Learning in an Uncertain World: MIMO Covariance Matrix Optimization with Imperfect Feedback

Panayotis Mertikopoulos, *Member, IEEE*, and Aris L. Moustakas, *Senior Member, IEEE*

Abstract

In this paper, we present a distributed learning algorithm for the optimization of signal covariance matrices in Gaussian multiple-input and multiple-output (MIMO) multiple access channels with imperfect (and possibly delayed) feedback. The algorithm is based on the method of matrix exponential learning (MXL) and it has the same information and computation requirements as distributed water-filling. However, unlike water-filling, the proposed algorithm converges to the system's optimum signal covariance profile even under stochastic uncertainty and imperfect feedback. Moreover, the algorithm also retains its convergence properties in the presence of user update asynchronicities, random delays and/or ergodically changing channel conditions. Our theoretical analysis is complemented by extensive numerical simulations which illustrate the robustness and scalability of MXL in realistic network conditions. In particular, the algorithm retains its convergence speed even for large numbers of users and/or antennas per user.

Index Terms

Imperfect feedback; MIMO; covariance matrix optimization; matrix exponential learning.

I. Introduction

Following the seminal prediction that the use of multiple antennas can lead to substantial performance gains [1, 2], multiple-input and multiple-output (MIMO) technologies have become an integral component of most state-of-the-art wireless communication protocols (ranging from 3G LTE and 4G to HSPA+ and WiMax). To capitalize on these gains, the emerging massive MIMO paradigm "goes large" by scaling up existing multiple-antenna transceivers through the use of inexpensive service antennas and time-division duplexing (TDD) [3–5]. In so doing, massive MIMO arrays can increase throughput by a factor of 10× (or more), bring about significant latency reductions over the air interface, and greatly improve the system's robustness to ambient noise [5, 6].

In this context, it is crucial to optimize the input signal covariance matrix of each user, especially for moderate signal-to-interference-and-noise ratio (SINR) values. This optimization is typically achieved via water-filling (WF)

This research was supported in part by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM# (contract no. 318306), and by the French National Research Agency (ANR) under the contract NETLEARN (ANR–13–INFR–004). Part of this work was presented in ISIT 2012 [11] and ISIT 2014 [12].

P. Mertikopoulos is with the French National Center for Scientific Research (CNRS) and the Laboratoire d'Informatique de Grenoble, Grenoble, France; A. L. Moustakas is with the Physics Department, University of Athens, Greece and Supélec, Gif-sur-Yvette, France, supported by the Digiteo Senior Chair "ASAPGONE".

methods [7–9] that rely on accurate channel state information (CSI) and multi-user interference-plus-noise (MUI) measurements. However, a major challenge occurs when this information is subject to measurement errors, delays and/or other imperfections (e.g. due to pilot contamination in massive MIMO systems [3, 10]). In this case, the convergence of WF methods is no longer guaranteed so the efficient deployment of MIMO-enabled devices calls for flexible and robust optimization algorithms that are capable of dealing with feedback uncertainty on several levels.

In this paper, we propose a distributed optimization algorithm based on the so-called matrix exponential learning (MXL) method that was first introduced in the continuous-time setting of [11]. Essentially, rather than updating their signal covariance matrices directly, transmitters update the logarithm of these matrices based on (possibly imperfect) measurements of a matrix analogue of the transmitter's SINR. The benefit of updating the logarithm of a user's covariance matrix is that the algorithm's updates do not need to satisfy the problem's semidefiniteness constraints. Furthermore, in contrast to WF methods, the proposed algorithm proceeds by *aggregating* the users' feedback over time: in this way, measurement errors, noise and asynchronicities effectively vanish in the long run thanks to the law of large numbers. As a result, the proposed algorithm has the following desirable attributes:

1) It is *distributed:* user updates are based on local information and channel measurements.
2) It is *robust:* measurements and updates may be subject to random errors, noise and delays.
3) It is *stateless:* users do not need to know the state (or topology) of the system.
4) It is *reinforcing:* each user tends to increase his own rate.
5) It is *flexible:* users can employ it synchronously or asynchronously, and in both static and fast-fading channels.

A good paradigm to test the performance of the proposed algorithm is the widely studied Gaussian vector multiple access channel (MAC) [8]. This system is the MIMO equivalent of the parallel multiple access channel (PMAC) and consists of several (independent) MIMO transceivers that are linked to a common multi-antenna receiver. In this framework, it is well known that iterative water-filling (IWF) converges to the system's optimum transmit profile provided that the transmitter has access to perfect CSI [8]; however, the algorithm's convergence speed decreases proportionally with the number of transmitting users. Simultaneous water-filling methods SWF can be much faster but, unfortunately, they may fail to converge altogether; to make matters worse, even IWF may fail to converge under noisy observations and feedback. By contrast, the proposed MXL algorithm converges to the system's optimum transmit profile within a few iterations (even for large numbers of users), and it remains convergent irrespective of the measurement noise.

After introducing our system model in Section II, the MXL algorithm is derived in Section III and our main convergence results in the presence of imperfect feedback and measurement errors are presented in Section IV. Section V is devoted to asynchronous/delayed feedback and the evolution of the users' covariance eigenvalues/eigenvectors, while Section VI extends our analysis to fast-fading channels. Finally, our theoretical results are validated and supplemented by numerical simulations in Section VII. To streamline the flow of the paper, proofs and technical details have been delegated to a series of appendices at the end.

Our work here greatly extends our recent conference paper [11] where we introduced a continuous-time matrix

exponential learning method for rate maximization in vector Gaussian multiple access channels. Compared to [11], the current paper provides the theoretical foundations for the properties of MXL that were announced in [11], and we provide several completely novel results – including (but not limited to) explicit estimates for the algorithm's convergence time and a proof of the algorithm's robustness to noisy feedback. We also provide three new algorithms: *a*) an asynchronous variant (Algorithm 2); *b*) an eigen-based variant that dispenses with the exponentiation step (Algorithm 3); and *c*) a learning algorithm for ergodic channels with imperfect statistical feedback (Algorithm 4).

II. System Model

Consider a Gaussian vector multiple access channel (MAC) where a finite set of wireless users $k \in \mathcal{K} \equiv \{1, \dots, K\}$ transmit simultaneously over a common channel to a base receiver with N antennas. If the k-th transmitter is equipped with M_k transmit antennas, we get the familiar signal model

$$\mathbf{y} = \sum\nolimits_{k=1}^{K} \mathbf{H}_k \mathbf{x}_k + \mathbf{z}, \tag{1}$$

where:

1) $\mathbf{x}_k \in \mathbb{C}^{M_k}$ is the message transmitted by user $k \in \mathcal{K}$.
2) $\mathbf{y} \in \mathbb{C}^N$ denotes the aggregate signal at the receiver.
3) $\mathbf{H}_k \in \mathbb{C}^{N \times M_k}$ is the $N \times M_k$ channel matrix of user k.
4) $\mathbf{z} \in \mathbb{C}^N$ is the ambient noise in the channel, including thermal, atmospheric and other peripheral interference effects (and modeled for simplicity as a zero-mean, circulant Gaussian vector with unit covariance).

In this context, the average *transmit power* of user k is simply

$$p_k = \mathbb{E}\left[\|\mathbf{x}_k\|^2\right] = \mathrm{tr}(\mathbf{Q}_k), \tag{2}$$

where $\mathbf{Q}_k$ denotes the user's *signal covariance matrix*

$$\mathbf{Q}_k = \mathbb{E}\left[\mathbf{x}_k \mathbf{x}_k^\dagger\right] \tag{3}$$

and the expectation is taken over the Gaussian codebook of user k. Hence, assuming that each user's maximum transmit power is finite, we obtain the feasibility constraints:

$$\mathbf{Q}_k \succcurlyeq 0 \quad \text{and} \quad \mathrm{tr}(\mathbf{Q}_k) \leq P_k, \tag{4}$$

where $P_k > 0$ denotes the maximum transmit power of user k.

The first part of our analysis focuses on *static channels*, i.e. $\mathbf{H}_k$ will be assumed to remain constant (or nearly constant) throughout the transmission horizon (fast-fading channels will be treated in Section VI). In this case, assuming single user decoding (SUD) at the receiver (i.e. interference by all other users is treated as additive noise), each user's achievable transmission rate will be given by the familiar expression [2]:

$$R_k(\mathbf{Q}) = \log\det\left(\mathbf{I} + \textstyle\sum_\ell \mathbf{H}_\ell \mathbf{Q}_\ell \mathbf{H}_\ell^\dagger\right) - \log\det\left(\mathbf{W}_{-k}\right), \tag{5}$$

where $\mathbf{Q} = (\mathbf{Q}_1, \ldots, \mathbf{Q}_K)$ and

$$\mathbf{W}_{-k} = \mathbf{I} + \textstyle\sum_{\ell \neq k} \mathbf{H}_\ell \mathbf{Q}_\ell \mathbf{H}_\ell^\dagger \tag{6}$$

represents the multi-user interference-plus-noise (MUI) covariance matrix of user k. We will thus say that a transmit profile $\mathbf{Q}^* = (\mathbf{Q}_1^*, \ldots, \mathbf{Q}_K^*)$ is at *Nash equilibrium* when no user can unilaterally improve his individual achievable rate R_k, i.e.

$$R_k(\mathbf{Q}^*) \geq R_k(\mathbf{Q}_k; \mathbf{Q}_{-k}^*) \quad \text{for all } \mathbf{Q}_k \in \boldsymbol{\mathcal{Q}}_k,\ k \in \mathcal{K}, \tag{NE}$$

where $(\mathbf{Q}_k; \mathbf{Q}_{-k}^*)$ is shorthand for $(\mathbf{Q}_1^*, \ldots, \mathbf{Q}_k, \ldots, \mathbf{Q}_K^*)$ and

$$\boldsymbol{\mathcal{Q}}_k = \{\mathbf{Q}_k \in \mathbb{C}^{M_k \times M_k} : \mathbf{Q}_k \succcurlyeq 0, \operatorname{tr}(\mathbf{Q}_k) \leq P_k\} \tag{7}$$

denotes the set of feasible signal covariance matrices for user k.

Dually to the above, if the receiver employs successive interference cancellation (SIC) techniques to decode the received messages, the users' achievable sum rate will be [8]:

$$R(\mathbf{Q}) = \log\det\left(\mathbf{I} + \textstyle\sum_k \mathbf{H}_k \mathbf{Q}_k \mathbf{H}_k^\dagger\right). \tag{8}$$

In this way, we obtain the sum rate maximization problem:

$$\begin{aligned} &\text{maximize} && R(\mathbf{Q}), \\ &\text{subject to} && \mathbf{Q}_k \in \boldsymbol{\mathcal{Q}}_k,\ k = 1, \ldots, K. \end{aligned} \tag{RM}$$

As can be easily checked, the users' sum rate (8) is a *potential function* for the game (5) in the sense that

$$R_k(\mathbf{Q}_k; \mathbf{Q}_{-k}) - R_k(\mathbf{Q}_k'; \mathbf{Q}_{-k}) = R(\mathbf{Q}_k; \mathbf{Q}_{-k}) - R(\mathbf{Q}_k'; \mathbf{Q}_{-k}). \tag{9}$$

Hence, with R concave, it follows that the solutions of the Nash equilibrium problem (NE) coincide with the solutions of (RM); put differently, *optimizing the users' achievable sum rate* (8) *under SIC is equivalent to reaching a Nash equilibrium with respect to their individual achievable rates* (5) *under SUD*.

For concreteness, in the rest of this paper, we will focus on the sum rate maximization problem (RM); however, owing to the above observation, our results apply verbatim to the unilateral equilibration problem (NE) as well.

III. Learning with Imperfect Feedback

The sum rate maximization problem (RM) is traditionally solved by water-filling (WF) methods [7], either iterative [8, 13] or simultaneous [14]. More precisely, transmitters are typically assumed to have perfect knowledge of the channel matrices $\mathbf{H}_k$ and the aggregate signal-plus-noise covariance matrix

$$\mathbf{W} = \mathbb{E}[\mathbf{y}\mathbf{y}^\dagger] = \mathbf{I} + \textstyle\sum_\ell \mathbf{H}_\ell \mathbf{Q}_\ell \mathbf{H}_\ell^\dagger, \tag{10}$$

which is in turn used to calculate the MUI covariance matrices $\mathbf{W}_{-k} = \mathbf{W} - \mathbf{H}_k \mathbf{Q}_k \mathbf{H}_k^\dagger$ and "water-fill" the effective channel matrices $\widetilde{\mathbf{H}}_k = \mathbf{W}_{-k}^{-1/2} \mathbf{H}_k$ at the transmitter [8]. At a multi-user level, this water-filling process could take place either iteratively (with users updating their covariance matrices in a round robin fashion) [8] or simultaneously (with all users updating at once) [14]. The former (iterative) scheme converges always (but slowly for large numbers of

users) [8], whereas the latter (simultaneous) algorithm is much faster [14] but it may fail to converge, even in simple, 2-user parallel multiple access channels [15].

