KIET IJCE KIET International Journal of Communications &Electronics Volume. No. 3, Issue No. 1, Jan-March 2015, ISSN: 2320 - 8996

Decoding of LDPC Block Codes overConvolutional Codes with Channels

Gaurav Vijay[1]*, Prof. (Dr.) R.P Gupta*[2]
Research Scholar Career Point University,Kota[1]*, Principal and Professor Manda Institute of technology, Bikaner*[2]
Department of Electronics and Communication Engineering,
(Email:gaurav8a@gmail.com[1]*, rpg12@rediffmail.com*[2]*)*

***Abstract*–Decoding of LDPC block codes over Convolutional codes with channels have been shown to be capable of achieving the same capacity-approaching performance as LDPC block codes with iterative message-passing decoding.However, for comparing block and convolutional codes tied to the implementation complexity of trellis based decoding are irrelevant for message-passing decoders. In this paper, we shows a comparison of LDPC block and convolutional codes based on several factors.In this paper the erasure channel are studied. Of special interest will be maximum distance profile (MDP) convolutional codes. These are codes which have a maximum possible column distance increase.This is shown how this strong minimum distance condition of MDP convolutional codes help us to solve error situations that maximum distance separable (MDS) block codes fail to solve. For this, two subclasses of MDP codes are defined: reverse-MDP convolutional codes and complete-MDP convolutional codes. Reverse-MDP codes have the capability to recover a maximum number of erasures using an algorithm which runs backward in time. Complete-MDP convolutional codes are both MDP and reverse-MDP codes. They are capable to recover the state of the decoder under the mildest condition.It is shown that complete-MDP convolutional codes perform in many cases better than comparable MDS block codes of the same rate over the erasure channel.**

***Index Terms*—Convolutional codes, maximum distance separable (MDS) block codes, decoding, erasure channel,maximum distance profile (MDP) convolutional codes, reverse-MDP convolutionalcodes, complete-MDP convolutional codes. maximum distance profile (MDP) convolutional codes, reverse-MDP convolutionalcodes.**

I. INTRODUCTION

As an erasure channel is transmitting, one of the problems encountered is the delay experienced on the received information due to the possible re-transmission of lost packets. One way to eliminate these delays is by using forward error correction. After the invention of turbo codes, researchers became aware that Gallager's low-density parity check (LDPC) block codes [1], were also capable of capacity-approaching performance on a variety of channels. Low-density parity-check (LDPC) codes, although introduced in the early 1960’s [20], were established as stateof-the-art codes only in the late 1990’s with the application of statistical inference techniques [21] to graphical models representing these codes [22], [23]. The promising results from LDPC block codes encouraged the development of convolutional codes defined by sparse parity-check matrices.Analysis and design of these codes quickly attracted considerable attention in the literature, beginning with the work of Wiberg [2], MacKay and Neal [3], and many others.The convolutional counterparts of LDPC block codes, namely LDPC convolutional codes, were subsequently proposed in [4].Analogous to LDPC block codes, LDPC convolutional codesare defined by sparse parity-check matrices that allow them to be decoded using a sliding window-based iterative message passing decoder. The use of convolutional codes over the erasurechannel has been studied much less.Recent studies have shown that LDPC convolutional codes are suitable for practical implementation in a number of different communication scenarios, including continuous transmission as well as block transmission in frames of arbitrary size [5], [6], [7].In this paper, we define a class of convolutional codes with strong distance properties, which we call complete maximum distance profile (complete-MDP) convolutional codes, and we demonstrate how they provide an attractive alternative. They are also known for their encoding simplicity,since the original code construction method proposed in [4] yields a shift-register based systematic encoder for real time encoding of continuous data. This is an advantage when compared to randomly constructed LDPC block codes.Given their excellent bit error rate (BER) performance along with their simplicity ofencoding, it is quite natural to compare LDPC convolutional codes with corresponding LDPC block codes. In this paper, we compare these codes under several different assumptions: equal decoding computational complexity, equal decoding processor (hardware) complexity,equal decoding memory requirements, and equal decoding delay.

The paper is organized as follows. In Section II, we provide a brief overview of LDPC convolutional codes. The maincontribution of the paper is Section III, where comparisonsof LDPC block and convolutional codes based on severalcriteria are presented. In the next section, we focus on finiteblock length comparisons between LDPC block codes andterminated LDPC convolutional codes. Finally, we providesome conclusions in Section V.

