

Zoha, A., Saeed, A., Imran, A., Imran, M. A., and Abu-Dayya, A. (2014) A Wen, L., Razavi, R., Xiao, P., and [Imran, M. A.](#) (2014) Fast Convergence and Reduced Complexity Receiver Design for LDS-OFDM System. In: 2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC), Washington, DC, USA, 02-05 Sep 2014, pp. 918-922. ISBN 9781479949120 (doi:[10.1109/PIMRC.2014.7136297](https://doi.org/10.1109/PIMRC.2014.7136297))

This is the author's final accepted version.

There may be differences between this version and the published version. You are advised to consult the publisher's version if you wish to cite from it.

<http://eprints.gla.ac.uk/136678/>

Deposited on: 28 April 2017

Fast convergence and reduced complexity receiver design for LDS-OFDM system

Lei Wen, Razieh Razavi, Pei Xiao and Muhammad Ali Imran
Centre for Communication Systems Research
University of Surrey
Guildford GU2 7XH, Surrey, United Kingdom
Email: {l.wen, r.razavi, p.xiao, m.imran}@surrey.ac.uk

Abstract—Low density signature for OFDM (LDS-OFDM) is able to achieve satisfactory performance in overloaded conditions, but the existing LDS-OFDM has the drawback of slow convergence rate for multiuser detection (MUD) and high receiver complexity. To tackle these problems, we propose a serial schedule for the iterative MUD. By doing so, the convergence rate of MUD is accelerated and the detection iterations can be decreased. Furthermore, in order to exploit the similar sparse structure of LDS-OFDM and LDPC code, we utilize LDPC codes for LDS-OFDM system. Simulations show that compared with existing LDS-OFDM, the LDPC code improves the system performance.

Keywords—LDS-OFDM; iterative multiuser detection; serial schedule; forward error correction

I. INTRODUCTION

Multi-carrier code division multiple access (MC-CDMA) is a multiple access method used in OFDM-based systems, allowing the system to support multiple users at the same time [1]. It has been considered as a suitable approach to coping with increasing data rate in wireless communications [2]. However, when the number of users or parallel data symbols exceeds that of available chips, which is referred to as the overloaded condition, the performance of MC-CDMA degrades dramatically. In that condition, multipath fading becomes a severe problem and the orthogonality of spreading sequences is destroyed, thus the system performance is limited by serious multiuser interference (MUI) and inter-symbol interference (ISI). Classic multiuser detection (MUD) fails to eliminate the MUI in overloaded conditions [3][4]. In order to deal with the problem, low density signature (LDS) for MC-CDMA, namely LDS-OFDM, has been proposed [5][6][7].

In LDS-OFDM, due to the low density signature, each data symbol is only spread over a limited number of chips (effective processing gain), and each chip is transmitted over an orthogonal sub-carrier. Each sub-carrier is only used by a limited number of data symbols that may belong to different users. Consequently, each user, transmitting on given sub-carriers, will experience interference from only a small number of other users' data symbols. By applying message passing algorithm, the LDS-OFDM shows satisfactory performance and outperforms similar well-known systems over multi-path fading channels. But the receiver complexity of LDS-OFDM is relatively high. Therefore, it is challenging to design a LDS-OFDM receiver to achieve satisfactory performance while reducing complexity. Message passing schedule on sparse graphs not only influences the convergence rate, but also affects the system performance. In [7] and [8],

flooding schedule is adopted for MUD of LDS-OFDM, where all the nodes update messages simultaneously. Its convergence behavior is not ideal. In this paper we develop a serial schedule to perform MUD. By using more reliable information, the serial schedule improves the convergence rate and reduces receiver complexity. In addition, inspired by sparse structure of LDS-OFDM, we utilize LDPC codes for forward error correction (FEC) in LDS-OFDM system.

The structure of this paper is organized as follows. Section II introduces the system model of LDS-OFDM. In section III, flooding and serial schedules for iterative MUD are presented. LDPC decoding algorithm adopted by LDS-OFDM receiver is presented in section IV. Section V shows simulation results and comparisons. Finally, conclusions are drawn in section VI.

