On the Performance-Complexity Tradeoff of Convolutional Codes for Broadband FWA Systems

I. Chatzigeorgiou[†], A. Demosthenous[‡], M. R. D. Rodrigues[*] and I. J. Wassell[†]

[†] Digital Technology Group, Computer Laboratory, University of Cambridge

Cambridge CB3 0FD, United Kingdom

{ic231, ijw24}@cam.ac.uk

[‡] Department of Electronics and Electrical Engineering, University College London

London WC1E 7JE, United Kingdom

a.demosthenous@ee.ucl.ac.uk

[*] Instituto de Telecomunicações and Department of Computer Science, University of Porto

4169-007 Porto, Portugal

mrodrigues@dcc.fc.up.pt

Abstract

In this paper, we investigate the performance-complexity tradeoff of convolutional codes for broadband fixed wireless access systems by considering the effects of quantization and path metric memory in practical Viterbi decoding implementations. We show that in systems with limited antenna diversity, low memory codes achieve a better error-rate performance compared to that of high memory codes. Only in systems with considerable antenna diversity, can the performance of a convolutional code be improved by increasing its memory size. Nevertheless, we demonstrate that the coding advantage offered by the high memory codes is not large enough to justify the significant increase in implementation complexity. In particular, memory-2 convolutional codes achieve a coding gain of up to 1.2 dB over their memory-8 counterparts in single-input single-output fixed wireless access systems. The situation is reversed when multiple antennas are used, but the decoder of memory-8 codes occupies at least 130 times more silicon area than that of memory-2 codes.

Index Terms – Fixed wireless access, complexity, convolutional codes, Viterbi decoding, performance.

1 Introduction

Broadband fixed wireless access (FWA) systems enable high data-rate communications where traditional landlines are either unavailable or too costly to be installed. These systems also enable operators in a competitive environment to rollout broadband services in a rapid and cost effective manner [1]. In this context, broadband FWA standardization activities have been performed under the auspices of the IEEE 802.16 and the ETSI HIPERMAN working groups. In particular, the IEEE 802.16a standard proposes a number of transmission techniques to combat multipath fading in broadband FWA systems, for example orthogonal frequency-division multiplexing (OFDM). This standard also proposes a number of coding techniques to further improve performance in broadband FWA systems, including the concatenation of Reed-Solomon and convolutional codes, block turbo codes and convolutional turbo codes [2].

Broadband FWA systems are characterized by low mobility/multipath propagation conditions, and hence by low time and frequency diversity. This lack of diversity significantly compromises the performance of coding techniques, since occasional deep fades cause severe error bursts that cannot be counteracted by interleaving/deinterleaving operations due to delay and latency considerations [3]. In a previous paper [4] we demonstrated that the use of powerful coding techniques provides only limited performance gains in broadband FWA systems. In particular, we have compared the performance of various coding schemes under the condition of identical decoding complexity which is measured in terms of the total number of computational operations.

Although the total number of operations gives an indication of the overall computational complexity of a decoding algorithm, it does not provide an appreciation of its hardware requirements. In contrast to [4] in which ideal decoding of non-quantized receive sequences is assumed, in this paper we consider convolutional coding and a practical implementation of the Viterbi algorithm; decoding parameters such as the quantization precision of the receive bits, the size of the memory allocated for the path metrics and the size of the sliding window are taken into account. We then characterize the hardware complexity of the Viterbi decoder based on the number of digital gates required for its implementation and we investigate the performance-complexity tradeoff of convolutional-coded FWA systems, both with and without antenna diversity, while exploring the effects of the decoding parameters on performance and complexity.

2 System Description

This section briefly describes the system model under consideration. In particular, we consider systems based on OFDM transmission, which lies at the heart of current broadband FWA standards [2]. We also consider single antenna FWA systems, which do not exploit space diversity, as well as multiple antenna FWA systems, which do exploit space diversity. Figure 1 depicts the system block diagram, where N_T and N_R represent the number of transmit and receive antennas, respectively.

At the transmitter, the information bits are convolutionally encoded and block interleaved. The mapper converts groups of $\log_2 M$ bits into one of M complex symbols from a unit power M-phase shift keyed (M-PSK) or M-quadrature amplitude modulation (M-QAM) constellation. In single antenna systems ($N_T = 1$), the space-time processing block does not further process the modulation symbols; instead, the modulation symbols are passed directly to the OFDM block. However, in multiple transmit antenna systems ($N_T > 1$), the space-time processing block will further process the modulation symbols before passing them to the OFDM block [5]. In particular, the space-time processor generates for each particular OFDM sub-carrier, a space-time block code (STBC) according to the generator matrices $\mathbf{G}_2$, $\mathbf{G}_3$ or $\mathbf{G}_4$ given by [6], [7][1]. Essentially, a total of $K \times N_T$ symbols obtained from the original K' modulation symbols are transmitted during K separate time slots by N_T transmit antennas by each particular OFDM sub-carrier. Note that $\mathbf{G}_2$, $\mathbf{G}_3$ and $\mathbf{G}_4$ are appropriate for two, three and four transmit antennas, respectively, and for an arbitrary number of receive antennas. Note also that $\mathbf{G}_2$ is rate $K'/K = 1$, whereas $\mathbf{G}_3$ and $\mathbf{G}_4$ are rate $K'/K = 1/2$. Finally, at each transmit antenna chain, N complex symbols corresponding to the elements for a particular time slot for the N different STBCs are imposed onto N orthogonal sub-carriers by means of a serial-to-parallel (S/P) conversion and an inverse fast Fourier transform (FFT). A parallel-to-serial (P/S) converter multiplexes the N parallel signals, a cyclic prefix is inserted with duration longer than the impulse response of the channel to combat intersymbol and intercarrier interference and the OFDM signal is then digital-to-analogue converted.