II. AN LDPC CONVOLUTIONAL CODES

Here defines the LDPC-CC as a rate R = b/c binary, time-varying LDPC-CC is definedas the set of semi-infinite binary row vectors v[∞] satisfying equation $vH^T=0$. An (m_s; J;K) regular LDPC convolutional code is the setof sequences v satisfying the equation $vH^T= 0$, where

$$H^T=\begin{pmatrix} H_0^T(0) & \cdots & H_{m_s}^T(m_s) & \\ & \ddots & & \\ & & H_0^T(t) \ldots & H_{m_s}^T(t+m_s) \end{pmatrix} \text{..(1)}$$

As the given parameter m_s is called the memory of the code and $v_s = (m_s + 1)$ c is referred to as the constraint length$H^T$is the (time-varying) semi-infinite syndrome former (transposed parity-check) matrix. For a rate R = b/c, b < c, LDPC convolutional code, the elements $H^T{}_i(t)$, i =0,1,2,3,…m_s are binary c X (c - b) sub matrices defined as

$$H_i^T(t)=\begin{bmatrix} h_i^{(1,1)}(t) & \cdots & h_i^{(1,c-b)}(t) \\ \vdots & \ddots & \vdots \\ h_i^{(c,1)}(t) & \cdots & h_i^{(c,c-b)}(t) \end{bmatrix} \text{..(2)}$$

Starting from the m_s (c - b)th column, H^T has J ones in each row and K ones in each column. The value ms,called the syndrome former memory, is determined by themaximal width of the nonzero area in the matrix H^T, and the associated constraint length is defined as.$v_s= (m_s +1)$ c. In practical applications, periodic syndrome former matricesare of interest. Periodic syndrome formers are said to have a period T if they satisfy H_i^T(t) = H_i^T(t+T), i=0, 1,.m_s

The advantage that convolutional codes to blockcodes, which will be exploited in our algorithms, is the flexibilityobtained through the "sliding window" characteristic ofconvolutional codes. The received information can be groupedin appropriate ways, depending on the erasure bursts, andthen be decoded by decoding the "easy" blocks first. Thisflexibility in grouping information brings certain freedom inthe handling of sequences; we can split the blocks in smallerwindows, we can overlap windows and we can proceed todecode in a less strict order.

Although the corresponding Tanner graph has an infinitenumber of nodes, the distance between two variable nodes that are connected to the same check node is limited by the syndrome former memory of the code. This allows continuous decoding that operates on a finite window sliding along the received sequence, similar to a Viterbi decoder with finite path memory [8]. The decoding of two variable nodes that are at least (m_s+1) time units apart can be performed independently, since the corresponding bits cannot participate in the same parity-check equation. This allows the parallelization of theI iterations by employing I independent identical processors working on different regions of the Tanner graph simultaneously. Alternatively, since the processors implemented in the decoder hardware are identical, a single .hopping. Processor that runs on different regions of the decoder memory successivelycan also be employed. A pipeline decoding architecture that is based on the ideas summarized in the previous paragraph was introduced by Jimenez Felstrom and Zigangirov in [4]. The pipeline decoderoutputs a continuous stream of decoded data once an initial decoding delay has elapsed. The operation of this decoder on the Tanner graph for a simple time-invariant rate R = 1/3 LDPC convolutional code with m_s = 2 is shown in Figure 1. (Note that, to achieve capacity-approaching performance, an LDPC convolutional code must have a large value of m_s) .

