Research Article

Parameter Estimation of Turbo Code Encoder

Mehdi Teimouri¹ and Ahmadrza Hedayat²

¹ Faculty of New Sciences and Technologies, University of Tehran, Tehran 14399, Iran

² Newracom, Irvine, CA 92618, USA

Correspondence should be addressed to Mehdi Teimouri; mehditeimouri@ut.ac.ir

Received 29 May 2014; Revised 15 August 2014; Accepted 27 August 2014; Published 18 September 2014

Academic Editor: Nikos D. Lagaros

Copyright © 2014 M. Teimouri and A. Hedayat. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The problem of reconstruction of a channel code consists of finding out its design parameters solely based on its output. This paper investigates the problem of reconstruction of parallel turbo codes. Reconstruction of a turbo code has been addressed in the literature assuming that some of the parameters of the turbo encoder, such as the number of input and output bits of the constituent encoders and puncturing pattern, are known. However in practical noncooperative situations, these parameters are unknown and should be estimated before applying reconstruction process. Considering such practical situations, this paper proposes a novel method to estimate the above-mentioned code parameters. The proposed algorithm increases the efficiency of the reconstruction process significantly by judiciously reducing the size of search space based on an analysis of the observed channel code output. Moreover, simulation results show that the proposed algorithm is highly robust against channel errors when it is fed with noisy observations.

1. Introduction

Channel code reconstruction problem is usually considered in noncooperative communication scenarios, where finding the structure and parameters of the channel code is a major goal of the receiver [1–14]. Since turbo codes are introduced in many standards and applications [15–22], blind reconstruction of turbo codes in practical noncooperative situations is a well motivated problem. This problem has been investigated in [1–4, 8, 11], where some of the code parameters are assumed to be known to the receiver, such as the number of input and output bits of each constituent convolutional code and the structure of puncturing matrix. Having the above parameters, the receiver becomes capable of separating the received coded stream into three sets [3]: systematic bits, parity bits of the first convolutional encoder, and parity bits of the second convolutional encoder. Now the turbo code reconstruction problem converts into three simpler reconstruction problems: two convolutional code reconstruction problems and one interleaver reconstruction problem.

However, in most practical noncooperative situations, the parameters of turbo code are unknown and thus should be estimated before applying reconstruction process or to

test each possible set of parameters which is highly time consuming. To the best of the authors' knowledge, no algorithm has yet been proposed for estimating these parameters, and this paper is the first to propose a novel method to estimate above-mentioned parameters before applying the turbo code reconstruction process. The proposed method reduces the search space of the reconstruction process significantly compared to [3] and makes the reconstruction process faster and more efficient. The proposed method is first formulated for noiseless streams and then it is adjusted for noisy observations.

The rest of the paper is organised as follows. In Section 2, system model is described. The method for estimating the turbo code parameters is proposed in Section 3 and experimental results are presented in Section 4. Finally, the paper is concluded in Section 5.

2. System Model

We consider the system model of turbo code reconstruction as shown in Figure 1. In this model, input bits and their interleaved version are fed to the first and second convolutional encoders, respectively. The first encoder is

FIGURE 1: System model of turbo code reconstruction.

an (n_1, k_1, m_1) convolutional code, where n_1 is the number of output bits, k_1 is the number of input bits, and m_1 is the memory length of the longest shift register of the encoder. Since convolutional codes used for parallel turbo codes are systematic [15], k_1 bits out of n_1 output bits are systematic bits, and the remaining $n_1 - k_1$ bits are parity bits. With such configuration, the value of any $n_1 - k_1$ parity bits is a function of recent k_1 systematic output bits and at most $k_1 \times m_1$ values of shift registers content. In the system model of Figure 1, the second convolutional code is also assumed to be an (n_2, k_2, m_2) convolutional code, but unlike the first convolutional encoder, its systematic bits are removed and only the remaining $n_2 - k_2$ parity bits are used.

Since parallel turbo codes have the same number of input bits per constituent convolutional code [15], we assume k_1 and k_2 are equal and denote both by k . Assume the length of input frames is N ; then N is a multiple of k . Using this assumption, at each time step, k bits out of N input bits and k bits out of the interleaver are fed to the first and second convolutional encoders, respectively. These input bits result in $n_1 + n_2 - k$ output bits; thus we can say that N/k sets of input bits form an output frame with length $L = N/k \times (n_1 + n_2 - k)$.