[1] We consider space-time coded OFDM systems where redundancy spans space and time domains, rather than space-frequency coded OFDM systems where redundancy spans space and frequency domains.

The OFDM signal is distorted by a broadband FWA channel as well as additive white Gaussian noise (AWGN). The broadband FWA channel is time-dispersive but not significantly time-varying. Hence, we assume that the channel is constant during the transmission of a frame of data.

At the receiver, which we implicitly assume maintains perfect channel state information, the signal is analogue-to-digital converted at each receive antenna chain, the cyclic prefix is removed and the N complex symbols corresponding to the elements for a particular time slot for the N different STBC are removed from the N orthogonal sub-carriers by means of an FFT, as is depicted in Figure 1. A soft demapper converts the complex symbols into soft bits or log-likelihood ratios (LLR)[2], which are then block de-interleaved and Viterbi decoded. In our work, we consider an implementation of the Viterbi algorithm that takes into account various practical aspects influencing performance and complexity, including soft bit quantization, path metric memory and sliding window size, as described in [8].

3 Performance-Complexity Tradeoff

This section studies the tradeoff between the performance of various convolutional codes and the complexity of the corresponding Viterbi decoders, both in single antenna and multiple antenna broadband FWA systems. In the study, we consider terminated non-recursive non-systematic convolutional (NRNSC) encoders with rate 1/2, generator polynomials (5,7), (53,75), (133,171) and (561,753) in octal form, which correspond to memory sizes 2, 5, 6 and 8 respectively, and an output frame size of 2048 bits. Note that the selected polynomials correspond to maximum free distance convolutional codes [9], [10]. We consider both ideal and practical Viterbi decoders; the practical decoders have a quantization precision of 1, 2 or 3 bits and employ best-state decoding with a finite sliding window of size equal to four times the code constraint length. Note that the coding gain achieved by a quantization precision of 3 bits is comparable to that of infinitely fine quantization [8]; additional quantization bits yield little performance improvement.

In this paper, we also consider quadrature phase shift keying (QPSK) and OFDM with $N = 256$ sub-carriers; the OFDM signals are characterized by a symbol duration of $T = 12.8$ μs and a cyclic

[2] Note that the channel state information is embodied in the LLR expressions [4].

prefix duration of T_{CP} = 3.2 μs. Consequently, the data-rate is 16Mb/s for both single antenna systems and multiple antenna systems using the STBC specified by $\mathbf{G}_2$, whilst the rate drops to 8 Mb/s when multiple antenna systems using the STBC specified by $\mathbf{G}_3$ or $\mathbf{G}_4$ are considered. Six interim broadband FWA channel models have been adopted by the IEEE 802.16a standard [11]. We use the SUI3 model, which corresponds to average suburban conditions. This model includes three fading taps with delays 0 μs, 0.5 μs and 1.0 μs, with relative powers 0 dB, –5 dB and –10 dB, and with K-factors 1, 0 and 0, respectively. The SUI3 channel model specifies an antenna correlation coefficient of value 0.4. Note that in our simulations the channel is considered constant during the transmission of a data frame owing to the extremely low Doppler spread value of 0.4 Hz.

3.1 Description of a practical implementation of the Viterbi decoding algorithm

We use uniform quantization [8] with μ_Q-bit precision, 2^{μ_Q} quantization levels and optimum normalized step size[3]. The optimum normalized step sizes obtained for the single antenna and the multiple antenna broadband FWA scenarios approximately correspond to the optimum ones for the AWGN scenario; performance is essentially constant over a range of normalized step values centered on 1 for 2-bit precision and 0.5 for 3-bit precision [8]. We have also observed that the optimum normalized step size is approximately independent of the convolutional code and the ratio of the energy per bit, E_b, to the spectral noise density N_0, denoted as E_b/N_0.

The convolutional decoder uses the Viterbi algorithm, initially proposed in [12]. The Viterbi algorithm is suitable for the high data-rate applications targeted in our paper because it provides maximum-likelihood solutions, whilst the delay introduced by the decoding process is independent of the channel conditions. In contrast, algorithms based on sequential decoding essentially follow a trial-and-error approach, which makes them unsuitable for high data-rate applications since their decoding complexity increases as the channel quality degrades [13]. Furthermore, Viterbi decoding is the dominant technique employed for the constraint lengths considered in our paper [10], [14]; Viterbi decoders for convolutional codes having a similar range of constraint lengths have already been developed by Qualcomm (e.g., Q1900) [15] and other telecoms chipset manufacturers.

[3] The soft bits are scaled prior to quantization in order to normalize the average noise power to one.

