Adaptive Channel Estimators with Hybrid Beamforming for Single-Carrier Massive MIMO

Sadjad Sedighi[1], Gokhan M. Guvensen[2], and Ender Ayanoglu[1]

[1]CPCC, Dept of EECS, UC Irvine, Irvine, CA, USA, [2]Dept of EEE, METU, Ankara, Turkey

Abstract— **The number of wireless devices connected to the cellular wireless network is anticipated to increase heavily in the next few years, eventually reaching tens of billions. Thus, there is a need for a new cellular wireless network, which is able to handle tens of billions of wireless devices. This phenomenon has an associated effect in terms of a huge increase in traffic. A number of means is under investigation to respond to that need. Among several alternatives, the category of massive multiple input multiple output (MIMO) systems is a great candidate for this purpose. Because of the large number of antennas in massive MIMO, there is a need to reduce the dimension of the MIMO channel effectively to decrease the complexity by considering the sparsity in the channel in terms of angle of arrival and delay, as introduced by the technique of Joint Spatial Division and Multiplexing. This can be achieved effectively by using a particular statistical pre-beamforming technique introduced recently. In this paper, different adaptive algorithms for estimating the channel vector coefficient and their performance, based on this recent pre-beamforming technique, are studied. Different approaches are used in order to find the best algorithm based on the channel estimation accuracy and the complexity of the algorithm. As a byproduct, with the help of the provided analysis, in single-carrier time-varying massive MIMO channels, the optimal number of RF chains (required spatial dimensions) can be determined in hybrid beamforming in terms of the achievable information rate by taking the estimation accuracy of different adaptive acquisition algorithms into account.**

I. Introduction

By using the two-stage beamforming concept, one can reduce the dimension of the MIMO channel effectively while at the same time enabling the massive MIMO gains [1]. The concept is also known as Joint Spatial Division and Multiplexing (JSDM) [2], [3]. Reference [4] applied this technique to both downlink and uplink transmission in Time Division Duplex (TDD) by considering the channel estimation accuracy. It uses user-grouping which partitions the users in a cell supported by a base station (BS) into different groups with approximately the same channel covariance eigenspaces. By using a spatial pre-beamformer, one can decompose the MIMO beamformer at the BS into two steps. The pre-beamformer distinguishes the intra-group signals from the other groups by suppressing the inter-group interference, while simultaneously reducing the signal dimension. The pre-beamformer is designed based only on the long-term parameters.

The main contributions of this paper are summarized as follows. First a model for channel state update is proposed based on a general framework on reduced dimensional channel state information (CSI) estimation [4], [5]. According to the new model for channel coefficient vector update, three different methods from adaptive signal processing are introduced for estimating the channel in a reduced dimension. Analytical techniques are developed to calculate transient Mean Squared Error (MSE) and the capacity of the adaptive systems. The methods are compared on the basis of complexity, MSE, and capacity, determining the required reduction in dimensionality.

This work was partially supported by NSF under Grant No. 1547155.

II. System Model

We consider a massive MIMO system in which an N-antenna BS communicates with K single-antenna User Terminals (UTs), operating at millimeter-wave (mm-wave) bands in the Time Division Multiplex (TDD) mode employing Single Carrier (SC) modulation. Users are categorized into G different groups, where K_g is the number of users in group g. These users have statistically independent but identically distributed (i.i.d.) channels [2]–[4], and transmit training sequences with length T at the beginning of every coherent interval.

We assume a linear modulation (e.g., QAM or PSK) and transmission over a frequency selective channel for all UTs with a slow evolution in time relative to the signaling interval (symbol duration). Assuming these conditions, the baseband equivalent received signal samples, taken at symbol rate after pulse matched filtering, are expressed as [4], [5]

$$\mathbf{y}_n = \underbrace{\sum_{\{k=1, g_k \in \Omega_g\}}^{K_g} \sum_{l=0}^{L_g-1} \mathbf{h}_{n,l}^{(g_k)} x_{n-l}^{(g_k)}}_{\text{Intra-Group Signal}} + \underbrace{\sum_{\{\forall g'_k \in \Omega_{g'} | g' \neq g\}} \left(\sum_{k=1}^{K_{g'}} \sum_{l=0}^{L_{g'}-1} \mathbf{h}_{n,l}^{(g'_k)} x_{n-l}^{(g'_k)} \right) + \mathbf{n}_n}_{\eta_n^{(g)}\text{:Inter-Group Interference + AWGN}} \quad (1)$$

for $n = 0, ..., T-1$, where $\{x_n^{(g_k)} : -L_g + 1 \leq n \leq T-1\}$ are the training symbols for the k^{th} user in group g and $\mathbf{h}_{n,l}^{(g_k)}$ is the $N \times 1$ time-varying multi-path channel vector, namely, the array impulse response of the serving BS stemming from the l^{th} multi-path component (MPC) of k^{th} user in group g. Also, Ω_g is the set of all UTs belonging to group g with cardinality $|\Omega_g| = K_g$, L_g is the channel memory of group g multi-path channels, and $\{g_k\}_{k=1}^{K_g}$ are UT indices that form Ω_g. We select the training symbols from a signal constellation $S \in \mathbb{C}$ and

$\mathbb{E}\{|x_n^{(g_k)}|^2\}$ is set to E_s for all g_k [4], [5]. In (1), $\mathbf{n}_n$ are the additive white Gaussian noise (AWGN) vectors during uplink pilot segment with spatially i.i.d. as $\mathcal{CN}(\mathbf{0}, N_0\mathbf{I}_N)$, and N_0 is the value of the noise power spectral density.

