Decoding the ‘Nature Encoded’ Messages for Distributed Energy Generation Control in Microgrid

Shuping Gong, Husheng Li, Lifeng Lai and Robert. C. Qiu

Abstract—The communication for the control of distributed energy generation (DEG) in microgrid is discussed. Due to the requirement of realtime transmission, weak or no explicit channel coding is used for the message of system state. To protect the reliability of the uncoded or weakly encoded messages, the system dynamics are considered as a ‘nature encoding’ similar to convolution code, due to its redundancy in time. For systems with or without explicit channel coding, two decoding procedures based on Kalman filtering and Pearl’s Belief Propagation, in a similar manner to Turbo processing in traditional data communication systems, are proposed. Numerical simulations have demonstrated the validity of the schemes, using a linear model of electric generator dynamic system.

I. INTRODUCTION

In recent years, smart grid has attracted significant studies due to its capability of improving the efficiency and robustness of power grid [2]. An important component in smart grid is the microgrid [1][6], which is a network containing multiple distributed energy generators (DEGs, like solar panels, micro-turbines or wind turbines) and multiple loads. A microgrid can either be connected to or be separated from the power grid. In normal situations, the microgrid is connected to the power grid and obtains energy according to its own energy generation. When the power grid experiences an emergency, e.g., a large area blackout, the microgrid can be disconnected from the power grid and work in an island mode. At this time, the power load within the microgrid will be supported by the DEGs. Such a microgrid has been implemented in many testbeds, e.g., the Electric Reliability Technology Solutions (CERTS) Microgrid supported by the US Department of Energy.

An important aspect of the microgrid is the communication infrastructure for controlling the DEGs since they could be located at different places. In such a communication infrastructure, the observations at sensors are sent to a processor (either centralized or decentralized) for system state monitoring or control. In traditional microgrids, wired communications are employed. For example, an RS-485 (digital computer based) link or an ethernet is used in the Oak Ridge National Lab (ORNL) microgrid system [3]. An alternative approach for the communication for DEG control is to use wireless communication systems due to its fast deployment. In wireless medium, fading, noise and interference may cause damage to the received signal of messages for DEG control. Therefore, a channel coding scheme is needed to protect the messages. However, there is a stringent requirement on the delay of messages due to the dynamics of DEGs. Hence, it is impossible to accumulate sufficiently many bits for a long codeword length and the corresponding coding gain since this will incur a substantial delay. Therefore, an effective approach is to use short codeword length which loses the protection on the bits in the message.

In this paper, we study the decoding procedure (including the demodulation procedure) for the messages with no or weak channel coding protection in the context of DEG control in microgrid. The key point is to exploit the redundancy in the system state, which can be considered as a ‘nature encoding’ for the messages. In this paper, a linear system model is adopted, where \( x(t+1) = A x(t) + B u(t) + n(t) \) is used to describe the dynamics of system state \( x \) subject to the control \( u(t) \) and noise \( n(t) \). We observe that the system state is similar to the convolutional codes except that the ‘encoding’ of the system state is in the real field instead of the Galois field. Hence, the message actually has been channel coded by the nature although no explicit or little explicit channel coding is used at the transmitter. We use both schemes of Kalman filtering and Pearl’s Belief Propagation (BP) for the soft decoding, combined with the soft demodulation to improve the reliability of demodulation and decoding. A practical dynamic system for the DEGs will be used for numerical simulation, which demonstrates the performance gain of incorporating the inherent redundancy in the ‘nature encoding’. The procedure is illustrated in Fig. 1. Note that such a scheme is similar to the Turbo processing techniques like Turbo decoding [8], Turbo multiuser detection [11] and Turbo equalization [12] in wireless communication systems. However, the ‘nature encoding’ is analog and implicit, thus resulting in a different processing procedure.

The remainder of this paper is organized as follows.
system model is introduced in Section II. The decoding procedure is discussed for the schemes of Kalman filtering based heuristic approach and the BF approach in Sections III and IV respectively. Numerical simulation results are shown in Section V. Finally, the conclusions are drawn in Section VI.

