Xu, R., Koçak, T., Woodward, G., & Morris, KA. (2010). Throughput improvement on bidirectional Fano algorithm. In *6th International Wireless Communications and Mobile Computing Conference, Caen, France* (pp. 276 - 280). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1145/1815396.1815460

Throughput Improvement on Bidirectional Fano Algorithm

Ran Xu
Centre for Communications Research
Department of Electrical and Electronic Engineering
University of Bristol
Bristol BS8 1UB, UK
Ran.Xu@bristol.ac.uk

Taskin Kocak
Centre for Communications Research
Department of Electrical and Electronic Engineering
University of Bristol
Bristol BS8 1UB, UK
T.Kocak@bristol.ac.uk

Graeme Woodward
Telecommunications Research Laboratory
Toshiba Research Europe Ltd
32 Queen Square
Bristol BS1 4ND, UK
Graeme.Woodward@toshiba-trel.com

Kevin Morris
Centre for Communications Research
Department of Electrical and Electronic Engineering
University of Bristol
Bristol BS8 1UB, UK
Kevin.Morris@bristol.ac.uk

ABSTRACT

Recently, we introduced a bidirectional Fano algorithm (BFA) [10] which can achieve much higher decoding throughput compared to the regular unidirectional Fano algorithm (UFA), especially at low signal-to-noise-ratio (SNR). However, the decoding throughput improvement of the conventional BFA with respect to the UFA reduces as the SNR increases and converges to 100% at high SNR. In this paper, two parameters in the BFA, which are known as the number of merged states (NMS) and the threshold increment value Δ, are exploited to improve the decoding throughout of the conventional BFA. The improved BFA can achieve much higher decoding throughput compared to the UFA and the conventional BFA, especially at high SNR. For example at E_b/N_0=5dB, the throughput improvement achieved by the improved BFA is about 280% compared to the UFA and about 80% compared to the conventional BFA, and its computational complexity is only 4% of the Viterbi algorithm.

Categories and Subject Descriptors

E.4 [**Coding and Information Theory**]: Error control codes

General Terms

Algorithms, Performance

IWCMC'10, June 28–July 2, 2010, Caen, France

Keywords

Bidirectional Fano algorithm, convolutional code, high throughput decoding, sequential decoding

1. INTRODUCTION

Sequential decoding is known as suboptimum decoding of convolutional codes [1]. Compared to the well-known Viterbi algorithm (VA) which can achieve maximum likelihood decoding performance, sequential decoding is a variable complexity algorithm and has much lower computational complexity than the VA at medium to high SNR. Since the computational complexity of sequential decoding is independent from the constraint length of the convolutional code, it can decode very long constraint length convolutional codes which are impossible for the VA to decode with the same computational effort. There are mainly two types of sequential decoding algorithms which are known as the Stack algorithm [2, 3] and the Fano algorithm [4]. Compared to the Stack algorithm, the Fano algorithm has low storage and sorting requirements, which makes it more suitable for hardware implementations [5, 6].

The availability of a very wide unlicensed bandwidth near 60 GHz makes multi-gigabit per second wireless communications an attractive research area nowadays [7]. The very high data rate (e.g. 1–6Gbps) makes the baseband signal processing more power and area hungry compared to that for low data rates [8]. For example, to achieve the required high throughput, the WirelessHD specification proposes simultaneous transmission of eight interleaved codewords, each encoded by a convolutional code [9]. It is straightforward to use eight parallel Viterbi decoders to achieve multi-Gbps decoding throughput. Alternative decoding techniques for convolutional codes are of interest in high throughput systems due to the high power consumption and large silicon area of Viterbi decoders.