The first term of (1) is the transmitted signal of the intended group g, which is labeled as the *intra-group* signal of group g users. The second term, $\boldsymbol{\eta}_n^{(g)}$, is labeled as the *inter-group interference*, and includes all of the interfering signals, which stem from all inner or outer cell users belonging to different groups other than g. Finally, the average received SNR can be defined as $SNR \triangleq \frac{E_s}{N_0}$. We assume users come in groups, either by nature or by the application of the proper *user grouping* algorithms in [3], [6], which are out of scope of this work.

The training vector (or convolution vector), comprising of the transmitted pilots for the kth user in group g at the nth signaling interval is defined as $\mathbf{x}_{n,k}^{(g)} \triangleq [x_n^{(g_k)}, x_{n-1}^{(g_k)}, \ldots, x_{n-L_g-1}^{(g_k)}]^H$. Then, the complete training vector that consists of the training data of all users in group g at the nth signaling interval is given by

$$\mathbf{x}_n^{(g)} \triangleq \text{vec}\Big\{\big[\mathbf{x}_{n,1}^{(g)}, \mathbf{x}_{n,2}^{(g)}, \ldots, \mathbf{x}_{n,K_g}^{(g)}\big]_{L_g \times K_g}\Big\}. \tag{2}$$

The pre-beamforming is applied in order to distinguish intra-group signal of group g users from other groups by suppressing the inter-group interference. At the pre-beamforming stage as shown in [4], [5], a D-dimensional vector $\mathbf{y}_n^{(g)}$ can be found for all groups by a linear transformation through $(\mathbf{S}_D^{(g)})^H$ on (1) as

$$\mathbf{y}_n^{(g)} = \big(\boldsymbol{\Psi}_n^{(g)}\big)^H \mathbf{h}_n^{(g)} + \big(\mathbf{S}_D^{(g)}\big)^H \boldsymbol{\eta}_n^{(g)} \tag{3}$$

where $\boldsymbol{\Psi}_n^{(g)} \triangleq \mathbf{x}_n^{(g)} \otimes \mathbf{S}_D^{(g)}$, $\mathbf{h}_n^{(g)}$ is the extended channel vector (including all the related channel parameters of users in group g to be estimated simultaneously), and $\boldsymbol{\xi}_n^{(g)} \triangleq (\mathbf{S}_D^{(g)})^H \boldsymbol{\eta}_n^{(g)}$ is defined as residual interference after pre-beamforming. Here, $\mathbf{S}_D^{(g)}$ is an $N \times D$ statistical pre-beamforming matrix that projects the N-dimensional received signal samples in $\{\mathbf{y}_n\}$ in (1) on a proper D-dimensional subspace. Therefore, $\mathbf{S}_D^{(g)}$ can be visualized as the analog beamforming stage in Hybrid Beamforming Framework [4], [5]. In this paper, the optimized pre-beamformer $\mathbf{S}_D^{(g)}$ given in [5] is used for better interference suppression by taking the spatially correlated interference into account.

We assume that the channel is block-fading and that the channel remains stationary during the lth block. The channel temporal variation across the blocks is modeled using a state-space framework as a first-order stationary Gauss-Markov process [7] with

$$\mathbf{h}_n^{(g)} = \sqrt{\alpha}\mathbf{h}_{n-1}^{(g)} + \sqrt{1-\alpha}\mathbf{b}_n^{(g)} \tag{4}$$

where $\alpha \in (0, 1]$ is the temporal fading coefficient and $\mathbf{b}_n^{(g)} \sim \mathcal{CN}(\mathbf{0}, \mathbf{R}_{\mathbf{h}}^{(\mathbf{g})})$ is the process noise. The channel spatial correlation is given by

$$\mathbf{R_h}^{(g)} = \mathbb{E}\{\mathbf{h}_n^{(g)}(\mathbf{h}_n^{(g)})^H\} = \alpha\mathbf{R_h}^{(g)} + (1-\alpha)\mathbf{R_b}^{(g)} \tag{5}$$

where $\mathbf{R_b}^{(g)}$ is easily calculated as $\mathbf{R_b}^{(g)} = \mathbb{E}[\mathbf{b}_n^{(g)}(\mathbf{b}_n^{(g)})^H] = \mathbf{R_h}^{(g)}$. Under Jakes' model, $\sqrt{\alpha} = J_0(2\pi f_D T_S)$ where $J_0(.)$ is the zero order Bessel function, f_D is the Doppler frequency, and T_s is the transmit symbol interval [8]. For transmission at 2.5 GHz and $T_s = 500\ \mu$s, $\sqrt{\alpha} = 0.9999$ corresponds to 3 km/h and $\sqrt{\alpha} = 0.9975$ corresponds to 150 km/h.

Based on these assumptions, we develop algorithms to estimate the channel with the pre-beamforming technique of [5]. Then, by analysis, we can find the optimum reduced dimensionality D, which can be different for different methods. The work in [5] has to do with the derivation of a Wiener filter, while that in [4] and in this paper, adaptive algorithms have been developed. While [4] discusses a Kalman filtering approach, in this paper we discuss Least Mean Square (LMS), Recursive Least Squares (RLS), and Kalman filtering approaches together with their effects on capacity. Note that all of these algorithms operate with dimensionality reduction based on [5] and are therefore different than the conventional forms of these algorithms. In the remaining sections, the emphasis is on deriving efficient ways to estimate the channel and then by using different tools, we will provide the minimum beamspace dimension (D) required for the dimension reduction in terms of achievable information rate. As in [4], [5] this dimension will in general be much smaller than without performing this reduction.

III. Adaptive Algorithms with Pre-Beamforming

We will now consider channel estimation techniques for multiuser massive MIMO systems and employ the signal models of Section II. Although the techniques are based on LMS, RLS, and Kalman filtering, they involve a step for dimension reduction, and therefore they are not the conventional forms of these techniques. We call these algorithms reduced rank (RR) versions of the standard algorithms and abbreviate them as RR-LMS, RR-RLS, and RR-Kalman filtering in the sequel.

