A Burst-Form CSI Estimation Approach for FDD Massive MIMO Systems

Mohammad Javad Azizipour, *Student Member, IEEE,* Kamal Mohammed-pour, and A. Lee Swindlehurst, *Fellow, IEEE*

***Abstract*—Pilot and channel state information (CSI) feedback overhead in the downlink and uplink paths are two major implementation challenges in frequency-division duplex (FDD) based massive MIMO systems. When the massive MIMO channel satisfies the burst-sparsity property, we can acquire the channel with compressed pilots and CSI feedback in a more efficient approach. This paper proposes a burst-form estimation approach, referred to as the burst-form least squares (BFLS) algorithm, to fully utilize the burst-sparsity property of massive MIMO channels. The proposed algorithm is based on knowledge of the starting location of each burst at the user side. For situations where the starting locations change quickly or are otherwise initially unknown at the user, a starting point estimation (SPE) algorithm is proposed to provide the position of each burst in the channel vector. Numerical results demonstrate that the BFLS algorithm acquires the channel better than competing approaches and reaches the performance upper bound. It is shown that the SPE algorithm can find the location of bursts with high accuracy and using the estimated values do not significantly degrade the estimation quality.**

***Index Terms*—Massive MIMO, Compressed sensing, Channel estimation, Pilot overhead, Burst-form least square**

I. Introduction

Massive multi-input multi-output (MIMO) has been introduced as a promising technology for the physical layer of wireless systems in order to enhance the spectral and also energy efficiency of future 5G networks [1], [2]. To achieve the gains promised by the use of a large number of antennas, the channel state information (CSI) should be acquired at the transmitter side with high precision. The time-division duplex (TDD) protocol assumes channel reciprocity, in which the downlink and uplink channels have the same frequency response. Based on reciprocity between the channels, the BS side is able to estimate the CSI by using pilot training in the uplink path, where the number of pilot symbols is proportional to the number of users [3]–[5]. This property does not hold for frequency-division duplex (FDD) systems because of the difference in the downlink and uplink frequencies. As a result, FDD systems must in general estimate the CSI via pilots whose length is on the order of the number of antennas [4], [6], which is very large for massive MIMO systems. In such cases the pilot and CSI feedback would be correspondingly large, and thus it is thought that massive MIMO will not have high throughput when operating in FDD mode [4], [7]. Since FDD-based systems are used as the dominant duplex mode in current wireless networks, many researchers have tackled this challenge in order to find a better solution for migration from previous wireless systems to 5G networks [8]–[15].

In many practical scenarios, particularly at millimeter-wave frequencies, due to limited local scattering the massive MIMO channel will exhibit sparsity in the spatial domain [15]; *i.e.,* the propagation paths between the BS antennas and the users is composed of only a few directions in the angular domain, a property referred to as *angular sparsity*. In such cases, compressed sensing (CS) methods can be applied to estimate the massive MIMO channel with fewer pilots and CSI measurements compared to conventional channel estimation procedures [16]–[19]. Thus, CS-based channel estimation algorithms and CS-based pilot design methods are considered to be promising directions for solving the pilot overhead issue in FDD-based massive MIMO systems [20], [21]. As one of the first contributions in this area, [11] improved the well-known orthogonal matching pursuit (OMP) algorithm by exploiting the common support shared by users located close to each other. The authors of [12] proposed a structured compressed sensing algorithm and used the spatial and temporal sparsity in each sub-channel to enhance the resulting estimation quality. In another work [15], they used an adaptive algorithm for pilot and CSI feedback compression by utilizing the same properties in the sub-channels. The challenge yet to be overcome is that these algorithms have been developed for certain special cases of massive MIMO channels, e.g., where the users' channels share the same scatterers in the propagation environment or the same sparsity pattern in the sub-channels, properties that may not hold in many scenarios. Another direction taken in the literature [22]–[24] is to improve CS-based methods by exploiting partial support information. In all these works, the channel recovery problem is formulated as a weighted linear optimization problem. The results show that the estimation quality is dependent on the given partial support information and the quality is decreased when the information is not accurate.

