Memory Error Resilient Detection for Massive MIMO Systems

Victor Tomashevich and Ilia Polian
Faculty of Mathematics and Computer Science, University of Passau, Germany, Innstr. 41, 94032
Email: {victor.tomashevich, ilia.polian}@uni-passau.de

***Abstract*—Massive MIMO systems employing hundreds of antennas at the base station (BS) are considered a breakthrough technology to provide users with high data rates. However, large number of antennas demands memories of large size which are prone to faults due to current aggressive technology downscaling. This paper introduces a novel nonlinear minimum mean square error (NMMSE) based detection algorithm that takes memory errors into account. The proposed detection method is able to handle multiple memory errors with low computational overhead. Simulation reports that the proposed solution significantly reduces the impact of multiple memory errors on the bit error rate (BER).**

I. Introduction

Massive MIMO systems have recently gained attention as potential means to improve spectral efficiency, link quality and coverage compared to contemporary small-scale MIMO [1]. This is achieved by employing hundreds of antennas at the BS, simultaneously serving tens of users. The improvements come at the cost of the computational complexity increase at the BS. Compared to small-scale MIMO, which usually employs up to 4 antennas at both ends of the link, massive MIMO detection has to process data from tens of users received by hundreds of antennas. Hence, optimal maximum-likelihood (ML) detection like Sphere decoding is not applicable due to its exponential complexity increase in number of antennas [2]. Therefore, it has to be resorted to linear methods or to quasi-ML methods that build up on them [3].

Even with linear detection the actual implementation is not straightforward due to large number of antennas [4], [5]. Continuous technology downscaling is expected to allow implementation of such complex receiver designs. At the same time, this downscaling denies the assumption of deterministic hardware operation [6]. The underlying hardware is subjected to process variations due to radiation effects [7] or voltage over-scaling [8].

It has been shown that the traditional small-scale MIMO receivers are memory dominated, as different buffering memories occupy up to 50% of chip area [9]. The authors of [10] introduced the statistical model that combines the effect of memory errors and the channel noise. Based on this model a memory error resilient breadth-first tree search detector that represents the modification of the standard algorithm, taking the knowledge of memory errors into account, has been proposed [11].

In massive MIMO uplink this memory dominance is even more pronounced due to much larger system size. The developed modified algorithm in [11] is suited for traditional small-scale MIMO systems. Massive MIMO systems have not yet been addressed. The authors of [10], [11] allow a simplifying assumption of the receive vector having a single element affected by a single bit flip. This assumption holds in small-scale MIMO, but not in massive MIMO systems due to much larger memory sizes.

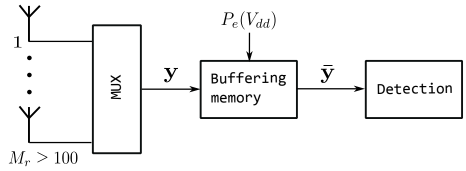

Fig. 1: MIMO receiver with faulty buffer memory

This paper introduces a novel memory error resilient NMMSE-based detection algorithm for memory dominated massive MIMO uplink. The algorithm builds up on the linear MMSE detection with moderate additional computational effort. The single-element single-bit flip assumption is relaxed, and the proposed solution targets the multiple-element single-flip scenario. The simulation results show that even for high memory error rates, the proposed algorithm is able to significantly reduce the impact of the memory errors on the overall receiver performance in terms of BER.

The rest of this paper is organized as follows: the statistical memory error model is outlined in Sec. II, Sec. III introduces the proposed memory error resilient detection algorithm, Sec. IV provides simulation results and Sec. V concludes the paper.

II. Memory error model

Consider the uplink of a massive MIMO system. The receiver side is equipped with $M_r \geq 100$ receive antennas. The real-valued system model is given as

$$\mathbf{y} = \mathbf{Hx} + \mathbf{n} \tag{1}$$

where $\mathbf{y}$ is the receive symbol vector, $\mathbf{x}$ is the transmit symbol vector, $\mathbf{H}$ is the channel matrix, assumed to be known at the receiver, and $\mathbf{n}$ is the AWGN vector with zero mean and covariance $\sigma_n^2\mathbf{I}$.