For deriving RR-LMS, RR-RLS, and Kalman algorithms, the following equations are being used (dropping "(g)")

$$\mathbf{B}_n = \big(\mathbf{x}_n \otimes \mathbf{I}_D\big) \tag{6}$$

$$\mathbf{h}_{n,\mathit{eff}} \triangleq \mathbf{Q}\mathbf{h}_n \tag{7}$$

$$\hat{\mathbf{h}}_{n,\mathit{eff}} \triangleq \mathbf{Q}\hat{\mathbf{h}}_n \tag{8}$$

where $\mathbf{x}_n$ is defined in (2) and its size is $K_g L_g \times 1$, $\mathbf{Q} = \Big(\mathbf{I}_{K_g L_g} \otimes \mathbf{S}_D^H\Big)$, the assigned vectors are the dimension reduced vectors, i.e., effective channel vectors, and both $\mathbf{h}_n$ and $\hat{\mathbf{h}}_n$ are $K_g L_g N \times 1$ vectors.

A. RR-LMS Algorithm

The conventional LMS algorithm needs to be altered considering the subspace reduction. We define J_n as the cost function which we want to minimize over $\mathbf{h}_n$:

$$J_n = E\{||\mathbf{e}_n||^2\} = E\{||\mathbf{y}_n - \boldsymbol{\Psi}_n^H \mathbf{h}_n||^2\} \tag{9}$$

By using the general form of LMS algorithm which is

$$\hat{\mathbf{h}}_n = \hat{\mathbf{h}}_{n-1} - \mu\nabla_{\mathbf{h}_n^H} J_n \tag{10}$$

one can easily calculate the minimum of cost function by using two above equations.

Based on (6) and (7) one can rewrite (3) without index (g)

as

$$\mathbf{y}_n = (\mathbf{x}_n^H \otimes \mathbf{S}_D^H)\mathbf{h}_n + \boldsymbol{\xi}_n = \mathbf{B}_n^H \mathbf{h}_{n,eff} + \boldsymbol{\xi}_n. \quad (11)$$

Based on (9), (10) and (11) one can get the RR-LMS algorithm

$$\begin{aligned}\hat{\mathbf{h}}_{n,eff} &= \hat{\mathbf{h}}_{n-1,eff} - \mu \nabla_{\mathbf{h}^H_{n,eff}} J_n \\ &= \hat{\mathbf{h}}_{n-1,eff} \\ &\quad -\mu \nabla_{\mathbf{h}^H_{n,eff}} \left[(\mathbf{y}_n^H - \mathbf{h}_{n,eff}^H \mathbf{B}_n)(\mathbf{y}_n - \mathbf{B}_n^H \mathbf{h}_{n,eff}) \right] \\ &\overset{(a)}{=} \hat{\mathbf{h}}_{n-1,eff} - \mu(-\mathbf{B}_n)(\mathbf{y}_n - \mathbf{B}_n^H \hat{\mathbf{h}}_{n-1,eff}) \\ &= \hat{\mathbf{h}}_{n-1,eff} + \mu \mathbf{B}_n (\mathbf{y}_n - \mathbf{B}_n^H \hat{\mathbf{h}}_{n-1,eff}) \\ &= \hat{\mathbf{h}}_{n-1,eff} + \mu \mathbf{B}_n \mathbf{e}_n \quad (12)\end{aligned}$$

where $\mathbf{e}_n = \mathbf{y}_n - \mathbf{B}_n^H \hat{\mathbf{h}}_{n-1,eff}$. In step (a), we used the approximation $\hat{\mathbf{h}}_{n,eff} \approx \hat{\mathbf{h}}_{n-1,eff}$ due to double multiplication on $\nabla_{\mathbf{h}^H_{n,eff}} J_n$ by μ. Note that in this derivation we removed the expectation operator. This is because of employing the "stochastic gradient" in deriving the LMS algorithm as in, e.g., [9].

B. RR-RLS Algorithm

In this section, the RR-RLS estimator is presented. The RR-RLS algorithm recursively finds the coefficients that minimize a weighted linear least squares cost function relating to the input signals. This is in contrast to RR-LMS, that aims to reduce the mean square error where the mean calculation is in a statistical sense. Whereas, RR-RLS employs time averages. The channel estimates are updated recursively upon receiving new training symbols. The channel estimation problem corresponds to solving the following least-squares (LS) optimization problem

$$\hat{\mathbf{h}}_{n,eff} = \underset{\mathbf{h}_{n,eff}}{\arg\min} \left(\sum_{i=0}^{n} \lambda^{n-i} \|\mathbf{y}_i - \mathbf{B}_i^H \mathbf{h}_{n,eff}\|^2 \right) \quad (13)$$

The parameter λ is a forgetting factor chosen between 0 and 1, and gives exponentially less weight to older error samples. This problem can be solved by computing the gradient of (13), equating them to a zero matrix and manipulating the terms, which yields our solution. We rewrite (13) as

$$J_n = \sum_{i=0}^{n} \lambda^{n-i} \|\mathbf{e}_i\|^2$$

where

$$\begin{aligned}\mathbf{y}_i &= (\mathbf{x}_i^H \otimes \mathbf{S}_D^H)\mathbf{h}_i + \boldsymbol{\xi}_i = \boldsymbol{\Psi}_i^H \mathbf{h}_i + \boldsymbol{\xi}_i \\ \mathbf{e}_i &= \mathbf{y}_i - \mathbf{B}_i^H \hat{\mathbf{h}}_{n,eff}.\end{aligned}$$