The sparsity observed in massive MIMO channels does not mean that the signal arrivals are confined to a single direction, but instead to a cluster of directions whose angular spread may or may not be narrow, depending on the limited number of physical scatterers near the user in the propagation environment. We refer to this type of sparsity structure as *burst-sparsity*, a property that has been exploited, for example, in [13], [25], [26]. In [25], the conventional LASSO algorithm [27] was redesigned to be compatible with the burst structure in massive MIMO channels. Moreover, in [13], authors improved the proposed burst LASSO approach by exploiting partial support information assumed to be available at the user side. Despite using burst sparsity to enhance the recovery performance, the resulting burst LASSO algorithm shows satisfactory results only when the support information is accurately known at the user side. Furthermore, the algorithm suffers from high computational complexity especially when the number of antennas and the burst size are large.

In this paper, to overcome the issues raised above, we propose a new approach based on the burst property that yields a simple least squares (LS) solution, referred to as the burst-form LS (BFLS) algorithm. Two scenarios are considered. In the first, the user is aware of the starting location of the bursts, and in the second, the starting location of the bursts is unknown and must be estimated. In the first scenario, the proposed BFLS algorithm estimates the channels and refines the estimates by exploiting the burst structure. In this case, the estimated CSI quality is shown in the simulations to provide performance equivalent to that of a genie-aided LS algorithm and has very low computational complexity. In the second scenario, an algorithm is presented for estimating the burst starting points, which are then used by the BFLS algorithm for channel acquisition. Furthermore, an error correction mechanism is designed to enhance the estimation accuracy of the burst starting points before halting the estimation procedure, because the algorithm may produce large errors due to the small number of assigned pilots at the base station (BS) side.

By estimating the starting location of each burst, the proposed algorithm is not dependent on the existence of any supporting information and is capable of channel estimation whether or not the starting location of the bursts is known. In addition, our proposed algorithms do not impose heavy complexity at the user side and also can estimate the channel with less complexity than algorithms of the LASSO family. The rest of the paper is organized as follows. Section II presents the downlink signal model for FDD massive MIMO systems and also the burst structure of the channel. In Section III, the proposed burst-form LS algorithm is introduced. In this section, we also describe how to utilize the burst sparsity when the starting location is known. In Section IV, the algorithm for estimating the burst starting locations is developed when they are unknown. The performance of BFLS and the starting point estimation algorithms are then evaluated in Section V. Finally, Section VI concludes the paper.

Notation: Matrix and vectors are denoted in upper and lower boldface, respectively. The operators $(.)^*$, $(.)^{\mathrm{T}}$, $(.)^{\mathrm{H}}$, $|.|$, $(.)^{\dagger}$, $\mathbb{E}\{.\}$, and symbols $\mathbb{Z}$, $\mathbf{I}$, and $\mathbf{0}$ are used for conjugate, transpose, conjugate transpose, cardinality, Moore-Penrose pseudo-inverse, expectation, the set of integers, identity matrix, and zero vector respectively. The ℓ_p-norm of vector $\mathbf{a}$ is defined as $\|\mathbf{a}\|_p \triangleq \left(\sum_{i=1}^{n} |a_i|^p\right)^{1/p}$. The j-th element of given set $\mathcal{A}$ is denoted by $\mathcal{A}[j]$, the specific i-th column of matrix $\mathbf{A}$ and the columns with indices in set $\mathcal{I}$ are denoted by $\mathbf{A}[i]$ and $\mathbf{A}[\mathcal{I}]$ respectively, and $\mathbf{a}[i]$ refers to the i-th entry of vector $\mathbf{a}$ and those elements whose indices are in set $\mathcal{I}$ are denoted by $\mathbf{a}[\mathcal{I}]$.