Sequential decoding can achieve a similar error rate performance compared to the VA and its computational effort can be much lower than that of the VA. The low hardware

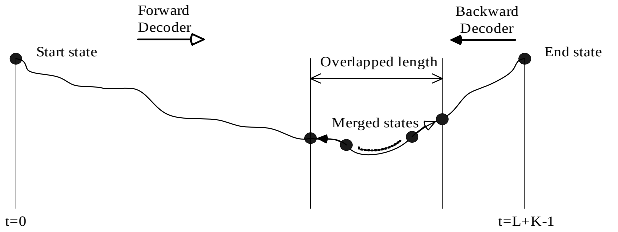

Figure 1: Illustration of bidirectional Fano algorithm decoding, where L is the information length and K is the constraint length of the convolutional code. The more rigorous merging check requires more than one merged state and the overlapped length should be not less than 1.[10]

complexity and low power consumption of sequential decoding make it very attractive for battery powered portable devices in high data rate applications. However, due to the irregular and unstructured operations and variable decoding delay, it is more difficult for sequential decoding algorithms to achieve high throughput compared to the VA. Some techniques in sequential decoding can be used to reduce the decoding delay and thus improve the throughput. For example, bidirectional searching can be used in sequential decoding to involve parallel processing in decoding one codeword by adding one more decoder compared to the UFA. In [10] a bidirectional Fano algorithm (BFA) was proposed for high throughput sequential decoding. It was shown that the proposed BFA could achieve much higher decoding throughput compared to the UFA, especially at low SNR. The throughput improvement comes from the parallel processing and also the reduction in computational effort. However, the throughput improvement of the BFA with respect to the UFA reduces as the SNR increases and it converges to 100% at high SNR.

In this paper, two parameters in the BFA, known as the number of merged states (NMS) and threshold increment value Δ are exploited to improve the decoding throughput of the conventional BFA. A look-up table of the optimal pairs of NMS and Δ can be constructed offline for a specific convolutional code based on the error rate and decoding throughput statistics at the SNR of interest. The BFA decoder can choose the optimal pair of NMS and Δ according to the estimated SNR in real-time decoding. The throughput improvement with respect to the UFA achieved by the improved BFA is higher than that achieved by the conventional BFA, especially at high SNR. The rest of the paper is organized as follows. In Section II, the BFA is briefly reviewed and the trade-off between error rate and decoding throughput is discussed. How to construct the look-up table which stores the optimal pairs of NMS and Δ is also presented in this section. The simulation results are given in Section III, and the conclusions are drawn in Section IV.

2. THROUGHPUT IMPROVEMENT

2.1 Trade-offs in the BFA

In the BFA, there is a forward decoder (FD) and a backward decoder (BD) which search towards each other simultaneously as illustrated in Fig. 1. Both the FD and the BD start decoding from the known state zero and finish decoding when the FD and the BD merge with each other. Merging means that the FD and the BD have the same state and the same position within the codeword. In order to increase the reliability of merging and thus improve the error rate performance of the BFA, more than one merged state is required for the merging condition. A trade-off between error rate performance and decoding throughput can be made by changing the number of merged states (NMS) in the BFA. A detailed description of the BFA and its performance can be found from [10]. Another parameter in the BFA which can be changed to make a trade-off between error rate performance and decoding throughput is the threshold increment value Δ. Generally, a “finer” search of the code tree can be achieved by reducing the Δ value at the cost of higher computational effort. A lower decoding delay can be achieved by increasing the Δ value, but the error rate performance degrades due to a “coarser” search of the code tree. A more detailed discussion about the choice of Δ can be found from [11]. The trade-off between error rate performance and decoding throughput by changing NMS and Δ was examined by simulation first. The bit error rate performance and throughput improvement (TI) for different values of NMS and Δ at E_b/N_0=4dB are shown in Fig. 2 (a) and (b), respectively. TI is defined as:

$$TI = \left(\frac{Delay_{UFA}}{Delay_{BFA}} - 1\right) \times 100\%, \qquad (1)$$

where $Delay_{UFA}$ and $Delay_{BFA}$ are the average decoding delays to decode one codeword by using the UFA with Δ=1 and by using the BFA with different NMS and Δ values, respectively. The decoding delay of the UFA is defined as:

$$Delay_{UFA} = \sum_{i=1}^{N} NoI_i/N. \qquad (2)$$

And the decoding delay of the BFA is defined as:

$$Delay_{BFA} = \sum_{i=1}^{N} \max\left(NoI_{i,FD}, NoI_{i,BD}\right)/N. \qquad (3)$$