II. SYSTEM MODEL

In this section, a single cell uplink LDS-OFDM system model is presented. Let K be the number of users, M be the identical number of modulated symbols transmitted from each user, and N be the number of chips. Each chip is transmitted over an orthogonal sub-carrier, the ISI can be avoided by the insertion of cyclic prefix (CP) provided that the length of CP is longer than the channel delay spread.

Fig. 1 shows transmitter block diagram of an uplink LDS-OFDM system. Each user has an independent link as shown in the figure. Denote $x_{k,m}$ as the m^{th} data symbol of user k ($k \in [1, K], m \in [1, M]$), c_n is the n^{th} chip ($n \in [1, N]$). It can be seen that similar to the MC-CDMA spreading process, after FEC encoding and symbol mapping, we multiply the symbol with a spreading signature (a random sequence of chips) and perform the OFDM modulation afterwards. However, in the LDS-OFDM case, the main difference is that the spreading signature has a low density (a large number of chips in the sequence are equated to zero), i.e., zero padding and interleaving are added after the spreading procedure. In other words, the number of users' modulated symbols that superimposed on each chip is much less than the total number of modulated symbols, $d_c \ll (K \times M)$, where d_c is the number of symbols that superimposed on one chip. Similarly, the number of chips that spread by each symbol is much less than the total number of chips, $d_v \ll N$, where d_v is the number of chips that spread by one symbol. Thus the spreading sequence becomes very sparse, which means there are many zero in the sequence. In fact, d_v is the effective spreading factor. If d_c and d_v are both constants, it is a regular LDS-OFDM, otherwise it is an irregular LDS-OFDM.

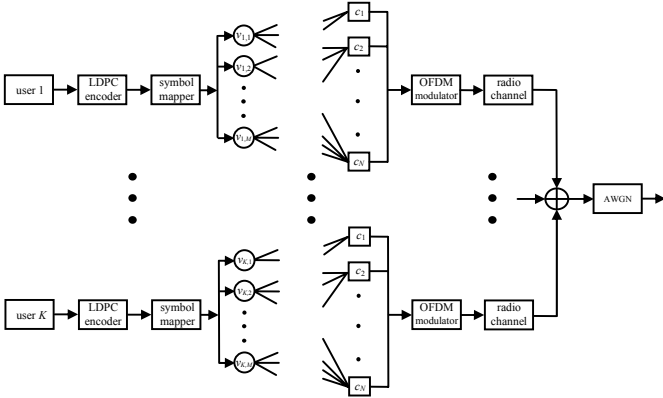


Fig. 1 Transmitter structure of LDS-OFDM

Fig. 2 shows receiver block diagram of an uplink LDS-OFDM. The receiver is a single base station. Users' signals that are using the same chip will be superimposed. As the number of symbols that interfere with each other at one chip is much less than total number of symbols, the LDS-OFDM can perform well under overload conditions. In Fig. 2, we can see that the despreading is performed over a low density signature (the dashed rectangle). There are two types of nodes in the signature: chip node c_n and variable node $x_{k,m}$. Different types of nodes are connected by sparse edges. Message passing algorithm can be employed for MUD of LDS-OFDM, which is presented in the next section.