II. Problem Formulation

A. FDD Massive MIMO

As illustrated in Fig. 1, we consider an FDD massive MIMO system with N antennas at the BS and one antenna at the user side. In the FDD protocol, the BS broadcasts M pilot signals to the users in the downlink path and the estimated CSI is sent back to the BS for downlink channel precoding. Without loss of generality, we only focus on the reference user for downlink channel estimation. The received signal at the user can be written as

$$\mathbf{y} = \mathbf{h}^{\mathrm{H}}\mathbf{X} + \mathbf{n}, \tag{1}$$

where $\mathbf{h} \in \mathbb{C}^N$ is the channel vector between each antenna element and the user, $\mathbf{X} \in \mathbb{C}^{N\times M}$ is the pilot matrix consisting of M pilots with total transmit power $\mathbb{E}\{\mathbf{x}_i^{\mathrm{H}}\mathbf{x}_i\} = P = Mp$, where $\mathbf{x}_i^{\mathrm{T}}$ is the i-th row of $\mathbf{X}$, and $\mathbf{n} \in \mathbb{C}^{1\times M}$ is complex additive Gaussian noise in the downlink path with zero mean and covariance $\mathbb{E}\{\mathbf{n}\mathbf{n}^{\mathrm{H}}\} = \sigma_n^2\mathbf{I}$. Now, the user is able to apply a suitable estimation algorithm to the measurement vector $\mathbf{y}$ for downlink channel estimation. Note that conventional channel estimation approaches require $M \geq N$ pilots, which results in an overwhelming pilot overhead for the downlink path. By considering CS-based methods, the BS can compress the required number of pilots and the resulting measurements can be adjusted to the relatively small number of non-zero channel entries. Although the channel support is an unknown variable, the sparsity upper bound changes slowly in each time slot and the BS can acquire it using prior information.

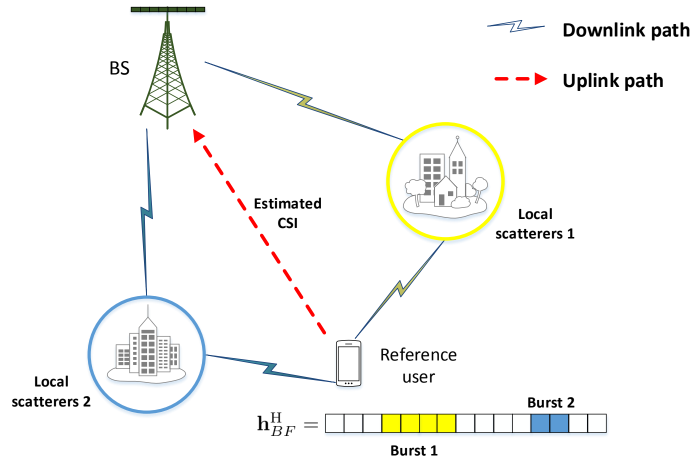

Fig. 1. FDD massive MIMO for uplink and downlink paths in a burst-sparse channel.

B. Burst-sparse Channel Model

Since the BS antennas are often installed at high elevations surrounded by few local scatterers, the angular support of the channel seen at the BS side is relatively small. Accordingly, some dominant elements in the channel vector can represent nearly all multipath directions between the BS and the users. More precisely, the channel has an angular representation with a small number of non-zero coefficients, and owing to the large number of antennas in massive MIMO, we can consider it to be a sparse vector in the angular domain. As depicted in Fig. 1, each cluster of local scatterers (denoted by rings) near the user yields a contiguous block of non-zero bins in the angular domain of the channel response with a specific starting and ending point, which we refer to as a burst area.

When the massive MIMO channel is represented in the angular domain, the burst channel model can be expressed as [28]

