Study of Bilinear Generalized Approximate Message Passing by Applying it to a MIMO Model.

Georgios Vougioukas, A.M: 2016039002

I. Introduction

This project aimed in better understanding message passing algorithms in general by reproducing results of [1]. Due to the difficulty arising from the problem's nature (i.e., continuous distributions, loopy graph), approximate techniques applied on the message passing procedure, were also studied [2]. A multiple in-multiple out (MIMO) system, following the model of Eq. (1), is considered and the goal of the reproduced algorithm is to estimate all the channel parameters (matrix $\mathbf{H}$) and jointly detect the transmitted data (matrix $\mathbf{X}$), utilizing the minimum mean squared error (MMSE) criterion. In contrast to the conventional method of first estimating the channel parameters using a known training sequence and then performing detection using the channel's estimate, the presented joint channel and data estimation (JCD) algorithm, which is based on bilinear generalized approximate message passing (Bi-GAMP) [2], simultaneously estimates and detects the channel and symbol matrices respectively.

Notation: $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{\mathrm{H}}$ denote the transpose and the conjugate transpose (Hermitian) of the argument. Simple lowercase letters denote scalars, bold lowercase letter denote vectors and bold upper case letters denote matrices. Scalar symbols in the form $x_{k,t}$ denote the element of matrix $\mathbf{X}$ located at row-k, column-t. Symbols in the form $\nu_{N,K}(\lambda)$ denote the matrix formed at iteration λ, by elements $\nu_{n,k}(\lambda)$ for $n = 1 \dots N,\ k = 1 \dots K$. Operators $\odot$, $\div$ denote the element-wise multiplication and division respectively.

II. Signal Model

As stated earlier, a MIMO system is considered as illustrated in Fig. 1. The receiver is equipped with N antennas and is capable of serving K, single antenna, users. Each user transmits in total $T = T_{\mathrm{t}} + T_{\mathrm{d}}$ symbols, of which T_{t} are used as a training sequence (the sequence is available to the receiver/base station) and the rest T_{d} are information bearing symbols. The signal model can then be expressed as:

$$\mathbf{Y} = \frac{1}{\sqrt{K}}\mathbf{H}\mathbf{X} + \mathbf{W}, \tag{1}$$

where $\mathbf{Y} \in \mathbb{C}^{N\times T}$ is the complex baseband equivalent of the received signal, $\mathbf{X} \in \mathbb{C}^{K\times T}$ contains the user-transmitted symbols for (K rows of T symbols each) a time duration of T and $\mathbf{H} \in \mathbb{C}^{N\times K}$ contains the coupling/channel parameters between the K users and each of the N antennas (K columns of N rows each). $\mathbf{W}$ represents the zero mean additive white Gaussian noise (AWGN). Each element $W_{n,t}$ of $\mathbf{W}$ is assumed such that $W_{n,t} \sim \mathcal{CN}\big(0, \sigma_w^2\big)$. In the rest of the report, the product $\frac{1}{\sqrt{K}}\mathbf{H}\mathbf{X}$ may appear as $\mathbf{Z} \triangleq \frac{1}{\sqrt{K}}\mathbf{H}\mathbf{X}$.

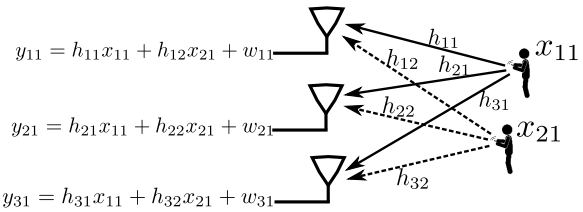

Fig. 1. Representation of the considered system model as per Eq. (1) for $N = 3$, $K = 2$ and $T = 1$.