The receive symbol vector is stored in a buffering memory prior to detection as illustrated in Fig. 1. Since the faulty memory may introduce errors, the receive vector after the memory is denoted $\bar{\mathbf{y}}$. The authors in [10] introduced a statistical model that describes the distribution of $\bar{\mathbf{y}}$ after the faulty buffering memory.

The model proposed in [10] is reviewed briefly. The induced error due to voltage over-scaling is modeled as a spatially uniformly distributed random variable where the probability of failure, $P_e(V_{dd})$, for each cell in the memory is the same

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

for a fixed supply voltage, V_{dd}, and linearly increases in the logarithmic domain with reduction of the supply voltage [12].

The receive symbols are stored in the buffering memory in fixed point representation. The fixed point representation is defined by two parameters:

- d - length (number of bits) of integer part.
- f - length (number of bits) of fractional part.

The length l of the fixed point number is $l = d + f$. Therefore any bit stored in the buffering memory can be flipped with the same probability P_e. Conversely, for a fixed point number of length l, each bit position from 0 to $l-1$ can be flipped with probability P_e.

Hence, the probability to have k bits flipped is given by

$$p_k = \binom{l}{k} P_e^k (1-P_e)^{l-k} \tag{2}$$

The value of the error at a particular bit position j depends on the fixed point format $[d].[f]$ and the error free bit value y_i^j of i-th element of the receive symbol vector $\mathbf{y}$

$$e_i^j = \begin{cases} +2^{(j-f)} & \text{when } y_i^j = 0 \\ -2^{(j-f)} & \text{when } y_i^j = 1 \end{cases} \tag{3}$$

The pdf of the i-th element of the receive vector after the memory is in general given as

$$f(\bar{y}_i) = \sum_{k=0}^{l} p_k f(y_i, k) \tag{4}$$

The pdf $f(y_i, 0)$ defines the distribution of $\bar{y}_i$ when there is no bit flip, and in this case $\bar{y}_i = y_i$. The pdf $f(y_i, 1)$ defines the distribution of $\bar{y}_i$ when there is a single bit flip. This pdf is given as

$$f(y_i, 1) = \sum_{j=0}^{l-1} f^j(y_i, 1) \tag{5}$$

where $f^j(y_i, 1)$ is the pdf of $\bar{y}_i$ given the error at bit position j, where the error value is given in Eq. 3. In general, in case of k bit flips

$$f(y_i, k) = \frac{1}{k!} \sum_{j_1=0}^{l-1} \sum_{\substack{j_2=0,\\ j_2\neq j_1}}^{l-1} \cdots \sum_{\substack{j_k=0,\\ j_k\neq j_1,\\ \dots, j_k\neq j_{k-1}}}^{l-1} f^{j_1, j_2, \dots, j_k}(y_i, k) \tag{6}$$

As P_e is typically a low value, the probability to have multiple bit flips within same y_i is negligible. Hence, pdf of i-th element of receive vector after the memory $\bar{y}_i = y_i + e_i^j$, j = $0, \dots, l-1$ is

$$\begin{aligned} f(\bar{y}_i) &= p_0 f(y_i, 0) + p_1 f(y_i, 1) \\ &= p_0 f(y_i, 0) + p_1 \sum_{j=0}^{l-1} f^j(y_i, 1) \end{aligned} \tag{7}$$

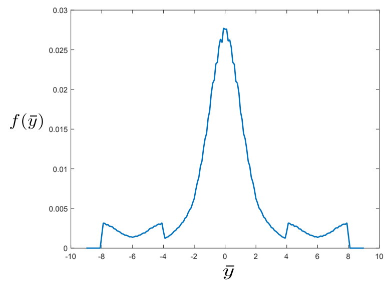

Fig. 2: pdf of receive vector element

where $p_0 = (1-P_e)^l$ and $p_1 = P_e(1-P_e)^{l-1}$.