$$\mathbf{h}^{\mathrm{H}} = \mathbf{h}_{BF}^{\mathrm{H}}\mathbf{G}, \tag{2}$$

where $\mathbf{G} \in \mathbb{C}^{N\times N}$ is a transformation matrix that is dependent on the geometrical structure of the antenna array, and $\mathbf{h}_{BF}^{\mathrm{H}} \in \mathbb{C}^{1\times N}$ is the burst-sparse channel vector in the angular domain. For a uniform linear array, the transformation matrix can be taken to be the discrete Fourier transform (DFT) matrix, which implies that the channel vector is sampled with uniform angular intervals at the BS side [28]. The burst-sparse channel vector $\mathbf{h}_{BF}^{\mathrm{H}} = [\mathbf{0},\ldots,\mathbf{0},\mathbf{h}_1^*,\ldots,\mathbf{0},\ldots,\mathbf{h}_B^*,\ldots,\mathbf{0}]$ consists of B bursts, with each burst forming a sub-vector $\mathbf{h}_l^* \in \mathbb{C}^{1\times d_l}$, where d_l is the length of l-th burst. Two parameters uniquely identify the l-th burst, the starting point s_l, and the length d_l which satisfies $d_l \leq U$, for some (known) upper bound U on the length of all bursts. Therefore, the indices of the non-zero elements in the l-th burst can be extracted from the set $\mathcal{S}_l = \{s_l + j|\ 0 \leq j \leq (U-1),\ j \in \mathbb{Z}\}$. Among the defined parameters, the BS can determine the number of bursts B, and the upper bound U on the burst length since these parameters change relatively slowly. However, the case of the starting point s_l is different since it can vary significantly with only a small change in the user position. It is worth noting that the defined burst model is not equivalent to the well-known block sparse model [29], since the bursts do not necessarily have the same length, and the blocks of zero elements in the channel angular response are also of varying lengths and in random positions.

C. CS Formulation for Downlink Channel Estimation

The downlink channel estimation problem can be formulated as a compressed sensing problem using previous methods for massive MIMO systems and sparse channels. By substituting (2) into (1), the received signal at the user side is given by

$$\mathbf{y} = \mathbf{h}_{BF}^{\mathrm{H}}\mathbf{G}\mathbf{X} + \mathbf{n}. \tag{3}$$

By taking the conjugate transpose of both sides and defining $\mathbf{\Psi} \triangleq \mathbf{X}^{\mathrm{H}}\mathbf{G}^{\mathrm{H}}$, the received signal in the downlink path can be written in the compressed sensing model as

$$\mathbf{y}_{\mathrm{cs}} = \mathbf{\Psi}\mathbf{h}_{BF} + \mathbf{n}_{\mathrm{cs}}, \tag{4}$$

where $\mathbf{y}_{\mathrm{cs}} = \mathbf{y}^{\mathrm{H}}$ is the measurement signal, $\mathbf{\Psi}$ is the sensing matrix and $\mathbf{n}_{\mathrm{cs}} = \mathbf{n}^{\mathrm{H}}$ is the resized vector of additive noise.

There are several recovery algorithms that can be used to reconstruct the channel vector $\mathbf{h}_{BF}$ by solving the M linear equations in (4). A well-known solution to this problem, the LASSO approach[1], is formulated as the following convex optimization problem:

$$\min_{\hat{\mathbf{h}}_{BF}} \|\mathbf{y}_{\mathrm{cs}} - \mathbf{\Psi}\hat{\mathbf{h}}_{BF}\|^2 + \xi\|\hat{\mathbf{h}}_{BF}\|_1 \tag{5}$$

where $\xi > 0$, and $\hat{\mathbf{h}}_{BF}$ is the reconstructed channel vector. However, the standard LASSO algorithm can only exploit non-specific sparse structure in the channel vector, and does not take into account the special burst nature of the channels we consider. Instead, one may consider solving the following optimization problem which exploits burst sparsity:

$$\begin{aligned} &\min_{\hat{\mathbf{h}}_{BF}} \|\mathbf{y}_{\mathrm{cs}} - \mathbf{\Psi}\hat{\mathbf{h}}_{BF}\|_2^2 \\ &\text{subject to} \quad \|\hat{\mathbf{h}}_{BF}\|_1 \leq \epsilon, \quad \hat{\mathbf{h}}_{BF} = \sum_{l=1}^{B}\mathbf{h}_l, \end{aligned} \tag{6}$$

where ϵ is a positive parameter. Applying block-sparse recovery algorithms directly to (6) will not produce satisfactory results, since the burst constraint in (6) can not be handled by the block-sparse model. By converting the burst-sparse model into a block-sparse model, a burst-LASSO approach was introduced in [13], [25] which can handle (6). However, as discussed in Section II-B, the zero channel elements do not share the block structure, and thus the penalty function based on such transformations can incur large performance losses especially for bursts of different lengths. In this paper, a novel least squares approach will be proposed to solve (6) in a simple and efficient way by utilizing the burst model directly.

