On Fixed-point Implementation of Log-MPA for SCMA Signals

Jiaqi Liu, Gang Wu, *Member, IEEE,* Shaoqian Li, *Fellow, IEEE,* Olav Tirkkonen, *Member, IEEE*

***Abstract*—In this paper, we present a design framework for fixed-point implementation of the log-domain message passing algorithm (Log-MPA) for sparse code multiple access signals, and make a detailed comparative analysis on the complexity of Log-MPA and traditional MPA. We also investigate the impact of the number of massage-passing iterations within the Log-MPA decoding process on the performance of Log-MPA. Simulation results demonstrate that Log-MPA achieves favorable performance with low-complexity hardware implementation as compared to MPA. We also implement Log-MPA on FPGA evaluation board to verify the performance of Mog-MPA.**

***Index Terms*—Sparse code multiple access, iterative decoding, massage passing, log-Likehood ratio, bipartite graph**

I. INTRODUCTION

SPARSE code multiple access (SCMA) [1] has attracted a lot of attention as a multiple-access technique candidate for the fifth generation (5G) mobile systems. Originating from low-density signature techniques [2], [3], SCMA improves spectral efficiency of wireless radio access by designing special codebooks [4], [5] to support overloading. SCMA provides 50% to 200% more connections compared with Orthogonal Frequency-Division Multiple-Access.

A major challenge with SCMA is its decoding complexity. A message-passing algorithm (MPA) exploiting the sparsity of the SCMA signal, is proposed in [4], [10]–[12]. MPA reaches near maximum-likelihood perfromance, and is able to achieve the shaping gain of the multi-dimensional SCMA constellation. In [5], a simplified log-domain MPA (Log-MPA) decoder was proposed, where the extrinsic information (EI) is computed in log domain.

In this letter, we present a detailed description on implementation of SCMA decoder based on fixed-point platforms, e.g., the FPGA systems. We analyze decoding complexity of Log-MPA and MPA in detail by comparing the number of operations involved in decoding process. Finally, The bit error rate (BER) performance of Log-MPA for uncoded SCMA, as well as the the block error rate (BLER) performance of Log-MPA for turbo coded SCMA, are verified via both floating and fixed-point simulation, with different numbers of Message Passing Iterations (MPIs). Moreover, we verified the previously mentioned performance with Verilog HDL program and based on Altera Stratix® V FPGA device.

The main contribution of this paper includes the design of the fixed-point implementation of Log-MPA, the complexity analysis of MPA and Log-MPA, the performance comparison of MPA and Log-MPA for SCMA by extensive simulations and hardware experiments, and a recommended quantization format of the EI.

This work was supported by the National Science Foundation of China under Grant 61461136003, and the Academy of Finland under Grant 284725.

J. Liu, G. Wu, and S. Li are with the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China

O. Tirkkonen is with the Department of Communications and Networking, Aalto University, Aalto FI-00076, Finland.

Corresponding author: Gang Wu, wugang99@uestc.edu.cn

II. SIGNAL MODEL OF SCMA

A. SCMA Encoder

An SCMA encoder can be regarded as a mapper from input bits to multi-dimensional constellations, which combines a bit-to-symbol mapping and sparse code spreading. The incoming bits are directly mapped to a multi-dimensional codeword in an SCMA codebook. An SCMA encoder contains J separate transmit layers, each encoding $\log_2 M$ bits. The m-th codeword of the j-th transmit layer, $\mathbf{c}_j^m = \mathbf{V}_j g_j(\mathbf{b}_j^m), \forall m \in \mathcal{M}$, is a K-dimensional complex sparse vector with N non-zero entries. Here $\mathcal{M} = \{1, 2, \ldots, M\}$ is the set of codeword indexes, and $\mathbf{b}_j^m$ denotes the incoming binary bits with index m. The mapping from bits to a multidimensional constellation is denoted by $g_j : \mathbb{B}^{\log_2 M} \rightarrow \mathcal{C}$, $\mathbf{s}_j^m = g_j(\mathbf{b}_j^m)$. Here $\mathbf{s}_j$ denotes an N-dimensional complex constellation point taken from the constellation set $\mathcal{C} \subset \mathbb{C}^N$, and $\mathbf{V}_j \in \mathbb{B}^{K \times N}$ denotes the binary mapping matrix which spreads the N-dimensional constellation points to the K-dimensional SCMA codewords. Since $N < K$, the $\mathbf{c}_j^m$ is sparse. We define the codebook matrix of the j-th layer as $\mathbf{C}(j) = \left[\mathbf{c}_j^1, \mathbf{c}_j^2, \ldots, \mathbf{c}_j^M\right]$.