An added complication in the use of WF methods is that they rely on perfect channel state information at the transmitter (CSIT) and accurate measurements of $\mathbf{W}$ at the receiver (who is usually assumed to feed this information back to the transmitters via a dedicated radio channel or as part of the TDD downlink phase). When such measurements are not available, it is not known whether WF methods converge; accordingly, our goal in this section will be to describe a distributed learning method that allows users to attain the system's sum capacity under imperfect feedback.

Instead of relying on fixed-point methods, we will track the direction of steepest ascent of the system's sum rate in a dual, unconstrained space, and then map the result back to the problem's feasible space via matrix exponentiation. Formally, assuming for the moment perfect feedback, we will consider the matrix exponential learning scheme:

$$\begin{aligned} \mathbf{Y}_k(n+1) &= \mathbf{Y}_k(n) + \gamma_n \mathbf{V}_k(\mathbf{Q}(n)), \\ \mathbf{Q}_k(n+1) &= P_k \frac{\exp(\mathbf{Y}_k(n+1))}{\mathrm{tr}\left[\exp(\mathbf{Y}_k(n+1))\right]}, \end{aligned} \tag{MXL}$$

where:

1) $n = 0, 1, \ldots$, denotes the current iteration of the algorithm.
2) $\mathbf{V}_k \equiv \mathbf{V}_k(\mathbf{Q})$ denotes the (matrix) derivative of the system's sum rate with respect to each user's covariance matrix:

$$\mathbf{V}_k(\mathbf{Q}) \equiv \nabla_{\mathbf{Q}_k} R(\mathbf{Q}) = \nabla_{\mathbf{Q}_k} R_k(\mathbf{Q}) = \mathbf{H}_k^\dagger \mathbf{W}^{-1} \mathbf{H}_k. \tag{11}$$

3) $\mathbf{Y}_k$ is an auxiliary "scoring" matrix which tracks the direction of steepest sum rate ascent.
4) γ_n is a decreasing step-size sequence (typically, $\gamma_n \sim 1/n$).

Remark. Intuitively, (MXL) assigns more power to the spatial eigendirections that perform well while the variable step-size γ_n keeps the eigenvalues of $\mathbf{Q}(n)$ from approaching zero too fast. In this way, (MXL) can be seen as a "primal-dual" gradient method [16] which reinforces the spatial directions that lead to higher sum rates by allocating more power to the corresponding eigen-directions of the users' covariance matrices.

Of course, to employ the recursion (MXL), each user $k \in \mathcal{K}$ needs to know his individual gradient matrix $\mathbf{V}_k$. In turn, this matrix can be calculated at the transmitter by measuring $\mathbf{H}_k$. This can be easily achieved for example in the case of TDD wireless networks [17, 18] where both uplink and downlink modes share the same frequency band (so the channel can be obtained from the pilot downlink phase). Additionally, the receiver also broadcasts the received signal precision matrix

$$\mathbf{P} = \mathbf{W}^{-1} = \left(\mathbf{I} + \textstyle\sum_k \mathbf{H}_k \mathbf{Q}_k \mathbf{H}_k^\dagger\right)^{-1}, \tag{12}$$

which is the only feedback required for the update process.[1] However, since measurements and feedback are often subject to noise and uncertainty,[2] we will assume that the gradient matrices $\mathbf{V}_k$ of (11) are only known up to a *noisy* estimate $\hat{\mathbf{V}}_k$ at the transmitter. In particular, we will assume that:

[1] Importantly, these measurement and feedback requirements are the same as in distributed water-filling [8, 13, 14].

[2] Such errors in the estimation of the channel matrix can be traced in the mobility of the users, as well as pilot contamination [18].

1) At every update period $n = 1, 2, \ldots$, each user $k \in \mathcal{K}$ can observe a noisy estimate $\hat{\mathbf{V}}_k(n)$ of $\mathbf{V}_k(\mathbf{Q}(n))$.

2) Users update their signal covariance matrices according to (MXL) and the process repeats.

More concretely, this recurring process may be encoded in algorithmic form as follows:

Algorithm 1 Matrix Exponential Learning (MXL).

Parameter: decreasing step-size sequence γ_n

Initialize: $n \leftarrow 0$; $\mathbf{Y}_k \leftarrow 0$; $\mathbf{Q}_k \leftarrow \frac{P_k}{M_k}\mathbf{I}$

Repeat

$n \leftarrow n + 1$;

for each user $k \in \mathcal{K}$ **do simultaneously**

get estimate $\hat{\mathbf{V}}_k$ of $\mathbf{V}_k = \mathbf{H}_k^\dagger \mathbf{W}^{-1}\mathbf{H}_k$;

update score matrix:

$\mathbf{Y}_k \leftarrow \mathbf{Y}_k + \gamma_n \hat{\mathbf{V}}_k$;

update covariance matrix:

$\mathbf{Q}_k \leftarrow P_k \exp(\mathbf{Y}_k)/\operatorname{tr}[\exp(\mathbf{Y}_k)]$;

until termination criterion is reached.

The MXL algorithm above will be the main focus of our paper, so a few remarks are in order:

a) Implementation: From an implementation point of view, MXL has the following desirable properties:

(P1) *Distributedness:* users have the same information requirements as in distributed water-filling methods [8, 13, 14].

(P2) *Robustness:* the algorithm does not assume perfect CSIT or precise signal measurements at the receiver.

(P3) *Statelessness:* users do not need to know the state of the system (e.g. its topology).

(P4) *Reinforcement:* users reinforce the transmit directions that lead to higher transmission rates.

b) Assumptions on the measurement errors: Throughout this paper, we will work with the following statistical hypotheses for the noise process $\mathbf{Z}_k(n) = \hat{\mathbf{V}}_k(n) - \mathbf{V}_k(\mathbf{Q}(n))$:

(H1) *Unbiasedness:*

$$\mathbb{E}\left[\mathbf{Z}(n+1) \,|\, \mathbf{Q}(n)\right] = 0. \tag{H1}$$

(H2) *Finite mean squared error (MSE):*

$$\mathbb{E}\left[\, \|\mathbf{Z}(n+1)\|^2 \,|\, \mathbf{Q}(n)\right] \leq \Sigma^2 \quad \text{for some } \Sigma > 0. \tag{H2}$$

The statistical hypotheses above allow us to account for a very wide range of error processes: in particular, we will *not* be assuming independent and identically distributed (i.i.d.) errors, or even errors that are a.s. bounded.[3] In fact, Hypotheses (H1) and (H2) simply amount to asking that the gradient estimate $\hat{\mathbf{V}}$ be unbiased and bounded in mean square:

$$\mathbb{E}\left[\|\hat{\mathbf{V}}_k(n+1)\|^2 \,|\, \mathbf{Q}(n)\right] \leq V_k^2 \quad \text{for some } V_k > 0. \tag{13}$$

[3]This observation is crucial in the context of wireless networks because measurement errors are typically correlated with the state of the system.

Our convergence results will be stated with only these two mild requirements in mind. That being said, even sharper results can be obtained under the hypothesis:

(H2′) *Finite errors:*

$$\|\mathbf{Z}(n)\| \leq \Sigma_n \quad \text{for some } \Sigma_n > 0. \tag{H2′}$$

From a theoretical viewpoint, (H2′) is not satisfied by error distributions with unbounded support. However, given that real-world measurements are necessarily bounded and Σ_n can become arbitrarily large in (H2′), this assumption is not particularly restrictive from a practical point of view.

Finally, we should note here that Algorithm 1 does not detail how the system's users can get an unbiased estimate $\hat{\mathbf{V}}_k$ of $\mathbf{V}_k$. To streamline our discussion, we will state our convergence results below under the assumption that there is an oracle-like mechanism that returns such estimates upon request; the construction of such a mechanism is detailed in Appendix B.

c) Computational cost: Complexity-wise, each iteration of Algorithm 1 is polynomial (with a low degree) in the number of transmit and receive antennas (for calculations at the transmitter and receiver side respectively). Specifically, the complexity of the required matrix inversion and exponentiation steps is $\mathcal{O}(N^\omega)$ and $\mathcal{O}(M_k^\omega)$ respectively, where the exponent ω is as low as 2.373 if the processing units employ fast Coppersmith–Winograd matrix multiplication methods [19].[4] The Hermitian structure of $\mathbf{W}$ can be exploited to reduce the computational cost of each iteration even further but such issues lie beyond the scope of this paper. In practice, the number of transmit and receive antennas are physically constrained by the size of the wireless array, so these operations are quite light.

By comparison, the computational bottleneck of each iteration in distributed water-filling is the calculation of the effective channel matrix $\widetilde{\mathbf{H}}_k = \mathbf{W}_k^{-1/2}\mathbf{H}_k$ of each user and, subsequently, sorting the singular values of $\widetilde{\mathbf{H}}_k$. The computational complexity (per user) of these operations is $\mathcal{O}(\max\{M_k, N\}^\omega)$ and $\mathcal{O}(M_k \log M_k)$ respectively, leading to an overall complexity of $\mathcal{O}(\max\{M_k, N\}^\omega)$.[5] In other words, (MXL) and water-filling methods not only have the same feedback requirements but also the same computational cost per iteration.

IV. Convergence Analysis

In this section, we focus on the convergence properties of Algorithm 1 under imperfect feedback. Our main result in this context is as follows:

Theorem 1. *Assume that Algorithm 1 is run with nonincreasing step sizes γ_n such that $\sum_n \gamma_n^2 < \sum_n \gamma_n = \infty$ and noisy measurements $\hat{\mathbf{V}}(n)$ satisfying Hypotheses* (H1) *and* (H2)*. Then, $\mathbf{Q}(n)$ converges to the solution set of the sum rate maximization problem* (RM) *with probability* 1.

More generally, let $\bar{R}_n = \sum_{j=1}^n \gamma_j R_j / \sum_{j=1}^n \gamma_j$ denote the empirical mean of the users' sum rate with respect to an arbitrary step-size sequence γ_n. Then:

[4] In particular, the complexity of each iteration of Algorithm 1 is that of matrix multiplication.

[5] In the PMAC case, the diagonal structure of the problem reduces the computational cost of MXL and IWF/SWF methods to linear and linearithmic time respectively, so MXL is strictly lighter in this case.

i)
$$\mathbb{E}\left[\bar{R}_n\right] \geq R_{\max} - \varepsilon_n, \tag{14}$$

ii)
$$\mathbb{P}\left(R_{\max} - \bar{R}_n \geq z\right) \leq \varepsilon_n / z \tag{15}$$

where $R_{\max} = \max_{\mathbf{Q}} R(\mathbf{Q})$ *is the system's sum capacity and*

$$\varepsilon_n = \frac{\sum_{k=1}^{K} \log M_k + \frac{1}{2} L^2 \sum_{j=1}^{n} \gamma_j^2}{\sum_{j=1}^{n} \gamma_j} \tag{16}$$

denotes the algorithm's mean performance guarantee at the n-th update period (*in the above,* M_k *is the number of transmit antennas of user k and* $L^2 = \sum_{k=1}^{K} P_k^2 V_k^2$ *is a positive constant*).

Finally, if (H2′) *also holds, the algorithm's worst-case performance is bounded by the mean guarantee* (14) *to exponential accurarcy:*

$$\mathbb{P}\left(R_{\max} - \bar{R}_n \geq \varepsilon_n + z\right) \leq \exp\left(-\frac{t_n^2 z^2}{8K^2 \sum_{j=1}^{n} \gamma_j^2 \Sigma_j^2}\right), \tag{17}$$

where $t_n = \sum_{j=1}^{n} \gamma_j$ *and* Σ_j *is given by* (H2′).

Proof: See Appendix A-2. ∎

Theorem 1 will be our core convergence result for Algorithm 1 so some remarks are in order:

a) On the choice of step-size: The use of a decreasing step-size sequence γ_n in (MXL) might appear counter-intuitive because it implies that new gradients enter the algorithm with *decreasing weights* (after all, intuition suggests that one should put more weight on recent observations rather than older, obsolete ones). However, if the algorithm has reached a near-optimal point, a constant step size might cause it to overshoot: this can be seen clearly from the mean error bound (16) which does not vanish as $n \to \infty$ for step sizes of the form $\gamma_n = \gamma$.