III. Algorithm Foundation

The goal of the algorithm is to find the posterior distributions $p(x_{k,t}|\mathbf{Y})\ \forall\, k, t$ and $p(h_{n,k}|\mathbf{Y})\ \forall\, n, k$. Then, computing the expected values of those distributions results to the MMSE estimates of $x_{k,t}$ and $h_{n,n}$ respectively [3, pp. 316]. Based on the signal model on Sec. II, the statistical model for the posterior of the system can be expressed as:

$$\begin{aligned} p(\mathbf{X}, \mathbf{H}|\mathbf{Y}) \propto & p(\mathbf{Y}|\mathbf{Z} = \mathbf{H}\mathbf{X})\, p(\mathbf{X})\, p(\mathbf{H}) \\ &= \left[\prod_{n=1}^{N}\prod_{t=1}^{T} p(y_{n,t}|z_{n,t} = \mathbf{h}_{n,1:K}\mathbf{x}_{1:K,t})\right] \\ &\times \left[\prod_{k=1}^{K}\prod_{t=1}^{T} p(x_{k,t})\right]\left[\prod_{n=1}^{N}\prod_{k=1}^{K} p(h_{n,k})\right]. \end{aligned} \tag{2}$$

The above factorization is made due to (by formulation) independence of the involved random variables and is represented in Fig. 2 using a factor graph. The factors appear as square boxes while random variables as circles. Using Eq. (2), when a random variable is involved in a factor, an edge is used to connect the variable's node (circle) with the factor's node (box).

The factor graph of Fig. 2 is the basis for running the loopy sum-product message passing algorithm which will eventually result to the aforementioned posteriors. Exchange of messages between factor nodes and variable nodes results to those posteriors:

- The message transmitted from a variable node towards a given edge (towards a factor), is the product of all the incoming messages to that node, except the one coming from the recipient of the message.
- A message transmitted from a factor node towards a given edge (towards a variable) is defined as the integral of the

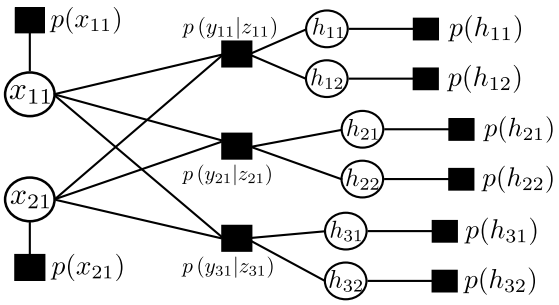

Fig. 2. Graphical representation of factorization in Eq. (2). The representation is made for $N = 3$, $K = 2$ and $T = 1$.

product of the factor and all incoming messages from other nodes, except the incoming from the recipient.

It should be noted that message exchange is performed only between a variable and a factor. The posterior for a given variable is the product of all the incoming messages. The message passing procedure is executed in an iterative manner (messages are continuously updated) until a convergence criterion is satisfied.

In the studied works [1], [2], the messages take the form of log-pdfs with arbitrary constant offsets. For example, based on Fig. 2, the message from factor node $p(y_{11}|z_{11})$ impinging on variable x_{11} at iteration ξ is defined as:

$$\mu^{(\xi)}_{p_{11}\to x_{11}}(x_{11}) = \log \int_{h_{11},h_{12},x_{21}} \overbrace{p(y_{11}|z_{11}=h_{11}x_{11}+h_{12}x_{21})}^{\text{factor}} \times \underbrace{\mathrm{e}^{\mu^{(\xi)}_{p_{11}\leftarrow x_{21}}(x_{21})+\mu^{(\xi)}_{p_{11}\leftarrow h_{11}}(h_{11})+\mu^{(\xi)}_{p_{11}\leftarrow h_{12}}(h_{12})}}_{\text{incoming messages}} + \mathrm{c}, \tag{3}$$

where c arbitrary constant introduced due to the modeling of messages. As it can be seen, when the number of variables involved is large, integration could be challenging. To overcome this problem, approximate expressions are used for evaluating the messages (and of course updating them) based on central limit theorem (CLT) and Taylor series expansion. The above approximations result to Gaussian modeling of the messages and tractable expressions for update rules, involving just the first two moments (mean and variance).