SCMA codebook design is the key to achieve the shaping gain of multi-dimensional constellation and hence to improve the performance of the SCMA system. During implementation, SCMA codebook is designed and stored into a look-up table. The SCMA encoder just finds the SCMA codeword from the stored table for the input bits, and maps it to the physical resource elemens (REs). One group of K REs constitutes an SCMA block.

The structure of an SCMA encoder with J transmit layers and K REs can be represented by a factor graph defined by a sparse matrix $\mathbf{F}$. Taking each RE as a function node (FN) and each transmit layer as a variable node (VN), transmit layer j and RE k are connected if $(\mathbf{F})_{kj} = 1$. The degree ofd_f of FNs and the degree N of VNs are two key parameters to design the codebook which determine the decoding complexity. The overloading factor of the code is $\lambda = J/K$.

A simple method to construct the codebook, i.e. the multi-dimensional constellation of each layer, is to rotate a mother

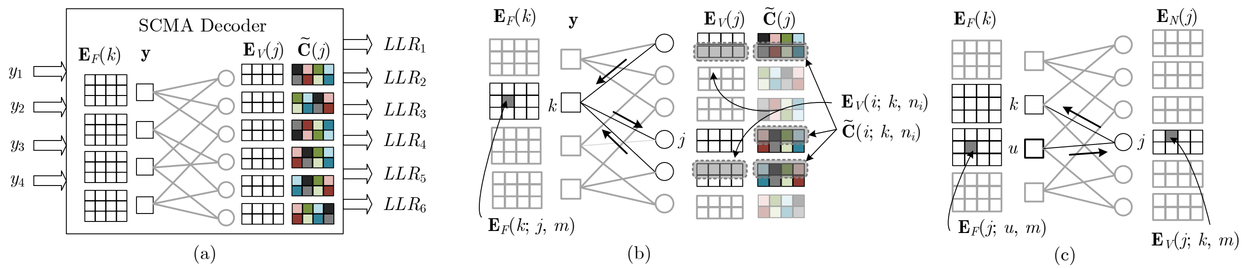

Fig. 1. Structure and detailed description of SCMA decoder with MPA detection: (a) Modules of an SCMA decoder; (b) Detailed description of Step 1, i.e. the EI update at FNs; (c) Detailed description of Step 2, i.e. the EI update at VNs.

constellation. This is described in detail in [1]. In our work, we use standard QPSK as the mother constellation, and phase rotations from the set $\{0, \frac{\pi}{6}, \frac{\pi}{3}\}$, according to a regular factor graph matrix, $\mathbf{F}$, as defined in [1].

B. SCMA for OFDM

Here we consider OFDM as a basic underlying waveform so the basic physical resource element is an LTE RE [6]—one subcarrier in one OFDM symbol. In an uplink scenario, J users are multiplexed over K REs, and each user is assigned a transmit layer, or in other words, a codebook. Here we assume that both the base station and the users are equipped with a single antenna. The uplink [7], [8] received signal at the base station on the received subcarriers is

$$\mathbf{y} = \sum_{j=1}^{J} \text{diag}(\mathbf{h}_j)\mathbf{x}_j + \mathbf{n}, \tag{1}$$

where $\mathbf{y}$ is a K-dimensional vector denoting the received signal on the K REs, $\mathbf{x}_j$ is the transmitted codeword of user j, $\mathbf{h}_j$ is the channel vector of user j, and $\mathbf{n}$ is additional white Gaussian noise (AWGN). Here, diag$(\mathbf{v})$ is a diagonal matrix with the vector $\mathbf{v}$ on the diagonal.

In a downlink scenario, the codewords from the base station to J users are multiplexed on K REs, and each user is allocated a transmit layer. The received signal at user j is

$$\mathbf{y}_j = \text{diag}(\mathbf{h}_j) \sum_{j=1}^{J} \mathbf{x}_j + \mathbf{n}_j, \tag{2}$$

where the K-dimensional vector $\mathbf{y}_j$ denotes the received complex signal, $\mathbf{x}_j$ is the codeword for user j, $\mathbf{h}_j$ is the channel vector and AWGN of user j, respectively.