III. Burst-Form Least Square Approach

In this section, a least squares algorithm is investigated for compressed channel estimation instead of convex optimization approaches. First, we exploit the burst-form of the channel to design an LS-based algorithm. The main challenge of the proposed algorithm is related to the redundant elements which arise due to the use of the upper bound parameter U in forming the burst areas. In other words, the proposed algorithm assumes that the sparsity order of each burst is U, while in general they satisfy[2] $d_l \leq U$. This issue becomes worse when the transmit SNR is low and the incorrectly estimated elements have significant amplitudes. To eliminate them, we apply a refinement step to the algorithm in order to remove the estimated channel coefficients that do not correspond to the actual channel. The details of algorithm are explained later in the section.

A. BFLS Algorithm

At the user side, the measurement signal $\mathbf{y}_{\mathrm{cs}}$ and the sensing matrix $\mathbf{\Psi}$ are available due to availability of the received downlink signal and knowledge of the pilot matrix. If the precise angular support of the channel was known by the user, the least squares estimator for the downlink channel $\mathbf{h}_{BF}$ would be given by

$$\hat{\mathbf{h}}_{BF}[\Omega] = \mathbf{\Psi}^{\dagger}[\Omega]\mathbf{y}_{\mathrm{cs}}, \tag{7}$$

where Ω is the index set of the channel support with cardinality $|\Omega| = \sum_{l=1}^{B} d_l$, and $\mathbf{\Psi}^{\dagger} = (\mathbf{\Psi}^{\mathrm{H}}\mathbf{\Psi})^{-1}\mathbf{\Psi}^{\mathrm{H}}$ is the Moore-Penrose pseudo-inverse of the sensing matrix. If only the starting point s_l but not the length d_l of each burst is known, an initial

[1]Also known as standard LASSO to be distinct from other versions, for example group LASSO, fused LASSO, etc.

[2]Note that the exact number of entries of vector $\mathbf{h}_l^*$ is an unknown parameter for which we only have an upper bound.

estimate can be obtained by using the superset Ω_U:

$$\Omega \subseteq \Omega_U = \bigcup_{l=1}^{B} \mathcal{S}_l, \tag{8}$$

where $\mathcal{S}_l$ is the set of channel elements obtained by starting from s_l and including the subsequent U angles. In this case, the burst-form LS (BFLS) estimator is given by

$$\hat{\mathbf{h}}_{BF}[\Omega_U] = \mathbf{\Psi}^{\dagger}[\Omega_U]\mathbf{y}_{\mathrm{cs}} \ . \tag{9}$$

Note that the BFLS approach estimates only the channel elements whose indices are in the set Ω_U, and the remaining coefficients will be zero based on the sparsity property of the channel. The burst-form LS estimator minimizes the energy between the measurement signal and the compressed samples of the channel such that these samples are aligned with the bursts in the channel. In other words, we fully exploit the burst structure and then solve the estimation problem. In the mathematical expression, $\hat{\mathbf{h}}_{BF}$ is the solution of the optimization problem (6), where the sparsity order in each burst has been set to the upper limit $d_l = U$. Therefore, the estimated channel has $(BU - \sum_{l=1}^{B} d_l)$ erroneously estimated coefficients, and these elements should be removed because their amplitude may not be negligible for some transmit SNR regimes.

To refine the channel estimate and remove the erroneous elements, we introduce the following step in the BFLS algorithm. The core of this step is based on comparing the residual vector with a predetermined threshold parameter. The residual vector is defined as

$$\mathbf{r}_0 = \mathbf{y}_{\mathrm{cs}} - \mathbf{\Psi}[\Omega_U^1]\hat{\mathbf{h}}_{BF}, \tag{10}$$

where $\Omega_U^1 = \Omega_U$ and $\mathbf{r}_0$ are respectively the indices of the channel support and the residual vector before applying the refinement step. We see from this definition that the norm of the residual vector should approach the norm of the additive noise as the redundant channel elements are properly removed. On the other hand, if we eliminate one of the elements corresponding to the original channel support, the norm of the residual will increase according to the energy of the removed element. Therefore, the difference of residual norms in consecutive iterations, i.e. $(\|\mathbf{r}_i\|_2 - \|\mathbf{r}_{i-1}\|_2)$ can be compared with a predefined threshold η_{th} to determine whether or not a coefficient with significant energy has been removed. If the difference of residual norms does not exceed the threshold, then the last presumed support location in the l-th burst is removed, and the norm of the residual is checked again. This process continues until the difference of the norms of the residuals rises above the threshold η_{th}.

