BLIND (TRAINING-LIKE) DECODER ASSISTED BEAMFORMING FOR DS-CDMA SYSTEMS

Ryan A. Pacheco and Dimitrios Hatzinakos

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto, 10 King's College Road, Toronto, ON M5S 3G4, Canada
{ryan,dimitris}@comm.toronto.edu

ABSTRACT

We propose an iterative blind beamforming strategy for short-burst high-rate DS-CDMA systems. The blind strategy works by creating a set of "training sequences" in the receiver that are used as input to a semi-blind beamforming algorithm, thus producing a corresponding set of beamformers. The objective then becomes to find which beamformer gives the best performance (smallest bit error). Two challenges we face are, to first find a semi-blind algorithm that requires very few training symbols (to minimize the search time), and then to find an appropriate criterion for picking the beamformer that offers the best performance. Different semi-blind algorithms and criteria are tested. The recently proposed SBCMACI (Semi-Blind CMA with Channel Identification) is demonstrated to be ideal because of how few training symbols it needs for convergence. Of the tested criteria, one based on feedback from the decoder (essentially using trellis information) is shown to achieve nearly optimal performance.

1. INTRODUCTION

Semi-blind algorithms have recently been developed for interference suppression in asynchronous short-burst DS-CDMA systems showing significant reduction in the required number of training symbols per burst in comparison to training-only (Least Squares (LS)) approaches (see [1]). Semi-blind algorithms usually require much less quasi-stationary data then their blind counterparts (for example, compare semi-blind CMA with CMA, as demonstrated in [2]) making them attractive solutions for wireless systems. Is it possible to retain this desirable feature, yet have no training? It is, if one is willing to trade training for complexity.

Considering one extreme, the method presented by Seshadri in [3] performs joint data and channel estimation by finding the least-squares channel estimate between the received signal and every possible input signal that may have been transmitted. The computational complexity of this approach is obviously not feasible, so sub-optimal procedures were described. The approach we take is simpler: create a set of of training-sequences of a certain length in the receiver and use these to train a semi-blind algorithm. Using an appropriate criterion, pick the sequence which trains to give the fewest bit errors. To maintain a reasonable complexity, a semi-blind algorithm that converges quickly is desired.

The SBMCA, first presented in [1], uses an extremely small number of training symbols for convergence. In [1] a dense scattering environment was considered, wherein the receiver consisted of an A element antenna array with element spacing far enough apart to allow for diversity combining. In contrast, the receiver in this paper has array elements spaced by $\lambda/2$ (where λ is the wavelength of the carrier) to allow for beamforming. The SBCMACI was first applied in a beamforming-scenario in [4]. As an example of the relative performance of the SBCMACI to other beamforming strategies, consider the result shown in Fig. 1 comparing the probability of symbol error for different training lengths (the simulation details are not important at this time). Observe the significant difference in performance of the SBCMACI in comparison to conventional LS beamforming and another semi-blind technique, SBCMA [5]. In this case, the eye diagrams in Fig. 2 show that with only 3 training symbols the SBCMACI is able to open the eye.

The number of training symbols required by the SBCMACI is so small that one wonders if it is feasible (or even possible) to perform completely blind beamforming using it. The objective of this paper is to investigate the answer to this question. The conclusion is that it is feasible, but the criterion for deciding which training-sequence to use is critical. Of the four criteria we tested, the one that incorporated trellis information, through feedback from the decoder, was found to almost always produce the correct decisions.

2. THE SETUP

We propose a blind strategy that operates according to the setup shown in Fig. 3. There are three main components: a block which generates "training sequences", a block which does semi-blind beamforming using both the received data and the generated (hypothetical) "training sequence", and a block which picks the sequence that trains to give the smallest number of errors.