From now on, we use $\hat{\mathbf{h}}_n$ instead of $\hat{\mathbf{h}}_{n,eff}$ for simplicity. By computing the gradient we have

$$\begin{aligned}\nabla_{\hat{\mathbf{h}}_n^H} J_n &= \nabla_{\hat{\mathbf{h}}_n^H} \left(\sum_{i=0}^{n} \lambda^{n-i} \|\mathbf{e}_i\|^2 \right) = \nabla_{\hat{\mathbf{h}}_n^H} \left(\sum_{i=0}^{n} \lambda^{n-i} \|\mathbf{y}_i - \mathbf{B}_i^H \hat{\mathbf{h}}_n\|^2 \right) \\ &= \nabla_{\hat{\mathbf{h}}_n^H} \left(\sum_{i=0}^{n} \lambda^{n-i} \left(\mathbf{y}_i^H - \hat{\mathbf{h}}_n^H \mathbf{B}_i \right) \left(\mathbf{y}_i - \mathbf{B}_i^H \hat{\mathbf{h}}_n \right) \right) \\ &= \sum_{i=0}^{n} \lambda^{n-i} \left(-\mathbf{B}_i \right) \left(\mathbf{y}_i - \mathbf{B}_i^H \hat{\mathbf{h}}_n \right) = 0.\end{aligned}$$

If we rearrange the above equation

$$\sum_{i=0}^{n} \lambda^{n-i} \mathbf{B}_i \mathbf{B}_i^H \hat{\mathbf{h}}_n = \sum_{i=0}^{n} \lambda^{n-i} \mathbf{B}_i \mathbf{y}_i \quad (14)$$

$$\mathbf{R}_n \hat{\mathbf{h}}_n = \mathbf{r}_n \quad (15)$$

where $\mathbf{R}_n = \sum_{i=0}^{n} \lambda^{n-i} \mathbf{B}_i \mathbf{B}_i^H$ is the weighted covariance matrix for $\mathbf{B}_i$ and $\mathbf{r}_n = \sum_{i=0}^{n} \lambda^{n-i} \mathbf{B}_i \mathbf{y}_i$ is the equivalent estimate for the cross-covariance between $\mathbf{B}_i$ and $\mathbf{y}_i$.

Based on this expression we find the coefficients which minimize the cost function as

$$\hat{\mathbf{h}}_n = \mathbf{R}_n^{-1} \mathbf{r}_n. \quad (16)$$

This is a form of the well-known Wiener-Hopf equation while employing time averages.

1) Recursive Algorithm

In this section we want to derive a recursive solution of the form

$$\hat{\mathbf{h}}_n = \hat{\mathbf{h}}_{n-1} + \Delta \hat{\mathbf{h}}_{n-1} \quad (17)$$

where $\Delta \hat{\mathbf{h}}_{n-1}$ is a correction factor at time $n-1$. We start the derivation of the recursive algorithm by expressing the cross covariance $\mathbf{r}_n$ in terms of $\mathbf{r}_{n-1}$

$$\begin{aligned}\mathbf{r}_n &= \sum_{i=0}^{n} \lambda^{n-i} \mathbf{B}_i \mathbf{y}_i = \sum_{i=0}^{n-1} \lambda^{n-i} \mathbf{B}_i \mathbf{y}_i + \lambda^0 \mathbf{B}_n \mathbf{y}_n \\ &= \lambda \mathbf{r}_{n-1} + \mathbf{B}_n \mathbf{y}_n. \quad (18)\end{aligned}$$

Similarly, we express $\mathbf{R}_n$ in terms of $\mathbf{R}_{n-1}$ by

$$\begin{aligned}\mathbf{R}_n &= \sum_{i=0}^{n} \lambda^{n-i} \mathbf{B}_i \mathbf{B}_i^H = \sum_{i=0}^{n-1} \lambda^{n-i} \mathbf{B}_i \mathbf{B}_i^H + \mathbf{B}_i \mathbf{B}_i^H \\ &= \lambda \mathbf{R}_{n-1} + \mathbf{B}_n \mathbf{B}_n^H. \quad (19)\end{aligned}$$

In order to calculate the channel coefficients vector, we are interested in the inverse of auto-covariance matrix. The matrix inversion lemma comes in handy for this task [9]

$$\begin{aligned}\mathbf{R}_n^{-1} &= \lambda^{-1} \mathbf{R}_{n-1}^{-1} \\ &\quad - \lambda^{-1} \mathbf{R}_{n-1}^{-1} \mathbf{B}_n \left(\mathbf{I}_D + \mathbf{B}_n^H \lambda^{-1} \mathbf{R}_{n-1}^{-1} \mathbf{B}_n \right)^{-1} \mathbf{B}_n^H \lambda^{-1} \mathbf{R}_{n-1}^{-1}.\end{aligned}$$

To come in line with the standard literature, we define

$$\mathbf{P}_n \triangleq \mathbf{R}_n^{-1} = \lambda^{-1} \mathbf{P}_{n-1} - \mathbf{K}_n \mathbf{B}_n^H \lambda^{-1} \mathbf{P}_{n-1} \quad (20)$$

where the gain matrix $\mathbf{K}_n$ is

$$\begin{aligned}\mathbf{K}_n &= \lambda^{-1} \mathbf{P}_{n-1} \mathbf{B}_n \left(\mathbf{I}_D + \mathbf{B}_n^H \lambda^{-1} \mathbf{P}_{n-1} \mathbf{B}_n \right)^{-1} \\ &= \mathbf{P}_{n-1} \mathbf{B}_n \left(\lambda \mathbf{I}_D + \mathbf{B}_n^H \mathbf{P}_{n-1} \mathbf{B}_n \right)^{-1}. \quad (21)\end{aligned}$$

Before we move on, it is necessary to bring $\mathbf{K}_n$ into another form

$$\begin{aligned}\mathbf{K}_n \left(\mathbf{I}_D + \lambda^{-1} \mathbf{B}_n^H \mathbf{P}_{n-1} \mathbf{B}_n \right) &= \lambda^{-1} \mathbf{P}_{n-1} \mathbf{B}_n \\ \mathbf{K}_n + \lambda^{-1} \mathbf{K}_n \mathbf{B}_n^H \mathbf{P}_{n-1} \mathbf{B}_n &= \lambda^{-1} \mathbf{P}_{n-1} \mathbf{B}_n\end{aligned}$$