Executing this approximate message passing procedure results to breaking the initial problem of finding the posterior means (i.e., the MMSE estimates) of $\mathbf{H}, \mathbf{X}$, to two Gaussian scalar estimation problems (each scalar problem corresponds to each element of $\mathbf{X}, \mathbf{H}$ to be estimated) with a trivial solution. At each iteration, the algorithm emulates the observation of each element $h_{n,k} \in \mathbf{H}$ as though it was observed through a Gaussian channel:

$$\hat{q}_{n,k} = h_{n,k} + n^h_{n,k}, \tag{4}$$

where $n^h_{n,k} \sim \mathcal{CN}\left(0, \nu^q_{n,k}\right)$. $h_{n,k}$ is assumed to be a circularly symmetric complex normal r.v, i.e. $h_{n,k} \sim \mathcal{CN}\left(0, \sigma^2_h\right)$. That way $\hat{q}_{n,k}$ is also a complex normal random variable with $\hat{q}_{n,k} \sim \mathcal{CN}\left(0, \sigma^2_h + \nu^q_{n,k}\right)$. Having stated the above, the bayes-optimal MMSE estimator (which is defined as the posterior mean $\mathbb{E}[h_{n,k} \mid \hat{q}_{n,k}]$) coincides with the linear MMSE estimator [3, pp.350] and the MMSE estimate of $h_{n,k}$ can be acquired by:

$$\hat{h_{n,k}} = \frac{\sigma_{\hat{q}h}}{\sigma^2_h + \nu^q_{n,k}}\hat{q}_{n,k} = \frac{\sigma^2_h}{\sigma^2_h + \nu^q_{n,k}}\hat{q}_{n,k}. \tag{5}$$

In a similar manner, each $x_{k,t} \in \mathbf{X}$, during each iteration is considered to be observed through another gaussian channel:

$$\hat{r}_{k,t} = x_{k,t} + n^x_{k,t}, \tag{6}$$

where $n^x_{k,t} \sim \mathcal{CN}\left(0, \nu^r_{n,k}\right)$. If the symbols are equiprobable and drawn from a 4-QAM constellation the detector can be shown to take the form [4]:

$$\hat{x_{k,t}} = \frac{1}{\sqrt{2}}\tanh\left(\frac{2\Re\{\hat{r}_{k,t}\}}{\sqrt{2}\nu^r_{k,t}}\right) + \mathrm{j}\frac{1}{\sqrt{2}}\tanh\left(\frac{2\Im\{\hat{r}_{k,t}\}}{\sqrt{2}\nu^r_{k,t}}\right). \tag{7}$$

At each iteration, the message passing procedure updates the parameters of the above equations which result to new estimates used in the next iteration, until a convergence criterion is satisfied or a maximum iteration number has been reached.

A. Description

The algorithm implemented is the one presented in [1]. This algorithm is based on the results of [2] appropriately modified to fit the MIMO model as described in Sec. II. A notable difference between the two, is the fact that a part of matrix $\mathbf{X}$ is known (training sequence). Due to the approximations described earlier, every calculation involves just matrix multiplications. In lines 9, 17 & 19 the MMSE estimates of $\mathbf{Z}$, $\mathbf{X}$ & $\mathbf{H}$ can be seen respectively.

Although the implementation, given the pseudo-code, may seem trivial, there is a number of problems when someone tries to reproduce the results of the work. In order to make it work, for the purposes of the project, a small number of modifications were made to the algorithm of [1]. These modifications are listed below.

B. Modifications

1) Variable Initialization: To initialize the variables, authors of [1] suggest to assign zero values to every involved variable except of course those who are related to or represent variances, which are assigned to value 1. Some of the variables initialized with the value 0, are part of denominators in fractions. When the simulation software tries to calculate those fractions, something will probably go wrong. And in fact that happens, simulator throws NaN (not a number) values and the algorithm just does not work. To overcome this problem, the respective variables are initialized to 10^{-5} instead of 0.