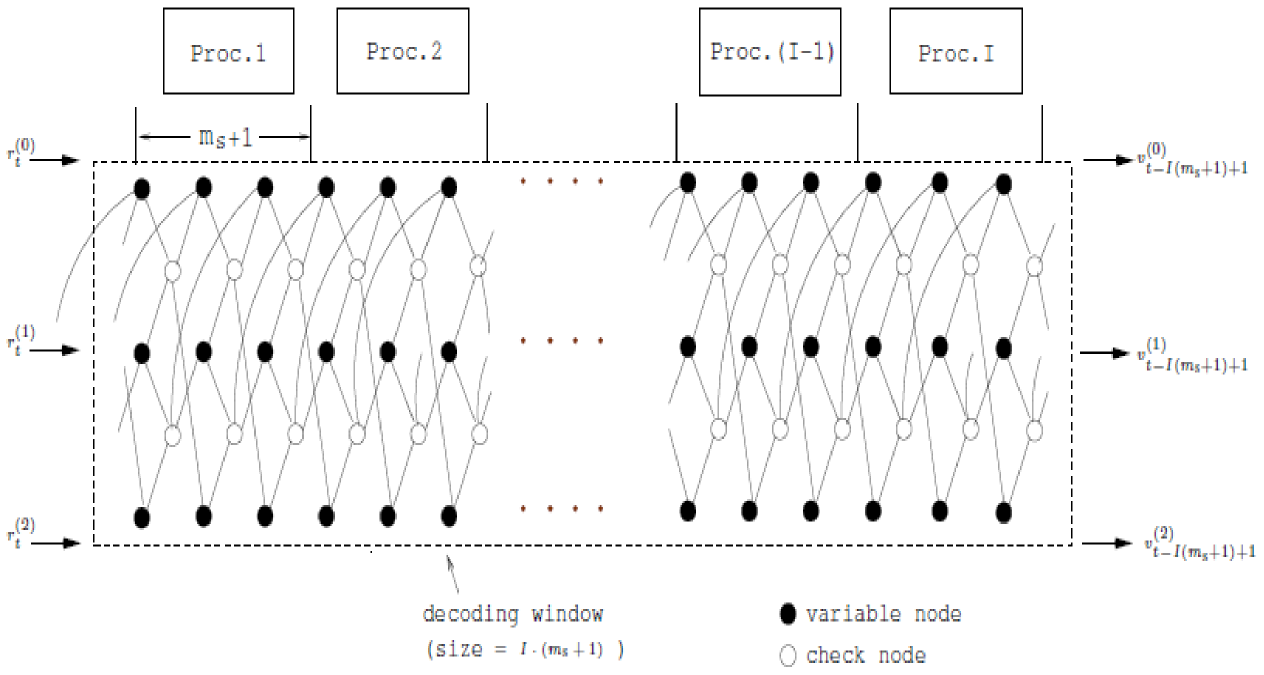


Fig. 1. Tanner graph of an R=1/3 LDPC convolutional code and an illustration of pipeline decoding.

III. COMPARISONS OF LDPC BLOCK AND CONVOLUTIONAL CODES WITH AN IMPLEMENTATION

Here, we compare several aspects of decoding LDPC convolutional and block codes.

A. *Complexity With Computational Methodology*

Let C_{check} (C_{var}) denote the number of computations required for a check (variable) node update for a check (variable) node of degree N (M). Regardless of the code structure, C_{check}andC_{var}only depend on the values M and N. For a rate R = b/c, (m_s; M, N)-LDPC convolutional code decoded using a pipeline decoder with I iterations/processors,at every time instant each processor activates c - b check nodes and c variable nodes. The computational complexity per decoded bit is therefore given by

$$C_{bit}^{conv} = ((c-b).C_{check} + c.C_{var}).\frac{I}{c} \ldots (3)$$
$$= ((c-b).C_{check} + c.C_{var}).I$$

which is independent of the constraint length v_s.Similarly, the decoding complexity for an (L,M,N)-LDPCblock code is given by

$$C_{bit}^{block} = (L.M/N.C_{check} + L.C_{var}).1/L) \ldots (4)$$
$$= (M/N.C_{check} + C_{var}).I$$
$$= ((1-R).C_{check} + C_{var}).I$$

which is again independent of the code length L. Thus, thereis no difference between block and convolutional LDPC codeswith respect to computational complexity.

B. *Processor (Hardware) Complexity*

The sliding window decoder implementation of an LDPCconvolutional code operates on I v_s symbols. Howeverdecoding can be carried out by using I identical independentparallel processors, each capable of handling only $v_s$symbols. Hence it is sufficient to design the processor hardware for v_s symbols. For an LDPC block code of length L, the processor must be capable of handling all L symbols. Therefore, forthe same processor complexity, the block length of an LDPCblock code must be chosen to satisfy L = v_s

C. *Memory Requirements*

For the pipeline decoder, we need a storage element foreach edge in the corresponding Tanner graph. Each variablenode also needs a storage element for the channel value. Thusa total of I. (M + 1) $v_s$storage elements are required for Iiterations of decoding. Similarly, we need L (M +1) storageelements for the decoding of an LDPC block code of lengthL. Thus, for the same memory requirements, an LDPC blockcode must satisfy L = I.v_s

D. Decoding Delay

Let T_{ss} denote the time between the arrival of successivesymbols, i.e., the symbol rate is 1=T_{ss}. Then the maximumtime from the arrival of a symbol until it is decoded is givenby

$$\Delta_{io}^{conv} = ((c-1) + (m_s+1)c.I)T_{ss} \ldots.. (5)$$

The first term (c - 1) in (5) represents the time between the arrival of the first and last of the c encoded symbols output by arate R = b/c convolutional encoder in each encoding interval. The dominant second term (m_s + 1) .c. I is the time eachsymbol spends in the decoding window. Since c symbols areloaded into the decoder simultaneously, the pipeline decoderalso requires a buffer to hold the first (c - 1) symbols.

With LDPC block codes, data is typically transmitted ina sequence of blocks. Depending on the data rate and theprocessor speed, several scenarios are possible. We considerthe best case for block codes, i.e., each block is decoded bythe time the first bit of the next block arrives. This results ina maximum input-output delay of$\Delta_{io}^{block} = K.T_{ss}^{1}$.Thus, for equal decoding delays, the block length must satisfy L =(c - 1) + $v_{s.}$ I, assuming the least possible delay for blockcodes.

E. VLSI implementation requirements

As previously noted, both LDPC block and convolutionalcodes can be decoded using message passing algorithms.Therefore decoder implementations in both cases consist ofidentical processing elements, namely variable nodes andcheck nodes. What differs between the two decoders is thetotal number of these elements and the way in which they areinterconnected.It is well known that VLSI implementations of parallelLDPC block decoders suffer from an interconnection problem[9]. This is due to the fact that processing nodes must be placed on the silicon at specific locations and connected as definedby H. Regardless of how the rows and columns of H arepermuted, long interconnections are still required. The sameobservation was also noted for LDPC block codes constructedusing algebraic techniques [10].However, VLSI implementations of LDPC convolutionaldecoders are based on replicating identical units, termedprocessors. As illustrated in Fig. 2, the complete decoder canbe constructed by concatenating a number of these processorstogether. For comparable BER performance, the size of anLDPC convolutional code processor needs to be about an orderof magnitude less than the block length of an LDPC block code[11]. Therefore the routing complexity within a processor is also an order of magnitude less than for a block code.