III. SCMA Decoder with Log-MPA Detection

Joint multi-user detection has to be adopted for a non-orthogonal system like SCMA, in which more than one OFDM symbols are overlaid on each RE. Maximum a posterior probability (MAP) detection is optimal but with very large complexity. By utilizing SCMA codeword sparsity, a message passing algorithm (MPA) on a factor graph can achieve near-optimal performance. Although MPA has considerably lower complexity than MAP, it still suffers from implementation complexity and a wide dynamic range of exponential operations when computing likelihood ratios. Inspired by the Log-MAP decoding algorithm for turbo codes [9], a further simplification can be achieved by realizing MPA in log domain. Here we consider a log-domain implementation of MPA, called Log-MPA.

Fig. 1(a) shows the structure of the Verilog HDL-based module for an SCMA decoder with $K = 4$ function and $J = 6$ variable nodes, and each layer encoding $\log_2 M = 2$ bits. The degree of the function nodes is $d_f = 3$, and the degree of the variable nodes is $N = 2$. The inputs of the module are the received complex signals on the $K = 4$ REs, and the outputs are bitwise log-likelihood ratios (LLRs) for the corresponding coded binary bits, which can be passed to a Forward error corection (FEC) decoder. In Fig. 1, each complex variable of the input signals, $\mathbf{y}$, as well as the efficient codebook, $\widetilde{\mathbf{C}}$, are shown as a block for simplicity. While in Verilog HDL program design, the in-phase and quadrature components of them are transmitted, received and saved separately.

The efficient codebooks considering the channel coefficient for each layer are saved as the matrix $\widetilde{\mathbf{C}}(j) = \text{diag}(\mathbf{h}_j)\mathbf{C}(j)$, with an entry $\widetilde{\mathbf{C}}(j; k, m)$ for each RE k that the layer uses, and each codeword m of the layer. These entries are multiplied with the channel coefficient of layer j at RE k.

At each FN k, EI is saved in a $d_f \times M$ matrix $\mathbf{E}_F(k)$, of which the (j, m)-th element, $\mathbf{E}_F(k; j, m)$, is the EI corresponding to the mth codeword on layer j. Rows of this matrix are sent to neighboring VNs. Similarly, at each VN j, the EI updated in each iteration is saved in a $N \times M$ matrix $\mathbf{E}_V(j)$, rows of which are sent to the neighboring FNs.

Each iteration of MPA consist of 2 steps, i.e. the EI update at FNs and the EI update at VNs. The detailed procedures are described as follows:

Step 1: Update the EI at FNs

The log-domain EI to be passed from FN k to VN j for codeword m is updated as

$$\begin{aligned} \mathbf{E}_F(k; j, m) \quad = \quad & \max_{\mathbf{n} \in \, \mathcal{N}_k|_{n_j = m}} -\frac{1}{2\sigma^2} \left| y_k - \sum_{i \in \mathcal{L}_k} \widetilde{\mathbf{C}}(i; k, n_i) \right|^2 \\ & + \sum_{i \in \mathcal{L}_k, i \neq j} \mathbf{E}_V(i; k, n_i), \end{aligned} \tag{3}$$

where $\mathcal{L}_k$ is the set of the VNs that are neighbours of FN k. The vector $\mathbf{n}$ denotes a combination of codeword indexes for VNs, taking values in $\mathcal{N}_k\,|_{n_j=m} = \left\{\mathbf{n} \in \mathcal{M}^{|\mathcal{L}_k|} | n_j = m\right\}$, which denotes the set of all possible combinations of the codeword indexes on the neighbouring VNs to FN k with the m-th codeword on VN j. AWGN power is indicated by σ^2. Fig. 1(b) shows the detailed iteration above. There are totally NMd_f EI variables to be updated. In Verilog HDL program design, these variables are designed to be updated in parallel.

Step 2: Update the EI at VNs

For each VN-FN pair, the log-domain EI to be sent from VN j to FN k for codeword m is updated as

$$\mathbf{E}_V(j;k,m) = \max_{u\in\mathcal{R}_j, u\neq k} \mathbf{E}_F(j;u,m), \tag{4}$$

where $\mathcal{R}_j$ denotes the set of all FNs that are neighbours of VN j. In the case where 2 FNs connect to each VN, a simplified form of this equation is as (11) in [5].

After the desired number of iterations, the log likelihood for each codeword of a layer can be obtained by the sum of corresponding EI at the FNs as

$$\log \text{Prob}\left(c_{j,m} \mid \mathbf{y}\right) = \sum_{k\in\mathcal{R}_j} \mathbf{E}_V(j;k,m). \tag{5}$$

Bitwise output LLRs are calculated according to the bit-to-codeword mapping.