As shown in Figure 1, a puncturing pattern can also be applied to the output frames to form smaller frame lengths and higher code rates. We use an $(n_1 + n_2 - k) \times p$ matrix $\mathbf{P} = [q_{i,j}]$ to represent puncturing pattern where p is the puncturing period. Every row of $\mathbf{P}$ corresponds to one output bit; that is, the first row corresponds to the first systematic bit, the second row corresponds to the second systematic bit (or the first parity bit of the first convolutional encoder), and so on. Note that $q_{i,j} \in \{0, 1\}$, where 0 implies that the corresponding bit is punctured. If n_0 elements out of $(n_1 + n_2 - k) \times p$ of $\mathbf{P}$ take the value of 1, then the output frame length is $L = (N \times n_0)/(k \times p)$. Note that we have implicitly assumed that N is a multiple of $k \times p$, since the input frame needs to be a multiple of the puncturing period. Since the turbo codes employed in many applications such as CDMA 2000 [16] and WCDMA [17] are systematic parallel turbo codes, in this paper we consider systematic parallel turbo codes. The term “systematic” is used for a turbo code when none of its systematic bits are punctured. In systematic turbo codes all the elements of the first k rows of $\mathbf{P}$ take the value of 1. Other elements of $\mathbf{P}$ can be either 0 or 1; however, it should be noted that none of the rows can be an all zeros vector.

All the previous works that address the turbo code reconstruction problem implicitly assume that the received coded stream is partitioned into three sets: systematic bits, parity bits of the first convolutional encoder, and parity bits of the second convolutional encoder. However, it should be

noted that in order to do such partitioning we need to know the number of output bits of both encoders (i.e., n_1 and n_2) and the puncturing pattern (i.e., $\mathbf{P}$). But, in most practical noncooperative situations, these parameters are unknown and thus should be estimated before applying reconstruction process or to test each possible set of parameters which is highly time consuming. In this paper, a novel method is proposed to estimate the above-mentioned parameters before applying the turbo code reconstruction process. The proposed method reduces the search space of the reconstruction process significantly and makes the reconstruction process faster and more efficient.

3. Proposed Method

Assume that the number of elements with the value of 1 in puncturing matrix $\mathbf{P}_{(n_1+n_2-k) \times p}$ is equal to n_0 . Using the same arguments as those detailed in [5], if the input data to the turbo encoder behaves as random data, we can say that the number of linearly independent blocks of received coded stream with length $n_0 \times i$, assuming i to be a positive integer, is at most equal to $(p \times k + p \times k) \times i + m_0$, where nonnegative integer m_0 takes at most the value of $m_1 + m_2$. This can be easily seen by noting that each block, out of i blocks with length n_0 , is generated by linear combination of $p \times k + p \times k$ information bits and $m_1 + m_2$ initial memory bits. When the Hamming weight in the last $n_2 - k$ rows of $\mathbf{P}$ is smaller than $p \times k$, the value of the second $p \times k$ should be interchanged with this Hamming weight. Let us denote the minimum value between $p \times k$ and the Hamming weight in the last $n_2 - k$ rows of $\mathbf{P}$ by p_2 . Thus, we can say that the number of linearly independent blocks of received coded stream with length $n_0 \times i$ is at most equal to $(p \times k + p_2) \times i + m_0$. When $p_2 < p \times k$ we can say that p_2 parity bits of the second convolutional encoder in each puncturing period act like random independent data, so it can be concluded that in this case m_0 can take at most the value of m_1 . Since only the values of i that results in $((p \times k + p_2) \times i + m_0)/(n_0 \times i) < 1$ are acceptable, we have

$$i \geq \left\lfloor \frac{m_0}{n_0 - p_0} \right\rfloor + 1, \quad (1)$$

where $p_0 = p \times k + p_2$ and $\lfloor x \rfloor$ denotes the largest integer less than or equal to x .

Starting from the first bit we partition the received bit stream into blocks with length n and consider these blocks as the rows of a matrix named $\mathbf{Z}_n$ with size $a \times n$. Choosing the value of a is arbitrary, but it should be selected in order

to be much bigger than n . Now, if we define function $R(n)$ for the received bit stream in the domain of positive integer as follows:

$$R(n) = \frac{\text{Rank}(\mathbf{Z}_n)}{n}, \quad (2)$$

where $\text{Rank}(\mathbf{Z}_n)$ denotes the rank of matrix $\mathbf{Z}_n$, according to the aforementioned discussion in the beginning of this section, it is obvious that this function should have local minimum points at $n = n_0 \times i$, where i is given by (1). Moreover, the function values at these local minimum points are given as follows:

$$R(i \times n_0) = \frac{p_0 \times i + m_0}{n_0 \times i}; \quad i \geq \left\lfloor \frac{m_0}{n_0 - p_0} \right\rfloor + 1. \quad (3)$$

By obtaining local minimum points and function values at these points for function $R(n)$ it is possible to estimate the values of m_0 , p_0 , and n_0 parameters of turbo code, which can facilitate the turbo code reconstruction process.