The Viterbi algorithm computes and stores a metric for each branch of the trellis diagram. If μ_Q quantization bits are used to represent a symbol, which is input to the Viterbi decoder of a rate 1/2 code, it can be shown that no more than μ_Q+1 bits are required to store a branch metric [16]. The decoder has been configured in such a way that a small branch metric represents a highly probable event, while larger metrics represent less likely events. When all branch metrics at a particular trellis step have been computed, an add-compare-select process identifies the path having the lowest metric, at each memory state. This metric is stored in the path metric memory, the size of which depends on the constraint length of the code as well as the quantization precision [16]. The optimal size of the path metric memory, denoted as μ_{PM}, for various code configurations when the quantization precision is set to 1, 2 or 3, is presented in Table I. The value for μ_{PM} was determined by monitoring the usage of a memory pool, which was allocated for storing the path metric at a memory state for each trellis step. A path metric memory of size higher than μ_{PM} would not yield any performance improvement; however, a lower memory size would degrade the performance of the Viterbi decoder. Note that path metric renormalization has been implemented, according to which the minimum path metric is subtracted from all path metrics, at each trellis step. Path metric renormalization avoids memory overflow and results in a limited range of path metric values [17]; hence, only a small memory size for path metric storage is required. However, approaches based on modulo arithmetic [17], [18] could also be used.

Practical implementations of the Viterbi decoding algorithm store a fixed length of past trellis transitions and output the oldest decoded bit, before making a step deeper into the trellis [8]. It has been found that this method, known as sliding window, approaches optimal decoding performance when the size of the window is 4 or 5 times the code constraint length, provided that best-state decoding is used. In best-state decoding [19], the metrics of all paths are compared at each trellis step and the path having the lowest metric is traced back. However, taking into account that any survival path will usually merge with the correct path within a few braches, we can trace back a path starting from any arbitrary fixed state [16]. Contrary to best-state decoding, fixed-state decoding requires a longer sliding window of size at least 7 or 8 times the code constraint length but performs no path metric comparisons, therefore it exhibits lower complexity than best-state decoding [16]. Nevertheless, best-state decoding is the most suitable candidate when path metric renormalization is used, since the best state is readily identified by the renormalization circuit. In general, when the size restrictions of

the end product are flexible, such as in broadband FWA systems, the designer can combine modulo-arithmetic with fixed-state decoding to minimize hardware complexity requirements for the path elimination process or path metric renormalization with best-state decoding to minimize memory requirements for storing the history of the winning path metrics. In our implementation we use best-state decoding in conjunction with a sliding window of size four times the code constraint length.

For comparison purposes, we have also considered ideal Viterbi decoders, which use infinitely fine quantization, no path renormalization, best-state decoding and a sliding window of size equal to the length of the input information sequence.

3.2 Complexity characterization

The functional block diagram of a practical Viterbi decoder employing best-state decoding is shown in Figure 2 [20]. The decoder consists of the following blocks: branch metric computer (BMC), the add-compare-select (ACS), storage survivor memory (SSM), and output decision (OD). The BMC block performs a correlation operation in order to compute the branch metrics using a suitable distance function, the ACS block performs addition, maximum or minimum selection, path elimination and path metric storage, the SSM block keeps track of the decisions made in the ACS block in a hypothesized digital representation, and the OD block outputs the decoded bits corresponding to the best-state information. Since the ACS block requires the highest computational processing, its hardware complexity dictates the Viterbi decoder complexity [20]-[23]. The complexity of the BMC block is negligible compared to that of the ACS and SSM blocks. For a given code memory size, denoted as ν, the complexity of the SSM block is fixed, irrespective of the number of bits used for the path metric memory; the SSM block essentially consists of $L\cdot(\nu+1)\cdot 2^{\nu}$ single-bit memory units where $L\cdot(\nu+1)$ is the size of the sliding window expressed in multiples of the code constraint length, which is given by $\nu+1$. However, the complexity of the ACS block does depend on the number of bits used for the path metric memory. We base the complexity characterization on the number of digital gates (each equivalent to the complexity of a 4-transistor NAND gate) required for the implementation of the ACS and SSM blocks.

The operations performed by the ACS block are better understood by breaking the trellis up into a number of identical “butterfly” modules as shown in Figure 3(a) for a binary convolutional code; observe that we have used a radix-2 implementation, which is sufficient for the data-rate requirements

of our system. Since the entire trellis is constructed from replicas of this "butterfly" module, the computational unit in Figure 3(b), which is referred to as the add-compare-select unit (ACSU) can be used recursively to implement the ACS block. Assuming the minimum path metric is selected, the ACSU computes the following recursive relationship:

$$\Gamma_{i,t+1} = \min_{j}\left(\Gamma_{j,t} + \lambda_{ji,t}\right), \text{ for } i = 1,2,\ldots,2^{\nu} \text{ and } j = 1,2,\ldots,2^{\nu}, \tag{1}$$

where $\Gamma_{j,t}$ is the survivor path metric for memory state j at time t and $\lambda_{ji,t}$ is the branch metric of the transition from memory state j at time t to memory state i at time $t + 1$.