Here we analysis the decoding complexity for the codebooks with regular factor graph matrix. To obtain each element of the efficient codebook matrix $\widetilde{\mathbf{C}}(j)$, one complex multiplication is involved. Thus the computation of efficient codebook matrices totally requires $2MKd_f$ additions and $4MKd_f$ multiplications. To update one variable of EI at FN according to (3), MPA needs $(2d_f+1)M^{d_f-1}$ additions, $(d_f+2)M^{d_f-1}$ multiplications, and M^{d_f-1} exponentials, while Log-MPA only requires $3d_fM^{d_f-1}$ additions, $3M^{d_f-1}$ multiplications, and M^{d_f-1} comparisons. To update one variable of EI at VN in (4), MPA requires $(N-2)$ additions, while Log-MPA needs only $(N-2)$ comparisons.

TABLE I shows the computational complexity of MPA and Log-MPA, where N_{iter} is the number of MPIs. In Fig. 2, we illustrate the comparison among two different codebooks, both with $M=4$, $N=2$ and $N_{\text{iter}}=4$. Fig. 2(a) shows the decoding complexity for the codebooks with $K=4$, and thus with $J=6$, $d_f=3$, and $\lambda=150\%$. Fig. 2(b) shows the decoding complexity for the codebooks with $K=6$, and thus with $J=15$, $d_f=5$, $\lambda=250\%$.

With 150% overload, Log-MPA requires 29% more additions than MPA, but 40% less multiplications. While for the case of 250% overload, Log-MPA needs only 36% more additions than MPA, whereas it saves 57% of the multiplications. Both the increase in additions and decrease in multiplications grow with the overload factor. Considering the complexity and running time of multiplications, Log-MPA has clear advantages in implementation complexity with higher overload factors.

TABLE I
COMPLEXITY OF MPA AND LOG-MPA.

Algorithms	MPA	Log-MPA
ADD	$(2d_f+1)M^{d_f}Kd_fN_{\text{iter}}$ $+(N-2)MKd_fN_{\text{iter}}$ $+2MKd_f$	$3M^{d_f}Kd_f^2N_{\text{iter}}$ $+2MKd_f$
MUL	$(d_f+2)M^{d_f}Kd_fN_{\text{iter}}$ $+4MKd_f$	$3M^{d_f}Kd_fN_{\text{iter}}$ $+4MKd_f$
EXP	$M^{d_f}Kd_fN_{\text{iter}}$	0
CMP	0	$M^{d_f}Kd_fN_{\text{iter}}$ $+(N-2)MKd_fN_{\text{iter}}$

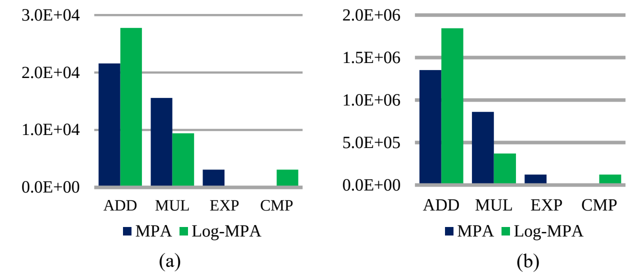

Fig. 2. Complexity comparison between MPA and Log-MPA with (a) $\lambda=150\%$ and (b) $\lambda=250\%$.

IV. SIMULATION AND EXPERIMENTAL RESULTS

We have done both float-point simulation and fixed-point simulation through careful design on choosing Fixed-point width and precision for Log-MPA. Furthermore, we verified our fixed-point simulation results via Altera Stratix® V FPGA evaluation board by implementing Verilog HDL-based Log-MPA.

TABLE II shows the configurations for fixed point simulation of Log-MPA and MPA. Note that the number of decimal bits for FPGA implementation can be controlled by the number of truncated bits. As mentioned in the previous section, storage for two variants of MKd_f EI variables is required. We assume automatic gain control so that the amplitude of received signal is normalized before ADC, and the reference codebooks for the SCMA decoder is also normalized with the same factor. In order to reduce the memory overhead for intermediate variables in iteration, we omit the decimal part of the EI. To ensure the same workspace overhead, in the fixed-point simulation of MPA, an EI variable is saved as an 9 bit unsigned decimal without integer bit.