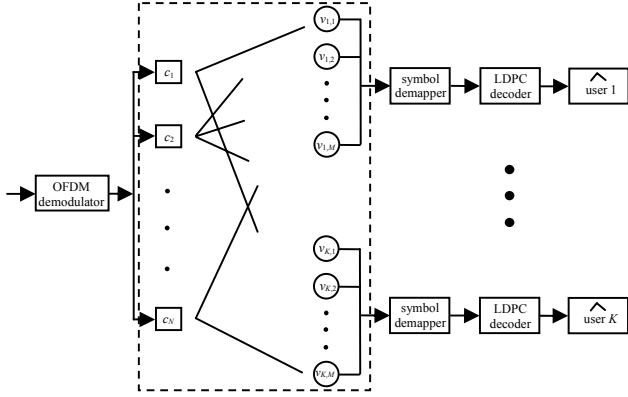


Fig. 2 Receiver structure of LDS-OFDM

III. ITERATIVE MULTIUSER DETECTION

The spreading matrix of user k is $\mathbf{S}_k = [\mathbf{s}_{k,1}, \dots, \mathbf{s}_{k,M}] \in \mathbb{C}^{N \times M}$, where \mathbb{C} represents the complex field and \mathbf{S}_k has only d_v non-zero elements at each column. Let us denote $\mathbf{S} = [\mathbf{S}_1, \dots, \mathbf{S}_K] \in \mathbb{C}^{N \times M \times K}$ as the low density signature matrix of the LDS-OFDM system, $\mathbf{A} = \text{diag}(A_1, \dots, A_K)$ as the transmit power gain of users and $\mathbf{G}_k = \text{diag}(g_{k,1}, \dots, g_{k,N})$ as the corresponding channel gain for the k^{th} user.

In LDS-OFDM, each user's generated chip will be transmitted over an orthogonal sub-carrier. The received spreading sequence for data symbol $m \in \{1, \dots, M\}$ of user k can be represented by $\mathbf{r}_{k,m} = T_k \mathbf{G}_k \mathbf{s}_{k,m}$. To be more specific, the

received signature gain at chip n of data symbol m of user k is $r_{k,m}^n = T_k g_{k,n} s_{k,m}^n$. Let $\psi_n = \{(k, m) : s_{k,m}^n \neq 0\}$ be the set of data symbols (which may belong to different users) that interfere on chip c_n , and $\varepsilon_{k,m} = \{n : s_{k,m}^n \neq 0\}$ be the set of different chips that the m^{th} symbol of user k is spread on.

For an uplink MC-CDMA system, the received signal at chip (sub-carrier) index n is written as

$$y_n = \sum_{k=1}^K \sum_{m=1}^M r_{k,m}^n v_{k,m} + z_n \quad (1)$$

where z_n is the noise over chip c_n . Because the spreading signature only has a limited number of non-zero positions in LDS-OFDM, we can express the received signal at the n^{th} chip (sub-carrier) as

$$y_n = \sum_{(k,m) \in \psi_n} r_{k,m}^n v_{k,m} + z_n \quad (2)$$

Let $L_{v_{k,m} \rightarrow c_n}^j$ be the log-likelihood ratio (LLR) delivered from variable node $v_{k,m}$ to chip node c_n at j^{th} iteration. Similarly, the LLR delivered from chip node c_n to variable node $v_{k,m}$ at j^{th} iteration is given by $L_{c_n \rightarrow v_{k,m}}^j$, and $L_{v_{k,m}}$ is the final estimation of variable node $v_{k,m}$. In message passing algorithm, the schedule is a key that determines convergence rate. The flooding schedule has been applied for LDS-OFDM in [7] and [8], which is summarized as follows.

A. Flooding Schedule

• Initialization

Assuming there is no *a priori* probability available, initial values at the first iteration are set to zero:

$$L_{v_{k,m} \rightarrow c_n}^1 = 0, \forall k, \forall m, \forall n \quad (3)$$

• Chip nodes updating

$$L_{c_n \rightarrow v_{k,m}}^j = f(x_{k,m} | y_n, L_{v_{k',m'} \rightarrow c_n}^{j-1}, (k', m') \in \psi_n \setminus (k, m)) \quad (4)$$

where $\psi_n \setminus (k, m)$ is the set of data symbols (excluding $x_{k,m}$) that interfere on chip c_n .