III. Resilient Nonlinear Detection

As indicated in Sec. II, linear detection is attractive for massive MIMO uplink, since its complexity grows linearly with the number of antennas. Among linear schemes, linear MMSE (LMMSE) estimation achieves the best performance [13]. The LMMSE estimate is given as

$$\hat{\mathbf{x}} = \sigma_x^2 \mathbf{I} \mathbf{H}^T \left(\mathbf{H}^T \sigma_x^2 \mathbf{I} \mathbf{H} + \sigma_n^2 \mathbf{I} \right)^{-1} \mathbf{y} \tag{8}$$

where $\sigma_n^2 \mathbf{I}$ is the AWGN covariance matrix and $\sigma_x^2 \mathbf{I}$ is the transmit symbol vector covariance matrix.

The LMMSE estimate in Eq. 8 is optimal in minimum mean square sense when the transmit symbol vector $\mathbf{x}$ and the noise vector $\mathbf{n}$ are independent and Gaussian distributed [14]. Due to memory errors, this assumption does not hold anymore. Figure 2 depicts the distribution of an element of $\bar{\mathbf{y}}$ for $[4].[8]$ fixed point format. Clearly, this distribution is not Gaussian. The difference to Gaussian bell curve is most prominent at the tails of the depicted distribution. Therefore, in presence of memory errors, the LMMSE estimate is expected to produce a poor result.

It is known, that any distribution can be well approximated by a mixture of Gaussian distributions [15]. Hence, the framework of Gaussian mixture (GM) model can be used as a starting point for derivation of a suitable detection scheme that takes memory errors into account. The pdf of a Gaussian mixture distributed random vector $\mathbf{r}$ is given as [16]

$$f(\mathbf{r}) = \sum_{m=1}^{M} p_m N(\mu_{\mathbf{r}}^m, C_{\mathbf{rr}}^m) \tag{9}$$

where $\mu_{\mathbf{r}}^m$ is the mean of m-th component, $C_{\mathbf{rr}}^m$ is the covariance matrix of m-th component and p_m is the probability of drawing the m-th component from available M components. Obviously, $\sum_{m=1}^{M} p_m = 1$.

At first glance, it makes sense to regard the joint effect of AWGN and memory errors as a GM distributed noise vector $\bar{\mathbf{n}}$, with component covariance matrices $\sigma_{n^m}^2 \mathbf{I}$. The MMSE estimate for a Gaussian mixture is given as [17]

$$\hat{\mathbf{x}} = \sum_{m=1}^{M} \alpha_m(\mathbf{y}) \sigma_x^2 \mathbf{I} \mathbf{H}^T (\mathbf{H} \sigma_x^2 \mathbf{I} \mathbf{H}^T + \sigma_{n^m}^2 \mathbf{I})^{-1} \mathbf{y} \tag{10}$$

with coefficients $\alpha_m(\mathbf{y})$

$$\alpha_m(\mathbf{y}) = \frac{p_m \frac{1}{(\sqrt{2\pi})^{2M_r}|\mathbf{H}\sigma_x^2\mathbf{I}\mathbf{H}^T+\sigma_{nm}^2\mathbf{I}|^{\frac{1}{2}}} e^{-w_m}}{\sum_{m=1}^{M} p_m \frac{1}{(\sqrt{2\pi})^{2M_r}|\mathbf{H}\sigma_x^2\mathbf{I}\mathbf{H}^T+\sigma_{nm}^2\mathbf{I}|^{\frac{1}{2}}} e^{-w_m}} \tag{11}$$

where the exponents w_m are given as

$$w_m = \frac{1}{2}\mathbf{y}^T(\mathbf{H}\sigma_x^2\mathbf{I}\mathbf{H}^T + \sigma_{nm}^2\mathbf{I})^{-1}\mathbf{y} \tag{12}$$

It is a nonlinear MMSE estimate (NMMSE), since the coefficients $\alpha_m(\mathbf{y})$ are non-linear in $\mathbf{y}$. These coefficients can be regarded as the probability that the noise is drawn from component pdf m, given the receive vector $\mathbf{y}$. It is to observe that if there is only one noise component, $M = 1$, the coefficient $\alpha_1(\mathbf{y})$ equals one and the NMMSE estimate coincides with the LMMSE estimate in Eq. 8.