The blocks shown in Fig. 3 fit inside the system shown in Fig. 4, which represents an M-user nonorthogonal coded DS-CDMA system. The encoder applies a convolutional code to the data bits for each user. Let $d_m[n] \in \{0, 1\}$ be the nth bit for the mth user, then in vector form $\mathbf{d}_m = [d_m[0], \cdots, d_m[N_d - 1]]^T$, where N_d is the number of bits per burst. Let $e(\cdot)$ and $\Pi_m(\cdot)$ represent the encoding and interleaving operations respectively, then as shown in Fig. 4, $\tilde{\mathbf{d}}_m = \Pi_m(e(\mathbf{d}_m))$, represents the output of the interleaver for the mth user which is then mapped to a 4-QAM alphabet through the following simple differential coding scheme: $b_m[n] = b_m[n-1]\mathrm{e}^{j\phi_n^m}$, $b_m[0] = \mathrm{e}^{j\phi_0^m}$, where $\phi_n^m = (\tilde{d}_m[2(n-1)] + 2 \cdot \tilde{d}_m[2n-1]) \cdot \pi/2$ and $\phi_0^m \in \{\pm\pi/4, \pm 3\pi/4\}$. ϕ_0^m is randomly assigned to each user. Differential encoding is used because the blind algorithm is only able to equalize up to a complex scalar, thus we need to design the system to be invariant to phase

This work was partially supported by an Ontario Graduate Scholarship (OGS).

rotations. The next step is then spreading with a unique normalized spreading code $c_m[n] \in \{\pm 1/\sqrt{N}\}, n = 0, \cdots, N-1$, where N is the processing gain. The signal is transmitted over a multipath channel of length L_m and received at an A element antenna array. Assuming that the inverse signal bandwidth is large compared to the propagation time across the array, the complex envelopes of the signals received by different antenna elements from a given path are identical except for phase and amplitude differences that depend on the path angle of arrival (AOA), array geometry and the element pattern. The AOA of the lth multipath signal from the mth user is Θ_m^l and $\mathbf{a}(\Theta_m^l) = [a_1(\Theta_m^l), \cdots, a_A(\Theta_m^l)]^T$ is the corresponding array response vector.

The received signal is sampled at the chip rate, to yield the following discrete-time signal at the ith antenna element:

$$r_i[n] = \sum_{m=1}^{M} \sum_{k=0}^{N_b-1} b_m[k] g_m^i[n-k] + v_i[n] \tag{1}$$

where N_b is the number of symbols per burst, $v_i[n]$ is additive white complex Gaussian noise of variance σ^2,

$$g_m^i[k] = \sum_{l=0}^{L_m-1} a_i(\Theta_m^l)\beta_m[l]c_m[k-l] \tag{2}$$

L_m is the number of resolvable multipath components for the mth user and $\beta_m[l]$ are complex Gaussian random variables. The received signal, corresponding to the kth symbol can be written in vector form as follows:

$$\mathbf{r}(k) = [\mathbf{G}(L-1), \cdots, \mathbf{G}(0)] \cdot [\mathbf{b}(k-L+1)^T, \cdots, \mathbf{b}(k)^T]^T + \mathbf{v}(k) \tag{3}$$

where, $(L-1)N \geq \max(L_m)$
$\mathbf{r}(k) = [\mathbf{r}_1(k)^T, \cdots, \mathbf{r}_A(k)^T]^T$, $\mathbf{r}_a(k) = [r_a[kN], \cdots, r_a[(k+1)N-1]]^T$, $\mathbf{G}(l) = [\mathbf{g}_1(l), \cdots, \mathbf{g}_M(l)]$,
$\mathbf{g}_m(l) = [g_m^1[lN], \cdots, g_m^1[(l+1)N-1], \cdots, g_m^A[lN], \cdots, g_m^A[(l+1)N-1]]^T$, $\mathbf{b}(k) = [b_1[k], \cdots, b_M[k]]^T$, and
$\mathbf{v}(k) = [\mathbf{v}_1(k)^T, \cdots, \mathbf{v}_A(k)^T]^T$, $\mathbf{v}_a(k) = [v_a[kN], \cdots, v_a[(k+1)N-1]]^T$.