II. SYSTEM MODEL

A. Linear System

We consider a discrete time linear dynamic system, whose system state evolution is given by

\[ \begin{align*}
    x(t+1) &= Ax(t) + Bu(t) + n(t), \\
    y(t) &= Cx(t) + w(t)
\end{align*} \]

where \( x(t) \) is the \( N \)-dimensional vector of system state at time slot \( t \), \( u(t) \) is the \( M \)-dimensional control vector, \( y(t) \) is the \( K \)-dimensional observation vector and \( n(t) \) and \( w(t) \) are noise vectors, which are assumed to be Gaussian distributed with zero expectation and covariance matrices \( \Sigma_n \) and \( \Sigma_w \) respectively. For simplicity, in this paper we do not consider \( u(t) \).

We assume that the observation vector \( y(t) \) is obtained by a sensor. The sensor quantizes each dimension of the observation using \( B \) bits, thus forming a bit sequence which is given by

\[ b(t) = (b_1(t), b_2(t), ..., b_{KB}(t)). \]

B. Communication System

Suppose that binary phase shift keying (BPSK) is used for the transmission from the sensor to the controller. The bit sequence is passed through an optional channel encoder, which generates an \( L \)-bit sequence \( s(t) \). Then, the received signal at the controller is given by

\[ r(t) = s(t) + e(t), \]

where the additive white Gaussian noise \( e(t) \) has a zero expectation and variance \( \sigma_e^2 \). Note that we ignore the fading and normalize the transmit power to be 1. The algorithm and conclusion in this paper can be easily extended to the case with different types of fading.

III. KALMAN FILTERING BASED HEURISTIC APPROACH

In this section, we adopt a heuristic approach, which is based on Kalman filtering, to exploit the redundancy in the system state. We first carry out the Kalman filtering and then apply the prediction to the soft demodulation.

A. Kalman Filtering

When the observations \( y(t) \) are sent to the controller perfectly, the controller can use the following Kalman filtering to predict the future system state, whose expectation is given by

\[ x(t+1|t) = Ax(t|t), \]

where

\[ x(t|t) = x(t|t-1) + K(t) (y - Cx(t|t-1)), \]

and

\[ K(t) = \Sigma(t|t-1)C^T (C\Sigma(t|t-1)(C^T + \Sigma_o)^{-1}, \]

and the covariance matrix given by

\[ \Sigma(t|t) = \Sigma(t|t-1) - K(t)\Sigma(t|t-1), \]

where

\[ \Sigma(t+1|t) = \Sigma(t|t) + \lambda \Sigma(t|t-1) + \Sigma_p. \]

B. Soft Decoding and Demodulation

Based on the Kalman filtering, the controller can obtain the distribution of the observation, which is Gaussian distributed with the expectation \( x(t+1|t-1) \) given by Eq. (4) and the covariance \( \Sigma(t|t-1) \) given by Eq. (5).

Because different dimensions in the observation \( y_i \) are not independent, it is challenging to directly compute the \( a \ priori \) probability for each bit, which is given by (suppose that \( j \) is used to describe the \( i \)-th bit in \( b(t) \))

\[ P(b_i = 1 | y(\infty : t)) = \frac{1}{\sqrt{2\pi|\Sigma(t|t-1)|}} \int (b_i = 1, y(t) = y) \times \exp\left(-\frac{1}{2}(y - x(t|t-1))^T \Sigma^{-1}(t|t-1) (y - x(t|t-1))dy. \]  

We propose to use the Monte Carlo simulation to obtain a series of samples of \( \{y_i\} \) based on the prediction of Kalman filtering, then quantize these samples to obtain a series of samples \( b_i \) and calculate the prior probability \( \xi_i(t) \) for \( b_i \). We use \( \{\xi_i(t)\} \) as the \( a \ priori \) probability of being 1 for demodulating \( b(t) \). Then, the \( a \ posteriori \) probability of \( b_i(t) \) is given by

\[ P(b_i(t) = 1 | r(\infty : t)) = \frac{P(r(t)|b_i(t) = 1) \xi_i(t)}{P(r(t)|b_i(t) = 1) \xi_i(t) + P(r(t)|b_i(t) = 0)(1 - \xi_i(t))} \]

where

\[ P(y(t)|b_i(t) = 1) = \frac{1}{\sqrt{2\pi\sigma_e^2}} \exp\left(-\frac{(r_i(t) - 1)^2}{2\sigma_e^2}\right), \]

and

\[ P(y(t)|b_i(t) = 0) = \frac{1}{\sqrt{2\pi\sigma_e^2}} \exp\left(-\frac{(r_i(t) + 1)^2}{2\sigma_e^2}\right). \]