The general NMMSE estimator is next adapted to mitigate the effect of the memory errors. First, the attention is restricted to single bit flips within the integer part, as they cause the largest error values. In this case the mixture component probabilities p_m are: the probability to have no memory errors $p_{(k=0)}$ and d equal probabilities $p_{(k=1)}$ to have a single bit flip at one of the integer bit positions j, $j \in \{l-1, \dots, f\}$. Therefore, there would be $d+1$ noise components: nominal channel AWGN noise $\mathbf{n}$ in case there are no memory errors and d components where each noise term includes the bit flip at bit position j, $j \in \{l-1, \dots, f\}$.

The problem with this approach is that according to memory error model in Sec. II, the value of e_i^j actually depends on value of y_i^j. Thus, it is not possible to separate out e_i^j from $\bar{y}_i$ and form $\bar{n}_i = n_i + e_i^j$. Therefore, the same channel AWGN noise vector with covariance matrix $\sigma_n^2\mathbf{I}$ is present in all $d+1$ mixture components. The first component assumes the vector y to be memory error free, while remaining components assume that some elements of receive vector $\mathbf{y}$ are affected by bit flips at position j,

$$\bar{\mathbf{y}}^j = \mathbf{y} + e^j\mathbf{u} \tag{13}$$

where e^j are the error values given in Eq. 3. Vector $\mathbf{u}$ contains ones at i-th position for which the bit flip at bit position j occurs and zero otherwise. The simplifying assumption of having single-bit flip in just a single element i of the receive vector taken in [11] cannot be fulfilled in massive MIMO uplink due to large memory size required. Therefore, the receive vector $\bar{\mathbf{y}}$ may contain multiple elements i affected by a single bit flip at position j, $j \in \{l-1, \dots, f\}$. This implies that vector $\mathbf{u}$ may contain multiple ones.

Finally the mixture contains: receive vector $\mathbf{y}$ with no memory errors with component probability $p_{(k=0)}$ and d components where receive vector $\mathbf{y}^j$ has a bit flip at position $j, j \in \{l-1, \dots, f\}$ at some elements i with same component probability $p_{(k=1)}$. Since possible error values are known, the positions i of error-affected vector elements can be identified one by one. When an element of the receive vector contains a flip in the integer part, the pdf of $\bar{\mathbf{y}}$ will deviate from the nominal multivariate Gaussian distribution given in the numerator of the NMMSE coefficient $\alpha_{(m=1)}(\bar{\mathbf{y}})$ in Eq. 11. The exponent $w_{(m=1)}$ given in Eq. 12 will assume a large value and the coefficient $\alpha_1(\bar{\mathbf{y}})$ will be close to zero.

The receiver has the potentially error-affected vector at its disposal, but does not know whether the error has occurred, nor the affected element(s) of the vector. Therefore, for each possible bit flip position j (corresponding to mixture components, $m > 1$), the single-bit flip correction values β_i^j are computed for all i elements of the receive vector. Since the value of the assumed bit flip depends on the value of the receive vector, the correction values are obtained as

$$\beta_i^j = \begin{cases} (2\bar{y}_i^j - 1)2^{(j-f)} & \text{when } j = l-1 \\ (-2\bar{y}_i^j + 1)2^{(j-f)} & \text{when } j \neq l-1 \end{cases} \tag{14}$$