The decoding delay of the UFA or the BFA is measured in terms of average Number of Iterations (NoI) to decode one codeword [10]. N is the total number of codewords. The simulation setup is listed in Table 1. The codewords whose computational effort exceeded the limitation value (C_{lim}) were ignored in the simulation.

Table 1: Simulation setup

Code rate (R)	1/3
Generator polynomials	$g_0 = \{133\}_8$ $g_1 = \{171\}_8$ $g_2 = \{165\}_8$
Constraint length (K)	7
Branch metric calculation	Fano metric(hard decision)
NMS	[1 2 4 8 16 32]
Δ	[1 2 4 6 8]
Modulation	BPSK
Channel	AWGN
Information length (L)	200 bits
Number of frames (N)	50,000
C_{lim}	20,000 NoI

It can be seen from Fig. 2 (a) that for a fixed NMS value, the BER performance can be improved by decreasing the Δ value, and for a fixed Δ value the BER performance can be improved by increasing the NMS value. However, the BER curve for each Δ value becomes flat as NMS increases to a large value. This effect is more pronounced for a larger Δ value. From Fig. 2 (b) it can be seen that for a fixed NMS value, the TI performance can be improved by increasing the Δ value. However, the TI curve for each NMS value becomes flat as Δ increases to a large value. The TI performance can also be improved by decreasing the NMS value, and the improvement increases more rapidly when NMS gets smaller. It has been shown in Fig. 2 (a) and (b) that the directions to better performance are opposite to each other, which means that a trade-off between BER and TI can be made by selecting a pair of NMS and Δ.

The values of BER and TI corresponding to Fig. 2 (a) and (b) are listed in Table 2 and Table 3, respectively. It is shown that if the target BER (BER_{target}) is set the same as the BER achieved by the BFA with NMS=1 and Δ=1, which is 2.63×10^{-3} at E_b/N_0=4dB, the achievable maximum TI is 191% which corresponds to {NMS=8, Δ=2}. This pair of NMS and Δ is considered to be optimal among the considered pairs of NMS and Δ listed in Table 2 and Table 3. If another target BER is chosen, which depends on the system design requirement, the corresponding optimal pair of NMS and Δ can be chosen in a similar manner. For example, if the target BER is set as 3×10^{-3}, the achievable maximum TI will be 275% which corresponds to {NMS=16, Δ=4}. In summary, an optimal pair of NMS and Δ can be chosen from a selected set of NMS and Δ at each SNR in the BFA to achieve the maximum decoding throughput without degrading the target error rate performance. How to construct a look-up table which stores the optimal pairs of NMS and Δ for different SNR values will be discussed next.

Table 2: BER ($\times 10^{-3}$) look-up table

4dB	Δ=1	Δ=2	Δ=4	Δ=6	Δ=8
NMS=1	2.63	2.97	3.66	4.83	6.97
NMS=2	2.69	2.81	3.48	4.67	6.68
NMS=4	2.57	2.67	3.41	4.42	6.34
NMS=8	2.48	2.57	3.10	4.15	5.94
NMS=16	2.13	2.27	3.01	3.90	5.55
NMS=32	1.89	2.25	2.83	3.89	5.62

Table 3: TI look-up table

4dB	Δ=1	Δ=2	Δ=4	Δ=6	Δ=8
NMS=1	132%	254%	363%	380%	350%
NMS=2	117%	235%	345%	362%	331%
NMS=4	102%	216%	323%	342%	315%
NMS=8	84.2%	191%	302%	319%	298%
NMS=16	74.0%	171%	275%	297%	277%
NMS=32	65.9%	158%	254%	274%	257%