The maximum operating speed of the ACSU is limited by the propagation delay around the local feedback loop (i.e., ACS loop). Thus, for high-speed Viterbi decoding that is required in FWA applications, a state-parallel architecture [16], [21] is adopted where the number of ACSUs in the ACS block is equal to the number of memory states, i.e., 2^{ν}. For example, the memory-2 (5,7) code has 4 memory states and requires 4 ACSUs, while the memory-8 (561,753) code has 256 memory states and requires 256 ACSUs. The implementation of the ACSU in Figure 3(b), requires two adders, a comparator, a 2-to-1 line multiplexer, two registers and a subtractor for path metric renormalization. Furthermore, the ACS block requires $(2^{\nu} - 1)$ additional comparators, for selecting the minimum path metric for renormalization purposes. Since the renormalization circuit readily identifies the best state, no extra circuitry is required to perform best-state decoding in the SSM block. Note that each component of the ACS block is μ_{PM}-bit wide, where μ_{PM} is the precision of the path metric memory. The SSM block using the *register exchange* method requires $L\cdot(\nu+1)\cdot 2^{\nu}$ D-type flip-flops and $L\cdot(\nu+1)\cdot 2^{\nu}$ single-bit 2-to-1 line multiplexers [20]. If G_{add}, G_{comp}, G_{mux}, G_{sub}, G_{reg}, G_{FF} and $G_{mux\text{-}1bit}$ denote the equivalent number of gates for a multi-bit adder, multi-bit comparator, multi-bit 2-to-1 line multiplexer, multi-bit subtractor, multi-bit register, D-type flip-flop and single-bit 2-to-1 line multiplexer, respectively, we can compute the total number of gates G_{total} composing the ACS and SMM blocks, using

$$G_{total} = \underbrace{2^{\nu}\left(2G_{add} + G_{comp} + G_{mux} + G_{sub} + 2G_{reg}\right) + \left(2^{\nu} - 1\right)\cdot G_{comp}}_{\text{ACS block}} + \underbrace{L\cdot(\nu+1)\cdot 2^{\nu}\left(G_{FF} + G_{mux-1bit}\right)}_{\text{SSM block}}. \tag{2}$$

The values of G_{add}, G_{comp}, G_{mux}, G_{sub} and G_{reg}, when the path metric memory μ_{PM} is in the range between 2 and 7 bits, are presented in Table II. The equivalent number of gates for G_{FF} and $G_{mux\text{-}1bit}$ is fixed and equal to 6 and 3, respectively, since the complexity of the SSM block does depend on μ_{PM},

as previously mentioned. It is important to note that that the target data-rate of our system does not dictate the use of parallelism techniques, which are often introduced in very high data-rate schemes [18], [23]. For this reason, the equivalent number of gates for each component of an ACS block shown in Table II, was determined by using standard cells from the digital library of the 0.35-µm austriamicrosystems CMOS process technology [24]. These standard digital cells have worst case propagation delay of less than 5 ns, allowing the implementation of radix-2 ACSUs for state-parallel architecture that can be clocked at about 50 MHz (worst case, as we established from back annotating the layout parasitic capacitances into the circuit schematic). The selected operating frequency is sufficient for our target data rate of 16 Mb/s. The complexity information is summarized in Table I.

Similarly, we can calculate the silicon area occupation of the ACS and SSM blocks using the following expression:

$$A_{\text{total}} = a\left[\underbrace{2^{\nu}\left(2A_{\text{add}} + A_{\text{comp}} + A_{\text{mux}} + A_{\text{sub}} + 2A_{\text{reg}}\right) + \left(2^{\nu} - 1\right)\cdot A_{\text{comp}}}_{\text{ACS block}} + \underbrace{L\cdot\left(\nu+1\right)\cdot 2^{\nu}\left(A_{\text{FF}} + A_{\text{mux-1bit}}\right)}_{\text{SSM block}}\right], \quad (3)$$

where A_{add}, A_{comp}, A_{mux}, A_{sub}, A_{reg}, A_{FF} and $A_{\text{mux-1bit}}$ denote the area of a multi-bit adder, multi-bit comparator, multi-bit 2-to-1 line multiplexer, multi-bit subtractor, multi-bit register, D-type flip-flop and single-bit 2-to-1 line multiplexer, respectively, and the factor a is included to take into account the area of the chip interconnections. For the targeted 0.35-µm CMOS process technology, Table III presents the breakdown of the silicon area for each of the ACSU sub-blocks for μ_{PM} values between 2 and 7 bits (note that a Verilog code was written and linked to the process technology in order to extract this information in the Cadence tool environment). The silicon area for G_{FF} and $G_{\text{mux-1bit}}$ is 279.3 µm^2 and 111.7 µm^2, respectively. The total area, A_{total}, was calculated assuming a = 1.5. The resulting A_{total} values for the various decoder scenarios considered in our study are listed in Table I.