As a rule of thumb, the use of a (large) constant step size speeds up the algorithm at the cost of oscillations around the end state because it does not dissipate measurement noise and discretization errors. If the system's users seek to eliminate such phenomena, a decreasing step size should be preferred instead.

b) Large deviations and outage probabilities: The bounds (15) and (17) represent the probability of observing sum rates far below the channel's capacity so they can be interpreted as a measure of the system's outage probability.[6] In this context, the tail behavior of (15) shows that MXL hardens considerably around its deterministic limit: even though measurement errors can become arbitrarily large, the probability of observing sum rates much lower than what is obtainable with perfect gradient measurements decays very fast. In fact, this rate of decay is exponential if (H2′) holds: in this case, for large n, the factor $t_n^{-2} \sum_{j=1}^{n} \gamma_j^2$ which controls the width of non-negligible large deviations in (17) is of order $\mathcal{O}(1/n)$ for step-size sequences of the form $\gamma_n \propto n^{-a}$, $a \in (0, 1/2)$, and of order $\mathcal{O}(n^{2a-2})$ for $a \in (1/2, 1)$.

c) Convergence rate: If users employ a constant step-size sequence $\gamma_j = \gamma$ for a number of iterations n that is *fixed in advance*, an easy calculation shows that the minimum value of the mean guarantee (16) is attained for

[6] We note here briefly that (15) is obtained using's Markov's inequality while (17) relies on Azuma's inequality. For a detailed derivation, see Appendix A.

$\gamma_j = \gamma = L^{-1}\sqrt{2\sum_k \log M_k/n}$ and is equal to:

$$\varepsilon_n = L\sqrt{\frac{2\sum_k \log M_k}{n}}. \tag{18}$$

Put differently, running Algorithm 1 with a constant step-size as above allows user to get within $\varepsilon > 0$ of $R_{\max}$ in $n = 2L^2\sum_k \log M_k/\varepsilon^2 = \mathcal{O}(K\varepsilon^{-2})$ iterations. That said, this guarantee concerns the *empirical mean* of the system's sum rate $\bar{R}_n = n^{-1}\sum_j R_j$ and not the users' *achieved* sum rate R_n at epoch n. As we shall see in Section VII, R_n evolves much faster and converges to $R_{\max}$ within a few iterations, even for very large numbers of users and/or antennas per user.

d) Mirror descent and exponential learning: The proof of Theorem 1 relies on stochastic approximation techniques [20] and a deep connection between matrix exponentiation and the von Neumann quantum entropy. In fact, as we show in the appendix, (MXL) is closely related to the matrix regularization techniques of [21–23] for online learning and the mirror descent machinery of [16, 24] for (stochastic) convex programming. In particular, the "convergence-in-the-mean" bound (14) is derived in the same way as the corresponding results of [16, 24], but the techniques developed therein do not suffice for the much stronger almost sure convergence result that we present here. For a comprehensive treatment of mirror descent methods, see [16, 24] and references therein.

We should also note here that the exponential update map of (MXL) is reminiscent of the log-linear learning algorithm [25] which employs the Gibbs distribution in its update step. However, despite their formal similarity in the use of exponentiation, the two methods are fundamentally different: in log-linear learning, an action is chosen with probability proportional to its payoff; here, the gradient of each player's payoff function is first aggregated and *then* exponentiated. Moreover, log-linear learning was designed for (and primarily applied to) mixed strategy learning in *finite* games; by contrast, the users' rate maximization game has a continuum of actions and the players are only playing pure strategies. Applying a log-linear learning method to the problem at hand would involve players playing (Borel) mixed strategies over their continuous action sets and studying the convergence of these mixed strategies from a measure-theoretic perspective. As such, even though there have been recent advances to mixed strategy learning in abstract, infinite-dimensional settings [26, 27], there is essentially no overlap with our approach.

V. Asynchronous Learning and Further Analysis

We continue our analysis of (MXL) by discussing the algorithm's behavior under delays and/or asynchronicities and by providing a variant which removes exponentiation altogether.

A. Asynchronous updates and delays

Even though the update structure of (MXL) is fully local in nature, Algorithm 1 tacitly involves a fair degree of coordination between users in that they must all update their covariance matrices at the same time. To overcome this synchronicity limitation, we examine here an asynchronous variant of Algorithm 1 where each user updates his signal covariance matrix based on an individual – and independent – schedule.

To that end, assume that each transmitter has an update timer τ_k whose ticks trigger an update of $\mathbf{Q}_k$ (referred to as an `UpdateEvent` for user k).[7] Similarly, assume that the receiver has a timer τ_0 that triggers the measurements of the aggregate signal-plus-noise covariance matrix (a `FeedbackEvent`). Thus, at every tick of τ_k, user k measures $\mathbf{H}_k$ and updates $\mathbf{Q}_k$; likewise, at every tick of τ_0, the receiver measures the aggregate signal-plus-noise matrix $\mathbf{W}$ and feeds it back to the transmitters.

Of course, in this case, the users' gradient estimates $\hat{\mathbf{V}}_k$ may suffer from delays and asynchronicities, so the update structure of Algorithm 1 must be modified appropriately. To state this formally, let $\mathcal{K}_n \subseteq \mathcal{K}$ denote the subset of users that update their covariance matrices at the n-th overall `UpdateEvent` (typically $|\mathcal{K}_n| = 1$ if users update at random times) and let $\tau(n)$ denote the time at which this event occurs. Dually, given some $t \geq 0$, let $n^\sharp(t) = \sup\{n : \tau(n) \leq t\}$ denote the total number of `UpdateEvents` that have occurred up to time t, and define the corresponding counting functions $n_0^\sharp(t)$ and $n_k^\sharp(t)$ for `FeedbackEvents` at the receiver and `UpdateEvents` for user k. Finally, let the delay variable $d_k(n)$ denote the number of `UpdateEvents` that have elapsed between the n-th `UpdateEvent` and the last `UpdateEvent` of user k before the most recent `FeedbackEvent`.[8]

With all this in place, we obtain the following asynchronous variant of Algorithm 1 (for a pseudocode implementation, see Algorithm 2 below):

$$\begin{aligned} \mathbf{Y}_k(n+1) &= \mathbf{Y}_k(n) + \gamma_{n_k}\, \mathbb{1}(k \in \mathcal{K}_n)\, \hat{\mathbf{V}}_k(n), \\ \mathbf{Q}_k(n+1) &= P_k \frac{\exp(\mathbf{Y}_k(n+1))}{\operatorname{tr}[\exp(\mathbf{Y}_k(n+1))]}, \end{aligned} \tag{MXL-a}$$

where $n_k = \sum_{j=1}^{n} \mathbb{1}\{k \in \mathcal{K}_j\} = n_k^\sharp(\tau(n))$ is the number of updates that have been performed by user k up to epoch n and

$$\hat{\mathbf{V}}_k(n) = \mathbf{H}_k^\dagger \left[\mathbf{I} + \sum\nolimits_\ell \mathbf{H}_\ell \mathbf{Q}_\ell(n - d_\ell(n)) \mathbf{H}_\ell^\dagger\right]^{-1} \mathbf{H}_k + \mathbf{Z}_k(n), \tag{19}$$

is a noisy estimate of $\mathbf{V}_k$ (based on the most recent receiver feedback) and $\mathbf{Z}_k(n)$ is the feedback noise (assumed to satisfy Hypotheses (H1) and (H2) as before).

By construction, $\mathbf{Y}_k$ and $\mathbf{Q}_k$ are updated at the $(n+1)$-th `UpdateEvent` if and only if $k \in \mathcal{K}_n$, so every user only needs to keep track of his individual update timer τ_k. Remarkably, in this asynchronous context, we still get:

Theorem 2. *Assume that the users' delay processes $d_k(n)$ are bounded and their updates occur at a positive, finite rate – i.e. $\lim_{n\to\infty} n/\tau_k(n)$ is strictly positive and finite. Then, Algorithm 2 converges (a.s.) to the solution set of the sum rate maximization problem (RM).*

Proof: See Appendix A-3. ∎

Obviously, Algorithm 2 enjoys the same implementation properties as Algorithm 1, and, in addition:

(P5) *Asynchronicity:* there is no need for a global update timer to synchronize the network's wireless users.

[7] More precisely, we assume here that $\tau_k : \mathbb{N} \to \mathbb{R}_+$ is an increasing (and possibly random) sequence such that $\tau_k(n)$ marks the instance in time at which the k-th user updates his covariance matrix $\mathbf{Q}_k$ for the n-th time – so $\mathbf{Q}_k$ changes at $\tau_k(n)$ and remains constant throughout $[\tau_k(n), \tau_k(n+1))$.

[8] More formally (but far less intuitively), $d_k(n) = n - n^\sharp(\tau_k(n_k^\sharp(\tau_0(n_0^\sharp(\tau(n))))))$.

Algorithm 2 Asynchronous exponential learning (MXL-a).

Parameter: $\gamma > 0$.

Initialize: $n \leftarrow 0;\ n_k \leftarrow 0;\ \mathbf{Y}_k \leftarrow 0;\ \mathbf{Q}_k \leftarrow P_k/M_k\,\mathbf{I}$

Repeat

  At each `UpdateEvent`

    $n \leftarrow n + 1$;

    foreach user $k \in \mathcal{K}_n$ **do**

      $n_k \leftarrow n_k + 1$;

      get estimate $\hat{\mathbf{V}}_k$ of $\mathbf{V}_k$;

      update score matrix:

        $\mathbf{Y}_k \leftarrow \mathbf{Y}_k + \gamma/n_k\,\hat{\mathbf{V}}_k$

      based on latest `FeedbackEvent`;

      update covariance matrix:

        $\mathbf{Q}_k \leftarrow P_k \exp(\mathbf{Y}_k)/\operatorname{tr}[\exp(\mathbf{Y}_k)]$;

until termination criterion is reached.

In particular, the criteria that trigger an `UpdateEvent` could be completely arbitrary, so (MXL-a) is more suitable for scenarios where there can be no coordination between the transmitters' update periods. Otherwise, if `UpdateEvents` are triggered concurrently (e.g. once a `FeedbackEvent` occurs), Algorithm 2 reduces to synchronous MXL (Algorithm 1).

Finally, we should also note that the algorithm's information requirements are the same as in the synchronous case: the matrices $\mathbf{V}_k$ are updated based on information obtained at the latest `FeedbackEvent`, while, just as in synchronous mode, the matrix $\mathbf{H}_k$ can be estimated at the transmitter via reciprocal (downlink) transmission when operating in TDD mode.

B. Matrix exponential learning with no exponentiation

We close this section by describing an alternative, eigen-based implementation of Algorithm 1 which does not require a matrix exponentiation step. The key ingredient of our analysis is the following proposition:

Proposition 1. *Let* $\{q_{k\alpha}, \mathbf{u}_{k\alpha}\}_{\alpha=1}^{M_k}$ *be a smooth eigen-system for* $\mathbf{Q}_k$ *and let* $V^k_{\alpha\beta} \equiv \mathbf{u}^\dagger_{k\alpha}\mathbf{V}_k\mathbf{u}_{k\beta}$. *Then, the iterates of Algorithm 1 track the mean dynamics:*

$$\dot{q}_{k\alpha} = q_{k\alpha}\left(V^k_{\alpha\alpha} - P_k^{-1}\textstyle\sum_{\beta=1}^{M_k} q_{k\beta}V^k_{\beta\beta}\right), \tag{20a}$$

$$\dot{\mathbf{u}}_{k\alpha} = \sum\nolimits_{\beta\neq\alpha} V^k_{\beta\alpha}\left(\log q_{k\alpha} - \log q_{k\beta}\right)^{-1}\mathbf{u}_{k\beta}. \tag{20b}$$

Proof: See Appendix A-3. ∎

Algorithm 3 Eigen-based exponential learning (MXL-e).

Parameter: decreasing step-size sequence γ_n

Initialize: $n \leftarrow 0$; $q_{k\alpha}$; $\mathbf{u}_{k\alpha}$

Repeat

$n \leftarrow n + 1$;

for each user $k \in \mathcal{K}$ **do simultaneously**

measure $\mathbf{V}_k$;

update eigenvalues:

$q_{k\alpha} \leftarrow q_{k\alpha} + \gamma_n q_{k\alpha} \left(V^k_{\alpha\alpha} - P_k^{-1} \sum_{\beta=1}^{M_k} q_{k\beta} V^k_{\beta\beta} \right)$;

update eigenvectors:

$\mathbf{u}_{k\alpha} \leftarrow \mathbf{u}_{k\alpha} + \gamma_n \sum_{\beta \neq \alpha} V^k_{\beta\alpha} (\log q_{k\alpha} - \log q_{k\beta})^{-1} \mathbf{u}_{k\beta}$;

correct roundoff errors:

$\mathbf{u} \leftarrow \texttt{Orthonormal}(\mathbf{u})$;

update covariance matrix: $\mathbf{Q}_k \leftarrow \sum_{\alpha=1}^{M_k} q_{k\alpha} \mathbf{u}_{k\alpha} \mathbf{u}_{k\alpha}^\dagger$;

until termination criterion is reached.