In order to approximate the optimum maximum a posteriori probability (MAP) detector, the right hand side of (4) represents marginalization function, which is based on (2), and can be written as

$$\begin{aligned} & f(x_{k,m} | y_n, L_{v_{k',m'} \rightarrow c_n}^{j-1}, (k', m') \in \psi_n \setminus (k, m)) \\ &= \log(\sum p^{j-1}(y_n | \mathbf{x}) p_n^{j-1}(\mathbf{x} | x_{k,m})) \\ &= \log(\sum p^{j-1}(y_n | \mathbf{x}) \prod_{(k', m') \in \psi_n \setminus (k, m)} p_n^{j-1}(x_{k', m'})) \end{aligned} \quad (5)$$

where the conditional probability density function (PDF) $p^{j-1}(y_n | \mathbf{x})$ and *a priori* probability $p_n^{j-1}(x_{k', m'})$ are given as

$$p^{j-1}(y_n | \mathbf{x}) = \exp(-\frac{1}{2\sigma^2} \|y_n - \mathbf{r}_{[n]}^T \mathbf{v}_{[n]}\|^2) \quad (6)$$

$$p_n^{j-1}(x_{k', m'}) = \exp(L_{v_{k', m'} \rightarrow c_n}^{j-1}) \quad (7)$$

where $\mathbf{v}_{[n]}$ and $\mathbf{r}_{[n]}$ denote the vector containing the symbols transmitted by the users that spread their data on chip n and

their corresponding effective received signature values, respectively. As can be seen in (5) that based on the received chip y_n and *a priori* input information $p_n^j(x_{k',m'})$, extrinsic values are calculated for all the constituent bits involved in (2). Combining (6) and (7) into (5), the message update will be

$$L_{c_n \rightarrow v_{k,m}}^j = \kappa_{n,k,m} \max_{\mathbf{v}_{[n]}}^* \left(\sum_{(k',m') \in \mathcal{V}_n \setminus (k,m)} L_{v_{k',m'} \rightarrow c_n}^{j-1} - \frac{1}{2\sigma^2} \|y_n - \mathbf{r}_{[n]}^T \mathbf{v}_{[n]}\|^2 \right) \quad (8)$$

where $\kappa_{n,k,m}$ denotes the normalization coefficient and

$$\max^*(a, b) = \log(e^a + e^b) \quad (9)$$

• Variable nodes updating

$$L_{v_{k,m} \rightarrow c_n}^j = \sum_{n' \in \mathcal{E}_{k,m} \setminus n} L_{c_{n'} \rightarrow v_{k,m}}^j \quad (10)$$

where $\mathcal{E}_{k,m} \setminus n$ is the set of different chips(excluding c_n) that the m^{th} symbol of user k is spread on.

• Estimation

This technique is based on log-MAP detection. After the message-passing has converged or has reached the maximum number of iterations J , a posteriori probability of the transmitted symbol $v_{k,m}$ is estimated as

$$L_{v_{k,m}} = \sum_{n \in \mathcal{E}_{k,m}} L_{c_n \rightarrow v_{k,m}}^J \quad (11)$$

By making a hard decision, the estimated value of $v_{k,m}$ is

$$\hat{v}_{k,m} = \arg \max_{v_{k,m}} L_{k,m}(v_{k,m}) \quad (12)$$

B. Serial Schedule

The flooding schedule for message passing is in a parallel manner, i.e., all chip nodes update at the same time, then all variable nodes update simultaneously. In the case of cycle-free signature, the belief will converge to the exact a posterior probability after a finite number of iterations that is bounded by half length of the longest path in the signature. Generally speaking, signature can not avoid cycle, and the propagated information may lead to inaccurate a posterior probability [9]. In the flooding schedule, the updated message has to be stored until all the other nodes complete updating, which means the new message can not join the belief propagation immediately. Thus the convergence speed is slow and the detection performance is also limited. Furthermore, since all iterative messages are float-point numbers, high speed processors and large memory registers are required for hardware implementation.