TABLE II
FORMAT OF FIXED-POINT NUMBERS.

Variables	**Fixed-point number format**
Output for encoder before DAC	Q7 w/ 1 sign and 7 decimal bits
Input for decoder after ADC	Q7 w/ 1 sign and 7 decimal bits
Codebook	Q7 w/ 1 sign and 7 decimal bits
Storage of $\mathbf{E}_F$	Unsigned, 7 int bits w/ 2 decimal bits
Storage of $\mathbf{E}_V$	Unsigned, 7 int bits w/ 2 decimal bits

BER performance of SCMA systems in an AWGN channel is evaluated in Fig. 3. We consider an SCMA system with $K=4$, $N=2$, $J=6$, $M=4$, and thus the overload factor is $\lambda=150\%$. For the baseline uncoded LTE system, we assume that one RE is assigned to each transmit layer, and the symbol rate on each layer is the same as that in each layer of SCMA, i.e., 100 Mbps. Channel estimation and transmit power control are assumed to be perfect, and the 5 MPIs are used for both MPA and Log-MPA. SCMA loses about 2.5 dB in performance on average at BER 10^{-4} as compared to baseline LTE. However, SCMA has a 50% total symbol rate gain. The performance loss caused by replacing MPA with low complexity log-MPA is negligible. Fixed-point log-MPA loses less than 1dB to the floating-point version. While for the MPA, the BER performance of the fixed-point version is significantly degraded as compared to floating-point. This is a consequence

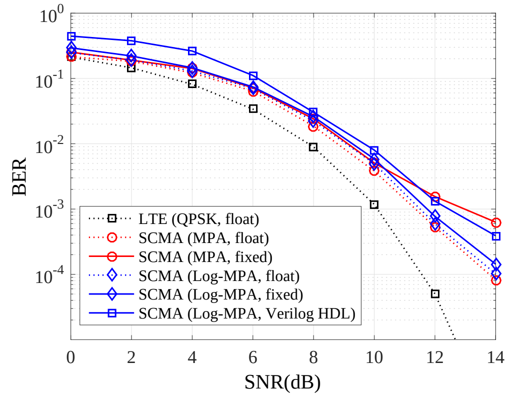

Fig. 3. BER performance evaluation, uplink LTE and SCMA systems with MPA and Log-MPA with floating and fixed-point simulations.

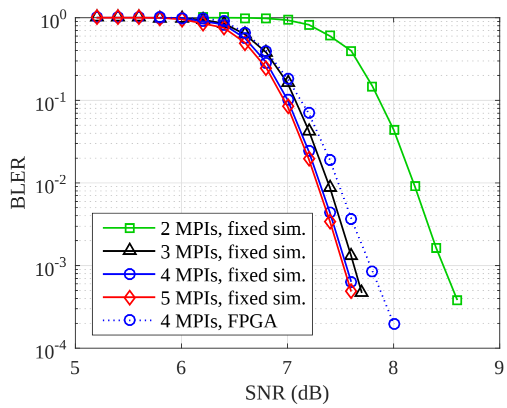

Fig. 4. BLER performance of SCMA with turbo FEC.

of the large dynamic range of the exponential operation, which enlarges the impact of quantization errors introduced by fixed-point expressions of small decimals. Therefore, Log-MPA outperforms MPA especially in fixed point realization, especially with few quantification bits. We also verify the BER performance of Log-MPA with FPGA implementation. The software simulation for Verilog HDL-based Log-MPA with 5 MPIs based on ModelSim® shows an acceptable gap with the previous results. This gap may be caused by the internal processing differences between the real Altera IP cores and MATLAB-simulated version of them.

In Fig. 4, BLER performance of SCMA with concatenated turbo FEC is evaluated. The BLER performance of SCMA decoder with different numbers of MPIs from 2 to 5 are given for comparison. The length of a code block is 1024, including 1000 data bits and 24 CRC bits. The turbo FEC is 1/3-rate according to the 3GPP LTE specifications. It is observed that using 4 MPIs still is recommendable, with about 0.1 dB gain at BLER 10^{-3}, as compared to 3 MPIs. While 5 MPIs shows no significant improvement compared to 4 MPIs. We also verify the results of turbo-coded SCMA with Altera Stratix® V FPGA device. Experimental results show an acceptable gap with the simulation. This gap may be caused by the internal functional difference between the Turbo FEC IP core provided by Altera and the Turbo FEC functions in MATLAB® LTE System Toolbox.