There is a wealth of other considerations than impact upon VLSI implementations of LDPC codes. These include the following.

- Any fully parallel LDPC block code decoder may suffer from routing congestion [9]. If so, the total area of such a decoder will be quite large and the maximum clock frequency will be limited by wiring delays.
- The fully parallel LDPC block code decoder can be replaced with a smaller decoder that implements a fraction of the circuit per clock cycle over a number of cycles. This reduces power, area, and throughput in a linear fashion.
- The LDPC convolutional code architecture is more amenable to pipelining because it is inherently feedforward architecture. Therefore it may achieve higher clock speeds.
- Since LDPC convolutional code decoders require fewer check and variable processing elements, the marginal cost of optimizing their critical paths in return for increasing their area is less than for LDPC block codes.
- Memory-based architectures have been proposed for both LDPC block codes [12] and LDPC convolutional codes [13].
- Algebraic code construction techniques can simplify LDPC block code decoder implementations [14], [15]. However since LDPC convolutional codes can often be constructed using the same methods these benefits may be applied to both types of codes [11].
- The choice may be affected by system level issues such as variable frame size, variable code rates, latency budgets, and target frame error rate (FER).

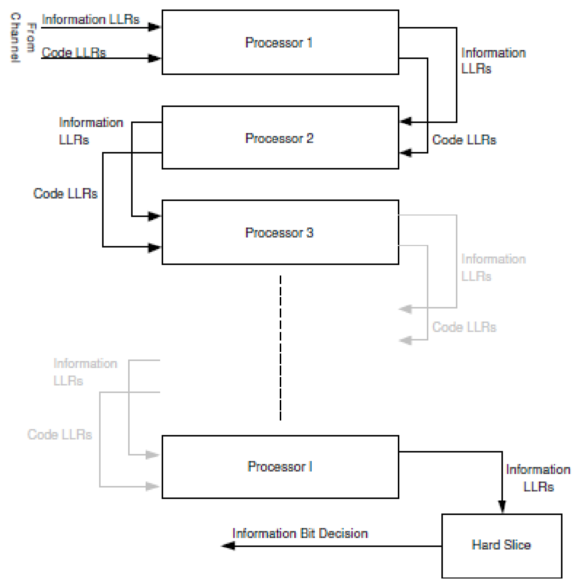


Fig. 2. LDPC convolutional code decoders can be implemented by concatenatingsub-units termed processors.

In Table I we present a brief summary of some existingLDPC block code and LDPC convolutional code VLSI implementations.In [16] a 54 MBPS decoder was implementedon a Xilinx Virtex-E FPGA. In [9] the decoder throughputwas much higher (500 MBPS) because the design was implementedas an ASIC. The BER was about 2X10^{-5} at 2dB. The decoder presented in [10] targets the IEEE 802.3anstandard [17], but it is a hard-decision decoder. This permits avery high throughput but compromises performance. Also note that the codes in [12] and [10] are high-rate, which makes iteasier to achieve higher throughput since less processing isrequired per information bit. The decoder presented in [7] isfor a very powerful LDPC convolutional code, and as such itachieves very good performance at the cost of relatively highcomplexity and low throughput. In [18] the first ever LDPCconvolutional code ASIC decoder is presented. It occupiesthree times less area (in a larger process) than the LDPCblock code ASIC in [9], however it has about three times lessthroughput and the BER performance is worse.The discussion in this section illustrates the point thatmaking a comparison between LDPC block codes and LDPCconvolutional codes that includes BER/FER performance andVLSI implementation complexity is not simple. It depends oncode choice, throughput, power, area, clock speeds, processingnode sizes, and system considerations. It is very possible thatthere is no single best choice between LDPC block codesand LDPC convolutional codes. Instead, LDPC block codesand LDPC convolutional codes may provide complementarysolutions and the appropriate choice may vary from one systemto another. It is worth noting that LDPC block codes havebeen the focus of extensive research for several years whereasLDPC convolutional codes are relatively understudied. The first attempts to implement decoders for LDPC convolutionalcodes noted here are encouraging enough to suggest thatfurther investigation is warranted.