In order to improve the convergence rate and reduce MUD complexity, we present a serial schedule for LDS-OFDM. In the serial schedule, the chip nodes update message sequentially. Unlike the flooding schedule where the new message can only be used in the next iteration, the serial schedule allows immediate propagation of new messages, and it is more efficient in terms of convergence rate and hardware cost. In the serial schedule, we use $L_{v_{k,m}}$ and $L_{c_n \rightarrow v_{k,m}}^j$ to

compute $L_{v_{k,m} \rightarrow c_n}^j$ on the fly, avoiding additional memory to store $L_{v_{k,m} \rightarrow c_n}^j$. Such processing is derived by combining (10) and (8). It gives the following expression for the updating of chip node c_n :

$$L_{c_n \rightarrow v_{k,m}}^j = \kappa_{n,k,m}$$

$$\max_{\mathbf{v}_{[n]}}^* \left(\sum_{(k',m') \in \mathcal{V}_n \setminus (k,m)} (L_{v_{k',m'}}^{j-1} - L_{c_n \rightarrow v_{k',m'}}^{j-1}) - \frac{1}{2\sigma^2} \|y_n - \mathbf{r}_{[n]}^T \mathbf{v}_{[n]}\|^2 \right) \quad (13)$$

$$L_{v_{k,m}}^{j_{\text{new}}} = L_{v_{k,m}}^{j_{\text{old}}} - L_{c_n \rightarrow v_{k,m}}^{j_{\text{old}}} + L_{c_n \rightarrow v_{k,m}}^{j_{\text{new}}} \quad (14)$$

The detailed procedures are described in the sequel.

• Initialization

$$L_{v_{k,m} \rightarrow c_n}^1 = 0, \forall k, \forall m, \forall n \quad (15)$$

$$L_{c_n \rightarrow v_{k,m}}^1 = 0, \forall k, \forall m, \forall n \quad (16)$$

• Chip nodes updating

1) Accumulating all the messages delivered to the chip node c_n :

$$S = \sum_{(k',m') \in \mathcal{V}_n} (L_{v_{k',m'}} - L_{c_n \rightarrow v_{k',m'}}^{j-1}) \quad (17)$$

2) For each variable node that is connected to the chip node c_n :

$$L_{\text{temp}} = L_{v_{k,m}} - L_{c_n \rightarrow v_{k,m}}^{j-1} \quad (18)$$

$$L_{c_n \rightarrow v_{k,m}}^j = \kappa_{n,k,m} \max_{\mathbf{v}_{[n]}}^* (S - L_{\text{temp}} - \frac{1}{2\sigma^2} \|y_n - \mathbf{r}_{[n]}^T \mathbf{v}_{[n]}\|^2) \quad (19)$$

$$L_{v_{k,m}} = L_{\text{temp}} + L_{c_n \rightarrow v_{k,m}}^j \quad (20)$$

3) Estimation

$$\hat{v}_{k,m} = \arg \max_{v_{k,m}} L_{k,m}(v_{k,m}) \quad (21)$$

Obviously, compared with flooding schedule, more fresh information can be utilized in the serial schedule. Thus the convergence rate and system performance can be improved, which will be shown in section V.

IV. ITERATIVE LDPC DECODING

As the low density matrices in LDPC codes are very similar to the low density signature of LDS-OFDM, we utilize LDPC codes for LDS-OFDM system. Message passing algorithm, also referred as the sum product algorithm (SPA), is well-known for LDPC decoding [10][11]. The normalized min-sum (NMS) algorithm is a simplified version of SPA that can reduce computation complexity significantly without loss of decoding performance. The NMS performs belief propagation iteratively and outputs the a posteriori probabilities of the coded bit. Similar to the low density signature of LDS-OFDM, the LDPC code is based on a low density parity check matrix \mathbf{H} with dimensions $M \times N$. Each row in \mathbf{H} represents a parity check equation, while each column corresponds to a coded bit. Let $R_{v \rightarrow c}$ be the LLR delivered from variable node to check node, $R_{c \rightarrow v}$ be the LLR delivered from check node to variable node, and R_v be the soft estimation of variable node. We present the NMS as follows.

- Initialization

The output of MUD, $L_{v_{k,m}}$, is sent to LDPC decoder as initial value.