In some cases, depending on the channel length, number of users (M), processing gain (N), and number of antenna elements (A), it might be necessary to process more than one received vector at a time in order to estimate the kth symbol. In general, stacking μ consecutive symbols, the vector that will be processed, $\mathbf{r}_\mu(k)$, is written as

$$\mathbf{r}_\mu(k) = \mathbf{G}_\mu \mathbf{b}_\mu(k) + \mathbf{v}_\mu(k) \tag{4}$$

where,
$\mathbf{r}_\mu(k) = [\mathbf{r}(k)^T, \cdots, \mathbf{r}(k+\mu-1)^T]^T$,
$\mathbf{b}_\mu(k) = [\mathbf{b}(k-L+1)^T, \cdots, \mathbf{b}(k+\mu-1)^T]^T$,
$\mathbf{v}_\mu(k) = [\mathbf{v}(k)^T, \cdots, \mathbf{v}(k+\mu-1)^T]^T$, and

$$\mathbf{G}_\mu = \begin{bmatrix} \mathbf{G}(L-1) & \cdots & \mathbf{G}(0) & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{G}(L-1) & \cdots & \mathbf{G}(0) \end{bmatrix}$$

It is assumed that $\mathbf{G}_\mu$ has full column rank. Thus, μ (called the smoothing factor) must satisfy the following inequality

$$\mu \geq \left\lceil \frac{M(L-1)}{AN-M} \right\rceil \tag{5}$$

Therefore, the dimensionality of the signal subspace, κ, becomes $\kappa = M(\mu + L - 1)$. The results shown in Figures 1 and 2 correspond to a synchronous non-coded DS-CDMA system with $M = 4$, $L = 2$, $A = 5$, $N = 7$ and SNR $= 10\text{dB}$ (SNR $\triangleq 10\log_{10}(1/\sigma^2)$), meaning that $\kappa = 8$. Given the phase ambiguity of the blind beamformer the number of training sequences of length N_t that give unique outputs is 4^{N_t-1}. So $4^7 = 16384$ training sequences of length 8 will need to be searched (in the infinite SNR case) if the LS algorithm is to be used. From Fig. 1 we see that many more sequences need to be searched when SNR is finite (at least 4^{15} of length 16 in this case). On the other hand, from Figures 1 and 2 we see that N_t should be at least 3 for the SBCMACI to work. This means that only $4^2 = 16$ different sequences need to be generated and used by the SBCMACI to create 16 different beamformers. The SBCMACI requires significantly less training symbols than the signal space dimensions because of enhanced initialization through semi-blind channel identification [1, 5]. The SBCMA [5] is similar to the SBCMACI but does not perform channel estimation.

It is the job of the "Choose Best Sequence" block in Fig. 3 to find the beamformer that yields the fewest number of errors. Once that's done, the recovered symbols from the selected beamformer are sent to the differential decoder to recover the transmitted (desired) bits. Looking at Fig. 1 we can see that if another semi-blind algorithm were used, such as the SBCMA, then about $4^8 = 65536$ different sequences would need to be tested to get the same results.

From the above discussion it is obvious that performance of this blind scheme depends greatly on two factors: 1. Ability of the semi-blind algorithm to open the eye with very few training symbols, and 2. Ability to choose the best beamformer of the 4^{N_t-1} different possibilities. The SBCMACI is needed to satisfy the first point. In order to satisfy the second, we tried four different criteria. Three are distance measures that compute the squared error between the output of the beamformer and either the source alphabet, the source modulus, or the generated training sequence. The other criterion uses the demapped and decoded outputs of the beamformer (see Fig. 4) to compute the number of bits that have been flipped by the decoder. If interference suppression by the semi-blind algorithm is not successful then the deinterleaved bits will not satisfy the constraints imposed by the encoder's trellis and the number of bits corrected (flipped) by the decoder will be large. So the beamformer whose output yields the smallest number of flips is chosen. This method of picking the best beamformer can be thought of as a further semi-blind enhancement where now we incorporate knowledge of the convolutional code and interleaver.

3. SELECTION CRITERIA

Using only the received data vectors, $\mathbf{X} = [\mathbf{r}_\mu(0), \cdots, \mathbf{r}_\mu(N_b-1)]$, the generated training sequence of length N_t, $\{\mathbf{b}_{\text{tr}}^k\}_{k=1}^{4^{N_t-1}}$, and the spreading code for the desired user, $\mathbf{c}_m$, the beamformer (weight vector) $\mathbf{W}_k$ ($k = 1, \cdots, 4^{N_t-1}$) is computed by

$$\mathbf{W}_k = \arg\min_{\mathbf{W}} F(\mathbf{b}_{\text{tr}}^k, \mathbf{X}, \mathbf{c}_m) \tag{6}$$

where $F(\cdot)$ is some optimization scheme (SBCMACI [1], SBCMA [5], LS, etc.). This setup is similar to that used by Kuzminskiy in [6]. Among the 4^{N_t-1} different weight vectors, the desired $\hat{\mathbf{W}}$ is chosen as follows

$$\hat{\mathbf{W}} = \mathbf{W}_{k_o}, \quad k_o = \arg\min_k \rho_k \tag{7}$$

where ρ_k is some criterion. We tried the following four different criteria for deciding on the correct weight vector:

