in this paper is more suitable. This effect might be interesting for optimised Turbo Decoders, which in the first iteration reach only a slight improvement of the BER in further iterations of the decoder the final wanted BER is reached. Thus using as well the presented decoder for the first iteration and than if the BER is low enough the conventional Viterby decoder should be used. The idea shown in this paper is a first step in using fuzzy-logic to design a decoder unit which is designated for use in moble transmission environments. The use of fuzzy-logic in telecommunication systems is not well investigated yet, but with this paper a starting point for further investigation may be settled. Performance of the algorithm can be further improved which is currently under investigation. Details of this research work related to the above discussed fuzzy based decoder will be shown in the full paper, which includes as well a comparision with the classical Viterby decoder.

7. REFERENCES

[1] Biglieri, E.; Proakis, J.G.; Shamai S.; Fading Channels : Information Theoretic and Communication Aspects; IEEE Trans. on Inform. Theory; October 1998.

[2] Glisic S., Vucetic B. ; Spread Spectrum CDMA Systems for Wireless Communications; Artech House Inc. 1997.

[3] Lukasiewicz, J; Die logischen Grundlagen der Wahrscheinlichkeitsrechnung; Krakow, 1913.

[4] Peterson, W. W. ; Weldon, E. J. ; Error-Correcting Codes; MIT Press 1972.

[5] Proakis, J.G.; Digital Communications; McGraw-Hill Series in Electrical and Computer Engineering; Third Edition 1995.

[6] Sklar, B.; Rayleigh Fading Channels in Mobile Digital Communication Systems - Part I : Characterization, and Part II : Mitigation; IEEE Comm. Mag.; September 1997, pp. 136- 146.

[7] Viterby, A.J.; Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, IEEE Trans. Inform. Theory, Vol IT-13 , pp 260-269 April 1967.

[8] Zadeh, L. A. ; Fuzzy sets; Infomation and Control; 8:338-353, 1965.

[9] see http://www.3gpp.org, series 25.2xy-series