- Parity check nodes updating

$$R_{v \rightarrow c} = \alpha \times \beta \quad (22)$$

where $\alpha = \text{sign}(R_{v \rightarrow c})$ and $\beta = \text{abs}(R_{v \rightarrow c})$, then

$$R_{c \rightarrow v} = \prod_{\text{excluding self edges}} \alpha \times \beta \quad (23)$$

- Variable nodes updating

$$R_{v \rightarrow c} = L_{v_{k,m}} + \sum_{\text{excluding self edges}} R_{c \rightarrow v} \quad (24)$$

- Estimation

$$R_v = L_{v_{k,m}} + \sum_{\text{all edges}} R_{c \rightarrow v} \quad (25)$$

The LDPC decoder can make hard decision according to the R_v .

- Syndrome computing

If syndrome equals to zero or the decoder reaches the maximum number iterations, the decoding is terminated; otherwise continue the iterations again.

V. SIMULATIONS AND COMPARISONS

In this section, the bit error rate (BER) performance of LDS-OFDM is evaluated over ITU Pedestrian Channel B with 6 channel taps [12]. Simulation parameters are chosen as follows. The number of users is 10, the FFT size is 64, the sub-channel bandwidth is 15 KHz and the system overloading is 200%. The low density signature of LDS-OFDM has 60 chip nodes and 120 variable nodes.

For fair comparisons and to exhibit the difference between different schedules of MUD, we first evaluate un-coded LDS-OFDM. Fig. 3 shows the BER results for several schemes of iterative MUD. As can be seen from the figure, at the first few iterations (iteration of 1 or 3), the serial schedule attains much better performance than the flooding schedule. This is due to

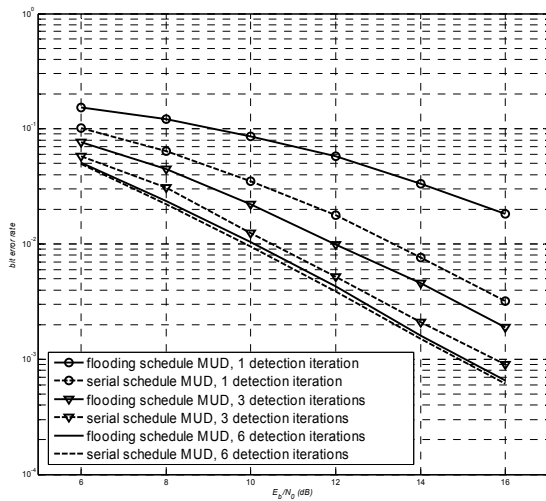


Fig. 3 Different iterative MUD for un-coded LDS-OFDM

the fact that the updated message can participate in the belief propagation when serial schedule is applied. Consequently, it is possible to gather more accurate information and accelerate the convergence rate. On the other hand, as the iterative process goes on (iteration of 6), the gap between flooding schedule and serial schedule becomes smaller. We can see that at the 6th iteration, the curves of two schedules almost overlap. This phenomenon informs that these two schedules will eventually converge to the same point. It is worth noting that the performance of the serial schedule with 3 iterations is very close to that of flooding schedule with 6 iterations.

One of the main advantages of the receiver for LDS-OFDM is its ability to support high loads while maintaining affordable complexity. For MUD, the complexity of LDS-OFDM is $O(|X|^{d_c})$, which is much less than $O(|X|^{K \times M})$ -- the complexity of conventional MC-CDMA (X denotes the constellation alphabet). According to Fig. 3, the serial schedule can significantly reduce detection iteration and complexity with marginal performance loss, i.e., only half detection complexity (3 iterations) is needed when compared with flooding schedule (6 iterations). Therefore, the advantages of the serial schedule are: 1) although detection performance of the serial schedule is nearly the same to that of the flooding schedule when the number of MUD iterations is large enough (more than 6), however, in some applications where there is a constraint on the number of affordable iterations due to the hardware cost, the serial schedule can achieve much better performance than the flooding schedule thanks to the faster convergence; 2) the memory requirement of the serial schedule is less than that of the flooding schedule, as the serial schedule saves the memory space for $L_{v_{k,m} \rightarrow c_n}^j$. The disadvantage of the serial schedule is that it causes longer processing delay than the flooding schedule.