Variables denoted as ν are initialized with value 1. Every other variable is initialized to 10^{-5} except the known part of $\hat{\mathbf{X}}$ which is initialized with the training sequence.

Algorithm 1 (slightly) Modified GAMP-based JCD

Input: $\mathbf{Y}$, priors $p(\mathbf{H})$ and $p(\mathbf{X})$, training sequence $\mathbf{X}_{T_\mathrm{t}}$

1: **for** $\lambda = 1 : \lambda_{max}$ **do**

2: $\nu^p_{N,T_\mathrm{t}}(\lambda) = \nu^h_{N,K}(\lambda)\,|\mathbf{X}_{T_\mathrm{t}}|^2$;

3: $\hat{p}_{N,T_\mathrm{t}}(\lambda) = \hat{\mathbf{H}}(\lambda)\,\mathbf{X}_{T_\mathrm{t}} - \hat{s}_{N,T_\mathrm{t}}(\lambda-1)\odot\nu^p_{N,T_\mathrm{t}}(\lambda)$;

4: $\bar{\nu}^p_{N,T_\mathrm{d}}(\lambda) = |\hat{\mathbf{H}}(\lambda)|^2\nu^x_{K,T_\mathrm{d}}(\lambda) + \nu^h_{N,K}(\lambda)\,|\hat{\mathbf{X}}_{T_\mathrm{d}}(\lambda)|^2$;

5: $\bar{p}_{N,T_\mathrm{d}}(\lambda) = \hat{\mathbf{H}}(\lambda)\,\hat{\mathbf{X}}_{T_\mathrm{d}}(\lambda)$;

6: $\nu^p_{N,T_\mathrm{d}}(\lambda) = \bar{\nu}^p_{N,T_\mathrm{d}}(\lambda) + \nu^h_{N,K}(\lambda)\,\nu^x_{K,T_\mathrm{d}}(\lambda)$;

7: $\hat{p}_{N,T_\mathrm{d}}(\lambda) = \bar{p}_{N,T_\mathrm{d}}(\lambda) - \hat{s}_{N,T_\mathrm{d}}(\lambda-1)\odot\bar{\nu}^p_{N,T_\mathrm{d}}(\lambda)$;

8: $\nu^z_{N,T}(\lambda) = \mathrm{Var}\Big\{\mathbf{Z}\,|\,\hat{p}_{N,T}(\lambda)\,,\nu^p_{N,T}(\lambda)\Big\}$;

9: $\hat{\mathbf{Z}}(\lambda) = \mathbb{E}\Big\{\mathbf{Z}\,|\,\hat{p}_{N,T}(\lambda)\,,\nu^p_{N,T}(\lambda)\Big\}$;

10: $\nu^s_{N,T}(\lambda) = \Big(1-\nu^z_{N,T}(\lambda)\div\nu^p_{N,T}(\lambda)\Big)\div\nu^p_{N,T}(\lambda)$;

11: $\hat{s}_{N,T}(\lambda) = \Big(\hat{\mathbf{Z}}(\lambda) - \hat{p}_{N,T}(\lambda)\Big)\div\nu^p_{N,T}(\lambda)$;

12: $\nu^r_{K,T}(\lambda) = 1 \div \Big(|\hat{\mathbf{H}}(\lambda)^\mathrm{T}|^2\nu^s_{N,T}(\lambda)\Big)$;

13: $\hat{r}_{K,T}(\lambda) = \hat{\mathbf{X}}(\lambda)\odot(1-\nu^r_{K,T}(\lambda)\odot(\nu^h_{N,K}(\lambda)^\mathrm{T}\nu^s_{N,T}(\lambda))) + \nu^r_{K,T}(\lambda)\odot\Big(\hat{\mathbf{H}}(\lambda)^\mathrm{H}\,\hat{s}_{N,T}(\lambda)\Big)$;