Note that the Kalman filtering is no longer rigorous in the networked control system due to the quantization error and possible decoding error. The proposed heuristic approach is based on the assumption that the Kalman filtering is very precise. As will be shown in the numerical results, this approach will be seriously affected by the propagation of decoding errors.
passing from \( U_m \) to \( X \) is \( \pi_{U_m,X}(U_m) \), which is the prior information of \( U_m \) conditioned on all the information \( U_m \) has received. Red arrows transmit \( \lambda \)-message which is from children to its parent. For instance, the message passing from \( Y_n \) to \( X \) is \( \lambda_{Y_n,X}(X) \), which is the likelihood of \( X \) based on the information \( Y_n \) has received. After \( X \) receives all \( \pi \)-message \( \pi_{U_m,X}(U_m) \) from its parents \( U_1, U_2, \ldots, U_M \) and all \( \lambda \)-message \( \lambda_{Y_n,X}(X) \), \( X \) from its children \( Y_1, Y_2, \ldots, Y_N \), \( X \) updates its belief information \( BEL_X(x) \) and transmits \( \lambda \)-messages \( \lambda_{X,U_m}(U_m) \) to its parents and \( \pi \)-message \( \pi_{X,Y_n}(x) \) to its children. The expressions of the quantities are given by

\[
\pi_X(x) = \sum_U p(x|U) \prod_{m=1}^M \pi_{U_m,X}(U_m) \tag{13}
\]

\[
\gamma_X(U) = \sum_x \prod_{n=1}^N \lambda_{Y_n,X}(x)p(x|U) \tag{14}
\]

\[
BEL_X(x) = \alpha \times \prod_{n=1}^N \lambda_{Y_n,X}(x) \times \pi_X(x) \tag{15}
\]

\[
\lambda_{X,U_m}(U_m) = \sum_U \gamma_X(U) \times \prod_{j \neq m} \pi_{U_j,X}(U_j) \tag{16}
\]

\[
\pi_{X,Y_n}(x) = \pi_X(x) \times \prod_{i \neq n} \lambda_{Y_i,X}(x) \tag{17}
\]

where \( U = (U_1, U_2, \ldots, U_M) \) and \( Y = (Y_1, Y_2, \ldots, Y_N) \).

In the initialization procedure of Pearl’s BP, some initial values are needed for the running of Pearl’s BP. The initial values are defined as

\[
\lambda_{X,U}(u) = \begin{cases} 
p(x_0|u), & X \text{ is evidence, } x = x_0 \\
1, & \text{X is not evidence} \end{cases} \tag{18}
\]

and

\[
\pi_{X,Y}(x) = \begin{cases} 
\delta(x,x_0), & X \text{ is evidence, } x = x_0 \\
p(x), & \text{X is source, not evidence} \end{cases} \tag{19}
\]

### B. Application of Pearl’s BP in Microgrid

The Bayesian network structure of the control system and communication system in the DEG networked control in microgrid is shown for three time slots in Fig. 4. In the Bayesian network, the system state is dependent on the previous system state and the control action; the observation is dependent on the system state; the uncoded bits are dependent on the observation vector; the received signal is dependent on the uncoded bits. Here we omit the coded bits as the relationship between the uncoded bits and the coded bits is deterministic. Fig. 4 shows the Bayesian network structure for the dynamic system with three observations: \( x_{t-2}, x_{t-1} \) and \( x_t \).

Based on the Bayesian network structure, the iterative decoding procedure can be derived. Fig 4 shows the message passing in the dynamic system. \( x_{t-2} \) summarizes all the information obtained from previous time slots and transmits \( \pi \)-message \( \pi_{x_{t-2},x_{t-1}}(x_{t-2}) \) to \( x_{t-1} \). The BP procedure can be implemented in synchronous or asynchronous manners. As the decoding process has a large overhead, we implement asynchronous Pearl’s BP. The updating order and message passing in one iteration is as follows: step 1): \( x_{t-1} \rightarrow y_{t-1} \); step 2): \( y_{t-1} \rightarrow b_{t-1} \); step 3): \( b_{t-1} \rightarrow y_{t-1} \); step 4): \( y_{t-1} \rightarrow \)
We run the simulations using Matlab to compare the performances of Kalman filtering and Pearl’s BP based algorithms for systems with and without channel coding. The baseline approach is the separated Kalman filtering and decoding process. In the following, these three algorithms are referred as ‘KF with Prior’, ‘Pearl BP’ and ‘KF’, respectively. The performance metrics are the mean square error (MSE) of each sample and the average bit error rate (BER). Each simulation runs 1000 times slots. The configuration for both systems are as follows: for the dynamic system of DEG described in (1): 
\[ \delta t = 0.01, \Sigma_u = 0.05, \Sigma_0 = 0.05. \]
Each dimension of the observation \( y_t \) is quantized with 16 bits, and the dynamic range for quantization is \([-200, 200]\). A 1/2 rate Recursive Systematic Convolutional (RSC) code is used as the channel coding scheme; and the code generator is \( g = [1, 1, 1] \). The decoding algorithm is Log-Map algorithm.