2.2 Construction of the Look-up Table

The following steps can be followed to find the optimal pair of NMS and Δ among a selected set of NMS and Δ to achieve the maximum decoding throughput for a target BER at each SNR:

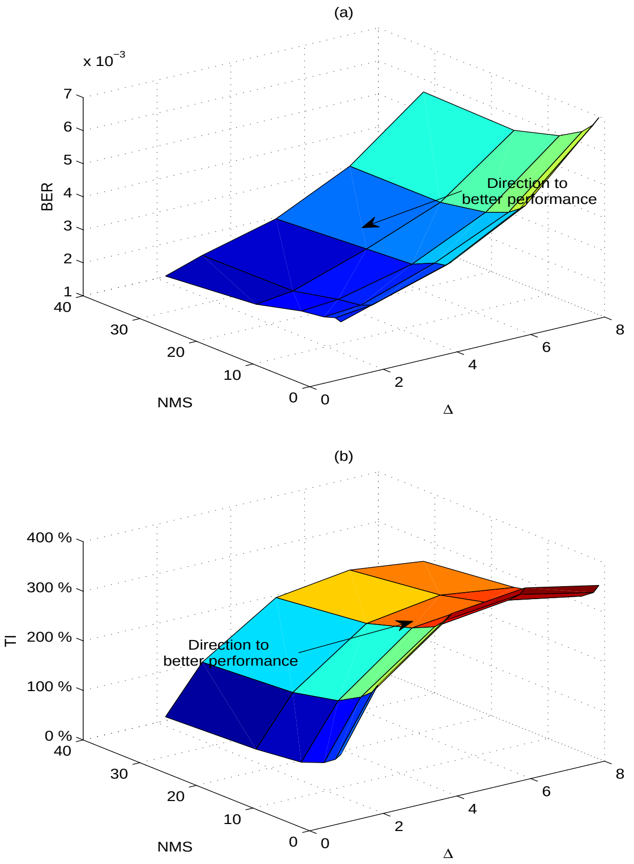

Figure 2: BER and TI for different NMS and Δ at E_b/N_0=4dB

Step 1: Determine two vectors selectively:

$$NMS = [NMS_1, \ldots, NMS_i, \ldots, NMS_I], \tag{4}$$

and

$$\Delta = [\Delta_1, \ldots, \Delta_j, \ldots, \Delta_J], \tag{5}$$

where $NMS_1 < \cdots NMS_i < \cdots NMS_I$ and $\Delta_1 < \cdots \Delta_j < \cdots \Delta_J$. In the paper NMS = [1, 2, 4, 8, 16, 32] and Δ = [1, 2, 4, 6, 8] were chosen, respectively. These two vectors are considered to be fine-grained enough for the SNR of interest;

Step 2: Get empirical statistics of $\boldsymbol{BER}$ and $\boldsymbol{TI}$ by simulations:

$$\boldsymbol{BER} = \begin{bmatrix} BER_{1,1} & BER_{1,2} & \cdots & BER_{1,J} \\ BER_{2,1} & BER_{2,2} & \cdots & BER_{2,J} \\ \vdots & \vdots & \ddots & \vdots \\ BER_{I,1} & BER_{I,2} & \cdots & BER_{I,J} \end{bmatrix}, \tag{6}$$

and

$$\boldsymbol{TI} = \begin{bmatrix} TI_{1,1} & TI_{1,2} & \cdots & TI_{1,J} \\ TI_{2,1} & TI_{2,2} & \cdots & TI_{2,J} \\ \vdots & \vdots & \ddots & \vdots \\ TI_{I,1} & TI_{I,2} & \cdots & TI_{I,J} \end{bmatrix}, \tag{7}$$

where $BER_{i,j}$ and $TI_{i,j}$ are the bit-error-rate of the BFA and the throughput improvement of the BFA with respect to