If in the construction process of $\mathbf{Z}_n$ (whilst $n = n_0 \times i$) we eliminate the first α bits of bit stream ($\alpha \in \{0, 1, 2, \dots, n_0 - 1\}$), the obtained matrix denoted by $\mathbf{Z}_{n,\alpha}$ can help us to estimate the puncturing matrix. In fact, if the value of α is chosen such that the first considered bit in construction of $\mathbf{Z}_{n,\alpha}$ is a systematic bit in the first row of $\mathbf{P}$, it is obvious that $\text{Rank}(\mathbf{Z}_{n,\alpha}) = \text{Rank}(\mathbf{Z}_n)$. On the other hand, if the value of α is chosen such that the first considered bit in construction of $\mathbf{Z}_{n,\alpha}$ is a systematic bit in other rows of $\mathbf{P}$ or a parity bit of the first convolutional code, we expect that $\text{Rank}(\mathbf{Z}_{n,\alpha}) > \text{Rank}(\mathbf{Z}_n)$. As a matter of fact, in such situations degree of freedom is increased which results in larger rank values. Finally, if the value of α is chosen such that the first considered bit in construction of $\mathbf{Z}_{n,\alpha}$ is a parity bit of the second convolutional code, we expect that in (at most) p positions the equality $\text{Rank}(\mathbf{Z}_{n,\alpha}) = \text{Rank}(\mathbf{Z}_n)$ holds, whereas, in the remaining positions, if any, the inequality $\text{Rank}(\mathbf{Z}_{n,\alpha}) > \text{Rank}(\mathbf{Z}_n)$ holds.

Thus, we can define puncturing signature as a vector with length n_0 in the following way. For any $\alpha \in \{0, 1, 2, \dots, n_0 - 1\}$, if $\text{Rank}(\mathbf{Z}_{n_0 \times i, \alpha}) > \text{Rank}(\mathbf{Z}_{n_0 \times i})$ the corresponding element in puncturing signature is set to 1; otherwise it is set to 0. Note that, by knowing the values of k , n_1 , n_2 , and p , construction of puncturing matrix $\mathbf{P}$ from puncturing signature is not unique, but definitely, the search space size for puncturing matrix is intensely reduced. Numerical results for implying this fact will be provided in the following section.

According to the above analysis, we are ready to present the proposed method for estimation of the number of input and output bits of both encoders (i.e., k , n_1 , and n_2) and the puncturing pattern (i.e., $\mathbf{P}$).

- (1) Plot $R(n)$ versus n for $n = n_{\min}, n_{\min} + 1, \dots, n_{\max}$ ($n_{\min}$ and $n_{\max}$ are two arbitrary integers). Denote the set of the points at which the local minimums occur by $\{\hat{n}_j\}$ and the set of the function local minimum values by $\{R_j\}$, where $j = 1, 2, 3, \dots, j_{\max}$. Define the set of $\{\hat{k}_j\}$ such that $\hat{k}_j = R_j \times \hat{n}_j$ for any j .

- (2) Estimate the value of n_0 as follows: $n_0 = \text{mode}(\{\hat{n}_{j+1} - \hat{n}_j; j = 1, 2, \dots, j_{\max} - 1\})$, where $\text{mode}(X)$ denotes the most frequent value in set X .
- (3) Estimate the value of p_0 as follows: $p_0 = \text{mode}(\{\hat{k}_{j+1} - \hat{k}_j; j = 1, 2, \dots, j_{\max} - 1\})$.
- (4) Estimate the value of m_0 as follows: $m_0 = \text{mode}(\{\hat{k}_j - (\hat{n}_j/n_0) \times p_0; j = 1, 2, \dots, j_{\max}\})$.
- (5) Plot function $R(\hat{n}_{j_{\max}}, \alpha) = \text{Rank}(\mathbf{Z}_{\hat{n}_{j_{\max}}, \alpha})/\hat{n}_{j_{\max}}$ versus α for $\alpha = 0, 1, \dots, n_0 - 1$, and then obtain the puncturing signature as stated before.
- (6) Using the estimated values of n_0 , p_0 , and puncturing signature we can consider feasible values of k , n_1 , and n_2 to construct all possible puncturing matrices. Then, since the construction process of the turbo code can be called with any sets of these possible parameters, somewhere we expect that one of these calls successfully results in a turbo code structure. Estimated value of m_0 can as well help to make the size of the search space for constituent convolutional codes smaller.