3.3 Results

Figures 4 and 5 depict the bit error rate (BER) performance of various convolutional codes in broadband FWA systems with and without antenna diversity for the cases of ideal and practical Viterbi decoding, respectively. It is interesting to note that, in contrast to the situation in the AWGN channel, lower memory convolutional codes can outperform a higher memory one in broadband FWA systems. In particular, in systems without antenna diversity (i.e., single transmit-single receive antenna systems) a lower memory convolutional code outperforms a higher memory code from low to high

E_b/N_0 values. On the other hand, in systems with significant antenna diversity higher memory convolutional codes outperform lower memory ones. We attribute these trends to the fact that high memory convolutional codes exhibit worse BER in the low E_b/N_0 regime than low memory ones on an AWGN channel (see Figures 4 and 5). Note that in a wireless channel the BER performance for a specific average E_b/N_0 value is obtained by averaging the BER performance for various instantaneous E_b/N_0 values. In systems with no antenna diversity low instantaneous E_b/N_0 values occur with relatively high probability, thus the decoders are being swamped by errors. However, low memory codes are at an advantage since they are able to rejoin the true trellis path more quickly than high memory codes [25]. Increasing the memory size at low instantaneous E_b/N_0 values also increases the length of the path that deviates from the true trellis path and causes more decoding errors [25]. In contrast, in systems with significant antenna diversity low instantaneous E_b/N_0 values occur with much lower probability, so that a high memory convolutional code outperforms a low memory one. These trends are observed not only in systems using ideal Viterbi decoding but also in systems using practical implementations of the Viterbi algorithm.

The relative benefit of convolutional coding for broadband FWA systems can be quantified by studying the performance-complexity tradeoff of convolutional codes of various memory sizes. Table I compares the performance of various convolutional codes (in terms of E_b/N_0 at a target BER of 10^{-4}) and the complexity of the corresponding Viterbi decoders (in terms of number of digital gates required to implement the ACS process) in various broadband FWA scenarios. The silicon area requirement for the implementation of the ACS block is also listed. As noted from the results in Table I, for single transmit-single receive antenna systems the advantage in coding gain of the (5,7) code increases from 0.2 dB to 1.2 dB compared to the (561,753) code, as the quantization precision increases from 1 bit to 3 bits, whilst the memory-2 code is about 130 to 180 times less complex than the memory-8 code with corresponding reduction in silicon area occupation. For two transmit-two receive antenna systems, the (53,75) code has a loss that varies between 0.1 dB and 0.3 dB compared to the (561,753) code, depending on the quantization precision, but it is 11 times less complex on average. For systems employing four antennas at the transmitter and an equal number of antennas at the receiver, the (561,753) code outperforms the (5,7) code by about 1 dB, when 3-bit soft-decision decoding is used. Nevertheless, the (561,753) code achieves better performance at the expense of about 130 times the

complexity of the (5,7) code. As expected, a 3-bit soft-decision decoder has an advantage in coding gain over a 1-bit hard-decision decoder, which varies between 1.8 dB and 3.3 dB, depending on the number of antennas and the quantization precision.

In general, our study shows that for multiple antenna systems the coding advantage offered by the high memory codes is not large enough to justify the exponential increase in complexity and silicon area occupation.

4 Conclusion

In this paper we compare the error rate performance of broadband FWA systems using convolutional codes of various memory sizes. In particular, we consider practical implementations of the conventional Viterbi decoder, we analyze their complexity and we investigate their performance in systems both with and without antenna diversity. Our results demonstrate that the selection of high memory convolutional codes in broadband FWA systems that do not exploit antenna diversity increases the decoding complexity without offering an advantage in performance. Less complex low memory convolutional codes can actually achieve a better performance, for bit error rates up to 10^{-4}. Only when antenna diversity is exploited, do high memory convolutionally coded systems outperform systems using low memory convolutional codes. Nevertheless, the advantage offered by the high memory codes is not large enough to justify the exponential increase in hardware complexity. These results are of practical interest for the deployment and design of broadband FWA systems. Future work will consider the study of the performance-complexity tradeoff for other coding techniques, including turbo codes and low density parity check (LDPC) codes, in broadband FWA systems.

References

[1] Bölcskei H., Paulraj A.J., Hari K.V.S. , Nabar R.U., and Lu W.W.: 'Fixed broadband wireless access: State of the art, challenges and future directions', IEEE Commun. Mag., 2001, **39**, (1), pp.100-108

[2] IEEE 802.16a: 'Part 16: Air interface for fixed broadband wireless access systems – Amendment 2: Media access control modifications and additional physical layer specifications for 2-11 GHz', 2003

[3] Xiao P., Carrasco R., and Wassell I.J.: 'Theoretical performance of single and multiple antenna BFWA systems', IET Commun., 2007, **1**, (3), pp. 496-506

[4] Chatzigeorgiou I., Rodrigues M.R.D., Wassell I.J., and Carrasco R.: 'Comparison of convolutional and turbo coding for broadband FWA systems', IEEE Trans. Broadcasting, 2007, **53**, (2), pp. 494-503

[5] Slaney A., and Sun Y.: 'Space-time coding for wireless communications: an overview', IEE Proc.-Commun., 2006, **153**, (4), pp. 509-518

[6] Tarokh V., Jafarkhani H., and Calderbank A.R.: 'Space-time block codes from orthogonal designs', IEEE Trans. Inf. Theory, 1999, **45**, (5), pp. 1456-1467

[7] Tarokh V., Jafarkhani H., and Calderbank A. R.: 'Space-time block coding for wireless communications: Performance results', IEEE J. Select. Areas Commun., 1999, **17**, (3), pp. 451-460