Table 4: Optimal pairs of NMS and Δ in the BFA

E_b/N_0	BER_{target}	NMS	Δ
3dB	1.63×10^{-2}	16	2
4dB	2.63×10^{-3}	8	2
5dB	2.64×10^{-4}	8	4
6dB	1.78×10^{-5}	4	4

the UFA corresponding to NMS_i and Δ_j, respectively. The simulation setup should comply with the conditions in which the BFA decoder will operate. In the paper the simulation setup is the same as that listed in Table 1;

Step 3: Find the indexes of the BER matrix whose values are below or equal to the target BER:

$$[\boldsymbol{I'}, \boldsymbol{J'}] = \mathit{find}\left(\boldsymbol{BER} \leq BER_{target}\right), \tag{8}$$

where BER_{target} is decided by the system design requirement. In the paper BER_{target} was set the same as the BER of the BFA with NMS=1 and Δ=1;

Step 4: Find the indexes of the TI matrix whose value is the maximum among those corresponding to the selected indexes $\boldsymbol{I'}$ and $\boldsymbol{J'}$:

$$[i', j'] = \mathit{find}\left[\max\left(\boldsymbol{TI}(\boldsymbol{I'}, \boldsymbol{J'})\right)\right]. \tag{9}$$

Step 5: $NMS_{i'}$ and $\Delta_{j'}$ are selected as the optimal pair of NMS and Δ among the selected set of NMS and Δ for the considered SNR and target BER:

$$[NMS_{i'}, \Delta_{j'}] = \Psi\left(SNR, BER_{target}\right). \tag{10}$$

The above steps were followed to get the optimal pairs of NMS and Δ at different SNR values which are listed in Table 4. It can be seen that as SNR increases the optimal NMS decreases while the optimal Δ increases. This look-up table is constructed offline which will be referred to by the BFA decoder in real-time decoding. During BFA decoding, different pairs of NMS and Δ can be selected according to the SNR estimation. Some SNR estimation techniques for the AWGN channel can be employed [12]. More entries can be produced for the look-up table to cover more fine-grained SNR values, which is at the cost of larger memory of the BFA decoder. Alternatively, the optimal pair of NMS and Δ corresponding to the SNR estimation which is not included in the look-up table can be interpolated by the NMS and Δ values of the nearby SNR values.

3. SIMULATION RESULTS

The BFA with the optimal pairs of NMS and Δ (named as improved BFA) is compared with the UFA, the conventional BFA and the VA in terms of error rate performance, throughput improvement and computational complexity.The simulation setup is the same as that listed in Table 1. From Fig. 3 it can be seen that the improved BFA and the conventional BFA with NMS=1 and Δ=1 have the same BER performance. This is because the target BER of the improved BFA was set the same as the BER of the conventional BFA with NMS=1 and Δ=1. There is about 0.2 dB coding gain loss at BER=10^{-4} compared to the UFA with Δ=1 which has almost the same BER performance as the VA.

The throughput improvement achieved by the improved BFA and the conventional BFA compared to the UFA is shown in Fig. 4. It can be seen that with NMS=1 and Δ=1,

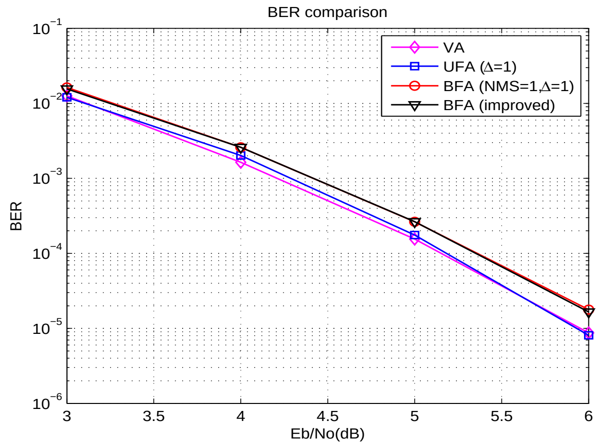

Figure 3: BER comparison between the VA, the UFA, the conventional BFA and the improved BFA