The exponents w_i^j are obtained with applied single-bit corrections

$$w_i^j = \frac{1}{2}(\bar{\mathbf{y}} + \beta_i^j\mathbf{u}_i)^T(\mathbf{H}\sigma_x^2\mathbf{I}\mathbf{H}^T + \sigma_n^2\mathbf{I})^{-1}(\bar{\mathbf{y}} + \beta_i^j\mathbf{u}_i) \tag{15}$$

where $\mathbf{u}_i$ is the unit vector containing one at position i and zeros at other positions. When the applied single-bit correction indeed corrects the error, the deviation from the nominal Gaussian density would decrease and the respective w_i^j will decrease too. Otherwise, the correction actually introduces an error, forcing respective w_i^j to increase. By finding the minimal exponent w_i^j for a given $j, j \in \{l-1, \dots, f\}$, the index i of the receive vector element where the bit flip at position j occurred is identified.

Next, the coefficients $\alpha_m(\bar{\mathbf{y}})$ are obtained with minimal exponent values. It is to note that as the division by the sum of numerators in Eq. 11 is just scaling, the numerator of coefficient and the coefficient itself will be used interchangeably. The direct computation of coefficients by Eq. 11 reveals a numeric instability that is caused by the multiple errors within the receive vector.

Assume that $\bar{\mathbf{y}}$ contains two bit flips: one at $(l-1)$-th bit position in element s and the other at $(l-2)$-th bit position in element z. The minimum exponent value for $(l-1)$-th bit flip position will be obtained with the bit flip corrected, however, due to the remaining bit flip in element z, the value of $w_{min}^{(l-1)}$ may still be large. This would result in the zero coefficient for $(l-1)$-th bit flip position. The same would happen for $w_{min}^{(l-2)}$, due to influence of bit flip in element s. This way all coefficients would be computed to zero, rendering the algorithm useless. This problem is avoided by transforming Eq. 11 to log domain.

$$\begin{aligned} \alpha_m(\bar{\mathbf{y}}) = {} & \log(p_m) - \\ & - \log\left((\sqrt{2\pi})^{2M_r}|\mathbf{H}\sigma_x^2\mathbf{I}\mathbf{H}^T + \sigma_n^2\mathbf{I}|^{\frac{1}{2}}\right) - w_{min}^j \end{aligned} \tag{16}$$

The maximum coefficient identifies the single error that caused the maximal deviation from the nominal Gaussian pdf.

The value of this error is given by the maximal coefficient index m that corresponds to the bit flip position j. The vector element i where this error has occurred is given by the index for which the minimal exponent was obtained for the maximal coefficient.

This way the correction is performed iteratively starting from the memory error which caused the largest deviation and correcting one error per iteration. The algorithm is summarized in Alg. 1. Its detailed function is outlined next.

- The algorithm starts on the potentially memory error affected receive vector.
- Next, the main loop is executed until maximum number of iterations m_{it} is reached.
- Within an iteration, the bit values $\bar{y}_i^j$ of all integer bit positions of all receive vector elements are extracted by BITGET function in line 6 of Alg. 1.
- For these bit values the correction values are computed by Eq. 14, assuming that a single flip occurred at exactly this position.
- Next, exponents w_i^j by Eq. 15 are computed with correction values obtained previously.
- For each integer bit position j, minimum exponent is obtained. Its index s identifies the receive vector element with assumed error at bit position j. Recall that the j-th bit flip position corresponds to the mixture component $m > 1$. For $m = 1$, minimization is not required, since $\bar{\mathbf{y}}$ is assumed memory error free and $w_{(m=1)}$ is obtained by Eq. 12.
- Component coefficients $\alpha_m(\bar{\mathbf{y}})$ with obtained minimum exponents w_{min}^j and $w_{(m=1)}$ in log domain are computed by Eq. 16.
- The maximum component coefficient is obtained. Its index identifies the bit position of the error. If $\alpha_{max}(\bar{\mathbf{y}})$ corresponds to $m = 1$, the algorithm is terminated as all errors are assumed to have been corrected and LMMSE estimate of the current $\bar{\mathbf{y}}^{(it)}$ is output:

$$\hat{\mathbf{x}} = \sigma_x^2 \mathbf{I}\mathbf{H}^T (\mathbf{H}\sigma_x^2 \mathbf{I}\mathbf{H}^T + \sigma_n^2 \mathbf{I})^{-1} \bar{\mathbf{y}}^{(it)} \tag{17}$$