The precise sense in which $\mathbf{Q}(n)$ "tracks" the mean dynamics (20) is explained in Appendix A-1; for our purposes, the most important consequence of Proposition 1 is that (20) leads to the eigen-based Algorithm 3 (see below). As in the case of the original MXL algorithm, we then obtain:

Theorem 3. *Assume that Algorithm 3 is run with sufficiently small, nonincreasing step-sizes γ_n such that $\sum_n \gamma_n^2 < \sum_n \gamma_n = \infty$. Then, $\mathbf{Q}(n)$ converges (a.s.) to $\arg\max_{\mathbf{Q}} R(\mathbf{Q})$.*

We close this section with a few remarks on Algorithm 3:

a) The orthonormalization step: Even though the eigenvector dynamics (20b) preserve orthonormality, Algorithm 3 introduces an $\mathcal{O}(\gamma_n^2)$ round-off error to orthogonality due to discretization. The call to `Orthonormal` performs a basis orthonormalization and it is intended to correct that error in order to yield a covariance matrix $\mathbf{Q}_k$ that satisfies the feasibility constraints (7) of (RM). Like matrix exponentiation, orthonormalization has the same complexity as matrix multiplication (fast Coppersmith–Winograd methods [19] provide an $\mathcal{O}(M_k^{2.373})$ bound), so this does not impact the algorithm's (polynomial) complexity.

b) Noisy measurements: Theorem 3 has been stated for simplicity for noiseless measurements. In the case of noisy measurements, the step-size of the algorithm must be tuned adaptively so that $q_{k\alpha} \geq 0$. It is not hard to do so by using the random step-size techniques of [20] but we chose to focus on the noiseless case for presentational clarity.

VI. The Case of Fast-Fading Channels

In the presence of fading, the users' channel matrices $\mathbf{H}_k$ evolve stochastically over time at a rate which is much faster than the characteristic length of each transmission block; as a result, the static sum rate function R of (8) is no

Algorithm 4 MXL for ergodic channels.

Parameter: decreasing step-size sequence γ_n

Initialize: $n \leftarrow 0$; $\mathbf{Y}_k \leftarrow 0$; $\mathbf{Q}_k \leftarrow \frac{P_k}{M_k}\mathbf{I}$

Repeat

$n \leftarrow n+1$;

for each user $k \in \mathcal{K}$ **do simultaneously**

get estimate $\hat{\mathbf{V}}_k$ of the instantaneous matrix $\mathbf{V}_k$ of (22);

update score matrix:

$\mathbf{Y}_k \leftarrow \mathbf{Y}_k + \gamma_n \hat{\mathbf{V}}_k$;

set covariance matrix:

$\mathbf{Q}_k \leftarrow P_k \ \exp(\mathbf{Y}_k) / \operatorname{tr}[\exp(\mathbf{Y}_k)]$;

until termination criterion is reached.

longer relevant. In this case, the users' achievable sum rate for fixed $\mathbf{Q}$ is given by the ergodic average [2, 28]:

$$R_{\text{erg}}(\mathbf{Q}) = \mathbb{E}_{\mathbf{H}}\left[\log\det\left(\mathbf{I} + \textstyle\sum_k \mathbf{H}_k\mathbf{Q}_k\mathbf{H}_k^\dagger\right)\right], \tag{21}$$

where the expectation is now taken with respect to the law of $\mathbf{H}$ (assumed here to follow a stationary, ergodic process). Accordingly, we obtain the ergodic rate maximization problem for fast-fading channels:

$$\begin{aligned} \text{maximize} \quad & R_{\text{erg}}(\mathbf{Q}), \\ \text{subject to} \quad & \mathbf{Q}_k \in \mathcal{Q}_k, \ k = 1, \dots, K, \end{aligned} \tag{ERM}$$

where the users' feasible sets $\mathcal{Q}_k$ are defined as in (7): $\mathcal{Q}_k = \{\mathbf{Q}_k \succcurlyeq 0 : \operatorname{tr}(\mathbf{Q}_k) \leq P_k\}$.[9]

Since expectation preserves convexity, the ergodic rate maximization problem (ERM) remains concave – in fact, it is straightforward to show that (ERM) is *strictly* concave [29]. However, given that the integration over the law of $\mathbf{H}$ is typically impossible to carry out, calculating the ergodic gradient $\mathbf{V}_{\text{erg}} = \nabla R_{\text{erg}}$ of R_{erg} is a likewise impractical task. Thus, instead of relying on intricate analytic calculations (that require substantial computation capabilities and a good knowledge of the channels' statistics), we will consider the same sequence of events as in the case of static channels:

1) At every update period $n = 1, 2, \dots$, each user $k \in \mathcal{K}$ gets an estimate $\hat{\mathbf{V}}_k(n)$ of the matrix

$$\mathbf{V}_k(n) = \mathbf{H}_k^\dagger(n)\left[\mathbf{I} + \sum\nolimits_\ell \mathbf{H}_\ell(n)\mathbf{Q}_\ell(n)\mathbf{H}_\ell^\dagger(n)\right]^{-1}\mathbf{H}_k(n), \tag{22}$$

where $\mathbf{H}_k(n)$ denotes the instantaneous realization of the channel matrix of user k at period n.

2) Users update their signal covariance matrices according to (MXL) and the process repeats.

Formally, writing $\mathbf{Z}_k(n) = \hat{\mathbf{V}}_k(n) - \mathbb{E}[\mathbf{V}_k(n)]$ for the difference between the users' observed estimate $\hat{\mathbf{V}}_k(n)$ and the expected value of (22), we will make the same statistical hypotheses for $\mathbf{Z}$ as in the static regime – though we

[9] Perhaps more appropriately for the case of interest, the above expression also holds over the long term for block-fading channels, in which case the channel is essentially fixed over each transmission length during which the instantaneous rate is used.

TABLE I

WIRELESS NETWORK SIMULATION PARAMETERS

Parameter	Value
Cell radius	1 km
Time frame duration	5 ms
Wireless propagation model	COST Hata [31]
Central frequency	2.5 GHz
Spectral noise density	−174 dBm/Hz
User speed (for mobility)	{3, 5, 50} km/h
Maximum transmit power	$P = 30$ dBm

should note here that fluctuations are now due to *both measurement errors and the channels' inherent variability.* Quite remarkably, despite the change of objective function, we obtain the following convergence result for fast-fading channels:

Theorem 4. *Assume that* (MXL) *is run with nonincreasing step-sizes* γ_n *such that* $\sum_n \gamma_n^2 < \sum_n \gamma_n = \infty$ *and noisy measurements* $\hat{\mathbf{V}}(n)$ *satisfying hypotheses* (H1) *and* (H2) *with respect to* (22)*. Then,* $\mathbf{Q}(n)$ *converges* (*a.s.*) *to the solution of the ergodic rate maximization problem* (ERM)*; moreover, the conclusions of Theorem 1 for an arbitrary step-size sequence* γ_n *also hold with the static sum rate* R *replaced by the ergodic sum rate* R_{erg}.

Proof: See Appendix A-4. ∎

Remark 1. In view of Theorem 4, we see that Algorithm 2 enjoys the additional property:

(P6) *Flexibility:* the MXL algorithm can be applied "as-is" in both static and fast-fading channels.

In particular, the same convergence rate and large deviation estimates that were derived for static channels in the previous section (cf. the remarks following Theorem 1) also carry over to the fast-fading regime. The only difference here is that the variance Σ that appears e.g. in (15) and (17) is not only due to imperfections in the estimation process of $\mathbf{V}$, but also stems from the inherent variability of the system's channels due to fast-fading. We will explore this issue in the following section.

Remark 2. In a very recent paper [30], Yang et al. proposed a successive stochastic-approximation-based algorithmic framework for solving general (not necessarily convex) multi-agent stochastic optimization problems. Their proposed method relies on an iterative stochastic best-response scheme, and they provide a set of sufficient conditions under which the method converges – at the cost of calculating a best response at each iteration of the algorithm. They subsequently apply this method to the problem of ergodic sum rate maximization in MIMO multiple access channels and show that any limit points of the method are (a.s.) solutions of (RM). A comprehensive comparison between the two methods would take us too far afield, so we delegate it to future work.

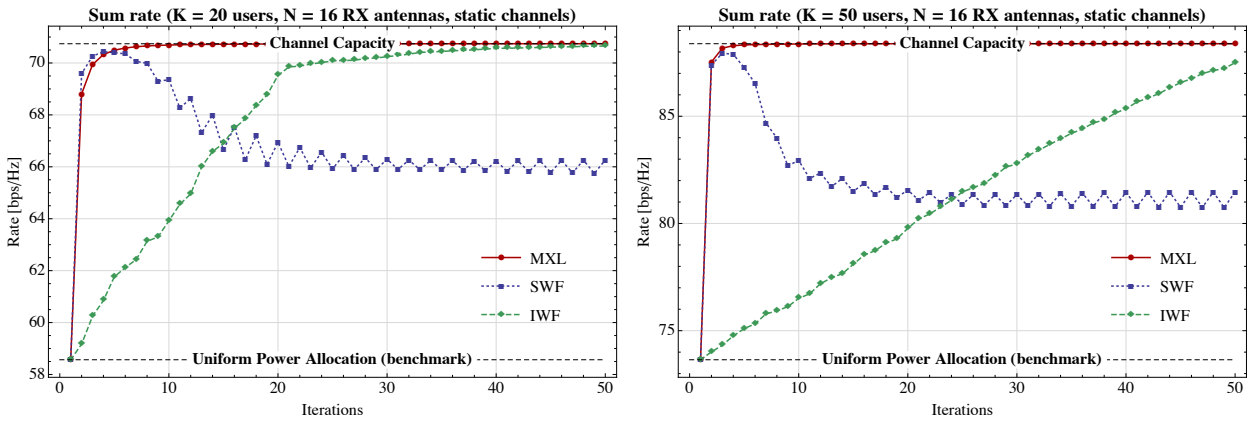


Fig. 1. Comparison of matrix exponential learning (MXL) to water-filling (WF) methods. The classical IWF algorithm converges relatively slowly (roughly within $\mathcal{O}(K)$ iterations) because only one user updates per cycle; the SWF variant is much faster (because all users updates simultaneously), but it may fail to converge due to the appearance of best-response cycles in the update process. By contrast, we see that MXL converges within a few iterations, even for large numbers of users.

VII. Numerical Results

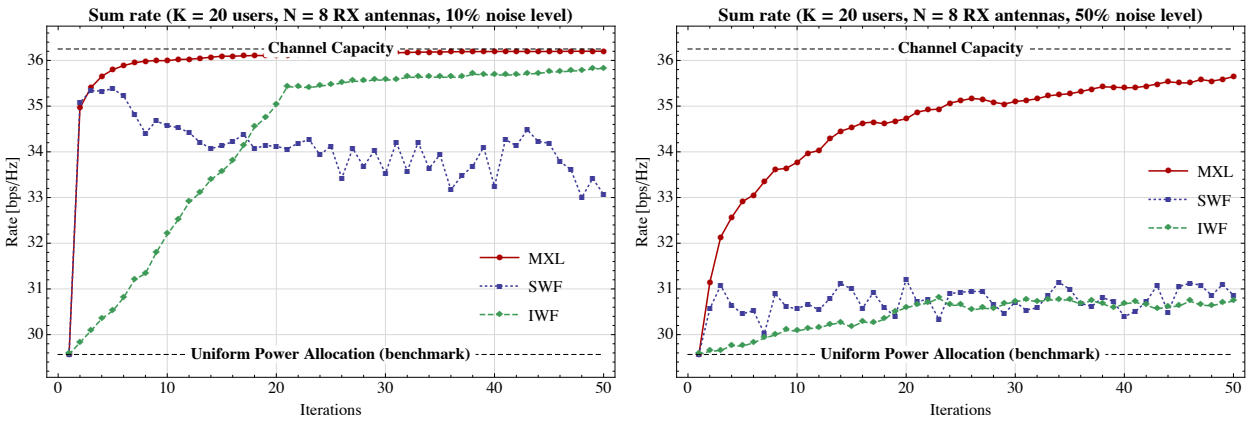
To assess the performance of MXL in practical scenarios, we conducted extensive numerical simulations from which we illustrate here a selection of the most representative cases. Specifically, we examined *a*) the algorithm's convergence speed; *b*) its robustness to feedback imperfections and mobility; *c*) its scalability; and *d*) its computational cost, comparing it at each case to state-of-the-art water-filling (WF) methods.

Our basic simulation setup is as follows: we consider a cellular wireless network occupying a 11 kHz band around a central frequency of $f_c = 2.5$ GHz. Wireless signal propagation is modeled following the COST Hata model [31, 32] for moderately dense urban environments with characteristics as in Table I. Network coverage is provided by a base station (BS) with a coverage radius of 1 km and we focus on the uplink of K wireless transmitters that are connected to said BS. Communication occurs over a TDD scheme with frame duration $T_f = 5$ ms and the transmitters have a maximum transmit power of 30 dBm. For a detailed overview of simulation parameters, see Table I.