Subtracting the second term on the left hand side yields

$$\mathbf{K}_n = \lambda^{-1} \left(\mathbf{P}_{n-1} - \mathbf{K}_n \mathbf{B}_n^H \mathbf{P}_{n-1} \right) \mathbf{B}_n. \quad (22)$$

With the recursive definition of $\mathbf{P}_n$ the desired form follows

$$\mathbf{K}_n = \mathbf{P}_n \mathbf{B}_n. \quad (23)$$

Now we are ready to complete the recursion. As discussed,

$$\hat{\mathbf{h}}_n = \mathbf{P}_n\mathbf{r}_n = \mathbf{P}_n(\lambda\mathbf{r}_{n-1} + \mathbf{B}_n\mathbf{y}_n) = \lambda\mathbf{P}_n\mathbf{r}_{n-1} + \mathbf{P}_n\mathbf{B}_n\mathbf{y}_n. \quad (24)$$

The second step follows from the recursive definition of $\mathbf{r}_n$ in (18). Next we incorporate the recursive definition of $\mathbf{P}_n$ in (20) together with the alternate form of $\mathbf{K}_n$ and get

$$\begin{aligned}\hat{\mathbf{h}}_n &= \lambda\left(\lambda^{-1}\mathbf{P}_{n-1} - \lambda^{-1}\mathbf{K}_n\mathbf{B}_n^H\mathbf{P}_{n-1}\right)\mathbf{r}_{n-1} + \mathbf{K}_n\mathbf{y}_n \\ &= \mathbf{P}_{n-1}\mathbf{r}_{n-1} - \mathbf{K}_n\mathbf{B}_n^H\mathbf{P}_{n-1}\mathbf{r}_{n-1} + \mathbf{K}_n\mathbf{y}_n \\ &= \mathbf{P}_{n-1}\mathbf{r}_{n-1} + \mathbf{K}_n\left(\mathbf{y}_n - \mathbf{B}_n^H\mathbf{P}_{n-1}\mathbf{r}_{n-1}\right).\end{aligned}$$

With (22), we arrive at the update equation

$$\hat{\mathbf{h}}_n = \hat{\mathbf{h}}_{n-1} + \mathbf{K}_n\left(\mathbf{y}_n - \mathbf{B}_n^H\hat{\mathbf{h}}_{n-1}\right) = \hat{\mathbf{h}}_{n-1} + \mathbf{K}_n\boldsymbol{\alpha}_n \quad (25)$$

where $\boldsymbol{\alpha}_n = \mathbf{y}_n - \mathbf{B}_n^H\hat{\mathbf{h}}_{n-1}$ is the a priori error. That means we found the correction factor

$$\Delta\hat{\mathbf{h}}_{n-1} = \mathbf{K}_n\boldsymbol{\alpha}_n \quad (26)$$

This intuitively satisfying result indicates that the correction factor is directly proportional to both the error and the gain vector, which controls how much sensitivity is desired, through the forgetting factor λ. When the channel is static over the transmission duration, it is logical to set the forgetting factor λ to one. On the other hand, when the channel is time-varying, in order to track the channel variations one needs to set λ to a value that corresponds to the coherence time of the channel.

By using the above equations, the recursive algorithm for this application can be summarized as

Initialization

$$\hat{\mathbf{h}}_{0,\mathit{eff}} = \mathbf{0} \quad \text{and} \quad \mathbf{P}_0 = \delta^{-1}\mathbf{I}_D$$

where $\mathbf{I}_D$ is the identity matrix of rank D and δ is a small positive number.

Computation

$$\begin{aligned}&\text{For } n = 1, 2, 3\ldots \\ \mathbf{y}_n &= \boldsymbol{\Psi}_n^H\mathbf{h}_n + \boldsymbol{\xi}_n \\ \mathbf{B}_n &= (\mathbf{x}_n \otimes \mathbf{I}_D) \\ \mathbf{e}_n &= \mathbf{y}_n - \mathbf{B}_n^H\hat{\mathbf{h}}_{n-1,\mathit{eff}} \\ \mathbf{S}_n &= \lambda\mathbf{I}_D + \mathbf{B}_n^H\mathbf{P}_{n-1}\mathbf{B}_n \\ \mathbf{F}_n &= \mathbf{P}_{n-1}\mathbf{B}_n \\ \mathbf{K}_n &= \mathbf{F}_n\mathbf{S}_n^{-1} \\ \mathbf{P}_n &= \lambda^{-1}\mathbf{P}_{n-1} - \lambda^{-1}\mathbf{K}_n\mathbf{F}_n^H \\ \hat{\mathbf{h}}_{n,\mathit{eff}} &= \hat{\mathbf{h}}_{n-1,\mathit{eff}} + \mathbf{K}_n\mathbf{e}_n.\end{aligned}$$

where we reinserted the subscript *eff* into the equations.