14: $\nu^q_{N,K}(\lambda) = 1 \div (\nu^s_{N,T_\mathrm{t}}(\lambda)\,|\mathbf{X}^\mathrm{T}_{T_\mathrm{t}}|^2 + \nu^s_{N,T_\mathrm{d}}(\lambda)\,|\hat{\mathbf{X}}^\mathrm{T}_{T_\mathrm{d}}|^2)$;

15: $\hat{q}_{N,K}(\lambda) = \hat{\mathbf{H}}(\lambda)\odot(1-\nu^q_{N,K}(\lambda)\odot(\nu^s_{N,T_\mathrm{d}}(\lambda)\nu^x_{K,T_\mathrm{d}}(\lambda)^\mathrm{T})) + \nu^q_{N,K}(\lambda)\odot\Big(\hat{s}_{N,T_\mathrm{t}}(\lambda)\,\mathbf{X}^\mathrm{H}_{T_\mathrm{t}} + \hat{s}_{N,T_\mathrm{d}}(\lambda)\,\hat{\mathbf{X}}^\mathrm{H}_{T_\mathrm{d}}(\lambda)\Big)$;

16: $\nu^x_{K,T_\mathrm{d}}(\lambda+1) = \mathrm{Var}\big\{\mathbf{X}_{T_\mathrm{d}}\,|\,\hat{r}_{K,T_\mathrm{d}}(\lambda)\,,\nu^r_{K,T_\mathrm{d}}(\lambda)\big\}$;

17: $\hat{\mathbf{X}}_{T_\mathrm{d}}(\lambda+1) = \mathbb{E}\big\{\mathbf{X}_{K,T_\mathrm{d}}\,|\,\hat{r}_{K,T_\mathrm{d}}(\lambda)\,,\nu^r_{K,T_\mathrm{d}}(\lambda)\big\}$;

18: $\nu^h_{N,K}(\lambda+1) = \mathrm{Var}\Big\{\mathbf{H}\,|\,\hat{q}_{N,K}(\lambda)\,,\nu^q_{N,K}(\lambda)\Big\}$;

19: $\hat{\mathbf{H}}(\lambda+1) = \mathbb{E}\Big\{\mathbf{H}\,|\,\hat{q}_{N,K}(\lambda)\,,\nu^q_{N,K}(\lambda)\Big\}$;

20: $\epsilon(\lambda) = ||\bar{p}_{N,T}(\lambda) - \bar{p}_{N,T}(\lambda-1)\,||^2_\mathrm{F}/||\bar{p}_{N,T}(\lambda)\,||^2_\mathrm{F}$

21: **end for**

22: $\lambda^* = \underset{\lambda}{\mathrm{argmin}}\,\epsilon(\lambda)$

Output: $\hat{\mathbf{H}}(\lambda^*)$, $\hat{\mathbf{X}}(\lambda^*)$

2) Damping: After several unsuccessful attempts of trying to run the algorithm in order to offer acceptable performance, damping as shown in [2, Sec.IV-A] and [4] was considered. Damping is a method for preventing specific variables from attaining large values. The following changes were applied the referenced lines of algorithm 1:

$$\text{Line 2: } \nu^p_{N,T_\mathrm{t}}(\lambda) = (1-\beta)\,\nu^p_{N,T_\mathrm{t}}(\lambda-1) + \beta\nu^h_{N,K}(\lambda)\,|\mathbf{X}_{T_\mathrm{t}}|^2 \tag{8}$$

$$\text{Line 4: } \bar{\nu}^p_{N,T_\mathrm{d}}(\lambda) = (1-\beta)\,\bar{\nu}^p_{N,T_\mathrm{d}}(\lambda-1) + \beta\Big(|\hat{\mathbf{H}}(\lambda)\,|^2\nu^x_{K,T_\mathrm{d}}(\lambda) + \nu^h_{N,K}(\lambda)\,|\hat{\mathbf{X}}_{T_\mathrm{d}}(\lambda)\,|^2\Big) \tag{9}$$