1. Distance from the source alphabet:

$$\rho_k^1 = \frac{1}{N_b} \sum_{i=0}^{N_b-1} \min_l (|\mathbf{W}_k^H \mathbf{r}_\mu(i) - a_l|), \quad (8)$$
$$\text{where} \quad a_l \in \{e^{\pm j\frac{\pi}{4}}, e^{\pm j\frac{3\pi}{4}}\}$$

2. Distance from the source modulus:

$$\rho_k^2 = \frac{1}{N_b} \sum_{i=0}^{N_b-1} (|\mathbf{W}_k^H \mathbf{r}_\mu(i)| - 1)^2 \quad (9)$$

3. Distance from the generated training sequence:

$$\rho_k^3 = \frac{1}{N_t} \sum_{i=0}^{N_t-1} |\mathbf{W}_k^H \mathbf{r}_\mu(i) - b_{\text{tr}}^k(i)|^2 \quad (10)$$

4. Number of flipped bits:
Let $f(\cdot)$ represent the differential encoding operation, then $\hat{\hat{\mathbf{d}}}_k = f^{-1}((\mathbf{W}_k^H \mathbf{X})^T)$, and $\hat{\mathbf{d}}_k = e^{-1}(\Pi_m^{-1}(\hat{\hat{\mathbf{d}}}_k))$ represent the outputs of the differential decoder and convolutional decoder respectively. The number of bits that have been flipped by the decoder, ρ_k^4 is the number of bits that are different between $\hat{\hat{\mathbf{d}}}_k$ and $\Pi_m(e(\hat{\mathbf{d}}_k))$.

4. SIMULATION EXAMPLES

In this section we present a few simulation results demonstrating the performance of the proposed scheme. We assume a synchronous coded CDMA system as shown in Fig. 4. All users use the same rate 1/2 constraint length 5 convolutional code (with generators 23, and 35 in octal notation), but have different interleavers which are generated randomly. Spreading codes are generated randomly as well. The receiver employs a circular antenna array with equally spaced elements ($\lambda/2$), thus $\mathbf{a}(\Theta_m^l) = [\exp(-j\frac{2\pi}{\lambda} r_{ant} \cos(\Theta_m^l - 2\pi)), \cdots, \exp(-j\frac{2\pi}{\lambda} r_{ant} \cos(\Theta_m^l - \frac{2\pi}{A}))]$, where $r_{ant} = A\frac{\lambda}{4\pi}$. AOAs are generated randomly for each user between 0 and 2π radians.

Figure 5 is an example of how the criteria for training sequence selection performs when the SBCMACI is used. The system parameters are $M = 4$, $N_d = 200$, $N = 7$, $A = 5$, SNR=5 dB, $\mu = 1$, and $L = 2$. The vertical axis of each plot represents the value of ρ_k. From top to bottom the results correspond to using criteria 1, 2, 3 and 4 respectively. k_0 and the number of bit errors in each case is ρ^1:[11,90], ρ^2:[11,90], ρ^3:[6,25], and ρ^4:[13,0]. So in this example, ρ^4 (minimum number of flipped bits) gave the smallest number of errors. It is also interesting to note that in this case the beamformer with index 6 (which yields 25 bit errors and was picked by ρ^3) was trained using the correct first 3 symbols of the data burst but did not yield the smallest number of errors. This means that the blind scheme can outperform the semi-blind scheme if the proper criterion is used.

Figure 6 is a comparison of the probability of bit error (estimated using 5000 iterations) when the four different criteria are used in conjunction with the SBCMACI. Same simulation parameters as before. We see that again ρ^4 outperforms the other three. Assume that from the 16 different beamformers we were always able to pick the beamformer that yields the smallest number of errors, then the probability of error curve that would result from this scheme is a lower bound on all possible criteria. This curve is shown, called "SB: Optimum" on the plot, and it is seen that ρ^4 comes very close to achieving this bound. If, from the 16 different beamformers, we were always able to pick the beamformer that was trained from the correct first three symbols of the data burst then the performance we would achieve is actually worse then that achieved through ρ^4 and this is shown with the curve labelled "SB: Training". Finally, we plotted the performance of the Wiener filter which is implemented assuming perfect knowledge of the channels for all users.