C. RR-Kalman Filtering

By using (4) one can write the RR-Kalman filter model, using the standard derivation as follows [4]

Initialization

$$\hat{\mathbf{h}}_{0|-1} = 0 \quad \text{and} \quad \mathbf{P}_{0|-1} = \mathbf{R}_{\mathbf{h}} \quad (27)$$

Prediction

$$\hat{\mathbf{h}}_{n|n-1} = \sqrt{\alpha}\hat{\mathbf{h}}_{n-1|n-1} \quad (28)$$

$$\mathbf{P}_{n|n-1} = \alpha\mathbf{P}_{n-1|n-1} + (1-\alpha)\mathbf{R}_{\mathbf{h}} \quad (29)$$

Update

$$\mathbf{z}_n = \mathbf{y}_n - \boldsymbol{\Psi}_n^H\hat{\mathbf{h}}_{n|n-1} \quad (30)$$

$$\mathbf{S}_n = \boldsymbol{\Psi}_n^H\mathbf{P}_{n|n-1}\boldsymbol{\Psi}_n + \mathbf{S}_D^H\mathbf{R}_\eta\mathbf{S}_D \quad (31)$$

$$\mathbf{K}_n = \mathbf{P}_{n|n-1}\boldsymbol{\Psi}_n\mathbf{S}_n^{-1} \quad (32)$$

$$\hat{\mathbf{h}}_{n|n} = \hat{\mathbf{h}}_{n|n-1} + \mathbf{K}_n\mathbf{z}_n \quad (33)$$

$$\mathbf{P}_{n|n} = (\mathbf{I} - \mathbf{K}_n\boldsymbol{\Psi}_n^H)\mathbf{P}_{n|n-1} \quad (34)$$

where $\boldsymbol{\Psi}_n^H = (\mathbf{x}_n^H \otimes \mathbf{S}_D^H)$ and $\mathbf{x}_n$ is defined in (2). Note that presence of $\boldsymbol{\Psi}_n$ is the reason for reduction in dimensionality. This makes this algorithm different than conventional Kalman filtering.

D. Transient MSE Analysis for General Adaptive Methods

Channel estimation update rule can be given in the following form

$$\begin{aligned}\hat{\mathbf{h}}_n &= \mathbf{A}_n\hat{\mathbf{h}}_{n-1} + \mathbf{W}_n\mathbf{y}_n \\ \mathbf{y}_n &= \boldsymbol{\Psi}_n^H\mathbf{h}_n + \boldsymbol{\xi}_n.\end{aligned}$$

Here, $\hat{\mathbf{h}}_n$ is the channel estimation vector at nth update interval for the effective channel $\mathbf{h}_n$, and the covariance of $\boldsymbol{\xi}$ is defined as $\mathbf{R}_{\boldsymbol{\xi}} \triangleq \mathbf{S}_D^H\mathbf{R}_\eta\mathbf{S}_D$.

Define

$$\mathbf{E}_n \triangleq \frac{1}{KLC}\mathbb{E}[(\hat{\mathbf{h}}_n - \mathbf{h})(\hat{\mathbf{h}}_n - \mathbf{h})^H] = \frac{1}{KLC}\mathbb{E}[\mathbf{v}_n\mathbf{v}_n^H] \quad (35)$$

where $\mathbf{v}_n = \hat{\mathbf{h}}_n - \mathbf{h}$, and for RR-Kalman filtering, $\mathbf{h}$ is the full dimensional channel vector, $C = N$; for both RR-RLS and RR-LMS algorithms, $\mathbf{h}$ is the effective channel vector, and $C = D$. One can compute (35) by adopting the channel model in (4) and defining

$$\begin{aligned}\mathbf{R}_{\hat{\mathbf{h}}}(n) &\triangleq \mathbb{E}\left[\hat{\mathbf{h}}_n\hat{\mathbf{h}}_n^H\right] \\ \mathbf{R}_{\mathbf{h}} &\triangleq \mathbb{E}\left[\mathbf{h}\mathbf{h}^H\right] \text{ (given in (5))} \\ \mathbf{P}(n) &\triangleq \mathbf{P}_{\hat{\mathbf{h}}\mathbf{h}}(n) \triangleq \mathbb{E}\left[\hat{\mathbf{h}}_n\mathbf{h}_n^H\right].\end{aligned}$$

It can be shown that

$$\mathbf{E}_n = \frac{1}{KLC}\left(\mathbf{R}_{\hat{\mathbf{h}}}(n) + \mathbf{R}_{\mathbf{h}} - (\mathbf{P}(n) + \mathbf{P}^H(n))\right). \quad (36)$$

E. Algorithm Complexity

It can be seen from Table I that, the complexity orders of three algorithms depend on the beamspace dimension (D), the number of antennas in the BS (N), and the number of iterations (n). The complexity orders of both RR-LMS and RR-RLS algorithms depend only on D and n.

TABLE I
ALGORITHM COMPLEXITY

RR-LMS	RR-RLS	RR-Kalman
$O(nD^2)$	$O(nD^3)$	$O(nN^2D + nND^2 + nD^3)$

IV. NUMERICAL RESULTS AND DISCUSSION

Based on the system model in Section II, we provide numerical results to evaluate the performance of the three adaptive filtering methods discussed in Section III-D. In the preparation of the results reported below, the following parameter settings were used. We consider a massive MIMO system in TDD mode with one BS and K users where each

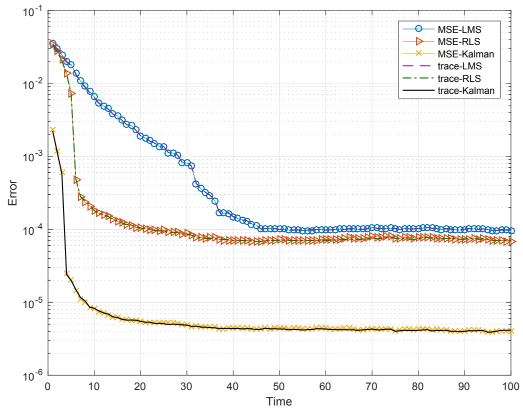

Fig. 1. MSE for different methods of estimating the channel coefficient vector, RR-LMS algorithm, RR-RLS algorithm and RR-Kalman filtering based on theory, i.e., the trace of (36), and Monte-Carlo simulations, i.e., MSEs of (37) and (38) with respect to time ($\sqrt{\alpha} = 0.9999$, $D = 8$).

user has a single receive antenna. By using the uplink training, we try to estimate the channel coefficient vector. Also, the BS is equipped with a uniform linear array (ULA) of $N = 100$ antenna elements along the y-axis. For grouping the users, they were clustered into eight different groups ($G = 8$). The location of each UT is at a specific azimuth angle θ along the ring centered at the origin in the $x - y$ plane. For grouping users one can use the proper user grouping algorithms in [3], [6]. In this work, we assume that users come in groups. User grouping has already been achieved by using such a method in JSDM framework.