[8] Heller J.A., and Jacobs I.M.: 'Viterbi decoding for satellite and space communication', IEEE Trans. Commun. Tech., 1971, **19**, (5), pp. 835-848

[9] Bossert M.: 'Channel Coding for Telecommunications' (Wiley, 2000), pp. 281-285

[10] Proakis J.G.: 'Digital Communications' (McGraw-Hill, 2001, 4th edn)

[11] IEEE 802.16a cont. IEEE 802.16.3c-01/29r4: 'Channel Models for Fixed Wireless Applications', 2003

[12] Viterbi A.J.: 'Error bounds for convolutional codes and an asymptotically optimum decoding algorithm', IEEE Trans. Inf. Theory, 1967, **13**, (2), pp. 260-269

[13] Feldman J., Frigo M., and Abou-Faycal I.: 'A fast maximum-likelihood decoder for convolutional codes', Proc. IEEE Vehicular Tech. Conf., Vancouver, Canada, Sep. 2002, pp. 371-375

[14] Wicker S.B.: 'Error Control Systems for Digital Communication and Storage' (Prentice-Hall, 1995)

[15] Qualcomm Inc, San Diego, http://www.qualcomm.com

[16] Clark G.C., and Cain J.B.: 'Error-Correcting Coding for Digital Communications' (Plenum Press, 1981)

[17] Hekstra A. P.: 'An alternative to metric rescaling in Viterbi decoders', IEEE Trans. Commun., 1989, **37**, (11), pp. 1220-1222

[18] Black P. J., and Meng T. H., 'A 140-Mb/s, 32-state, radix-4 Viterbi decoder', IEEE J. Solid-State Circuits, 1992, **27**, (12), pp. 1877-1885

[19] Onyszchuk I.M.: 'Truncation length for Viterbi decoding', IEEE Trans. Commun., 1991, **39**, (7), pp. 1023-1026

[20] Demosthenous A., and Taylor J.: 'A 100-Mb/s 2.8-V CMOS current-mode analog Viterbi decoder', IEEE J. Solid-State Circuits, 2002, **37**, (7), pp. 904-910

[21] Lou H.-L.: 'Implementing the Viterbi algorithm', IEEE Sig. Proc. Magazine., 1995, **12**, (5), pp. 42-52

[22] Kawazoe K., Honda S., Kubota S., and Kato S.: 'Ultra-high-speed and universal-coding-rate Viterbi decoder', IEICE Trans. Electronics, 1994, **E77-C**, (12), pp. 1888-1894

[23] Fettweis G., and Meyr H., 'Parallel Viterbi algorithm implementation: breaking the ACS-bottleneck' IEEE Trans. Commun., 1989, **37**, (8), pp. 785-790

[24] Austriamicrosystems, AG, Austria, http://asic.austriamicrosystems.com

[25] Carson W.R., Chatzigeorgiou I, Wassell I.J., Rodrigues M.R.D., and Carrasco R.: 'On the performance of iterative demapping and decoding techniques over quasi-static fading channels', Proc. IEEE Int. Symp. on Personal, Indoor and Mobile Radio Commun., Athens, Greece, Sep. 2007

Tables

Table I: Complexity and performance (E_b/N_0 at BER = 10^{-4}) of practical Viterbi decoding using bit quantization, best-state decoding and a sliding window of size equal to L=4 times the code constraint length. The SUI3 channel model has been considered.

NRNSC Generator Polynomials	Quant. Precision μ_Q (bits)	Path Memory μ_{PM} (bits)	Complexity G_{total} (gates)	Silicon Area A_{total} (mm^2)	E_b/N_0 (dB) $N_T = N_R = 1$	E_b/N_0 (dB) $N_T = N_R = 2$	E_b/N_0 (dB) $N_T = N_R = 4$
(5,7)	1	2	655	0.046	21.4	11.4	9.0
(53,75)	1	3	9817	0.692	21.5	10.9	8.3
(133,171)	1	4	24438	1.732	21.5	10.8	8.2
(561,753)	1	4	116214	8.130	21.6	10.6	7.8
(5,7)	2	4	942	0.070	19.4	10.1	7.4
(53,75)	2	5	11987	0.868	19.9	10.0	6.8
(133,171)	2	5	26291	1.887	20.1	9.9	6.7
(561,753)	2	6	132337	9.471	20.2	9.8	6.4
(5,7)	3	5	1055	0.079	18.1	9.1	7.0
(53,75)	3	6	13073	0.958	18.6	8.8	6.4
(133,171)	3	6	28465	2.067	18.8	8.7	6.2
(561,753)	3	7	141805	10.247	19.3	8.7	6.0

Table II: Equivalent number of gates for each μ_{PM}-bit wide component of an ACS block.

Path Memory (μ_{PM})	Adder (G_{add})	Comparator (G_{comp})	Mutliplexer (G_{mux})	Subtractor (G_{sub})	Register (G_{reg})
2	7	5	4	7	11
3	11	7	6	15	17
4	16	10	8	24	23
5	21	13	10	25	28
6	26	15	12	31	34
7	31	19	14	36	40

Table III: Breakdown of ACSU area in 0.35-µm CMOS for various μ_{PM} values. Area values are in µm^2.