Fig. 4 illustrates different FEC for coded LDS-OFDM, where the serial schedule with 3 iterations is chosen for the MUD. The compared FEC includes (2, 1, 7) convolutional code, (60, 30) LDPC code and (300, 150) LDPC code. The decoding algorithm for the convolutional code is the MAP algorithm which is originally proposed by Bahl, Cocke, Jelinek and Raviv (BCJR) [11]. The NMS algorithm presented in Section IV is adopted for LDPC decoding, where the maximum iteration number is set to 5. In Fig. 4, we can see that the performance of LDPC codes depends on the code length, and both LDPC codes outperform the convolutional code. In the medium to high SNR region, the (300, 150) LDPC code can attain about 0.2 dB gain over the (60, 30) LDPC code and about 0.5 dB gain over the convolutional code. Therefore, LDPC codes are more suitable for LDS-OFDM system than the convolutional code. This follows from the fact that both LDS-OFDM and LDPC code are based on sparse graph, and message passing algorithms can be efficiently applied for detection or decoding in LDPC coded LDS-OFDM systems. In terms of the decoding complexity, (60, 30) LDPC code is almost the same to the convolutional code, but (300, 150) LDPC code needs more than 5 times of decoding complexity than other two codes.

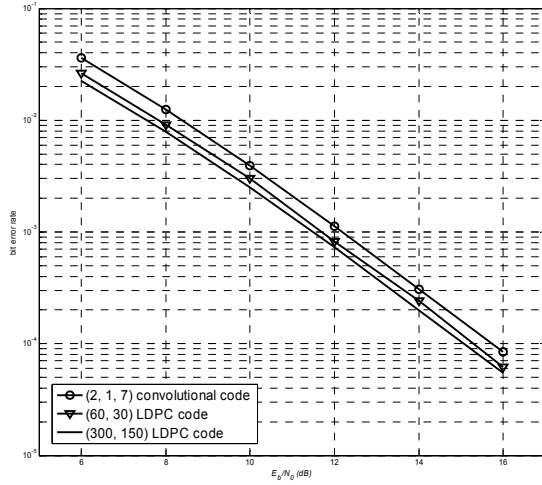


Fig. 4 Different FEC for coded LDS-OFDM

In order to fully investigate the effect of the serial scheduled MUD and LDPC decoding, we compare the LDS-OFDM system under different combinations of detection and decoding iterations, which is shown in Fig. 5. The serial schedule is chosen for the MUD, and the (60, 30) LDPC code is chosen for the FEC. As can be seen from this figure, BER performance fluctuates when different detection and decoding iterations are adopted. When 1 detection iteration and 5 decoding iterations are chosen, the BER performance is unsatisfactory. It justifies that the MUD is very important to the overall performance. Moreover, when 3 detection iterations and 5 decoding iterations are performed, the system performance is very close to that of the combination of 6 detection iterations and 5 decoding iterations. They can gain more than 5 dB compared with the schemes that only 1 detection is performed. Hence, taking account of system performance and receiver complexity, 3 detection iterations and 5 decoding iterations are the preferred choice.