In the simulations, we assume that each of the 8 groups has 3 MPCs, i.e., $L_{g_i} = 3$. We only focus on the group g, which have two users being served simultaneously, i.e., $K_g = 2$. Other 7 groups consists of 3 users, i.e., $K_{g'} = 3$, $g' \neq g$ where these users have interference with the intended group g. Also the first two MPCs of group g stem from an azimuth angular sector $[-1°, 1°]$ for delays at $l = 0, 1$. For the last MPC at $l = 2$ in group g, the angular sector is $[5°, 7°]$ in azimuth. The angular sector for other 7 groups are given by $[-29°,-26°]$, $[-21°,-19°]$, $[-12°,-9°]$, $[-5.5°,-3.5°]$, $[9.5°, 12.5°]$, $[15°, 17°]$, $[24°, 27°]$ in azimuth respectively [4], [5]. Also the noise power is set as $N_0 = 1$ so that all dB power values are relative to 1 and the input SNR is 30 dB. The channel covariance matrix of each group can be calculated via the same ways used in [1], [2]. For calculating the mean square error (MSE) per dimension, we used

$$e_{n1} = \frac{1}{KLN}\mathbb{E}[\mathbf{v}_n^H \mathbf{v}_n] \tag{37}$$

for RR-Kalman filtering and

$$e_{n2} = \frac{1}{KLD}\mathbb{E}[\mathbf{v}_n^H \mathbf{v}_n] \tag{38}$$

for $n = 1, 2, ..., T$, for RR-LMS and RR-RLS algorithms. As can be seen from Fig. 1, the MSE performance of the RR-Kalman filtering for both (35) and (37) is the best one compared to other algorithms. Its convergence speed is the fastest and its excess MSE is lowest, followed by the RR-LMS and RR-RLS algorithms, respectively, for both criteria. All algorithms will reach their final value prior to about 50 iterations, with RR-Kalman first, followed by RR-RLS, and RR-LMS being a distant third. Note that, although the three algorithms are different due to dimensionality reduction, this outcome is similar to the conventional versions of these algorithms. Fig. 1 is plotted for only one beamspace dimension ($D = 8$). By plotting the MSE with respect to beamspace dimension for each algorithm, where the value of MSE is the average of the last 50 elements of the error vector, the results for all three algorithms for different temporal fading coefficient (α) are in Fig. 2, 3, 4. As it can be seen, by increasing the beamspace dimension, the final value of the MSE in all algorithms saturates after some point. The saturation point is around $D = 8$ for the RR-Kalman filtering, but for other two algorithms, this value is greater than 8, and unlike RR-Kalman filtering, the difference between the final value of MSE after saturation and the value of MSE for $D = 1$ is less than 0.1. To be specific, by increasing the value of α, the saturation dimension increases. Also, for the time-varying channel at $\alpha = 0.7$ to the static channel where $\alpha = 1$, it can be seen that increasing α leads to better performance of the algorithm, which can be expected. For the RR-RLS algorithm, there is an optimum value for the forgetting factor (λ) for each channel fading coefficient (α). Also, the RR-RLS algorithm is not sensitive to δ. On the other hand, for the RR-LMS algorithm, the optimum value for μ is approximately constant for different values of channel fading coefficient (α) and is around 0.1.

Note that the dimensionalities of $\mathbf{h}_n$ and $\hat{\mathbf{h}}_n$ in (37) are the same, both equal to KLN. But these dimensions are different in the case of (38), where $\mathbf{h}_n$ has dimension KLN, and $\hat{\mathbf{h}}_n$ has dimension KLD. To set the two dimensions the same, we use the matrix multiplication $\mathbf{Q}\mathbf{h}_n$ in (3). This method gives good comparative results for MSE, as shown in Fig. 1. However, for determining the optimum value of D by using MSE, it does not. For that reason, we will use a capacity formulation for a determination of the optimal D.

On the other hand, when the achievable information rate (*AIR*) is plotted, a different behavior is observed. As it can be seen from Fig. 5 by increasing the value of α, i.e., the channel is more static, as we can see, the algorithms perform well in order to reach the genie-aided capacity bound. It is important to note that one can determine the optimal number of RF chains used in hybrid beamforming (required dimension in pre-beamforming) (depending on the adopted adaptive channel estimation technique and channel statistical information) based on the developed analytical tools.

V. Conclusion

In this paper, estimating the channel vector coefficient with different methods by using the pre-beamforming technique was investigated in massive MIMO systems. This work is based on the idea of dividing users in different groups. We came up with different algorithms to estimate the channel coefficients. The first algorithm is the RR-LMS algorithm which avoids matrix inversion. The trade-off for computational

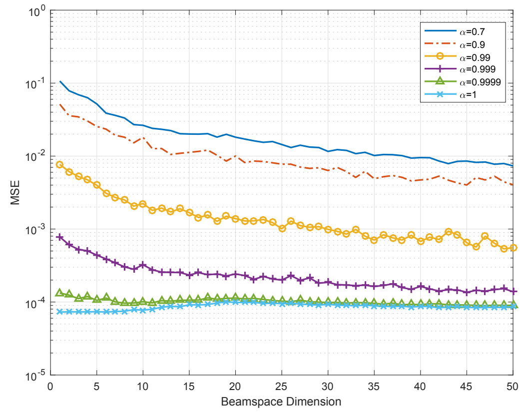

Fig. 2. MSE for RR-LMS algorithm with respect to beamspace dimension.