Note that, compared to the current methods which should blindly pass any possible set of parameters to the turbo code construction process, the proposed method here tries to reduce the size of such large search space. Numerical results for comparison reasons are provided in the following section.

In formulation of the proposed method for estimation of turbo code parameters it is assumed that the observed turbo coded stream is noiseless. In such situation, the rows of matrix $\mathbf{Z}_n$ form a linear space denoted by Γ . Similarly, the rows of matrix $\mathbf{Z}_{n,\alpha}$ form a linear space denoted by Γ_α . So, if the received data is noisy it is obvious that we can use the proposed algorithm on noiseless rows of $\mathbf{Z}_n$ and $\mathbf{Z}_{n,\alpha}$, and yet the same turbo code parameters as the noiseless case will be obtained. However, we should choose the number of the rows of $\mathbf{Z}_n$ and $\mathbf{Z}_{n,\alpha}$ large enough in order that the number of the noiseless rows of these matrices becomes much bigger than n .

In order to find the noiseless rows of $\mathbf{Z}_n$, first we apply Valembois' algorithm [7] on (noisy) rows of $\mathbf{Z}_n$ to find some words in the dual space of Γ . Now, any rows of $\mathbf{Z}_n$ that is orthogonal to all obtained words in the dual space of Γ can be considered as a noiseless row. A similar method can be used to find noiseless rows of $\mathbf{Z}_{n,\alpha}$.

Numerical and simulation results provided in the next section show that this algorithm is quite efficient in space search size reduction, and moreover it is also robust when it is fed with noisy and erroneous input.

4. Experimental Results

For obtaining experimental results we consider three different turbo code structures. The characteristics of these codes are given in Table 1. The first code is the code applied in WCDMA standard and the two others are introduced in CDMA 2000 standard. In this table, $\mathbf{G}_1(D)$ and $\mathbf{G}_2(D)$ denote

TABLE 1: The structure of the simulated turbo code.

	Constituent convolutional codes $G_1(D) = G_2(D) = G(D)$	Puncturing matrix	Overall code rate
Turbo code A	$G(D) = \left[1, \frac{1+D+D^3}{1+D^2+D^3} \right]$	$\mathbf{P} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	1/3
Turbo code B	$G(D) = \left[1, \frac{1+D+D^3}{1+D^2+D^3}, \frac{1+D+D^2+D^3}{1+D^2+D^3} \right]$	$\mathbf{P} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	1/5
Turbo code C	$G(D) = \left[1, \frac{1+D+D^3}{1+D^2+D^3}, \frac{1+D+D^2+D^3}{1+D^2+D^3} \right]$	$\mathbf{P} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$	1/4

FIGURE 2: Plot of $R(n)$ for turbo code B. FIGURE 3: Plot of $R(n)$ for turbo code C.

polynomial representation of generator matrix of the first and the second convolutional codes, respectively.

For comparison reasons, we set the first dimension of matrix $\mathbf{Z}_n$ to $3 \times n$ and set the value of $n_{\min}$ and $n_{\max}$ to 21 and 60, respectively. In Figure 2, function $R(n)$ for turbo code B is plotted. For this code the set of the points at which the local minimums occur is $\{\hat{n}_j\} = \{25, 30, 35, \dots, 55\}$. According to this set and as we expected, the distance between the consecutive points at which the local minimums occur is equal to $n_0 = 5$. For some turbo codes, more than one set of local minimums can be observed. For example, function $R(n)$ for turbo code C is shown in Figure 3. As it is apparent from this figure, two distinguishable sets of local minimum points are $\{\hat{n}_j\} = \{24, 32, 40, 48, 56\}$ and $\{\hat{n}_j\} = \{28, 36, 44, 52\}$. In such cases, the set with smallest local minimum values should be used to estimate turbo code parameters. The other local minimum set(s) in this case are formed due to the special structure of the constituent convolutional encoders and the puncturing pattern. For example, for code C, when the values

of n are selected from the set $\{8 \times 3 + 4, 8 \times 4 + 4, 8 \times 5 + 4, \dots\}$, linearly independent blocks of received coded stream would be less than n ; however, the relative number of these linearly independent blocks, that is, $R(n)$, is higher compared to the situation in which the values of n are selected from the right set which is $\{8 \times 3, 8 \times 4, 8 \times 5, \dots\}$.