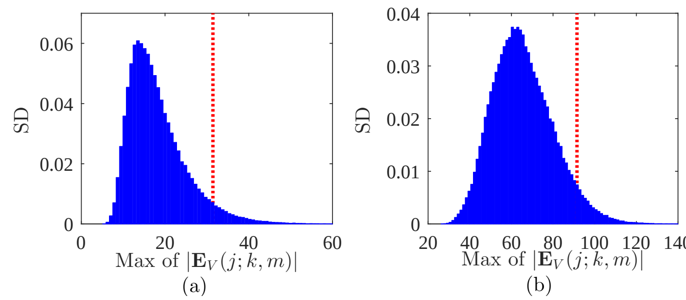

Fig. 5. SD for the maximum value of $|\mathbf{E}_V(j;k,m)|$ in Log-MPA decoding process with 5 MPIs, and (a) SNR = 3 dB, (b) SNR = 10 dB.

We also take the dynamic range of EI in Log-MPA decoding into consideration. As mentioned before, the EI variables are all negative numbers whose absolute values increase after each MPI. Fig. 5 shows the statistical distribution (SD) of the maximum absolute value of EI variables, i.e. $\mathbf{E}_V(j;k,m), \forall j,k,m$, after the last MPI of MPA decoding. The dashed line marks the value at which the cumulative frequency gets 0.95. It is observed that the dynamic range of EI variables gets larger with the increasing number of MPIs and SNR. Generally, 7 bits for the storage of the integer part of EI is enough.

V. Conclusion

In this letter, we have described the detailed implementation of Log-MPA for SCMA decoding, and analyzed its complexity. Software simulation results show that the performance of Log-MPA is close to MPA in fixed-point implementation with few quantization bits. We also evaluate the influence of MPIs on the performance of Log-MPA decoder. Moreover, experimental results based on FPGA verify the performance of Log-MPA.

References

[1] H. Nikopour, and H. Baligh, "Sparse code multiple access," in *Proc. IEEE PIMRC*, pp. 332–336, 2013.

[2] R. Hoshyar, R. Razavi,and M. Imari. "LDS-OFDM an efficient multiple access technique," in *Proc. IEEE VTC spring*, pp. 1–5, 2010.

[3] R. Hoshyar, F.P. Wathan, and R. Tafazolli, "Novel Low-Density Signature for Synchronous CDMA Systems Over AWGN Channel," *IEEE Trans. Signal Processing*, vol. 56, No. 4, pp. 1616–1626, Apr. 2008.

[4] M. Taherzadeh, H. Nikopour, A. Bayesteh, and H. Baligh, "SCMA Codebook Design," in *Proc. IEEE VTC Fall*, pp. 1–5, 2014.

[5] S. Zhang, X. Xu, L. Lu, Y. Wu, G. He, and Y. Chen, "Sparse code multiple access: An energy efficient uplink approach for 5G wireless systems," in *Proc. IEEE Globecom*, pp. 4782–4787, 2014.

[6] 3GPP TR36.814 (V9.0.0), "Further advancements for E-UTRA physical layer aspects," Mar. 2010.

[7] A. Bayesteh, E. Yi, H. Nikopour, and H. Baligh, "Blind detection of SCMA for uplink grant-free multiple-access," in *Proc. IEEE ISWCS*, pp. 853–857, 2014.

[8] K. Au, L. Zhang, H. Nikopour, E. Yi, A. Bayesteh, et. al., "Uplink contention based SCMA for 5G radio access," in *Proc. Globecom Workshops*, pp. 900–905, 2014.

[9] P. Robertson, E. Villebrun, P. Hoeher. "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *IEEE ICC*, Seattle, vol. 2, pp. 1009–1013, 1995.

[10] H. Mu, Z. Ma, M. Alhaji, P. Fan, and D. Chen, "A Fixed Low Complexity Message Pass Algorithm Detector for Up-Link SCMA System," *IEEE Wireless Commun. Lett.*, vol.4, no.6, pp.585-588, Dec. 2015

[11] K. Xiao, B. Xiao, S. Zhang and Z. Chen and B. Xia, "Simplified Multiuser Detection for SCMA with Sum-Product Algorithm," 2015 [Online]. Available: http://arxiv.org/abs/1508.00679.

[12] D. Wei, Y. Han, S. Zhang, and L. Liu, "Weighted message passing algorithm for SCMA," in *IEEE WCSP*, pp.1-5, Oct. 2015.