First, in Figure 1, we investigate the convergence speed of the MXL algorithm (Algorithm 1) as a function of the number of wireless transmitters and transmit/receive antennas, using state-of-the-art water-filling (WF) methods as a benchmark. For concreteness, we compared the evolution of MXL to that of IWF/SWF for a system consisting of a base MIMO terminal with $N = 16$ receive antennas and $K = \{20, 50\}$ wireless users. We then plotted the users' Shannon rate (8) at each iteration; for comparison, we also plotted the channel's sum capacity and the users' sum rate under uniform power allocation.

As can be seen in Figure 1, the MXL algorithm attains the system's sum capacity within a few iterations (essentially within a single iteration for $K = 50$ users).[10] This convergence behavior represents a marked improvement over

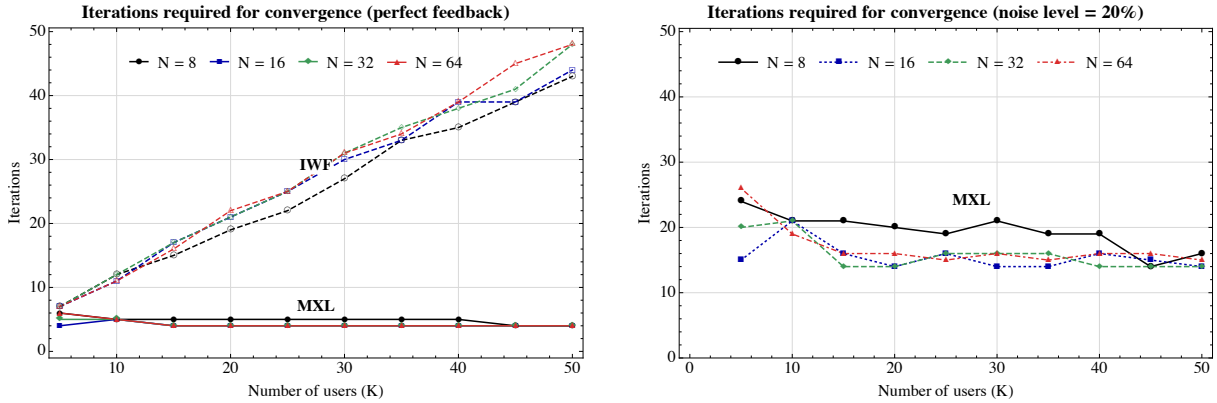
[10] Alternatively, in the game-theoretic context of (NE), this implies that the system's users reach a unilaterally stable Nash equilibrium.



(a) Learning with a relative error level of 10%.

(b) Learning with a relative error level of 50%.

Fig. 2. Performance of matrix exponential learning and water-filling (WF) methods under imperfect feedback. In contrast to WF methods, the MXL algorithm attains the channel's sum capacity, even in the presence of very high measurement errors and feedback noise.



(a) Learning with perfect feedback

(b) Learning with imperfect feedback

Fig. 3. Scalability of MXL under perfect and imperfect feedback (Figures 3(a) and 3(b) respectively). The convergence threshold was set to 99% of the system's sum capacity and the number of iterations required for convergence was averaged over 100 realizations. In Figure 3(a), we also plotted the corresponding data for the IWF algorithm (dashed lines with open markers). Similar convergence data for SWF was not plotted because SWF often fails to converge altogether (so the number of iterations required for convergence is effectively infinite); data for IWF in the case of imperfect feedback was not plotted for the same reason.

traditional WF methods, even in moderately-sized systems with $K = 20$ users. First, IWF is much slower than MXL (it requires $\mathcal{O}(K)$ iterations to achieve the same performance level as the first iteration of MXL). Second, SWF may fail to converge altogether due to "ping-pong" effects that occur when the users change transmit eigenvalues at the same time. By contrast, MXL converges very quickly, even for large numbers of users and/or antennas per user.

In Figure 2, we investigate the robustness of MXL under imperfect feedback and we compare it to IWF and SWF methods under similar conditions. Specifically, in Figure 2, we simulated a multi-user uplink MIMO system with $N = 16$ antennas at the receiver end and $K = 20$ wireless transmitters with antenna and channel characteristics as before (see also Table I). To simulate imperfections to the users' feedback, the measurement noise was controlled by

the relative error level of the estimator (deviation/mean). As a result, a relative error level of η means that, on average, the estimated matrix lies within $\eta\%$ of its true value. We then plotted the efficiency of MXL over time for relative noise levels of $\eta = 10\%$ and $\eta = 50\%$, and we ran the iterative and simultaneous WF algorithms with the same sample realizations for comparison.

As can be seen in Figure 2, the performance of IWF and SWF remains acceptable at low error levels, allowing users to attain between 90% and 95% of the channel's sum capacity. However, when the feedback noise gets higher, water-filling methods offer no perceptible advantage over uniform power allocation. By contrast, as predicted by Theorem 1, the MXL algorithm converges to the system's sum capacity, even with very noisy feedback – though, of course, the algorithm slows down when the measurement noise grows too high.

The scalability and robustness of MXL is further examined in Figure 3 where we plot the number of iterations required for users to attain 99% of the system's sum capacity. More precisely, for each value of K and N in Figure 3, we ran the MXL algorithm for 100 network instantiations (with simulation parameters as before) and we plotted the average number of iterations required to attain 99% of the network's capacity. This process was repeated for both perfect and imperfect feedback (with a 20% relative error level), and the results were plotted in Figures 3(a) and 3(b) respectively. For comparison purposes, we also plotted the number of iterations required for the convergence of IWF in the case of perfect feedback; since SWF often fails to converge, it was not included in our benchmark considerations (and likewise for IWF under imperfect feedback).

As can be seen in Figure 3, MXL scales very well with the number of users (and/or anntenas per user), achieving the system's sum capacity within (roughly) the same number of iterations. In fact, MXL is *faster* in larger systems because users can employ a more aggressive step-size policy.[11] Of course, in the case of imperfect feedback (Figure 3(b)), users have to be less aggressive because erroneous observations can perturb the algorithm's performance. For this reason, MXL with imperfect feedback converges more slowly, but it still attains the system's sum capacity within roughly the same number of iterations, independently of the number of users and/or antennas per user in the system. By contrast, IWF and SWF fail to converge altogether in this case, so the corresponding convergence data was not plotted in Figure 3(b).

The (per user) computational cost of each iteration of MXL is examined in Figure 4. Specifically, in Figure 4, we focused on a system with $N \in [4, 64]$ receive antennas and $K = 50$ transmitters, each with a number of transmit antennas drawn randomly between 2 and $N/2$. We then plotted the CPU time required to perform one iteration of MXL (per user) on a typical mid-range commercial laptop, averaging over 100 system realizations. For comparison, we also plotted the corresponding computation times for iterative and simultaneous water-filling (always per user and per iteration). As can be seen, the computational cost of MXL lies between that of IWF and SWF and is quite low, even for large number of antennas per user. Specifically, the computational time required to perform one iteration of MXL is well below the typical TDD frame duration ($\delta = 5$ ms), even for several tens of transmit/receive antennas.

[11] In large systems, the optimal signal covariance profile $\mathbf{Q}^*$ has many zero eigenvalues. As a result, using a very large step-size allows users to approach $\mathbf{Q}^*$ within very few iterations, with no danger of oscillations around $\mathbf{Q}^*$.

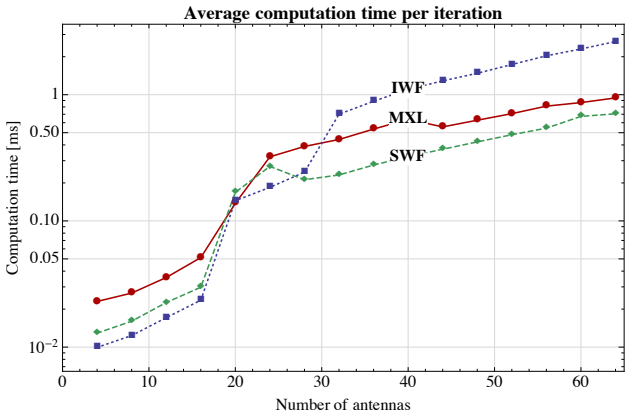


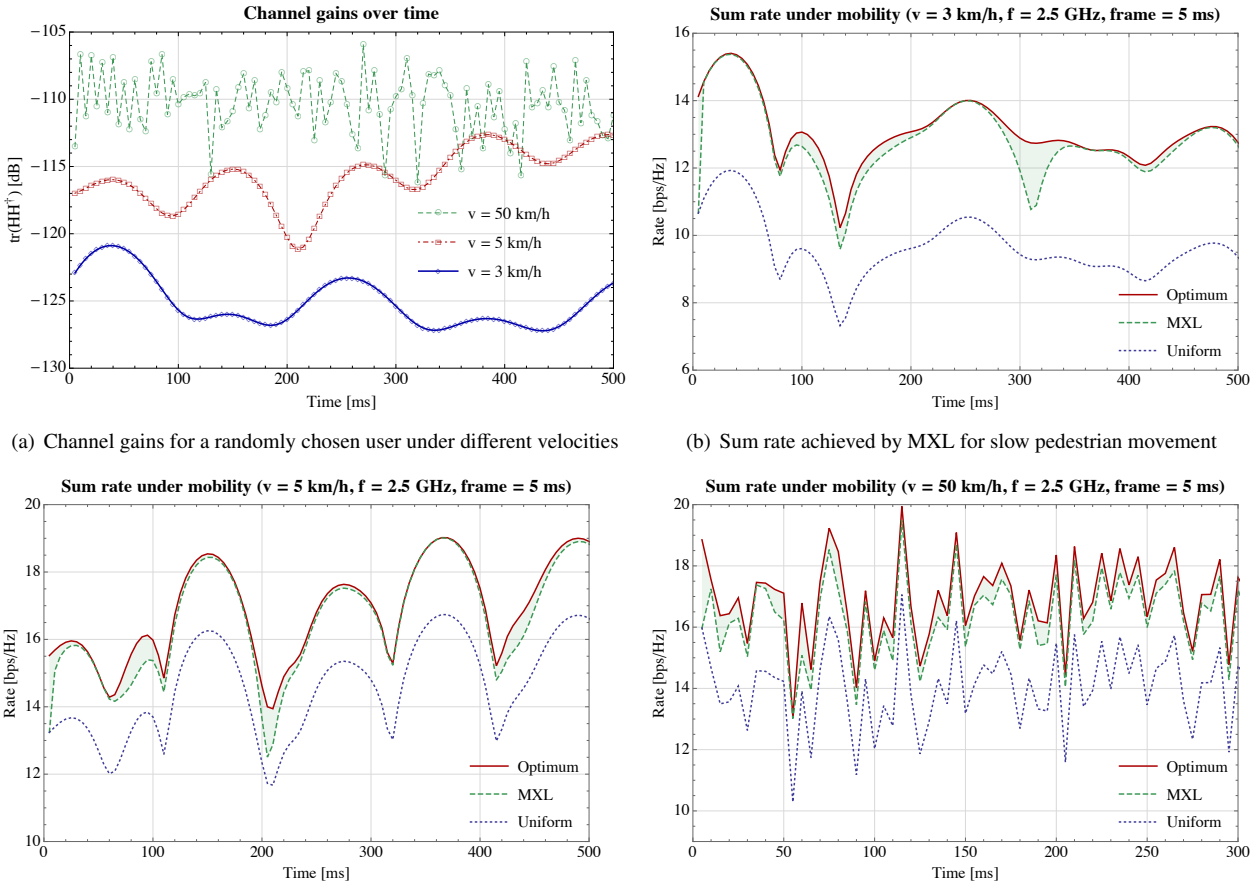
Fig. 4. Average computation time per user and per iteration. Each iteration of MXL exhibits the same complexity behavior as water-filling methods.

Finally, to account for mobility and time-varying network conditions, we also considered in Figure 5 the case of mobile users whose channels vary with time due to Rayleigh fading, path loss fluctuations, etc. As before, we focused on a MIMO MAC system with $M = 16$ antennas at the receiver and $K = 20$ mobile users, all moving with the same average speed. For simulation purposes, we used the extended typical urban (ETU) model for the users' environment and the extended pedestrian A (EPA) and extended vehicular A (EVA) models to simulate pedestrian (3–5 km/h) and vehicular (30–130 km/h) movement respectively [33]; for reference, the total channel gain $\mathrm{tr}(\mathbf{H}_k\mathbf{H}_k^\dagger)$ of a randomly selected user $k \in \mathcal{K}$ is shown in Figure 5(a).

We then ran the MXL algorithm with an update period of $\delta = 5$ ms (i.e. one update per frame), and we plotted the algorithm's sum rate R_n at the n-th iteration of the algorithm for user velocities $v = 3$, 5, and 50 km/h (corresponding to slow pedestrian, average pedestrian and average vehicular movement respectively). For comparison, we also plotted *a*) the system's sum capacity $R_n^{\max}$ (given the current realization of the channel matrices $\mathbf{H}(t)$ at time $t = n\delta$); and *b*) the users' sum rate under uniform power allocation. Thanks to its high convergence speed, the MXL algorithm tracks the system's sum capacity remarkably well, even under rapidly changing channel conditions. Moreover, the large difference in throughput between the learned covariance profile (under MXL) and uniform power allocation shows that there is a substantial benefit (of the order of 100% or more) in using MXL to track the system's (dynamically changing) optimum transmit profile.