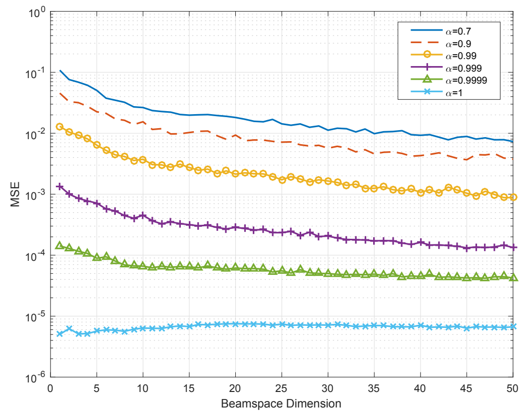

Fig. 3. MSE for RR-RLS algorithm with respect to beamspace dimension.

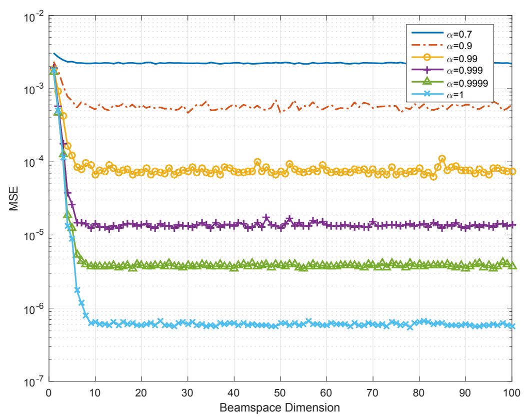

Fig. 4. MSE for RR-Kalman filtering with respect to beamspace dimension.

simplicity is in convergence time and performance. The second algorithm studied is the RR-RLS algorithm. The algorithm is a recursive method to estimate the channel. The third algorithm is the RR-Kalman filter. Comparing these three methods based on MSE and capacity leads to the best performance among them which is the RR-Kalman filter. The interesting conclusion is that for MSE, the RR-Kalman algorithm performance saturates around $D = 8$, while RR-LMS and RR-RLS require a larger D, especially when the channel is more time-varying. The reason for this is the expansion of noise subspace as the beamspace dimension (D) increases for RR-LMS and RR-RLS, since the a priori statistical information of the channel

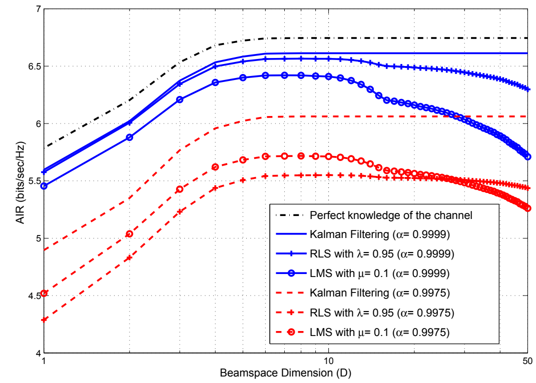

Fig. 5. Capacity of different methods of estimating the channel, RR-RLS algorithm, RR-LMS algorithm and RR-Kalman filtering with respect to beamspace dimension and for the actual channel for two different temporal fading coefficients $\sqrt{\alpha} = 0.9999$ and $\sqrt{\alpha} = 0.9975$ with SNR = 20 dB, interference-to-noise ratio = 30 dB.

after pre-beamforming is not utilized for these two methods. On the other hand, the capacity results for all three are very similar and indicate a saturation around $D = 8$. Considering the very large complexity advantages, this result indicates a preference towards simpler algorithms. Another important point is that while the RR-Kalman algorithm is matched to the channel model, RR-LMS and RR-RLS do not have this property.

REFERENCES

[1] D. Kim, G. Lee, and Y. Sung, "Two-stage beamformer design for massive MIMO downlink by trace quotient formulation," *IEEE Trans. Commun.*, vol. 63, pp. 2200–2211, Jun. 2015.

[2] A. Adhikary, J. Nam, J. Y. Ahn, and G. Caire, "Joint spatial division and multiplexing: The large-scale array regime," *IEEE Trans. Inf. Theory*, vol. 59, pp. 6441–6463, Oct. 2013.

[3] J. Nam, A. Adhikary, J. Y. Ahn, and G. Caire, "Joint spatial division and multiplexing: Opportunistic beamforming, user grouping and simplified downlink scheduling," *IEEE J. Sel. Topics Signal Process.*, vol. 8, pp. 876–890, Oct. 2014.

[4] G. M. Guvensen and E. Ayanoglu, "Beamspace aware adaptive channel estimation for single-carrier time-varying massive MIMO channels," *Proc. of IEEE International Conference on Communications*, pp. 1–7, May 2017.

[5] ——, "A generalized framework on beamformer design and CSI acquisition for single-carrier massive MIMO systems in millimeter wave channels," *Proc. of IEEE Globecom 2016 Workshops*, pp. 1–7, Dec. 2016.

[6] A. Adhikary, E. A. Safadi, M. K. Samimi, R. Wang, G. Caire, T. S. Rappaport, and A. F. Molisch, "Joint spatial division and multiplexing for mm-wave channels," *IEEE J. Sel. Areas Commun.*, vol. 32, pp. 1239–1255, Jun. 2014.

[7] L. Tong, B. M. Sadle, and M. Dong, "Pilot-assisted wireless transmissions: General model, design criteria, and signal processing," *IEEE Signal Process. Mag.*, vol. 21, no. 6, pp. 12–25, Nov. 2004.

[8] W. C. Jakes, *Microwave Mobile Communication*. New York, NY: Wiley, 1974.

[9] S. Haykin, *Adaptive Filter Theory, 4th ed.* Englewood Cliffs, NJ: Prentice- Hall, 2002.