IV. BER-FER PERFORMANCE COMPARISON OF LDPCBLOCK AND CONVOLUTIONAL CODES

In order to test the comparisons given in the previous section, in Figure 3 we plot the performance of a rate R = 1/2,(2048,3,6)-LDPC convolutional code with I = 50 iterationson an AWGN channel. Also shown is the performance of twoM = 3, N = 6 LDPC block codes with a maximum of 50iterations. The block lengths were chosen so that in one casethe decoders have the same processor complexity, i.e., L = $v_s$and in the other case the same memory requirements, i.e., L = v_s .I. For the same processor complexity, the convolutionalcode outperforms the block code by about 0.6 dB at a biterror rate of 10^{-5}. For the same memory requirements, theconvolutional and block code performance is nearly identical.LDPC convolutional codes are very efficient for the transmissionof streaming data since they allow continuous encodingdecoding. However, in some applications, it is preferable to have the data encoded in frames of pre-determined size in order to maintain compatibility with some standard format. Therefore, we now consider the performance of terminated LDPC convolutional codes in this context.

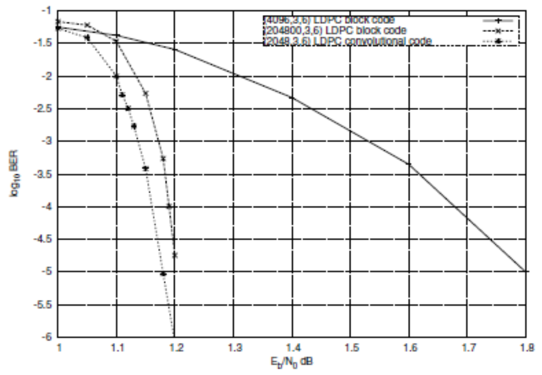


Fig. 3. BER performance comparison of LDPC block and convolutionalcodes.

The information sequence must be terminated with a tail ofsymbols to force the encoder to the zero state at the end of theencoding process. For conventional polynomial convolutionalencoders, the terminating tail consists of a sequence of zeros.For LDPC convolutional code encoders, the tail is, generallyspeaking, non-zero and depends on the encoded informationbits. Therefore, a system of linear equations must be solved[19].

In Figure 4, we show FER performance comparisons ofterminated LDPC convolutional codes versus LDPC blockcodes, assuming 100 decoding iterations. We terminate a rateR = 1=2 (2048; 3; 6) LDPC convolutional code at variousframe lengths, resulting in a variety of terminated blocklengths and rates. We also provide simulation results for LDPCblock codes of comparable block lengths. As shown in Figure4, a single LDPC convolutional code can be employed toconstruct a family of codes of varying frame length and errorperformance via termination. This is an advantage in terms of flexibility compared to LDPC block codes, where a newcode must be constructed each time a new transmission framelength is required.

Figure 4 also shows that, even though the LDPC convolutionalcode with syndrome former memory m_s = 2048 has ahardware complexity comparable to that of a length L = 4096LDPC block code, its performance is similar to much longerLDPC block codes. In particular, for a terminated frame lengthof L = 64512, the LDPC convolutional code outperforms theLDPC block code of length L = 10000 and performs almostas well as the LDPC block code of length L = 100000.

V. ERASURE CHANNELS WITH MEMORY

We now consider the performance of LDPC-CC ensemblesand codes over erasure channels with memory. We considerthe familiar two-state Gilbert-Elliott channel (GEC) [32], [33]as a model of an erasure channel with memory. In this model,the channel is either in a “good” state G, where we assume theerasure probability is 0, or in an “erasure” state E, in whichthe erasure probability is 1. The state process of the channel isa first-order Markov process with the transition probabilitiesP{E→G}=gand P{G→E}=b. With these parameters,we can easily deduce [34] that the average erasure rate ε and the average burst length Δ are given by

$$\varepsilon = P\{E\} = \frac{b}{b+g}\text{ , } \Delta = 1/g$$

We will consider the GEC to be parameterized by the pair(ε, Δ). Note that there is a one-to-one correspondence betweenthe two pairs (b, g) and(ε, Δ).

Discussion: The channel capacity of a correlated binaryerasure channel with an average erasure rate of " is given as(1-ε), which is the same as that of the memoryless channel,provided the channel is ergodic. Therefore, one can obtaingood performance on a correlated erasure channel through theuse of a capacity-achieving code for the memoryless channel with an interleaver to randomize the erasures [24], [27]. This isequivalent to permuting the columns of the parity-check matrixof the original code. We are not interested in this approachsince such permutations destroy the convolutional structure of the code and as a result, we are unable to use the WD forsuch a scheme.

Construction of LDPC block codes for bursty erasure channelshas been well studied. The performance metric of acode over a bursty erasure channel is related to the maximumresolvable erasure burst length (MBL) denoted Δ_{max} [27],which, as the name suggests, is the maximal length of a singlesolid erasure burst that can be decoded by a BP decoder.Methods of optimizing codes for such channels therefore focuson permuting columns of parity-check matrices to maximizeΔ_{max}, Instead of permuting columns of theparity-check matrix, in order to maintain the convolutionalstructure of the code, we will consider designing C_m(J;K)ensembles that Δ_{max}.