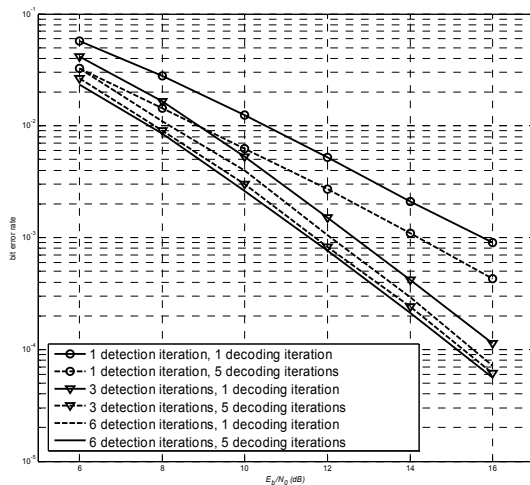


Fig. 5 Different detection and decoding iterations for LDS-OFDM

VI. CONCLUSIONS

LDS-OFDM is a promising candidate for future mobile communications, but its MUD convergence rate is not good enough, and its receiver complexity is relatively high. In this paper, a serial schedule is developed for the iterative MUD in LDS-OFDM. In the proposed serial schedule, updated message can be assimilated immediately in current iteration, hence the convergence rate is significantly improved. Meanwhile, compared with the conventional flooding schedule, the MUD iteration number can be saved, consequently the receiver complexity can be reduced, i.e., about half MUD complexity is saved with marginal performance loss. Furthermore, due to the similar sparse structure, LDPC codes are utilized for LDS-OFDM systems. Numeric results show that, by choosing proper numbers of iteration for serial scheduled MUD and LDPC decoding, it is possible to attain a satisfactory performance with affordable receiver complexity. The system can be extended to MIMO transmissions, which needs more detailed research in the future.

ACKNOWLEDGMENT

This work was supported by the UK Engineering and Physical Science Research Council (EPSRC) under grant number EP/J017655/1 and the fifth generation (5G) project.

REFERENCES

- [1] A. M. Tulino, L. Li, and S. Verd, "Spectral efficiency of multicarrier CDMA," *IEEE Transactions on Information Theory*, vol. 51, no. 2, pp. 479–505, February 2005.
- [2] B. A. Al-fuhaidi, H. E. A. Hassan, M. M. Salah and S. S. Alagooz, "Parallel interference cancellation with different linear equalisation and rake receiver for the downlink MC-CDMA systems," *IET Communications*, vol. 6, no. 15, pp. 2351–2360, March 2012.
- [3] K. Zhang, Y. L. Guan and Q. Shi, "Complexity reduction for MC-CDMA with MMSEC," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1989–1993, May 2008.
- [4] P. H. Tan and L. K. Rasmussen, "Asymptotically optimal nonlinear mmse multiuser detection based on multivariate gaussian approximation," *IEEE Transactions on Communications*, vol. 54, no. 8, pp. 1427–1438, August 2006.
- [5] A. Montanari and D. Tse, "Analysis of belief propagation for non-linear problems: The example of CDMA (or: How to prove tanaka's formula)," in *IEEE Information Theory Workshop*, March 2006, pp. 160–164.
- [6] R. Hoshyar, F. P. Wathan, and R. Tafazolli, "Novel low-density signature for synchronous CDMA systems over awgn channel," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1616–1626, April 2008.
- [7] R. Hoshyar, R. Razavi, and M. Al-Imari, "LDS-OFDM an efficient multiple access technique," in *Proc. IEEE Vehicle Technology Conference*, pp. 1–5, 2010.
- [8] R. Razavi, M. Al-Imari, Muhammad Ali Imran, R. Hoshyar and Dageng Chen, "On receiver design for uplink low density signature OFDM (LDS-OFDM)," *IEEE Transactions on Communications*, pp. 3499–3508, Vol 60, No.11, November 2012.
- [9] Clementi, A and Dipt. di Mat, "Information spreading in stationary markovian evolving graphs," *IEEE Transactions on Parallel and Distributed Systems*, pp. 1425–1432, Vol 22, No.9, September 2011.
- [10] R. G. Gallager, "Low-density parity-check codes," Cambridge, MA: MIT Press, 1963.
- [11] Yong Lin and Wensheng Niu, "High throughput LDPC decoder on GPU," *IEEE Communication Letter*, pp. 344 - 347, Vol 18, No.2, February 2014.
- [12] 3GPP, "3rd generation partnership project: technical specification group radio access network," *Technical report*. 25.890 V. 1.0.0, 2002.