5. CONCLUDING REMARKS

We have shown how it is possible to use a semi-blind technique, the SBCMACI, to perform blind beamforming in DS-CDMA systems subject to ISI and MAI. The results show that performance is very dependent on the choice of selection criterion and that if the decoder assists in the selection process then it is possible to get near optimum performance using the blind scheme. In fact, the blind scheme outperformed the original semi-blind technique. The complexity of this scheme is the major concern. It is only applicable to relatively small values of N_t and small alphabet sizes.

6. REFERENCES

[1] R.A. Pacheco and D. Hatzinakos, "Semi-blind spatio-temporal equalization and multiuser detection for DS-CDMA systems," in *2001 IEEE Third Workshop on Sig. Proc. Advances in Wireless Comm. (SPAWC'01)*, 2001, pp. 126–129.

[2] A.M. Kuzminskiy, "Finite amount of data effects in spatio-temporal filtering for equalization and interference rejection in short burst wireless communications," *Signal Processing*, vol. 80, no. 10, pp. 1987–1997, October 2000.

[3] Nambi Seshadri, *S. Haykin, ed., Blind Deconvolution*, chapter Joint data and channel estimation using blind trellis search techniques, pp. 259–286, Prentice Hall, 1994.

[4] Ivan R.S. Casella, E.S. Sousa, and P.J.E. Jeszensky, "Semi-blind beamspace-time interference cancellation using subspace channel identification for DS-WCDMA systems," in *PIMRC*, 2002, pp. 1972–1976.

[5] R.A. Pacheco and D. Hatzinakos, "Semi-blind strategies for interference suppression in DS-CDMA systems," *Submitted to IEEE Trans. on Communications*, 19 pages, 2002.

[6] A.M. Kuzminskiy and D. Hatzinakos, "Semi-blind estimation of spatio-temporal filter coefficients based on a training-like approach," *IEEE Signal Proc. Letters*, vol. 5, pp. 231–233, Sept. 1998.

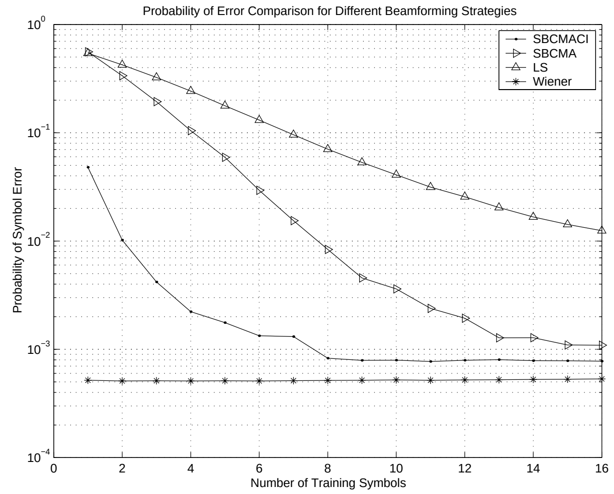

Fig. 1. Probability of symbol error comparison of different semi-blind algorithms. The relatively small number of training symbols required by the SBCMACI leads one to wonder if it can be used to perform completely blind interference suppression. SNR = 10dB, $M = 4$, $A = 5$, $N = 7$, $N_d = 200$, $L = 2$, $\mu = 1$. Thus, $\kappa = 8$.

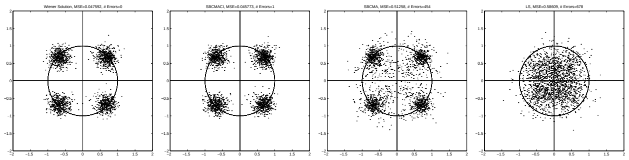

Fig. 2. Eye Diagrams: $N_t = 3$. Same parameters as in Fig.1. From left to right, eye diagrams correspond to outputs from Wiener filter, SBCMACI, SBCMA, and LS respectively. The corresponding Symbol Error Rates computed for each figure was: 0.513×10^{-3}, 4.2×10^{-3}, 0.194, and 0.325 respectively.