In Table 2, estimated parameters of turbo codes defined in Table 1 for the case of noiseless coded stream are given. As we can see, the given values for parameters n_0 , p_0 , and m_0 match the expected value. The last column of Table 2 represents the number of the potential sets of $\{k, n_1, n_2, \mathbf{P}\}$ which can be considered according to the estimated values for n_0 and p_0 . For turbo codes A and B, the only potential sets of $\{k, n_1, n_2, \mathbf{P}\}$ are those given in Table 1. But for turbo code C the number of the potential sets of $\{k, n_1, n_2, \mathbf{P}\}$ is equal to 56. Note that this value is much smaller compared to the number of the potential sets in blind reconstruction. For example, if the reconstruction process operates blindly in the case in which the values of k , n_1 , n_2 , and p are assumed to be at most

TABLE 2: Estimated parameters of turbo codes defined in Table 1 for noiseless case.

	n_0	p_0	m_0	Puncturing signature	Number of potential sets of $\{k, n_1, n_2, \mathbf{P}\}$
Turbo code A	3	2	3	[0, 1, 0]	1
Turbo code B	5	2	6	[0, 1, 1, 0, 1]	1
Turbo code C	8	4	5	[0, 1, 1, 0, 0, 1, 0, 1]	56

TABLE 3: Probability of success in parameters estimation of turbo codes defined in Table 1 for noisy channel.

	E_b/N_0 (dB)			
	5.2	4.2	3.7	3.2
Turbo code A	≈ 1	0.91	0.42	0.05
Turbo code B	≈ 1	0.52	0.05	≈ 0
Turbo code C	≈ 1	0.65	0.07	≈ 0

equal to 2, 3, 3, and 5, respectively, all possible puncturing matrices having these properties should be considered and the only constraint is that the number of 1s in the examined puncturing matrices should be a factor of the output frames length which is assumed to be equal to 1200 in this case. It can be seen that in this case the search space size is equal to 461794 which is much larger than 56.

To evaluate the capability of the proposed method against channel noise, we consider that the turbo coded stream is first modulated by a binary phase shift keying (BPSK) modulator and then passed through an additive white Gaussian noise (AWGN) channel. Also, we define E_b/N_0 as the ratio of energy per transmitted bit (E_b) to the spectral noise density (N_0). In Table 3, success probability for turbo code parameter estimation is given under the assumption of different E_b/N_0 values, considering the turbo codes defined in Table 1. The number of tests for each turbo code and SNR pair is set to 1000. As it is apparent from this table, when the value of E_b/N_0 is more than 5 dB, the probability of success is near 1, but for E_b/N_0 values around 3 dB the probability of success reaches 0.

Since turbo codes are supposed to perform in noisy channels, the low noise assumption for code reconstruction may seem too optimistic. However, it should be noted that in practical noncooperative situations, recognition of a communication system is usually done in higher values of signal to noise ratio compared to cooperative situations. In fact, in order to achieve such signal quality, we have no choice other than using ultra low noise receivers. Finally, it should be noted that the channel code reconstruction process is done once based on many observations of the channel code outputs and decision on which of the estimated code parameters to be picked as the true code parameters can be easily made based on the multiplicity and inferred confidence of the estimated sets.

5. Conclusion

A novel method is proposed in this paper to estimate the parameters of turbo codes. These parameters are the number

of input and output bits of both constituent convolutional encoders and the puncturing pattern while assuming that the target turbo code is systematic. It is assumed that receiver is only aware of the turbo coded bit stream. Numerical results confirm that the proposed method significantly reduces the size of the search space for reconstruction process of the turbo code. Moreover, simulation results indicate that when the proposed algorithm is fed with erroneous channel observations, the outcome is highly reliable as long as the E_b/N_0 value is more than 5 dB, demonstrating robustness of the proposed algorithm.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments.