A. Asymptotic Analysis

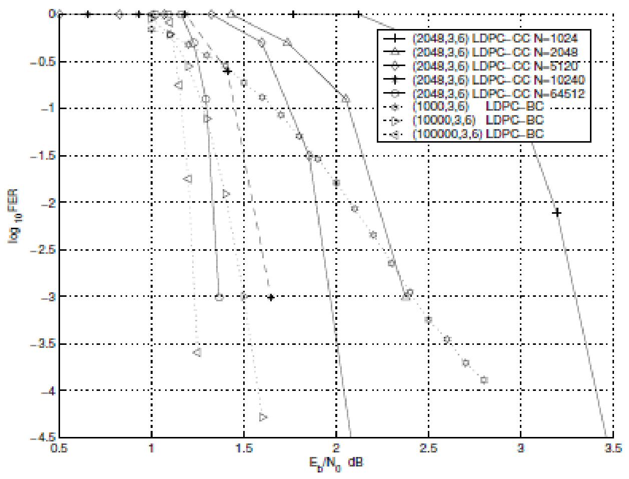
1) BP: As noted earlier, the performance of LDPC-CC ensemblesdepends on stopping sets. The structure of protographsimposes constraints on the code that limit the stopping set sizesand locations, as will be shown shortly.

Let us define a protograph stopping set to be a subsetS(B) of the VNs of the protograph B whose neighboringCNs are connected at least twice to S(B). These are alsodenoted as S(P), in terms of the set of polynomials definingthe protograph. We define the size of the stopping set asthe cardinality of S(B), denoted $|S(B)|$. We call the leastnumber of consecutive columns of B that contain the stopping set S(B) the span of the stopping set, denoted hS(B)i. Letus denote the size of the smallest protograph stopping setof the protograph B by |S(B)|*, and the minimum numberof consecutive columns of the protograph B that contain aprotograph stopping set by <S(B)>*. When the protographunder consideration is clear from the context, we will drop itfrom the notation and use $|S|$* and$\langle S\rangle$*. The minimum spanof a stopping set is of interest because we can give simplebounds for Δ_{max}based on <S(B)>*. Note that the stoppingset of minimal size and the stopping set of minimal span arenot necessarily the same set of VNs. However, we always have

|S(B)|* ≤ <S(B)>*.

TABLE I
A COMPARISON OF EXISTING BLOCK AND CONVOLUTIONAL LDPC CODE IMPLEMENTATIONS.NOTE PERFORMANCE FIGURES, WHERE AVAILABLE, AREGIVEN AT A CERTAINEb / N0

Ref.	Type	Code Params.	Code rate	Device	Perf.	Throughput
[9]	Block	(1024,3,6)	0.5	52.5mm^2 in 0.16mCMOS	BER=2e^{-5}@2dB	500MBPS
[16]	Block	(9216,3,6)	0.5	FPGA	BER=1.0e^{-6}@2dB	54MBPS
[12]	Block	(8176,7154)	0.875	FPGA	N/A	169MBPS
[10]	Block	(2048,1723)	0.875	17.6mm^2 in 0.18µmCMOS	BER=5e^{-5}@6dB	3200MBPS
[13]	Conv.	(128,3,6)	0.5	FPGA	BER=1e^{-3} (1e^{-4})@2dB	75MBPS(25MBPS)
[7]	Conv.	(2048,3,6)	0.5	FPGA	BER=1e^{-10}@2dB	2MBPS
[18]	Conv.	(128,3,6)	0.5	16mm^2 in 0.18µm CMOS	BER=3e^{-4}@3dB	150MBPS



.

Fig. 4. FER performance comparison of terminated LDPC convolutional andblock codes.

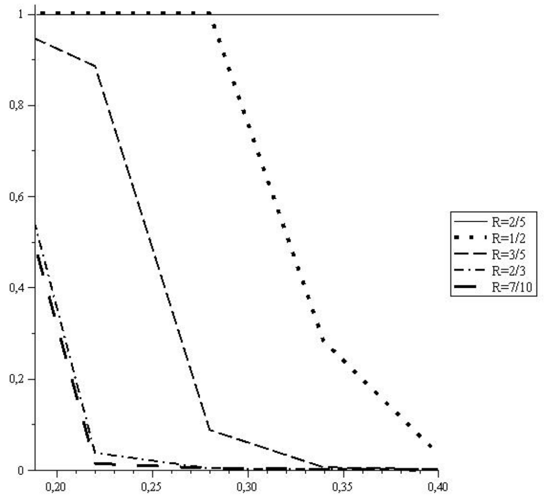


Fig.6. Recovering capability of reverse-MDP convolutional codes with different rates in terms of the erasure probability of the channel