  Otherwise, the correction value β_s^z corresponding to $\alpha_{max}(\bar{\mathbf{y}})$, where z indicates the bit flip position within the s-th vector element where the memory error has occurred, is applied to $\bar{\mathbf{y}}$

$$\bar{\mathbf{y}}^{(it+1)} = \bar{\mathbf{y}}^{(it)} + \beta_s^z \tag{18}$$

- The subsequent iteration is started with $\bar{\mathbf{y}}^{(it+1)}$
- In the last iteration, the LMMSE estimate with the last found correction value is output as the final solution

$$\hat{\mathbf{x}} = \sigma_x^2 \mathbf{I}\mathbf{H}^T (\mathbf{H}\sigma_x^2 \mathbf{I}\mathbf{H}^T + \sigma_n^2 \mathbf{I})^{-1} (\bar{\mathbf{y}}^{(m_{it})} + \beta_s^z \mathbf{u}_s) \tag{19}$$

Algorithm 1 Iterative log-NMMSE

1: Input: $\bar{\mathbf{y}}$, $\mathbf{H}$, P_e, m_{it}. Output: $\hat{\mathbf{x}}$.
2: $\bar{\mathbf{y}}^{(it=1)} = \bar{\mathbf{y}}$ ▷ Initialization
3: **for** $it = 1, \dots, m_{it}$ **do** ▷ maximum number of iterations
4: **for** $j = f, \dots, l-1$ **do** ▷ integer bit positions
5: **for** $i = 1, \dots, 2M_r$ **do** ▷ receive vector elements
6: $\bar{y}_i^j$ =BITGET$(\bar{\mathbf{y}}, j)$ ▷ obtain bit value
7: **end for**
8: **end for**
9: **for** $j = f, \dots, l-1$ **do**
10: **for** $i = 1, \dots, 2M_r$ **do**
11: $\beta_i^j \leftarrow$ (Eq. 14) ▷ compute correction value
12: **end for**
13: **end for**
14: **for** $j = f, \dots, l-1$ **do**
15: **for** $i = 1, \dots, 2M_r$ **do**
16: $w_i^j \leftarrow$ (Eq. 15) ▷ try out correction values
17: **end for**
18: **end for**
19: **for all** j **do**
20: $w_{min}^j \leftarrow \operatorname*{argmin}_i w_i^j$ ▷ obtain erroneous vector element
21: **end for**
22: **for** $m = 1, \dots, d+1$ **do**
23: $\alpha_m(\bar{\mathbf{y}}) \leftarrow$ (Eq. 16)
24: **end for**
25: $\alpha_{max}(\bar{\mathbf{y}}) \leftarrow \operatorname*{argmax}_m \alpha_m(\bar{\mathbf{y}})$ ▷ identify bit position of the error
26: **if** $it \neq m_{it}$ **then**
27: **if** $m = 1$ **then**
28: $\hat{\mathbf{x}} \leftarrow$ (Eq. 17) ▷ no correction is necessary
29: **else**
30: $\alpha_{max}(\bar{\mathbf{y}}) \rightarrow \beta_s^z$ ▷ pick correction value
31: $\bar{\mathbf{y}}^{(it+1)} \leftarrow$ (Eq. 18) ▷ correct found error before next iteration
32: **end if**
33: **else**
34: $\alpha_{max}(\bar{\mathbf{y}}) \rightarrow \beta_s^z$ ▷ pick correction value
35: $\hat{\mathbf{x}} \leftarrow$ (Eq. 19) ▷ output LMMSE estimate with final correction
36: **end if**
37: **end for**