VIII. Conclusions and Perspectives

In this paper, we introduced a distributed signal covariance optimization algorithm to maximize the uplink sum-capacity of multi-antenna users that transmit to a common multi-antenna receiver with only imperfect, possibly delayed and asynchronously updated channel state information at the users' disposal. Under very mild hypotheses on the statistics of the estimation imperfections, we showed that the proposed matrix exponential learning (MXL) algorithm converges rapidly, even for large numbers of users and/or antennas per user. Moreover, the probability that the algorithm deviates beyond a small error from the optimum after a fixed number of iterations is very small (and decays exponentially if the measurement errors are bounded in norm). In our view, these robustness properties of



(a) Channel gains for a randomly chosen user under different velocities

(b) Sum rate achieved by MXL for slow pedestrian movement

(c) Sum rate achieved by MXL for fast pedestrian movement

(d) Sum rate achieved by MXL for average vehicular speed

Fig. 5. Data rates achieved by MXL in a dynamic environment with time-varying channels. The dynamic transmit policy induced by the MXL algorithm allows users to track the system's sum capacity remarkably well, even under rapidly changing channel conditions. The users' achieved sum rate tracks its (evolving) maximum value remarkably well, even under rapidly changing channel conditions.

MXL make it an attractive alternative to water-filling methods which may fail to converge altogether in the presence of feedback noise. This is confirmed by extensive numerical simulations that illustrate the fast convergence and robustness properties of MXL in realistic channel conditions.

We focused on multiple access channels only for simplicity. The proposed MXL algorithm can be readily extended to a MIMO–OFDMA framework, different precoding schemes (such as MMSE or ZF-type precoders), or to account for other transmission features – such as spectral mask constraints, pricing, etc. The method can also be adapted to wide-range channel models (such as the interference channel) where a game-theoretic approach as in [9, 13] is more appropriate. In this context, a natural question that arises is whether the algorithm converges to a Nash equilibrium and whether this convergence is retained in the presence of noise. In fact, our method can be adapted to much more general constrained matrix optimization problems [34] arising in signal processing (such as prior estimation in phase retrieval problems [35]); we plan to explore these directions in future work.

Appendix A
Technical Proofs

Our goal in this appendix will be to prove the convergence results presented in the rest of our paper. To that end, we will first establish the convergence of a deterministic, "mean-field" dynamical system associated (MXL) and we will then show that (MXL) and its variants comprise a stochastic approximation of these dynamics in the sense of [20].

For notational clarity, in what follows (and unless explicitly stated otherwise), we will treat the case of a single user with maximum transmit power $P = 1$; the general case is simply a matter of taking a direct sum over $k \in \mathcal{K}$ and rescaling by the corresponding maximum power P_k.

1) Preliminaries: We begin by establishing some preliminary results that will be used in our discrete-time analysis later on. A key ingredient for our proofs will be the von Neumann (negative) entropy:

$$h(\mathbf{Q}) = \operatorname{tr}\left[\mathbf{Q}\log\mathbf{Q}\right], \quad \mathbf{Q} \in \mathcal{D}, \tag{23}$$

and its convex conjugate (Legendre transform) [36]:

$$h^*(\mathbf{Y}) \equiv \max_{\mathbf{Q}\in\mathcal{D}} \{\operatorname{tr}[\mathbf{Y}\mathbf{Q}] - h(\mathbf{Q})\} = \log\operatorname{tr}\left[\exp(\mathbf{Y})\right], \tag{24}$$

where $\mathcal{D}$ denotes the (compact) spectrahedron:

$$\mathcal{D} = \{\mathbf{Q} \succcurlyeq 0 : \operatorname{tr}(\mathbf{Q}) = 1\}. \tag{25}$$

It will also be convenient to introduce the following "primal-dual" coupling between $\mathbf{Q}$ and $\mathbf{Y}$:

$$F(\mathbf{Q},\mathbf{Y}) = h(\mathbf{Q}) + h^*(\mathbf{Y}) - \operatorname{tr}\left[\mathbf{Q}\mathbf{Y}\right]. \tag{26}$$

The above expression gathers all the terms of Fenchel's inequality [37], so, following [38], we will refer to it as the *Fenchel coupling* between $\mathbf{Q}$ and $\mathbf{Y}$. Below, we present some key properties of F:

Lemma 1. *With notation as above, we have $F(\mathbf{Q},\mathbf{Y}) \geq 0$ with equality iff $\mathbf{Q} = \exp(\mathbf{Y})/\operatorname{tr}[\exp(\mathbf{Y})]$. Moreover:*

$$\nabla_{\mathbf{Y}} h^*(\mathbf{Y}) = \frac{\exp(\mathbf{Y})}{\operatorname{tr}\left[\exp(\mathbf{Y})\right]}, \tag{27}$$

and

$$\nabla_{\mathbf{Y}} F(\mathbf{Q},\mathbf{Y}) = \frac{\exp(\mathbf{Y})}{\operatorname{tr}[\exp(\mathbf{Y})]} - \mathbf{Q}. \tag{28}$$

Proof: By standard matrix analysis results [39], we have:

$$\nabla_{\mathbf{Y}} h^*(\mathbf{Y}) = \frac{1}{\operatorname{tr}[\exp(\mathbf{Y})]}\nabla_{\mathbf{Y}}\operatorname{tr}[\exp(\mathbf{Y})] = \frac{\exp(\mathbf{Y})}{\operatorname{tr}[\exp(\mathbf{Y})]}, \tag{29}$$

and (28) follows trivially. The first part of our claim is then a consequence of the general theory of convex conjugation – see e.g. [37, Chap. 26]. ∎

With this result at hand, we proceed to establish the convergence of the following continuous-time version of (MXL):

$$\begin{aligned} \dot{\mathbf{Y}} &= \mathbf{V}, \\ \mathbf{Q} &= \frac{\exp(\mathbf{Y})}{\operatorname{tr}\left[\exp(\mathbf{Y})\right]}. \end{aligned} \tag{MXL-c}$$

As it turns out, (MXL-c) converges to the solution set of the rate maximization problem (RM):

Theorem 5. *Let* $\mathbf{Q}(t)$ *be a solution orbit of* (MXL-c)*. Then,* $\mathbf{Q}(t)$ *converges to a minimizer of* (RM)*.*

Proof of Theorem 5: Our proof relies on the fact that the Fenchel coupling $H(t) = F(\mathbf{Q}^*, Y(t))$ is a Lyapunov function for (MXL-c) whenever $\mathbf{Q}^*$ maximizes (RM). Indeed, the definition of the Fenchel coupling and Lemma 1 yield

$$\dot{H} = \operatorname{tr}\left[\nabla_{\mathbf{Y}} h^*(\mathbf{Y}) \cdot \dot{\mathbf{Y}}\right] - \operatorname{tr}\left[\mathbf{Q}^* \dot{\mathbf{Y}}\right] = \operatorname{tr}\left[(\mathbf{Q} - \mathbf{Q}^*) \cdot \nabla_{\mathbf{Q}} R\right] \leq 0, \tag{30}$$

where the inequality in the last step follows from the concavity of R and the fact that $\mathbf{Q}^*$ is a maximizer of R. Moreover, equality in (30) holds if and only if $\mathbf{Q}$ is also a maximizer of R, so H is a Lyapunov function for (MXL-c) with respect to $\arg\max R$.

The above reasoning shows that (MXL-c) converges to $\arg\max R$, but since R is not necessarily strictly concave, this does not imply that every trajectory of (MXL-c) converges to a specific point in $\arg\max R$. To show that this is indeed the case, let $\mathbf{Q}(t)$ be an orbit of (MXL-c) and let $\mathbf{Q}^*$ be an ω-limit of $\mathbf{Q}(t)$, i.e. $\mathbf{Q}(t_n) \to \mathbf{Q}^*$ for some increasing sequence $t_n \to \infty$. By Lemma 1, this implies that $F(\mathbf{Q}^*, \mathbf{Y}(t_n)) \to 0$, so, since $F(\mathbf{Q}^*, \mathbf{Y}(t))$ is nonincreasing, we also get $\lim_{t\to\infty} F(\mathbf{Q}^*, \mathbf{Y}(t)) = 0$. We conclude that $\mathbf{Q}(t) \to \mathbf{Q}^*$ (again by Lemma 1) and our proof is complete. ∎

We now proceed to show that the iterates of (MXL) are asymptotically close to solution segments of (MXL-c) of arbitrary length – more precisely, that they comprise an *asymptotic pseudotrajectory* of (MXL-c) in the sense of [20].

Proposition 2. *Assume that* (MXL) *is run with a nonincreasing step-size sequence* γ_n *such that* $\sum_n \gamma_n^2 < \sum_n \gamma_n = +\infty$ *and noisy measurements* $\hat{\mathbf{V}}_k$ *satisfying* (H1) *and* (H2)*. Then, the iterates* $\mathbf{Q}(n)$ *of* (MXL) *form an asymptotic pseudotrajectory of* (MXL-c)*.*

Proof: Simply note that the recursion (MXL) can be written in the form:

$$\mathbf{Y}(n+1) = \mathbf{Y}(n) + \gamma_n \left[\mathbf{V}(\mathbf{Q}(n)) + \mathbf{Z}(n)\right]. \tag{31}$$

Since the map $\mathbf{Y} \mapsto \mathbf{Q}$ is Lipschitz continuous[12] and the rate function $R(\mathbf{Q})$ is smooth over $\mathcal{D}$, it follows that the map $\mathbf{Y} \mapsto \mathbf{V}(\mathbf{Q}(\mathbf{Y}))$ is Lipschitz and bounded. Our claim then follows from Propositions 4.2 and 4.1 in [20]. ∎

2) Convergence of MXL with noisy feedback: In this section, we prove Theorem 1:

Proof of Theorem 1: Let $\mathcal{D}^* = \arg\max_{\mathbf{Q}\in\mathcal{D}} R(\mathbf{Q})$ denote the solution set of (RM) and assume ad absurdum that $\mathbf{Q}(n)$ remains a bounded distance away from $\mathcal{D}^*$. Furthermore, fix some $\mathbf{Q}^* \in \mathcal{D}^*$ and let $D_n = F(\mathbf{Q}, \mathbf{Y}(n))$; a Taylor expansion of F then yields:

$$\begin{aligned} D_{n+1} &= F(\mathbf{Q}^*, \mathbf{Y}(n+1)) = F(\mathbf{Q}^*, \mathbf{Y}(n) + \gamma_n \hat{\mathbf{V}}(n)) \\ &\leq D_n + \gamma_n \operatorname{tr}[(\mathbf{Q}(n) - \mathbf{Q}^*)\, \mathbf{V}(\mathbf{Q}(n))] + \gamma_n \xi_n + \tfrac{1}{2}\gamma_n^2 \|\hat{\mathbf{V}}(n)\|^2, \end{aligned} \tag{32}$$

[12]This follows from the fact that the von Neumann entropy (23) is strongly convex with respect to the nuclear norm [16, 22].

where $\xi_n = \mathrm{tr}[\mathbf{Z}(n)\,(\mathbf{Q}^* - \mathbf{Q}(n))]$ and we have used the fact that the convex conjugate h^* of the von Neumann entropy is 1-strongly smooth in the induced L^1 norm [22].

Our original assumption that $\mathbf{Q}(n)$ remains a bounded distance away from $\boldsymbol{\mathcal{D}}^*$ means that D_n is bounded away from zero; moreover, with R concave and smooth, we will also have $\mathrm{tr}[\mathbf{V}(n)\,(\mathbf{Q}(n) - \mathbf{Q}^*)] \leq -m$ for some $m > 0$. Thus, telescoping (32) yields:

$$D_{n+1} \leq D_0 - t_n\left(m - \sum\nolimits_{j=1}^{n} w_{j,n}\,\xi_j\right) + \frac{1}{2}\sum\nolimits_{j=1}^{n}\gamma_j^2\|\hat{\mathbf{V}}(j)\|^2, \tag{33}$$

where $t_n = \sum_{j=1}^{n}\gamma_j$ and $w_{j,n} = \gamma_j/t_n$. By the strong law of large numbers for martingale differences [40, Theorem 2.18], we have $n^{-1}\sum_{j=1}^{n}\xi_j \to 0$ (a.s.); hence, with $\gamma_{n+1}/\gamma_n \leq 1$, Hardy's Tauberian summability criterion [41, p. 58] applied to the weight sequence $w_{j,n} = \gamma_j/t_n$ yields $\sum_{j=1}^{n} w_{j,n}\,\xi_j \to 0$ (a.s.). Finally, since γ_n is square-summable and $\gamma_n\mathbf{Z}(n)$ is a martingale difference with finite variance, it follows that $\sum_{n=1}^{\infty}\gamma_n^2\|\hat{\mathbf{V}}(n)\|^2 < \infty$ (a.s.) by Theorem 6 in [42].