### B. Uncoded Case

Figures [5] and [6] show the simulation results for the uncoded case with different \( \frac{E_b}{N_0} \). When \( \frac{E_b}{N_0} \geq 5 \), the performance of ‘Pearl BP’ is much better than ‘KF’ and ‘KF with Prior’; when \( \frac{E_b}{N_0} \geq 7 \), the performance of ‘KF with Prior’ is better than ‘KF’. This demonstrates that the redundancy in the system state can improve the performance when \( \frac{E_b}{N_0} \) is high. Compared with ‘KF with Prior’, the performance gain of ‘Pearl BP’ stems from the utilization of soft decoding output in the system state estimation. However, in the low \( \frac{E_b}{N_0} \) regime, the performances of ‘Pearl BP’ and ‘KF with Prior’ are worse than ‘KF’. This is because the cascading effect of the decoding error and system state estimation error. The worst case is that some highest bits of observations are not decoded correctly, then the estimated observation largely deviates from the correct value; and then this error will propagate to succeeding time slots and thus will lead to the collapse of ‘KF with Prior’ or ‘Pearl BP’ algorithm.
\[ A' = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
-20.316 & 0 & 0 & -25.048 & -1.411 & 0 & 0 \\
-0.061 & 0 & -0.773 & -0.083 & 0.018 & 15.06 & 30.12 \\
-0.213 & 0 & 7.050 & -5.026 & 0.063 & 0 & 0 \\
-2.654 & 0 & 0 & -1.463 & -12.958 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
-0.008 & 0 & 0 & -0.565 & 0.971 & -3.33 & -33.33
\end{pmatrix} \]

(20)

Fig. 7: Mean Square Error comparison for system with channel coding

Fig. 8: Bit Error Rate comparison for system with channel coding

C. Coded Case

Figures 7 and 8 show the simulation results for the channel coded case with different \( E_b/N_0 \). As shown in the figures, the performance of ‘Pearl BP’ is always the best except that the bit error rate is worse than ‘KF with Prior’ occasionally. In a contrast to the ‘Pearl BP’ for the uncoded case, ‘Pearl BP’ does not collapse for coded case due to the gain of channel coding. When \( E_b/N_0 \geq 3 \), the performance of ‘KF with Prior’ is better than that of ‘KF’. These results demonstrate that using the redundancy in system state can improve performance in the coded case. The performance gain of ‘Pearl BP’ over ‘KF with Prior’ in low \( E_b/N_0 \) shows the gain of using soft decoding output in state estimation. When \( E_b/N_0 \geq 5 \), these three algorithms have good performance which is due to the gain of channel coding.

VI. Conclusions

In this paper, we have proposed to use Kalman Filter based and Pearl’s BP based decoding procedure (including the demodulation procedure) to exploit the redundancy, i.e., the nature encoding in the system state for the system with no or weak channel coding protection in the context of DEG control in microgrid. The numerical simulation results have shown that, when \( E_b/N_0 \) is high, the proposed algorithms achieve significant performance gain, compared with traditional separated decoding and system state estimation. However, when \( E_b/N_0 \) is low, the proposed algorithms may incur performance degradation due to the error propagation. This motivates us to study the mechanism to prevent such an adversarial effect.