IV. Results

The simulation is performed for bit flip probabilities of $P_e = 10^{-4}$ and $P_e = 10^{-3}$. The fixed point format is [6].[8]. Therefore, the possible error values are $\{\pm 2^5, \pm 2^4, \pm 2^3, \pm 2^2, \pm 2^1, \pm 2^0\}$ with probability $p_{(k=1)} = P_e(1-P_e)^5$. The channel matrix $\mathbf{H}$ is assumed constant for the duration of 100 receive symbol vectors. The uplink massive MIMO system employs $M_r = 256$ receive antennas. Modulation is 4-QAM with Gray mapping. The information bits are encoded by (133, 171) convolutional code with constraint length 7 and code rate $1/2$. At the receiver side, the detected bits are decoded by soft decision Viterbi decoder. The BER performance is depicted in Fig. 3, 4 for $P_e = 10^{-3}$ and $P_e = 10^{-4}$. The LMMSE curve for the case of no memory errors ($P_e = 0$) is plotted for reference and highlighted in blue. Both plots indicate the drastic BER degradation

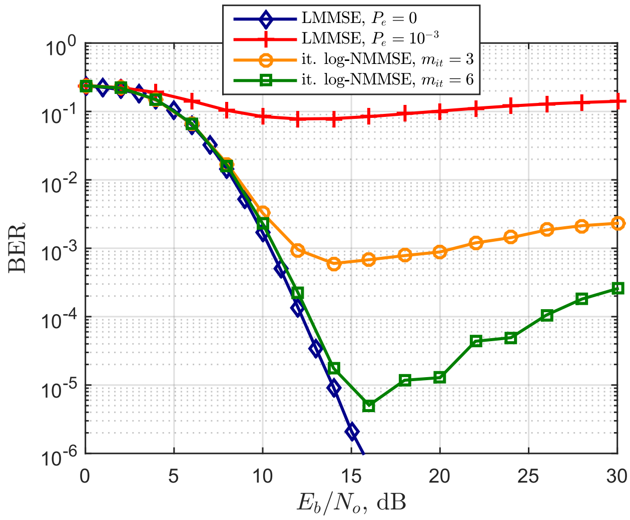

Fig. 3: BER performance, $P_e = 10^{-3}$

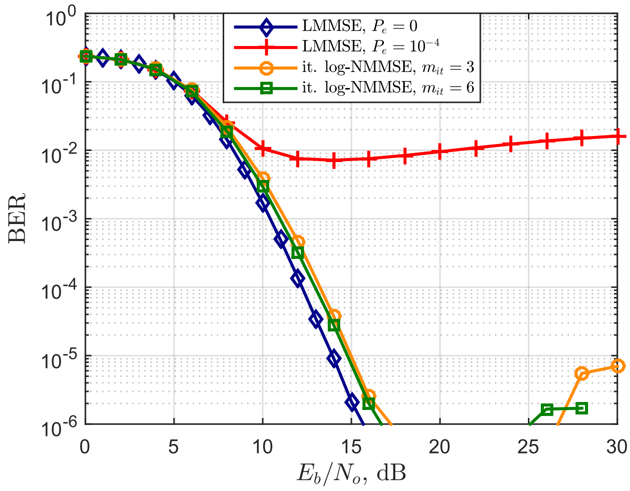

Fig. 4: BER performance, $P_e = 10^{-4}$

with LMMSE in case of memory errors (red curves). With $P_e = 10^{-3}$, proposed NMMSE-based method is able to lower the error floor by three orders of magnitude, performing six iterations $m_{it} = 6$. With $P_e = 10^{-4}$, performance close to the memory error-free case is achieved already with $m_{it} = 3$. The introduced computational overhead scales linearly in the number of receive antennas M_r, since the additional operations are solely matrix-vector multiplications. The run-time of the proposed algorithm is 2.5 times that of the standard linear detection, with comparison performed on the same host.