Path Memory (μ_{PM})	Adder (A_{add})	Comparator (A_{comp})	Mutliplexer (A_{mux})	Subtractor (A_{sub})	Register (A_{reg})
2	382.2	254.8	218.4	364.0	618.8
3	618.8	400.4	327.6	819.0	928.2
4	891.8	546.0	436.8	1310.4	1237.6
5	1164.8	691.6	546.0	1365.0	1547.0
6	1437.8	837.2	655.2	1674.4	1856.4
7	1710.8	1055.6	764.4	1983.8	2165.8

Figures

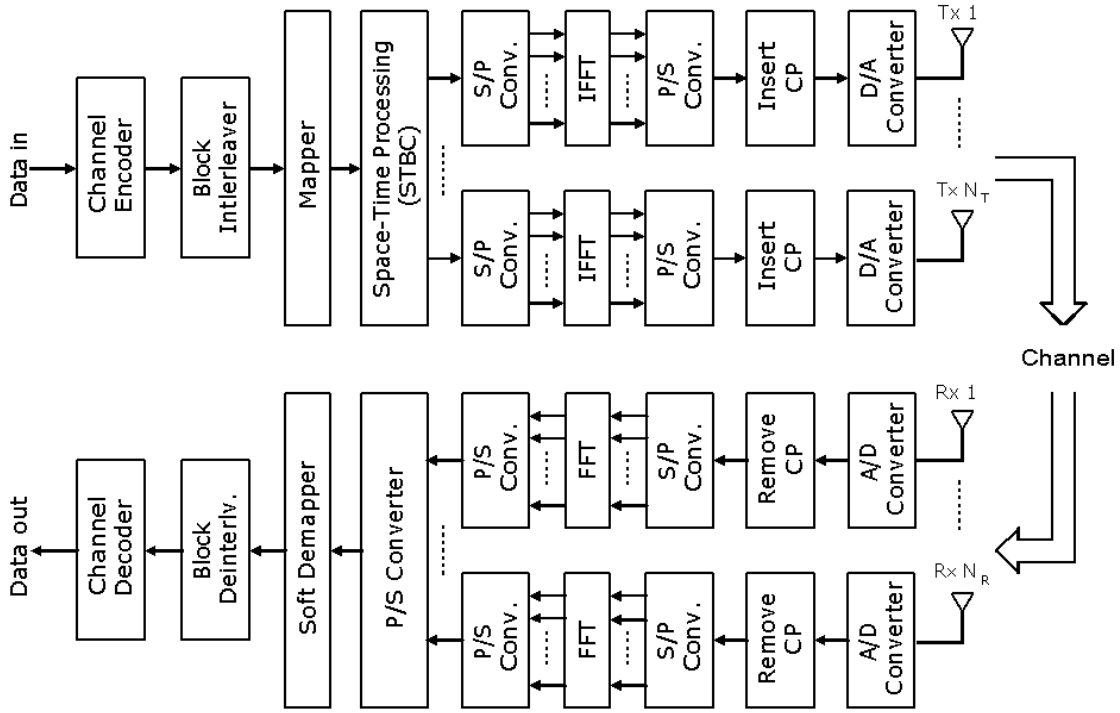


Figure 1: Communications system model.

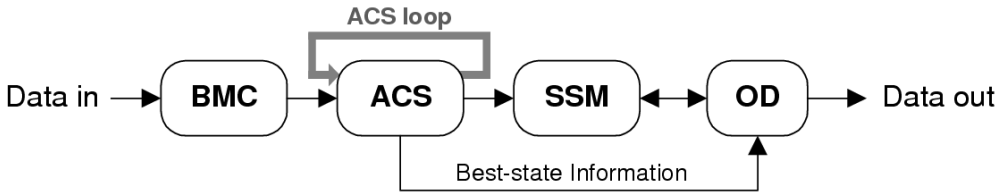
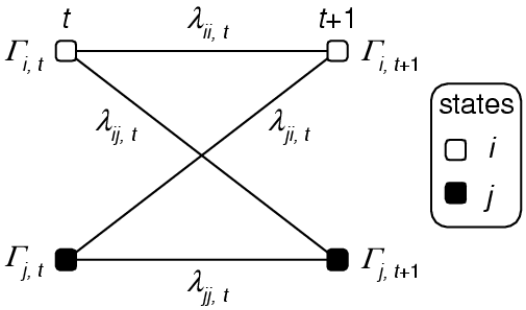
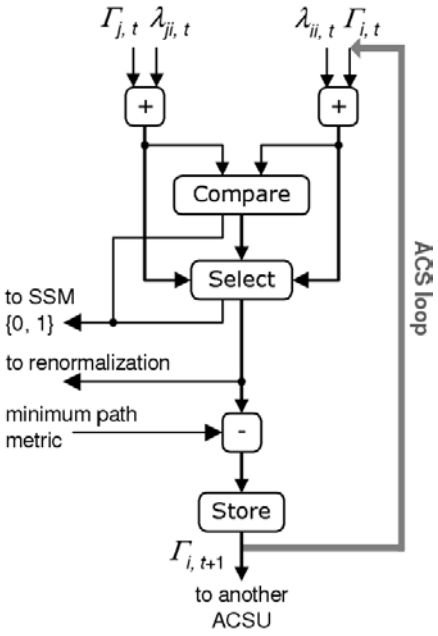


Figure 2: Simplified block diagram of a best-state Viterbi decoder.