References

- [1] J. Barbier, "Reconstruction of turbo-code encoders," in *Security and Defense, Space Communication Technologies Symposium*, Proceedings of SPIE, pp. 463–473, Orlando, Fla, USA, 2005.
- [2] R. Gautier, M. Marazin, and G. Burel, "Blind recovery of the second convolutional encoder of turbo-code when its systematic outputs are punctured," *MTA Review*, vol. 19, pp. 213–232, 2009.
- [3] M. Cluzeau, M. Finiasz, and J.-P. Tillich, "Methods for the reconstruction of parallel turbo codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '10)*, pp. 2008–2012, June 2010.
- [4] M. Côte and N. Sendrier, "Reconstruction of a turbo-code interleaver from noisy observation," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '10)*, pp. 2003–2007, June 2010.
- [5] J. Barbier, G. Sicot, and S. Houcke, "Algebraic approach for the reconstruction of linear and convolutional error correcting codes," *International Journal of Applied Mathematics and Computer Sciences*, vol. 2, pp. 113–118, 2006.
- [6] M. Côte and N. Sendrier, "Reconstruction of convolutional codes from noisy observation," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '09)*, pp. 546–550, July 2009.
- [7] A. Valembois, "Detection and recognition of a binary linear code," *Discrete Applied Mathematics*, vol. 111, no. 1-2, pp. 199–218, 2001.

- [8] Y. G. Debessu, H.-C. Wu, and H. Jiang, "Novel blind encoder parameter estimation for turbo codes," *IEEE Communications Letters*, vol. 16, no. 12, pp. 1917–1920, 2012.
- [9] M. Marazin, R. Gautier, and G. Burel, "Blind recovery of k/n rate convolutional encoders in a noisy environment," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, article 168, 2011.
- [10] H. Lee, C.-S. Park, J.-H. Lee, and Y. J. Song, "Reconstruction of BCH codes using probability compensation," in *Proceedings of the 18th Asia-Pacific Conference on Communications (APCC '12)*, pp. 591–594, Jeju Island, Republic of Korea, October 2012.
- [11] A. Naseri, O. Azmoon, and S. Fazeli, "Blind recognition algorithm of turbo codes for communication intelligence systems," *International Journal of Computer Science Issues*, vol. 8, pp. 68–72, 2011.
- [12] H. Xie, X.-M. Chai, F.-H. Wang, and Z.-T. Huang, "A method for blind identification of rate 1/2 convolutional code based on improved Euclidean algorithm," in *Proceedings of the 11th International Conference on Signal Processing (ICSP '12)*, pp. 1307–1310, Beijing, China, October 2012.
- [13] A. Zahedi and G.-R. Mohammad-Khani, "Reconstruction of a non-binary block code from an intercepted sequence with application to reed-solomon codes," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer*, vol. 95, pp. 1873–1880, 2012.
- [14] M. Marazin, R. Gautier, and G. Burel, "Algebraic method for blind recovery of punctured convolutional encoders from an erroneous bitstream," *IET Signal Processing*, vol. 6, no. 2, pp. 122–131, 2012.
- [15] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Transactions on Communications*, vol. 44, no. 5, pp. 591–600, 1996.
- [16] D. J. Shah, V. K. Patel, and H. A. Patel, "Performance analysis of turbo code for CDMA 2000 with convolutional coded IS-95 system in wireless communication system," in *Proceedings of the International Conference on Electronic Computer Technology (ICECT '10)*, pp. 42–45, Kuala Lumpur, Malaysia, May 2010.
- [17] K. Ramasamy, B. Balamuralithara, and M. U. Siddiqi, "Performance evaluation of proposed asymmetric turbo code in WCDMA HSDPA system," in *Proceedings of the 5th IEEE GCC Conference & Exhibition (GCC '09)*, pp. 1–5, March 2009.
- [18] O. Bayat, "Intersymbol interference cancellation in CDMA 1xEVDO network," *International Journal of Communication Systems*, 2012.
- [19] S. Y. Le Goff, "Bandwidth-efficient turbo coding over Rayleigh fading channels," *International Journal of Communication Systems*, vol. 15, no. 7, pp. 621–633, 2002.
- [20] O. N. Ucan, K. Buyukatak, E. Gose, O. Osman, and N. Odabasioglu, "Performance of multilevel-turbo codes with blind/non-blind equalization over WSSUS multipath channels," *International Journal of Communication Systems*, vol. 19, no. 3, pp. 281–297, 2006.
- [21] A. Ghayeb and T. Abualrub, "Asymptotic performance comparison of concatenated (turbo) codes over $GF(4)$," *International Journal of Communication Systems*, vol. 17, no. 5, pp. 479–490, 2004.
- [22] O. Osman, "Blind equalization of multilevel turbo coded-continuous phase frequency shift keying over MIMO channels," *International Journal of Communication Systems*, vol. 20, no. 1, pp. 103–119, 2007.