Appendix A

Derivation of Key Steps in the Pearl’s BP

In the following derivation part, we will use the following two basic equations, where \( N(x, \mu, \sigma) \) is the probability density function of Gaussian distribution with expectation \( \mu \) and variance \( \sigma \) at \( x \):

\[
\int_{-\infty}^{\infty} N(x, m_1, \Sigma_1)N(y, Cx, \Sigma_2)dx \propto N(y, Cm_1, C\Sigma_1C^T + \Sigma_2),
\]

and

\[
N(x, m_1, \Sigma_1)N(x, m_2, \Sigma_2) \propto N(x, m_3, \Sigma_3),
\]

where the variance is given by

\[
\Sigma_3 = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1},
\]

and the expectation is given by

\[
m_3 = \Sigma_3(\Sigma_1^{-1}m_1 + \Sigma_2^{-1}m_2).
\]
Below is the notation used throughout the derivation:

\[ \pi_{x_{t-1}, x_t}(x_{t-1}) = N(x_{t-1}, x_{\pi_x, t-1}, \Sigma_{\pi_x, t-1}) \]
\[ \pi_{x_{t-1}, y_{t-1}}(x_{t-1}) = N(x_{t-1}, x_{\pi_y, t-1}, \Sigma_{\pi_y, t-1}) \]
\[ \pi_{y_t, y_t}(y_t) = N(y_t, y_{\pi_y, t}, \Sigma_{\pi_y, t}) \]
\[ \pi_{x_t}(x_t) = N(x_t, x_{\pi_x, t}, \Sigma_{\pi_x, t}) \]
\[ \lambda_{y_t, y_t}(y_t) = N(y_t, y_{\lambda_y, t}, \Sigma_{\lambda_y, t}) \]
\[ \lambda_{x_t, x_t}(x_t) = N(x_t, x_{\lambda_x, t}, \Sigma_{\lambda_x, t}) \]
\[ \lambda_{b_t, y_t}(y_t) = N(y_t, y_{\lambda_b, t}, \Sigma_{\lambda_b, t}) \]
\[ \text{BEL}(x_t) = N(x_t, x_{\text{BEL}, t}, \Sigma_{\text{BEL}, t}) \]
\[ p(x_t | x_{t-1}) = N(x_t, A x_{\pi_x, t-1} + B u_{t-1}, \Sigma_p). \]

1) Step 1: \( x_{t-1} \rightarrow y_{t-1} \): we have

\[ \pi_{x_{t-1}, y_{t-1}}(x_{t-1}) = \int_{-\infty}^{\infty} p(x_{t-1} | x_{t-2}) \pi_{x_t, x_{t-1}}(x_{t-1} | x_t) dx_{t-2} \]
\[ = N(x_{t-1}, x_{\pi_x, t-1}, P_{\pi_x, t-1}), \quad (25) \]

where the expectation is given by

\[ x_{t-1} = A x_{\pi_x, t-2}, \quad (26) \]

and the variance is given by

\[ P_{\pi_x, t-1} = A \times P_{\pi_x, t-2} \times A^T + \Sigma_p \]
\[ \pi_{x_{t-1}, y_{t-1}}(x_{t-1}) = \pi_{x_{t-1}}(x_{t-1}) \lambda_{y_{t-1}, x_{t-1}}(x_{t-1}) \]
\[ = N(x_{t-1}, x_{\pi_x, t-1}, P_{\pi_x, t-1}) \times 1 \]
\[ = N(x_{t-1}, x_{\lambda_x, t-1}, P_{\lambda_x, t-1}) \quad (27) \]
\[ \text{where} \]
\[ x_{\pi_y, t-1} = x_{t-1}; P_{\pi_y, t-1} = P_{t-1}. \quad (28) \]

2) Step 4: \( y_{t-1} \rightarrow x_{t-1} \): we have

\[ \lambda_{y_{t-1}, x_{t-1}}(x_{t-1}) = \gamma_{y_{t-1}}(x_{t-1}) \]
\[ = \int_{-\infty}^{\infty} \lambda_{b_{t-1}, y_{t-1}}(y_{t-1}) p(y_{t-1} | y_{t-1}) dy_{t-1} \]
\[ = N(x_{t-1}, x_{\lambda_b, t-1}, P_{\lambda_y, t-1}), \quad (29) \]

where the expectation is given by

\[ x_{\lambda_b, t-1} = C^{-1} \times y_{\lambda_b, t-1}, \quad (30) \]

and the variance is given by

\[ P_{\lambda_y, t-1} = C^{-1} (S_{\lambda_y, t-1} + \Sigma_b) \times (C^{-1})^T. \quad (31) \]