$$\text{Line 6: } \nu^p_{N,T_\mathrm{d}}(\lambda) = (1-\beta)\,\nu^p_{N,T_\mathrm{d}}(\lambda-1) + \beta\Big(\bar{\nu}^p_{N,T_\mathrm{d}}(\lambda) + \nu^h_{N,K}(\lambda)\,\nu^x_{K,T_\mathrm{d}}(\lambda)\Big) \tag{10}$$

$$\text{Line 10: } \nu^s_{N,T}(\lambda) = (1-\beta)\,\nu^s_{N,T}(\lambda-1) + \beta\,\Big(\Big(1-\nu^z_{N,T}(\lambda)\div\nu^p_{N,T}(\lambda)\Big)\div\nu^p_{N,T}(\lambda)\Big) \tag{11}$$

$$\text{Line 11: } \hat{s}_{N,T}(\lambda) = (1-\beta)\hat{s}_{N,T}(\lambda-1) + \beta\,\Big(\Big(\hat{\mathbf{Z}}_{N,T}(\lambda) - \hat{p}_{N,T}(\lambda)\Big)\div\nu^p_{N,T}(\lambda)\Big)\,, \tag{12}$$

where $\beta \in (0,1]$ defines the amount of damping applied to the variables. As it can be seen, for $\beta = 1$ the algorithm runs without damping. Damping is also applied to lines $12-15$, where the variables $\hat{\mathbf{H}}(\lambda)$,$\hat{\mathbf{X}}(\lambda)$ are replaced by $\overline{\mathbf{H}}(\lambda)$,$\overline{\mathbf{X}}(\lambda)$, where:

$$\overline{\mathbf{H}}(\lambda) = \beta\hat{\mathbf{H}}(\lambda) + (1-\beta)\,\overline{\mathbf{H}}(\lambda-1)\,, \tag{13}$$

$$\overline{\mathbf{X}}(\lambda) = \beta\hat{\mathbf{X}}(\lambda) + (1-\beta)\,\overline{\mathbf{X}}(\lambda-1)\,. \tag{14}$$

3) Convergence Criterion: The convergence condition of [2, Table III] was utilized. However no stopping threshold was used. Instead a maximum number of iterations, λ_{max}, was defined and at the end of each iteration the convergence condition/error was saved in a vector $\epsilon(\lambda)$ (line 20 of Algorithm 1). After λ_{max} iterations, value λ^* corresponding to the position of minimum in vector ϵ, was chosen and $\hat{\mathbf{H}}(\lambda^*)$, $\hat{\mathbf{X}}(\lambda^*)$ were returned by the algorithm. This modification was made due to the fact, in the implementation performed for the project, that not a constant threshold criterion in the convergence condition was able to offer acceptable performance when various cases (i.e., different values for N, K, β etc.) were tested. Besides that, it was noticed that after iteration λ^* the algorithm diverged, i.e., $\epsilon(\lambda)$ attained large values compared to $\epsilon(\lambda^*)$. Value λ_{max} was chosen after observing the number of iterations needed for the algorithm to offer the best performance with respect to symbol error rate.

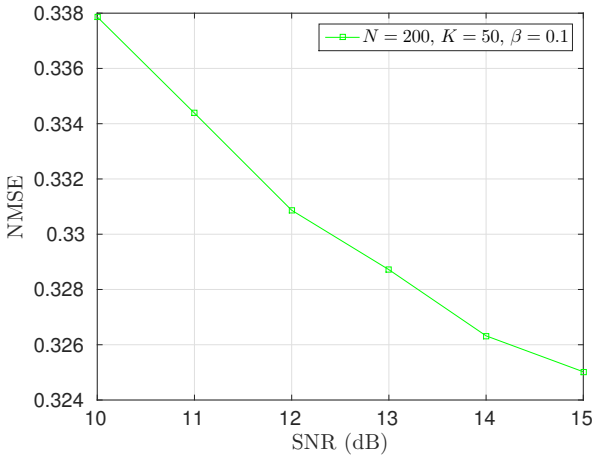

Fig. 3. Channel NMSE vs SNR for $N = 200$, $K = 50$, $\beta = 0.1$.