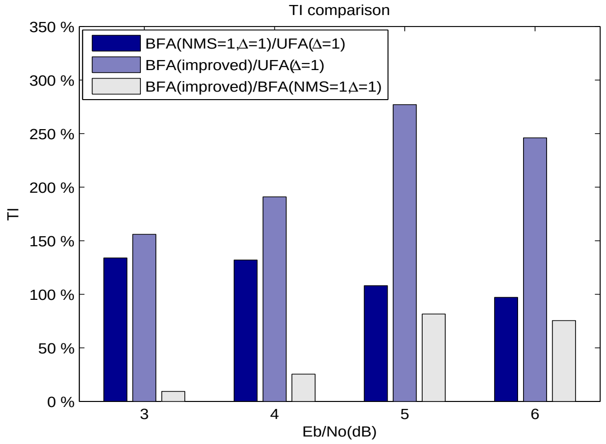

Figure 4: Throughput improvement by using the conventional BFA with respect to the UFA, using the improved BFA with respect to the UFA, and using the improved BFA with respect to the conventional BFA

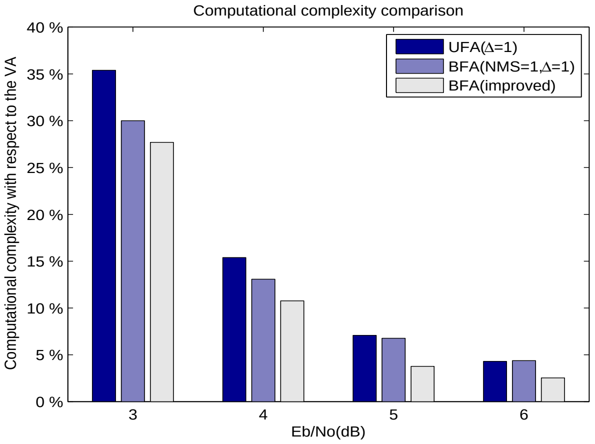

Figure 5: Computational complexity comparison between the UFA, the conventional BFA and the improved BFA with respect to the VA

the throughput improvement of the conventional BFA with respect to the UFA reduces as the SNR increases, and it converges to about 100% at high SNR. However, the improved BFA can achieve a much higher throughput improvement, especially at high SNR. For example at E_b/N_0=5dB, the improved BFA has about 280% throughput improvement compared to the UFA. The throughput improvement achieved by the improved BFA compared to the conventional BFA is also presented, which increases from about 10% at E_b/N_0=3dB to about 80% at E_b/N_0=6dB. The benefit of using the optimal pairs of NMS and Δ is more pronounced at high SNR.

The computational complexity comparison between the UFA, the conventional BFA and the improved BFA with respect to the VA is shown in Fig. 5. The computational complexity is measured in terms of average NoI to decode one codeword. In the BFA, the computational complexity is the sum of the NoI performed by the FD and that by the BD, and the average computational complexity can be calculated by:

$$C_{BFA} = \sum_{i=1}^{N} \left(NoI_{i,FD} + NoI_{i,BD} \right) / N. \tag{11}$$

The VA has a fixed computational complexity:

$$C_{VA} = (L + K - 1) \times 2^{K-1}. \tag{12}$$

where L is the information length and K is the constraint length. It can be seen from Fig. 5 that the computational complexity of the sequential decoding algorithms is much lower than that of the VA, especially at high SNR. For example above E_b/N_0=5dB, the computational complexity of the examined sequential decoding algorithms is less than 10% of the VA. This is because the sequential decoding algorithms only require local search of the code tree, unlike the VA which requires exhaustive search of the trellis diagram. By using bidirectional search, the BFA has lower computational complexity compared to the UFA. The improved BFA has the lowest computational complexity among the examined algorithms due to the usage of the optimal pair of NMS and Δ at each SNR. For example at E_b/N_0=5dB, the computational complexity of the improved BFA is only about 4% of the VA.