Combining all of the above, we see that the RHS of (33) tends to $-\infty$ (a.s.); this contradicts the fact that $D_n \leq 0$, so we conclude that $\mathbf{Q}(n)$ visits a compact neighborhood of $\boldsymbol{\mathcal{D}}^*$ infinitely often. Since $\boldsymbol{\mathcal{D}}^*$ attracts any initial condition $\mathbf{Y}(0)$ under the continuous-time dynamics (MXL), Theorem 6.10 in [20] shows that $\mathbf{Q}(n)$ converges to $\boldsymbol{\mathcal{D}}^*$, as claimed.

For the bound (14), note that (32) can be rewritten as

$$\gamma_n\,\mathrm{tr}[(\mathbf{Q}^* - \mathbf{Q}(n))\,\mathbf{V}(\mathbf{Q}(n))] \leq D_n - D_{n+1} + \gamma_n\xi_n + \frac{1}{2}\gamma_n^2\|\hat{\mathbf{V}}(n)\|^2, \tag{34}$$

so, recalling that R is concave and $\mathbf{V} = \nabla_{\mathbf{Q}}R$, we get:

$$\begin{aligned}\gamma_n\,[R_{\max} - R_n] &\leq \gamma_n\,\mathrm{tr}[(\mathbf{Q}^* - \mathbf{Q}(n))\,\mathbf{V}(\mathbf{Q}(n))]\\ &\leq D_n - D_{n+1} + \gamma_n\xi_n + \frac{1}{2}\gamma_n^2\|\hat{\mathbf{V}}(n)\|^2.\end{aligned} \tag{35}$$

Thus, taking expectations on both sides and telescoping, we obtain:

$$\sum\nolimits_{j=1}^{n}\gamma_j\left[R_{\max} - \mathbb{E}[R_j]\right] \leq D_0 + \frac{1}{2}V^2\sum\nolimits_{j=1}^{n}\gamma_j^2, \tag{36}$$

where we have used the fact that $\mathbb{E}[\xi_n] = 0$ and the finite mean square hypothesis $\mathbb{E}\left[\|\hat{\mathbf{V}}(n)\|^2\right] \leq V^2$. From (26), we have $D_0 = F(\mathbf{Q}^*, 0) \leq \max_{\mathbf{Q},\mathbf{Q}'}\{h(\mathbf{Q}) - h(\mathbf{Q}')\} = \log M$, so (14) follows by rearranging (36) and solving for $\mathbb{E}\left[\bar{R}_n\right] = t_n^{-1}\sum_{j=1}^{n}\gamma_j\,\mathbb{E}\left[R_j\right]$.

Having derived the mean performance guarantee (14), the tail bound (15) is a simple application of Markov's inequality:

$$\mathbb{P}(R_{\max} - \bar{R}_n \geq z) \leq \frac{\mathbb{E}[R_{\max} - \bar{R}_n]}{z} \leq \frac{\varepsilon_n}{z}. \tag{37}$$

Moreover, for the exponential bound (17), the inequality (35) yields $R_{\max} - \bar{R}_n \leq \varepsilon_n + t_n^{-1}\sum_{j=1}^{n}\gamma_j\xi_j$, so

$$\mathbb{P}\left(R_{\max} - \bar{R}_n \geq \varepsilon_n + z\right) \leq \mathbb{P}\left(\textstyle\sum_{j=1}^{n}|\gamma_j\xi_j| \geq t_n z\right). \tag{38}$$

Under the additional hypothesis (H2′), $\gamma_j\xi_j$ is a martingale difference with *finite* increments: $|\gamma_j\xi_j| \leq \gamma_j\|\mathbf{Z}_j\| \cdot \|\mathbf{Q}^* - \mathbf{Q}(j)\| \leq 2K\gamma_j\Sigma_j$. Thus, Azuma's concentration inequality [43] yields:

$$\mathbb{P}\left(\textstyle\sum_{j=1}^{n}|\gamma_j\xi_j| \geq t_n z\right) \leq \exp\left(-\dfrac{t_n^2 z^2}{8K^2\sum_{j=1}^{n}\gamma_j^2\Sigma_j^2}\right), \tag{39}$$

and (17) follows. ∎

3) Variants of MXL: In this section, we prove the convergence of the variant exponential learning schemes MXL-a and MXL-e (Theorems 2 and 3 respectively).

Proof of Theorem 2: We will show that the recursion (MXL-a) is an asynchronous stochastic approximation of (MXL-c) in the sense of [44, Chap. 7]. Indeed, by Theorems 2 and 3 in [44], the recursion (MXL-a) may be viewed as a stochastic approximation of the rate-adjusted dynamics

$$\begin{aligned} \dot{\mathbf{Y}}_k &= \eta_k \mathbf{V}_k \\ \mathbf{Q}_k &= \frac{\exp(\mathbf{Y}_k)}{\mathrm{tr}[\exp(\mathbf{Y}_k)]} \end{aligned} \tag{40}$$

where we have momentarily reinstated the user index k and $\eta_k = \lim_{n\to\infty} n_k/n > 0$ denotes the update rate of user k (the existence and positivity of this limit follows from the ergodicity of the update process $\mathcal{K}_n$). This multiplicative factor does not alter the rest points and internally chain transitive (ICT) sets [20] of the dynamics (MXL-c), so (40) converges to $\arg\max R$ from any initial condition and the proof of Theorem 1 carries through essentially verbatim. ■

To prove Theorem 3, we first need to derive the eigen-dynamics (20) induced by (MXL-c):

Proposition 3. *Let* $\mathbf{Q}(t)$ *be a solution orbit of* (MXL-c) *and let* $\{q_\alpha(t), \mathbf{u}_\alpha(t)\}$ *be a smooth eigen-decomposition of* $\mathbf{Q}(t)$*. Then,* $\{q_\alpha(t), \mathbf{u}_\alpha(t)\}$ *follows the eigen-dynamics* (20)*.*

Proof: By differentiating the identity $q_\alpha \delta_{\alpha\beta} = \mathbf{u}_\alpha^\dagger \mathbf{Q} \mathbf{u}_\beta$, we readily obtain:

$$\begin{aligned} \dot{q}_\alpha \delta_{\alpha\beta} &= \dot{\mathbf{u}}_\alpha^\dagger \mathbf{Q} \mathbf{u}_\beta + \mathbf{u}_\alpha^\dagger \dot{\mathbf{Q}} \mathbf{u}_\beta + \mathbf{u}_\alpha^\dagger \mathbf{Q} \dot{\mathbf{u}}_\beta \\ &= \mathbf{u}_\alpha^\dagger \dot{\mathbf{Q}} \mathbf{u}_\beta + (q_\alpha - q_\beta) \mathbf{u}_\alpha^\dagger \dot{\mathbf{u}}_\beta, \end{aligned} \tag{41}$$

where the last equality follows by differentiating the orthogonality condition $\mathbf{u}_\alpha^\dagger \mathbf{u}_\beta = \delta_{\alpha\beta}$. Thus, by *a)* taking $\alpha = \beta$ and *b)* solving for $\dot{\mathbf{u}}_\alpha^\dagger$ in (41), we respectively obtain:

$$\dot{q}_\alpha = \mathbf{u}_\alpha^\dagger \dot{\mathbf{Q}} \mathbf{u}_\alpha \tag{42a}$$

$$\dot{\mathbf{u}}_\alpha^\dagger = \sum\nolimits_{\beta \neq \alpha} \frac{\mathbf{u}_\alpha^\dagger \dot{\mathbf{Q}} \mathbf{u}_\beta}{q_\alpha - q_\beta} \mathbf{u}_\beta^\dagger \tag{42b}$$

However, by using the Fréchet derivative of the matrix exponential [45], we readily get:

$$\begin{aligned} \dot{\mathbf{Q}} &= \frac{1}{\mathrm{tr}[\exp(\mathbf{Y})]} \frac{d}{dt} \exp(\mathbf{Y}) - \exp(\mathbf{Y}) \frac{\mathrm{tr}[\dot{\mathbf{Y}} \exp(\mathbf{Y})]}{\mathrm{tr}[\exp(\mathbf{Y})]^2} \\ &= \frac{1}{\mathrm{tr}[\exp(\mathbf{Y})]} \int_0^1 \exp((1-s)\mathbf{Y}) \dot{\mathbf{Y}} \exp(s\mathbf{Y})\, ds - \mathbf{Q}\, \mathrm{tr}[\mathbf{V}\mathbf{Q}] \\ &= \int_0^1 \mathbf{Q}^{1-s} \mathbf{V} \mathbf{Q}^s\, ds - \mathbf{Q}\, \mathrm{tr}[\mathbf{V}\mathbf{Q}], \end{aligned} \tag{43}$$

and hence:

$$\begin{aligned} \mathbf{u}_\alpha^\dagger\, \dot{\mathbf{Q}}\, \mathbf{u}_\beta &= \int_0^1 \mathbf{u}_\alpha^\dagger \mathbf{Q}^{1-s} \mathbf{V} \mathbf{Q}^s \mathbf{u}_\beta\, ds - \mathrm{tr}[\mathbf{V}\mathbf{Q}]\, \mathbf{u}_\alpha^\dagger \mathbf{Q} \mathbf{u}_\beta \\ &= \int_0^1 q_\alpha^{1-s} V_{\alpha\beta} q_\beta^s\, ds - q_\alpha \delta_{\alpha\beta} \sum\nolimits_\gamma q_\gamma V_{\gamma\gamma}, \end{aligned} \tag{44}$$

where we have set $V_{\alpha\beta} = \mathbf{u}_\alpha^\dagger \mathbf{V} \mathbf{u}_\beta$. Thus, by carrying out the integration in (44), we finally obtain:

$$\mathbf{u}_\alpha^\dagger \dot{\mathbf{Q}} \mathbf{u}_\beta = \frac{q_\alpha - q_\beta}{\log q_\alpha - \log q_\beta} V_{\alpha\beta} - q_\alpha \delta_{\alpha\beta} \sum\nolimits_\gamma q_\gamma V_{\gamma\gamma}, \tag{45}$$

with the convention $(x-y)/(\log x - \log y) = x$ if $x = y$. The dynamics (20) then follow by substituting (45) in (42). ■

Proof of Proposition 1: Combining (MXL) and the derivative expression (43), we get:

$$\begin{aligned} \mathbf{Q}(n+1) &= \frac{\exp(\mathbf{Y}(n+1))}{\operatorname{tr}[\exp(\mathbf{Y}(n+1))]} = \frac{\exp(\mathbf{Y}(n) + \gamma_n \mathbf{V}(n))}{\operatorname{tr}[\exp(\mathbf{Y}(n) + \gamma_n \mathbf{V}(n))]} \\ &= \mathbf{Q}(n) + \gamma_n \int_0^1 \mathbf{Q}(n)^{1-s} \mathbf{V}(n) \mathbf{Q}(n)^s \, ds \\ &\quad - \gamma_n \operatorname{tr}[\mathbf{Q}(n)\mathbf{V}(n)] \, \mathbf{Q}(n) + \mathcal{O}\left(\gamma_n^2 \left\|\mathbf{V}(n)\right\|^2\right), \end{aligned} \tag{46}$$

where the term $\mathcal{O}\left(\gamma_n \left\|\mathbf{V}(n)\right\|^2\right)$ is bounded from above by $C\gamma_n^2 \left\|\mathbf{V}(n)\right\|^2$ for some constant C that does not depend on $\mathbf{Q}(n)$. Since $\gamma_n \to 0$ by assumption, Remark 4.5 in [20] shows that the quadratic error in (46) can be ignored in the long-run, so $\mathbf{Q}(n)$ is an asymptotic pseudotrajectory (APT) of the dynamics (43). Hence, by Proposition 3, the eigen-decomposition $\{q_\alpha(n), \mathbf{u}_\alpha(n)\}$ is an APT of (20), as claimed. ■