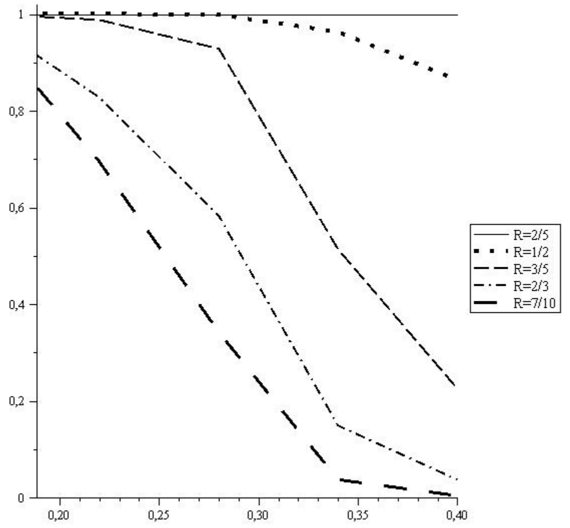


Fig. 5. Recovering capability (φ) of MDS block codes with differentrates in terms of the erasure probability of the channel

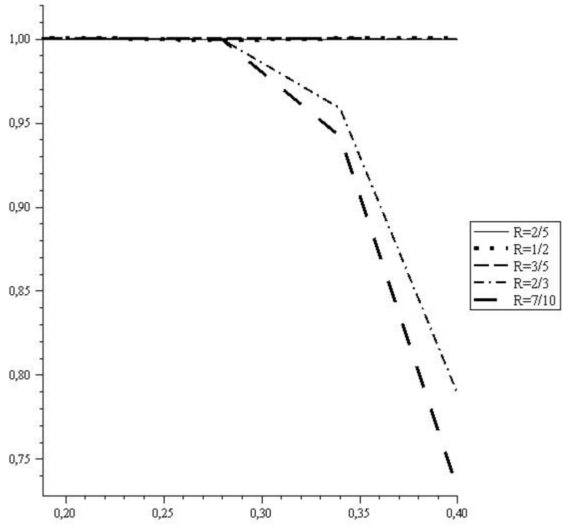


Fig.7Recovering capability of complete-MDP convolutional codes with different rates in terms of the erasure probability of the channel

Figure 5 reflects the behavior of MDS codes over the erasurechannel when choosing codes with different rates and overchannels with different erasure probabilities. The recoveringcapability is expressed in terms of φ = #erasures recovered /#erasures occurred .In Figures 6 and 7 we can see the performance of reveres-MDP and complete-MDP convolutional codes, respectively.The codes were chosen to have equal transmission rate andrecovering rate per window to those of the MDS block codesused in the simulations of Figure 5.The new simulation for reverse-MDP convolutional codes hows that reverse-MDP codes only outperform MDS codesat low rates. If we compare Figures 6 and 5, one can see thatonly for rates equal to R = 2/5 and R = 1/2 the results arebetter using reverse-MDP convolutional codes.

However, observing the results in Figure 7, one can seehow complete-MDP convolutional codes give much betterperformance than MDS block codes. Even though the rate decreasesfor convolutional codes when we increase the erasureprobability, the behavior is better than in the MDS case.

VI. COMPARISON BETWEEN MDS BLOCK CODES AND MDPCONVOLUTIONAL CODES

As we have already pointed out through several examplesMDP convolutional codes often are capable of decoding moreerasures than comparable MDS block codes. In this sectionwe would like to give some theoretical results on the decodingcapabilities of (complete) MDP convolutional codes andcompare these codes with MDS block codes of the same rate.

As a first goal we will show that a rate k/n convolutionalcode will not be able to decode erasures at a rate of more than(n-k)/=n. The following theorem serves this purpose.

VII. CONCLUSIONS

In this paper, we have discuss a comparison of LDPC block and convolutional codes based on several channels which shows some complexity, including computational complexity, hardware complexity, memory requirements, decoding delay, and BER/FER performance. It has been shown via computer simulations that LDPC convolutional codes have an error performance comparable to that of their block code counterparts. In addition, several interesting tradeoffs have been identified between the two different types of codes with respect to VLSI implementation.

For erasure channels, while close-to-optimal performance(in the sense of approaching capacity) was achievable for theBEC, we showed that the structure of LDPC-CC imposed constraintsthat bounded the performance over erasure channelswith memory strictly away from the optimal performance (inthe sense of approaching MDS performance). Nevertheless, thesimple structure and good performance of these codes, as wellas the latency flexibility and low complexity of the decodingalgorithm, are attractive characteristics for practical systems.

REFERENCES

[1] R. G. Gallager, .Low-density parity-check codes,.*IRE Trans. Inform.Theory*, vol. IT-8, pp. 21.28, Jan. 1962.

[2] N. Wiberg, *Codes and Decoding on General Graphs*. PhD thesis,Linkoping University, Sweden, 1996.

[3] D. J. C. MacKay and R. M. Neal, .Near Shannon limit performance oflow density parity check codes,.*Electronics Letters*, vol. 32, pp. 1645.1646, August 1996.

[4] A. Jim´enez-Feltstr¨om and K. Sh. Zigangirov, Time-varying periodicconvolutional codes with low-density parity-check matrix,.*IEEETrans.Inform. Theory*, vol. IT-45, pp. 2181.2191, Sept. 1999.