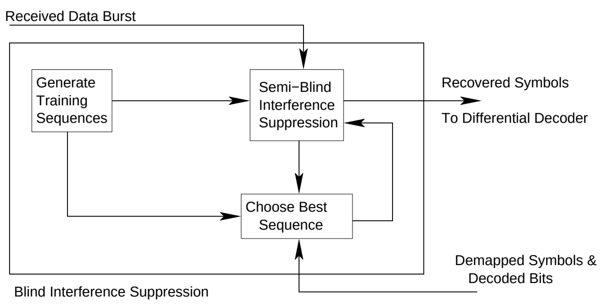

Fig. 3. Blind algorithm consists of three main components: Block which generates training sequences, block which does semi-blind recovery of the received data, block which picks the correct data sequence.

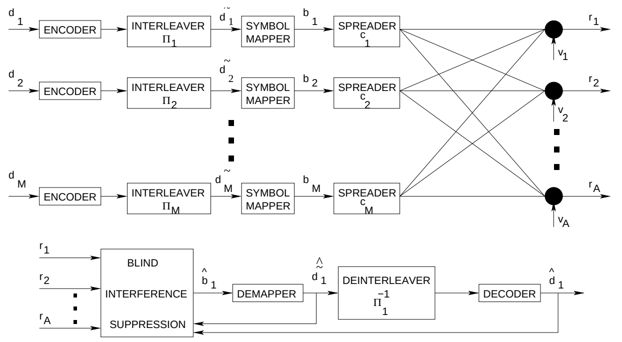

Fig. 4. Coded CDMA System with Blind Beamforming for Interference Suppression.

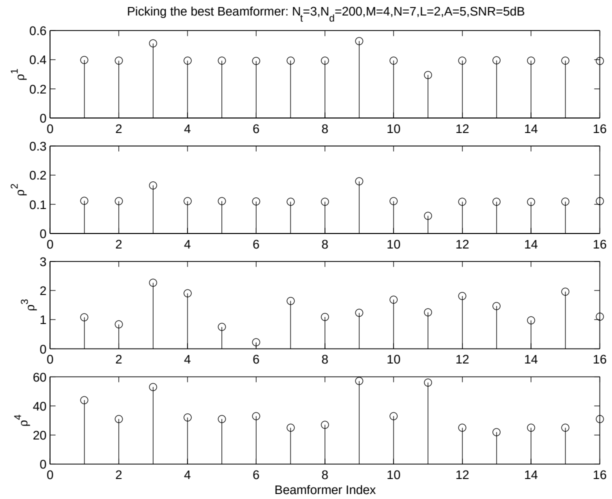

Fig. 5. Picking the best Beamformer: Top to Bottom, results for ρ^1 to ρ^4 respectively. [Minima, # of bit errors]: ρ^1:[11,90],ρ^2:[11,90],ρ^3:[6,25],ρ^4:[13,0]. $N_t = 3$, $N_d = 200$, $M = 4$, $A = 5$, $L = 2$, SNR=5 dB.

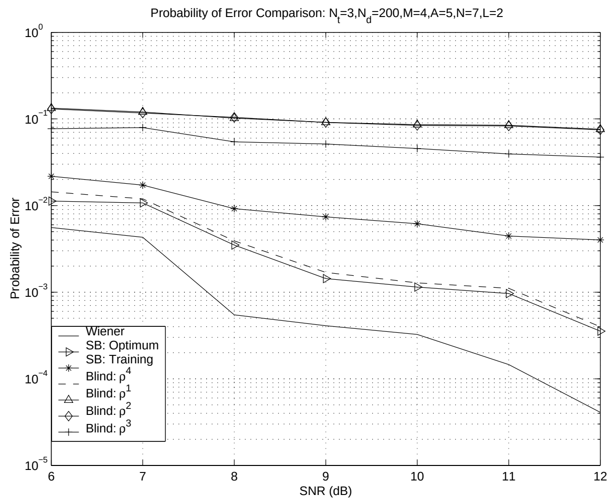

Fig. 6. Probability of Error Comparison: $N_t = 3$, $N_d = 200$, $M = 4$, $A = 5$, $L = 2$.