Proof of Theorem 3: Consider the following Euler discretization of the eigen-dynamics (20):

$$q_\alpha \leftarrow q_\alpha + \gamma_n q_\alpha \left(V_{\alpha\alpha} - \textstyle\sum_\beta q_\beta V_{\beta\beta}\right), \tag{47a}$$

$$\mathbf{u}_\alpha \leftarrow \mathbf{u}_\alpha + \gamma_n \textstyle\sum_{\beta \neq \alpha} \frac{V_{\beta\alpha}}{\log q_\alpha - \log q_\beta} \mathbf{u}_\beta, \tag{47b}$$

i.e. the update step of Algorithm 3 without the orthonormalization correction for $\mathbf{u}_\alpha$. We then obtain:

$$\mathbf{u}_\alpha^\dagger(n+1)\, \mathbf{u}_\beta(n+1) = \mathbf{u}_\alpha^\dagger(n)\, \mathbf{u}_\beta(n) + \mathcal{O}(\gamma_n^2), \tag{48}$$

which shows that the orthonormalization correction in Algorithm 3 is quadratic in γ_n. Thus, as long as γ_n is chosen small enough (so that $q_\alpha(n) \geq 0$ for all n), Remark 4.5 in [20] shows that the iterates of Algorithm 3 comprise an APT of (20). In turn, the same reasoning as in the proof of Prop. 1 can be used to show that $\mathbf{Q}(n) = \sum_\alpha q_\alpha(n) \mathbf{u}_\alpha(n) \mathbf{u}_\alpha^\dagger(n)$ is an APT of (MXL-c), so $\mathbf{Q}(n)$ converges to the solution set of (RM) by Theorem 1. ■

4) The fast-fading regime: In this section, we prove our convergence results (Theorem 4) for the ergodic (fast-fading) regime:

Proof of Theorem 4: Let $\mathbf{V}_{\text{erg}} = \nabla R_{\text{erg}}$ denote the gradient of the ergodic sum rate function R_{erg} and consider the dynamics:

$$\begin{aligned} \dot{\mathbf{Y}} &= \mathbf{V}_{\text{erg}}, \\ \mathbf{Q} &= \frac{\exp(\mathbf{Y})}{\operatorname{tr}[\exp(\mathbf{Y})]}. \end{aligned} \tag{49}$$

The same reasoning as in the proof of Theorem 5 shows that (49) converges to the unique minimizer of the (strictly concave) sum rate maximization problem (ERM). Moreover, given that R is concave for any fixed channel matrix $\mathbf{H}$ and R_{erg} is finite on $\mathcal{D}$, we have [46]:

$$\mathbf{V}_{\text{erg}}(\mathbf{Q}) = \nabla_{\mathbf{Q}} R_{\text{erg}}(\mathbf{Q}) = \mathbb{E}_{\mathbf{H}}\left[\nabla_{\mathbf{Q}} R(\mathbf{Q})\right] = \mathbb{E}_{\mathbf{H}}[\mathbf{V}(\mathbf{Q})], \tag{50}$$

with $\mathbf{V}(\mathbf{Q})$ defined as in (11). With $\mathbf{V}$ bounded, it follows that $\mathbf{V}_{\text{erg}}$ is Lipschitz, so Propositions 4.2 and 4.1 in [20] imply that the iterates of (MXL) comprise a stochastic approximation of the mean dynamics (49). The rest of the proof then follows as in the case of Theorem 1. ■

Appendix B
An Unbiased Estimator for $\mathbf{V}$

In this appendix, we briefly describe an unbiased estimator of the gradient matrix $\mathbf{V}$ based at each step on possibly imperfect signal and channel measurements – e.g. obtained through the exchange of pilot feedback signals. The first step will be to estimate the aggregate *signal precision* (inverse covariance) matrix $\mathbf{P} = \mathbf{W}^{-1} = \mathbb{E}[\mathbf{y}\mathbf{y}^\dagger]^{-1}$ of (12) by sampling the received signal $\mathbf{y} \in \mathbb{C}^N$. Since the channel is assumed Gaussian, this can be accomplished by means of the bias-adjusted estimator:

$$\hat{\mathbf{P}} = \frac{S - N - 1}{S}\hat{\mathbf{W}}^{-1}, \tag{51}$$

where $\hat{\mathbf{W}} = S^{-1}\sum_{s=1}^{S}\mathbf{y}_s\mathbf{y}_s^\dagger$ is an (unbiased) estimate for the received signal covariance matrix $\mathbf{W}$ [47].

In the absence of perfect CSIT, the transmitters must estimate the individual gradient matrices $\mathbf{V}_k = \mathbf{H}_k^\dagger\mathbf{W}^{-1}\mathbf{H}_k$ from the broadcast of $\hat{\mathbf{P}}$ and imperfect measurements of their individual channel matrices $\mathbf{H}_k$. To that end, if each transmitter takes S independent measurements $\hat{\mathbf{H}}_{k,1}, \dots, \hat{\mathbf{H}}_{k,S}$ of his channel matrix (e.g. via independent reverse pilot sampling), an unbiased estimate for $\mathbf{V}_k$ is given by the expression:

$$\hat{\mathbf{V}}_k = \frac{1}{S(S-1)}\sum\nolimits_{s\neq s'}\hat{\mathbf{H}}_{k,s}^\dagger\hat{\mathbf{P}}\hat{\mathbf{H}}_{k,s'}, \tag{52}$$

where $\hat{\mathbf{P}}$ is the latest estimate of (51) of $\mathbf{W}^{-1}$ that was broadcast by the receiver. Indeed, given that the sampled channel matrix measurements $\hat{\mathbf{H}}_{k,s}$ are assumed stochastically independent, we readily obtain:

$$\mathbb{E}[\hat{\mathbf{V}}_k] = \frac{1}{S(S-1)}\sum\nolimits_{s\neq s'}\mathbb{E}\left[\hat{\mathbf{H}}_{k,s}^\dagger\hat{\mathbf{P}}\hat{\mathbf{H}}_{k,s'}\right] = \mathbf{H}_k^\dagger\mathbf{W}^{-1}\mathbf{H}_k, \tag{53}$$

i.e. (52) constitutes an unbiased estimator of $\mathbf{V}$.

The construction above provides an estimator $\hat{\mathbf{V}}$ with $\mathbb{E}[\hat{\mathbf{V}}] = \mathbf{V}$ so Assumption (H1) holds. As for the variance of $\hat{\mathbf{V}}$, (52) can also be used to derive an expression for $\text{Var}(\hat{\mathbf{V}})$ in terms of the moments of $\hat{\mathbf{P}}$ and $\hat{\mathbf{H}}$. Since the system input and noise are assumed Gaussian, the former are all finite (and Gaussian-distributed) so the finite mean square error hypothesis (H2) boils down to measuring $\mathbf{H}$ with finite mean squared error.

References

[1] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, pp. 311–335, 1998.

[2] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Transactions on Telecommunications and Related Technologies*, vol. 10, no. 6, pp. 585–596, 1999.

[3] J. Hoydis, S. ten Brink, and M. Debbah, "Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?" *IEEE Trans. Wireless Commun.*, vol. 31, no. 2, pp. 160–171, February 2013.

[4] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, pp. 40–60, January 2013.

[5] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, February 2014.

[6] J. G. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[7] R. S. Cheng and S. Verdú, "Gaussian multiaccess channels with ISI: capacity region and multiuser water-filling," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 773–785, May 1993.

[8] W. Yu, W. Rhee, S. Boyd, and J. M. Cioffi, "Iterative water-filling for Gaussian vector multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 50, no. 1, pp. 145–152, 2004.

[9] G. Scutari, D. P. Palomar, and S. Barbarossa, "Competitive design of multiuser MIMO systems based on game theory: a unified view," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 7, pp. 1089–1103, September 2008.

[10] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, 2010.

[11] P. Mertikopoulos, E. V. Belmega, and A. L. Moustakas, "Matrix exponential learning: Distributed optimization in MIMO systems," in *ISIT '12: Proceedings of the 2012 IEEE International Symposium on Information Theory*, 2012, pp. 3028–3032.

[12] P. Coucheney, B. Gaujal, and P. Mertikopoulos, "Distributed optimization in multi-user MIMO systems with imperfect and delayed information," in *ISIT '14: Proceedings of the 2014 IEEE International Symposium on Information Theory*, 2014.

[13] G. Scutari, D. P. Palomar, and S. Barbarossa, "The MIMO iterative waterfilling algorithm," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 1917–1935, May 2009.

[14] ——, "Simultaneous iterative water-filling for Gaussian frequency-selective interference channels," in *ISIT '06: Proceedings of the 2006 International Symposium on Information Theory*, 2006.

[15] P. Mertikopoulos, E. V. Belmega, A. L. Moustakas, and S. Lasaulce, "Distributed learning policies for power allocation in multiple access channels," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 96–106, January 2012.

[16] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Mathematical Programming*, vol. 120, no. 1, pp. 221–259, 2009.

[17] J. Hoydis, S. ten Brink, and M. Debbah, "Making smart use of excess antennas: massive MIMO, small cells, and TDD," *Bell Labs Technical Journal*, vol. 18, no. 2, pp. 5–21, 2013.

[18] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath, "Pilot contamination and precoding in multi-cell TDD systems," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2640–2651, 2011.

[19] A. M. Davie and A. J. Stothers, "Improved bound for complexity of matrix multiplication," *Proceedings of the Royal Society of Edinburgh, Section A: Mathematics*, vol. 143, no. 2, pp. 351–369, 4 2013.

[20] M. Benaïm, "Dynamics of stochastic approximation algorithms," in *Séminaire de Probabilités XXXIII*, ser. Lecture Notes in Mathematics, J. Azéma, M. Émery, M. Ledoux, and M. Yor, Eds. Springer Berlin Heidelberg, 1999, vol. 1709, pp. 1–68.

[21] K. Tsuda, G. Rätsch, and M. K. Warmuth, "Matrix exponentiated gradient updates for on-line Bregman projection," *Journal of Machine Learning Research*, vol. 6, pp. 995–1018, 2005.

[22] S. M. Kakade, S. Shalev-Shwartz, and A. Tewari, "Regularization techniques for learning with matrices," *The Journal of Machine Learning Research*, vol. 13, pp. 1865–1890, 2012.

[23] P. Mertikopoulos and E. V. Belmega, "Transmit without regrets: online optimization in MIMO–OFDM cognitive radio systems," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 11, pp. 1987–1999, November 2014.

[24] A. S. Nemirovski, A. Juditsky, G. G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[25] L. E. Blume, "The statistical mechanics of strategic interaction," *Games and Economic Behavior*, vol. 5, pp. 387–424, 1993.

[26] S. Perkins and D. S. Leslie, "Asynchronous stochastic approximation with differential inclusions," *Stochastic Systems*, vol. 2, no. 2, pp. 409–446, 2012.

[27] S. Perkins, P. Mertikopoulos, and D. S. Leslie, "Game-theoretical control with continuous action sets," http://arxiv.org/abs/1412.0543, 2014.

[28] A. J. Goldsmith and P. P. Varaiya, "Capacity of fading channels with channel side information," *IEEE Trans. Inf. Theory*, vol. 43, no. 6, pp. 1986–1992, 1997.

[29] E. V. Belmega, S. Lasaulce, M. Debbah, and A. Hjørungnes, "Learning distributed power allocation policies in MIMO channels," in *EUSIPCO '10: Proceedings of the 2010 European Signal Processing Conference*, 2010.

[30] Y. Yang, G. Scutari, D. P. Palomar, and M. Pesavento, "A parallel stochastic approximation method for nonconvex multi-agent optimization problems," http://arxiv.org/abs/1410.5076, 2014.

[31] COST Action 231, "Digital mobile radio towards future generation systems," European Commission, final report, 1999.

[32] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Trans. Veh. Technol.*, vol. 29, no. 3, pp. 317–325, August 1980.

[33] 3GPP, "User equipment (UE) radio transmission and reception," White paper, Jun. 2014.

[34] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[35] T. E. Abrudan, J. Eriksson, and V. Koivunen, "Steepest descent algorithms for optimization under unitary matrix constraint," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1134–1147, March 2008.

[36] V. Vedral, "The role of relative entropy in quantum information theory," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 197–234, 2002.

[37] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ: Princeton University Press, 1970.

[38] P. Mertikopoulos and W. H. Sandholm, "Regularized best responses and reinforcement learning in games," http://arxiv.org/abs/1407.6267, 2014.

[39] J. Dattorro, *Convex Optimization & Euclidean Distance Geometry*. Palo Alto, CA, USA: Meboo Publishing, 2005.

[40] P. Hall and C. C. Heyde, *Martingale Limit Theory and Its Application*, ser. Probability and Mathematical Statistics. New York: Academic Press, 1980.

[41] G. H. Hardy, *Divergent Series*. Oxford University Press, 1949.

[42] Y. S. Chow, "Convergence of sums of squares of martingale differences," *The Annals of Mathematical Statistics*, vol. 39, no. 1, 1968.

[43] K. Azuma, "Weighted sums of certain dependent random variables," *Tôhoku Mathematical Journal*, vol. 19, no. 3, pp. 357–367, 1967.

[44] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press and Hindustan Book Agency, 2008.

[45] R. M. Wilcox, "Exponential operators and parameter differentiation in quantum physics," *Journal of Mathematical Physics*, vol. 8, no. 4, pp. 962–982, 1967.

[46] V. Strassen, "The existence of probability measures with given marginals," *The Annals of Mathematical Statistics*, vol. 38, pp. 423–439, 1965.

[47] T. W. Anderson, *An Introduction to Multivariate Statistical analysis*, 3rd ed. Wiley-Interscience, 2003.