[5] S. Bates, Z. Chen, and X. Dong, .Low-density parity check convolutionalcodes for Ethernet networks,.in*Proc. IEEE Paci_cRimConference on Communications, Computers and Signal Processing*,
(Victoria, B.C., Canada), Aug. 2005.

[6] S. Bates, D. Elliot, and R. Swamy, .Termination sequence generationcircuits for low-density parity-check convolutional codes,.*submittedtoIEEE Trans. Circuits and Systems I*, March 2005.

[7] S. Bates, L. Guntorphe, A. E. Pusane, Z. Chen, K. Sh. Zigangirov, andD. J. Costello, Jr., .Decoders for low-density parity-check convolutionalcodes with large memory,.in*Proc. 12th NASA Symposium on VLSIDesign*, (Coeur d'Alene, Idaho, U.S.A.), Oct. 2005.

[8] S. Lin and D. J. Costello, Jr., *Error Control Coding*. Englewood Cliffs,NJ: Prentice-Hall, 2nd ed., 2004.

[9] C. J. Howland and A. J. Blanksby, .A 690-mW 1-Gb/s 1024-b, rate1/2 low density parity check decoder,.*IEEE Trans. Solid-State Circuits*,vol. 37, pp. 404.412, March 2002.

[10] A. Darabiha, A. C. Carusone, and F. R. Kschischang, .Multi-gbit/sec lowdensity parity check decoders with reduced interconnect complexity,.in
Proc. IEEE Intl. Symposium on Circuits and Systems, (Kobe, Japan),May 2005.

[11] R. Tanner, D. Sridhara, A. Sridharan, T. Fuja, and D. Costello Jr., .LDPCblock and convolutional codes based on circulantmatricies,. *IEEE Trans.Information Theory*, vol. 50, pp. 2966.2984, December 2004.

[12] Z. Wang and Q. Jia, .Low complexity, high speed decoder architecturefor quasi-cyclic LDPC codes,.in*Proc. IEEE Intl. Symposium onCircuits and Systems*, (Kobe, Japan), May 2005.

[13] S. Bates and G. Block, .A memory based architecture for low-densityparity-check convolutional decoders,.in*Proc. IEEE Intl. Symposium onCircuits and Systems*, (Kobe, Japan), May 2005.

[14] H. Zhong and T. Zhang, .Block-LDPC: a practical LDPC coding systemdesignapproach,.*IEEE Trans. Circuits and Systems I: Fundamental
Theory and Applications*, vol. 52, pp. 766.775, April 2005.

[15] K. Yoshida, J. B. Brockman, D. J. Costello, Jr., T. E. Fuja, and M. R.Tanner, .VLSI implementation of quasi-cyclic LDPC codes,. in*Proc.Intl. Symposium on Information Theory and its Applications*, (Parma,Italy), pp. 551.556, Oct. 2004.

[16] T. Zhang and K. Parhi, .A 54 MBPS (3,6)-regular FPGA LDPCdecoder,. in*Proc. IEEE Workshop on Signal Processing*, pp. 127.132,October 2002.

[17] The Institute of Electrical and Electronic Engineers, *IEEE Standard802.3-2002 : Carrier Sense Multiple Access with Collision DetectionCSMA/CD Access Methods and Physical Layer Speci_cation*. The IEEE,2002.

[18] R. Swamy and S. Bates, .A (128,3,6) low density parity check convolutionalcode encoder and decoder implemented in 180nm cmos,. *inpreparation*.

[19] A. E. Pusane, A. Jim´enez-Feltstr¨om, A. Sridharan, M. Lentmaier, K. Sh.Zigangirov, and D. J. Costello, Jr., .Implementation aspects of LDPCconvolutional codes,. *submitted to IEEE Trans. Commun.*, Nov. 2005.

[20] R. G. Gallager, Low Density Parity Check Codes. Cambridge, Massachusetts:MIT Press, 1963.

[21] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco, 1988.

[22] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation,Linkoping University, Linkoping, Sweden, 1996.

[23] D. MacKay and R. Neal, "Near Shannon limit performance of low density parity check codes," Electronics Letters, vol. 33, no. 6, pp. 457–458, Mar 1997.

[24] D. Divsalar, S. Dolinar, and C. Jones, Channel," JPL INP, Tech. Rep., Tech. Rep., Oct 2006

[25] E. Gilbert, "Capacity of a burst-noise channel," Bell Syst. Tech. J, vol. 39,pp. 1253–1265, Sep. 1960.

[26] E. Elliott, "Estimates of error rates for codes on burst-noise channels,"Bell Syst. Tech. J, vol. 42, pp. 1977–1997, Sep. 1963.

[27] M. Yang and W. E. Ryan, "Design of LDPC codes for two-state fading channel models," in The 5th International Symposium on Wireless Personal Multimedia Communications, vol. 3, Oct. 2002, pp. 986–990