IV. Performance Results

The algorithm was implemented in MATLAB. Extensive trial & error, with respect to algorithm tuning, attempts were made in order to achieve acceptable performance. Different

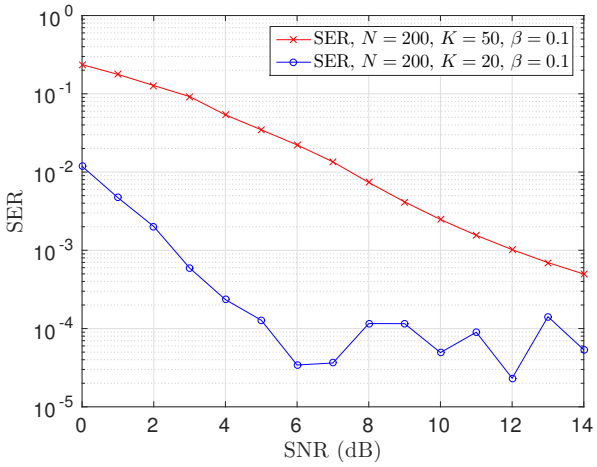

Fig. 4. Symbol error rate for $N = 200$, $\beta = 0.1$ and $T = T_{\mathrm{t}} + T_{\mathrm{d}} = 100 + 400$.

combinations of N, K, T_{t} and β were tried, various parts of the algorithm itself were experimentally modified in the quest for performance gain. The trials were made mainly by observing the symbol error rate (SER), giving less attention to channel normalized mean squared error (NMSE, defined as $\left(\frac{||\hat{\mathbf{H}}(\lambda^*)-\mathbf{H}||_{\mathrm{F}}}{||\mathbf{H}||_{\mathrm{F}}}\right)^2$).

To compile the plots, for each SNR (defined as SNR $= 1/\sigma_w^2$) value, both SER and channel NMSE was averaged across 10^4 channel realizations of Eq. (1) (different $\mathbf{H}$ and $\mathbf{W}$ per realization). The algorithm was run on each realization with $\lambda_{max} = 30$, value chosen after observing that the algorithm chose always $\lambda^* \leq 25$.

Fig. 4 offers the symbol error rate of the reproduced algorithm. Damping factor was $\beta = 0.1$. Attempts with $\beta = 1$ (i.e., no damping) resulted to unacceptable performance (high SER throughout the SNR range). In Fig. 3 the channel estimation performance can be seen. The obtained error is clearly high. This may be due to implementation error or misleading content of [1] (as informed by author of [4]), regarding channel estimation initialization (i.e., value of $\hat{\mathbf{H}}(0)$).

REFERENCES

[1] C. K. Wen, C. J. Wang, S. Jin, K. K. Wong, and P. Ting, "Bayes-optimal joint channel-and-data estimation for massive mimo with low-precision adcs," *IEEE Transactions on Signal Processing*, vol. 64, no. 10, pp. 2541–2556, May 2016.

[2] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing ;part i: Derivation," *IEEE Transactions on Signal Processing*, vol. 62, no. 22, pp. 5839–5853, Nov. 2014.

[3] S. M. Kay, *Fundamentals of statistical signal processing. [Volume I]. , Estimation theory*. Upper Saddle River (N.J.): Prentice Hall, 1993.

[4] N. Psaromanolakis, "Approximate message passing for joint data and channel estimation in multi-cell massive mimo," Nov. 2017, supervisor A. Bletsas.