4. CONCLUSIONS

The decoding throughput improvement of the conventional BFA with respect to the UFA reduces as SNR increases and converges to 100% at high SNR. In this paper, the decoding throughput of the BFA was improved by exploiting two parameters in the BFA which are known as the number of merged states (NMS) and threshold increment value Δ. A pair of NMS and Δ can be chosen at each SNR to achieve the maximum decoding throughput without losing the error rate performance compared to a target BER. A look-up table can be constructed offline which stores the optimal pairs of NMS and Δ at different SNR values. The BFA decoder can choose the optimal pair of NMS and Δ from the look-up table according to the estimated SNR during real-time decoding. It was shown that by using the optimal pairs of NMS and Δ at different SNR in the BFA, a higher throughput improvement with respect to the UFA can be achieved compared to the conventional BFA, and the improvement is more pronounced at high SNR which is much higher than 100%. The computational complexity of the improved BFA is lower than the conventional BFA and much lower than the VA, especially at high SNR. The proposed BFA decoding with the NMS and Δ looking-up strategy can be employed in high throughput wireless communication systems to decode convolutional codes with low hardware complexity and low power consumption.

5. ACKNOWLEDGMENTS

The authors would like to thank the Telecommunications Research Laboratory (TRL) of Toshiba Research Europe Ltd. and its directors for supporting this work.

6. REFERENCES

[1] S. Lin and D. J. Costello, Jr. *Error Control Coding: Fundamentals and Applications.* 2nd ed. Upper Saddle River, NJ: Pearson Prentice-Hall, 2004.

[2] K. Sh. Zigangirov. Some sequential decoding procedures. *Probl. Pered. Inform.*, vol. 2, no. 4, pp. 13-15, 1966.

[3] F. Jelinek. Fast sequential decoding using a stack. *IBM J. Res. Devel.*, vol. 13, pp. 675-685, Nov. 1969.

[4] R. M. Fano. A heuristic discussion of probabilistic decoding. *IEEE Transactions on Information Theory*, vol. IT-9, no. 2, pp. 64-74, Apr. 1963.

[5] R. O. Ozdag and P. A. Beerel. An asynchronous low-power high-performance sequential decoder implemented with QDI templates. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 9, pp. 975-985, Sep. 2006.

[6] M. Benaissa and Y. Zhu. Reconfigurable hardware architectures for sequential and hybrid decoding. *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 3, pp. 555-565, Mar. 2007.

[7] H. Singh, J. Oh, C. Kweon, X. Qin, H.-R. Shao and C. Ngo. A 60 GHz wireless network for enabling uncompressed video communication. *IEEE Communications Magazine*, vol. 46, no. 12, pp. 71-78, 2008.

[8] M. Piz, M. Krstic, M. Ehrig and E. Grass. An OFDM baseband receiver for short-range communication at 60 GHz. *IEEE International Symposium on Circuits and Systems*, pp. 409-412, 2009.

[9] Wireless High-Definition (WirelessHD); http://www.wirelesshd.org/.

[10] R. Xu, T. Kocak, G. Woodward, K. Morris and C. Dolwin. Bidirectional Fano Algorithm for High Throughput Sequential Decoding. *IEEE Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Tokyo, Japan, 2009.

[11] W. D. Pan and A. Ortega. Adaptive computation control of variable complexity Fano decoders. *IEEE Transactions on Communications*, vol. 57, pp. 1556-1559, June 2009.

[12] D. R. Pauluzzi and N. C. Beaulieu. A comparison of SNR estimation techniques for the AWGN channel. *IEEE Transactions on Communications*, vol. 48, pp. 1681-1691, Oct. 2000.