3) Step 5: \( x_{t-1} \rightarrow x_t \): we have

\[ \pi_{x_{t-1}, x_t}(x_{t-1}) = \pi_{x_{t-1}}(x_{t-1}) \times \lambda_{y_{t-1}, x_{t-1}}(x_{t-1}) \]
\[ = N(x_{t-1}, x_{\pi_x, t-1}, P_{\pi_x, t-1}), \quad (32) \]

where the variance is given by

\[ P_{\pi_x, t-1} = (P_{\pi_x, t-1}^{-1} + P_{\lambda_x, t-1}^{-1})^{-1}, \quad (33) \]

and the expectation is given by

\[ x_{\pi_x, t-1} = P_{\pi_x, t-1}^{-1} \times \frac{N(x_{t-1}, x_{\pi_x, t-1}(x_{t-1}))}{P_{\lambda_x, t-1}} \]
\[ + P_{\lambda_x, t-1}^{-1} x_{\lambda_x, t-1}. \quad (34) \]

The belief is thus given by

\[ \text{BEL}_{x_{t-1}} = \alpha \times \lambda(x_{t-1}, x_{t-1})(x_{t-1}) \times \pi_{x_{t-1}}(x_{t-1}) \]
\[ = N(x_{t-1}, x_{\text{BEL}, t-1}, P_{\text{BEL}, t-1}), \quad (34) \]

where the variance is given by

\[ P_{\text{BEL}, t-1} = (P_{\lambda_x, t-1}^{-1} + P_{t-1}^{-1})^{-1}, \quad (35) \]

and the expectation is given by

\[ x_{\text{BEL}, t-1} = P_{\text{BEL}, t-1} \times \frac{N(x_{t-1}, x_{\pi_x, t-1}(x_{t-1}))}{P_{\lambda_x, t-1}^{-1} x_{\lambda_x, t-1}} \]
\[ + P_{t-1}^{-1} \times x_{t-1}. \quad (36) \]

4) Step 10: \( x_t \rightarrow x_{t-1} \), we have

\[ \lambda_{x_{t-1}, x_t}(x_{t-1}) = \gamma_{x_{t-1}}(x_{t-1}) \]
\[ = \int_{-\infty}^{\infty} \lambda_{b_{t-1}, y_{t-1}}(x_{t-1}) p(x_{t-1} | y_{t-1}) dy_{t-1} \]
\[ = N(x_{t-1}, x_{\lambda_b, t-1}, P_{\lambda_x, t-1}), \quad (37) \]

where the variance is given by

\[ P_{\lambda_x, t-1} = A^{-1} (S_{\lambda_x, t-1} + C) (A^{-1})^T, \quad (38) \]

REFERENCES

[1] N. Hatzigiou, H. Sano, R. Iravani and C. Marnay, “Microgrids: An overview of ongoing research development and demonstration projects,” IEEE Power and Energy Magazine, vol.27, pp.78-94, July/Aug., 2007.
[2] ISO New England Inc., Overview of the Smart Grid: Policies, Initiatives and Needs, Feb. 17, 2009.
[3] J. D. Kueck, R. H. Staunton, S. D. Labinov and B. J. Kirby, “Microgrid energy management system,” Technical Report ORNL/TM-2002/242, Oak Ridge National Lab., 2002.
[4] J. Machowski, J. W. Bialek and J. R. Bumby, Power System Dynamics: Stability and Control, Wiley, 2008.
[5] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Network of Plausible Systems, Morgan Kaufmann Publishers, 1988.
[6] P. Piagi and R. H. Lasseter, “Antonomous control of microgrids,” in Proc. of IEEE Power and Energy Society (PES) Meeting, 2006.
[7] T. Richardson and R. Urbanke, “The capacity of low-density parity check codes under message passing decoding,” IEEE Trans. Inform. Theory, vol.47, pp.599-618, 2001.
[8] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo codes,” in Proc. Int. Conf. Communications, 1993.
[9] J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference”, San Mateo, CA: Morgan Kaufmann, 1988.
[10] R. McEliece, D. MacKay, and J. Cheng, "Turbo Decoding as an Instance of Pearls ‘Belief Propagation’ Algorithm”, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 16, NO. 2, pp.140-152, 1998.
[11] P. D. Alexander, M. c. Reed and c. B. Schlegel, "Iterative multiuser interference reduction: Turbo CDMA,” IEEE Trans. Commun., Vol. 47, No. 7, pp. 1008 - 1014, July 1999.
[12] Michael Tüchler, Ralf Koetter, and Andrew C. Singer, "Turbo Equalization: Principles and New Results”, IEEE Trans. Commun., Vol. 50, No. 5, pp. 754-767, MAY 2002.