V. Conclusion

This paper proposed a memory error resilient detection algorithm for massive MIMO uplink. The simulation results demonstrate that the proposed solution significantly reduces the impact of the multiple memory errors on the BER. In future work it would be relevant to investigate how to tune the maximum number of iterations, such that the optimum improvement is achieved with less computational effort, depending on the value of bit flip probability P_e and E_b/N_0.

References

[1] E. Larsson, O. Edfors, F. Tufvesson, and T. Marzetta, “Massive MIMO for next generation wireless systems,” *Communications Magazine, IEEE*, vol. 52, no. 2, pp. 186–195, February 2014.

[2] A. Chockalingam and B. S. Rajan, *Large MIMO Systems*. Cambridge University Press, 2014, cambridge Books Online. [Online]. Available: http://dx.doi.org/10.1017/CBO9781139208437

[3] A. Chockalingam, “Low-complexity algorithms for large-MIMO detection,” in *Communications, Control and Signal Processing (ISCCSP), 2010 4th International Symposium on*, March 2010, pp. 1–6.

[4] M. Wu, B. Yin, A. Vosoughi, C. Studer, J. Cavallaro, and C. Dick, “Approximate matrix inversion for high-throughput data detection in the large-scale MIMO uplink,” in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, May 2013, pp. 2155–2158.

[5] M. Wu, B. Yin, G. Wang, C. Dick, J. Cavallaro, and C. Studer, “Large-Scale MIMO Detection for 3GPP LTE: Algorithms and FPGA Implementations,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 8, no. 5, pp. 916–929, Oct 2014.

[6] G. Karakonstantis, N. Banerjee, and K. Roy, “Process-variation resilient and voltage-scalable dct architecture for robust low-power computing,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 18, no. 10, pp. 1461–1470, Oct 2010.

[7] P. Dodd and L. Massengill, “Basic mechanisms and modeling of single-event upset in digital microelectronics,” *Nuclear Science, IEEE Transactions on*, vol. 50, no. 3, pp. 583–602, June 2003.

[8] R. Hegde and N. Shanbhag, “A voltage overscaled low-power digital filter IC,” *Solid-State Circuits, IEEE Journal of*, vol. 39, no. 2, pp. 388–391, Feb 2004.

[9] F. Clermidy, C. Bernard, R. Lemaire, J. Martin, I. Miro-Panades, Y. Thonnart, P. Vivet, and N. Wehn, “A 477mW NoC-based digital baseband for MIMO 4G SDR,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, Feb 2010, pp. 278–279.

[10] A. Khajeh, K. Amiri, M. Khairy, A. Eltawil, and F. Kurdahi, “A Unified Hardware and Channel Noise Model for Communication Systems,” in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, Dec 2010, pp. 1–5.

[11] M. Khairy, C.-A. Shen, A. Eltawil, and F. Kurdahi, “Error resilient MIMO detector for memory-dominated wireless communication systems,” in *Global Communications Conference (GLOBECOM), 2012 IEEE*, Dec 2012, pp. 3566–3571.

[12] K. Agarwal and S. Nassif, “The Impact of Random Device Variation on SRAM Cell Stability in Sub-90-nm CMOS Technologies,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 16, no. 1, pp. 86–97, Jan 2008.

[13] C. Michalke, E. Zimmermann, and G. Fettweis, “Linear Mimo Receivers vs. Tree Search Detection: A Performance Comparison Overview,” in *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, Sept 2006, pp. 1–7.

[14] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[15] J. Q. Li and A. R. Barron, “Mixture Density Estimation,” in *IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 12*. MIT Press, 1999, pp. 279–285.

[16] J. Flam, S. Chatterjee, K. Kansanen, and T. Ekman, “On MMSE Estimation: A Linear Model Under Gaussian Mixture Statistics,” *Signal Processing, IEEE Transactions on*, vol. 60, no. 7, pp. 3840–3845, July 2012.

[17] A. Kundu, S. Chatterjee, A. Sreenivasa Murthy, and T. Sreenivas, “GMM based Bayesian approach to speech enhancement in signal / transform domain,” in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, March 2008, pp. 4893–4896.