(a)



(b)

Figure 3: (a) Basic trellis module for a rate 1/*n* convolutional code. (b) Block diagram of the ACSU used for the same code (for state *i*) for a best-state implementation using path metric renormalization. The '+' block is a binary adder and the '-' block is a binary subtractor.

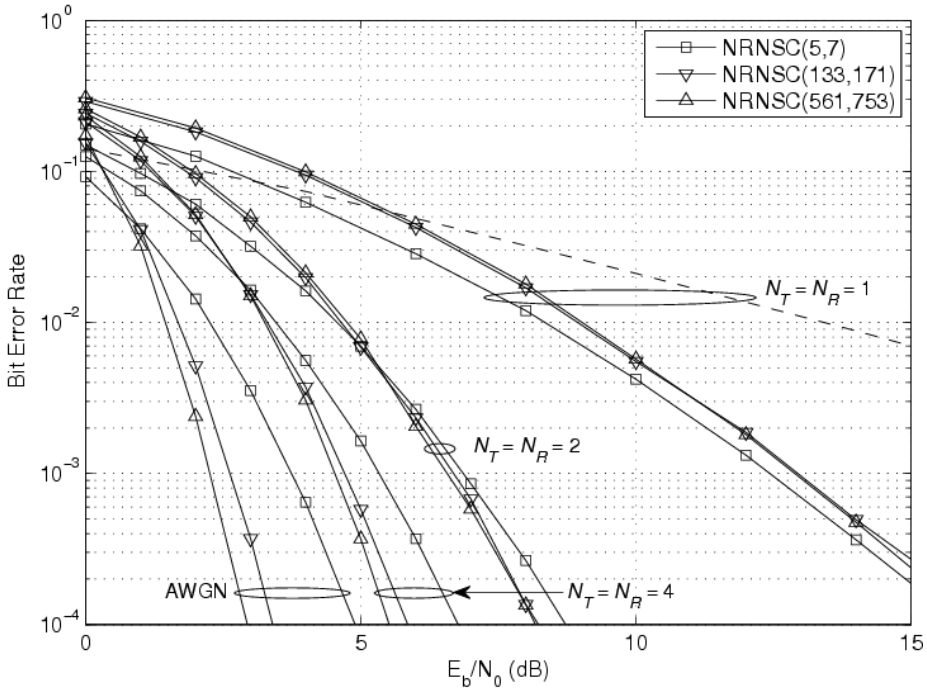


Figure 4: BER performance of convolutional codes in broadband FWA systems with and without antenna diversity. Case of ideal Viterbi decoding.

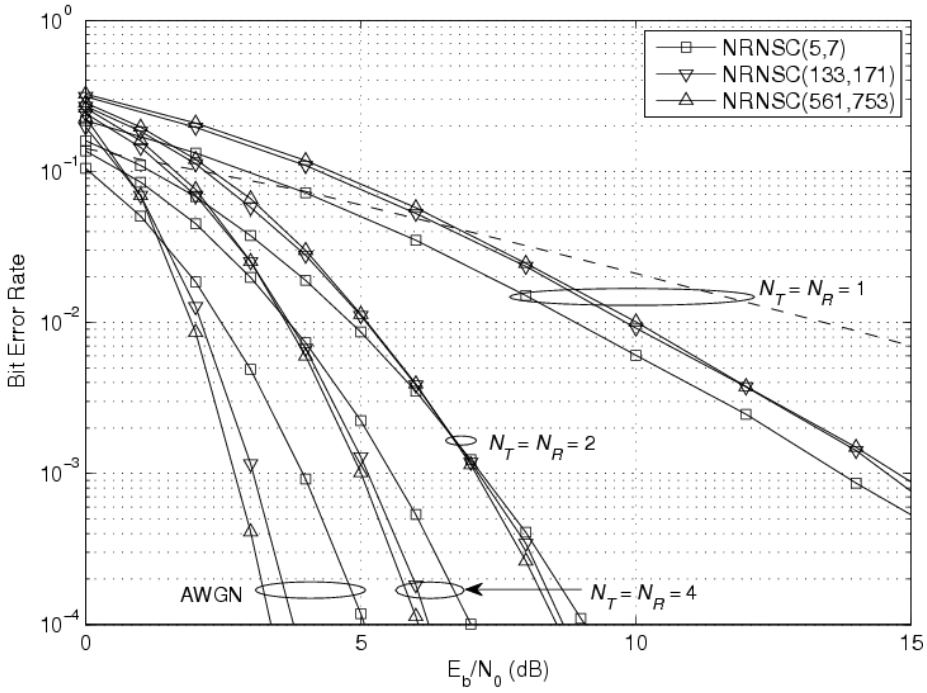


Figure 5: BER performance of convolutional codes in broadband FWA systems with and without antenna diversity. Case of practical Viterbi decoding with 3-bit quantization precision, best-state decoding and sliding window of size equal to 4 times the code constraint length.