More precisely, we let $(U - i)$ denote the last index of the arbitrary burst l in the i-th iteration of the refinement process, and the elements that will be eliminated from consideration at refinement step i are given by

$$\Sigma_i = \Big\{ s_l + (U-1), s_l + (U-2), \ldots, s_l + (U-i) \Big\} \tag{11}$$

where s_l is the starting location of the l-th burst. According to the above definitions, at step i the new burst $\mathcal{I}_i$ becomes the original set minus the i last channel elements:

$$\mathcal{I}_i = \mathcal{S}_l \backslash \Sigma_i, \tag{12}$$

where $\mathcal{I}_i$ is the refined set with cardinality $|\mathcal{I}_i| = U - i$, which determines the new support set for the l-th burst. The residual vector at the i-th iteration of the refinement step is computed by replacing $\mathcal{S}_l$ with the refined burst $\mathcal{I}_i$ to form the new burst areas Ω_{temp} and this is substituted into (10) as $\mathbf{r}_i = \mathbf{y}_{\mathrm{cs}} - \mathbf{\Psi}[\Omega_{\mathrm{temp}}]\hat{\mathbf{h}}_{BF}$. It should be noted that we select refined set $(i-1)$ as the final refined burst because the threshold will stop the algorithm when the original support is removed from the channel. The above procedure will be repeated for each burst individually until the halting condition is met.

Algorithm 1 shown below divides the BFLS algorithm into two main steps. **Step 1** determines the area of each burst and combines them into the initial set Ω_U (lines 4 and 5 in the algorithm). More precisely, the area of each burst is computed using their starting locations and the upper bound U, and then these B bursts are merged together to create the initial channel support that is of interest. Note that when the bursts are close to each other, adjacent bursts may overlap, but the common taps are included only once in Ω_U. In **Step 2** which includes the refinement step (lines $13-20$), the BFLS approach estimates the channel by utilizing the bursts formed in the previous step. The residual vector $\mathbf{r}_0$ is also computed. The computation of $\mathbf{r}_0$ in this step will be used as the previous residual norm at the refinement step. Therefore, in the case of $d_l = U$, the refinement process will be stopped with one iteration. In other words, when U equals the length of the support in each burst, the estimated channel vector $\hat{\mathbf{h}}_{BF}$ does not have any redundant elements and the BFLS algorithm can be stopped without refinement. Before the refinement procedure is performed, the l-th burst will be removed from the burst areas as $\Lambda_l = \Omega_U^l \backslash \mathcal{S}_l$ because it is regenerated with refined elements in line 21 to form the channel support Ω_U^l for the next burst. Due to the individual refinement, the algorithm needs to be repeated $B(U - d_l + 1)$ times to form the final burst-sparse channel estimate.

B. Computational Complexity of the BFLS Algorithm

The complexity of the BFLS algorithm is dominated by the least-squares and residual computations which are performed in lines $10-11$ and $18-19$ of the algorithm. The least-squares step in line 10 has a complexity of $\mathcal{O}(M|\Omega_U^l| + 2M|\Omega_U^l|^2 + |\Omega_U^l|^3)$ operations, and the residual update in line 11 has a complexity of $\mathcal{O}(M|\Omega_U^l|)$ operations, where $|\Omega_U^l|$ is the channel support which is a variable set in each iteration and satisfies $\sum_{l=1}^{B} d_l \leq |\Omega_U^l| \leq BU$. Lines $18-19$ have the same order of complexity with new channel support $|\Omega_{\mathrm{temp}}| = |\Lambda_l| + |\mathcal{I}_i|$, where $|\mathcal{I}_i|$ is decreased in each iteration of the refinement process and satisfies $d_l \leq |\mathcal{I}_i| < U$, and $|\Lambda_l|$ is changed in each burst estimation step, which satisfies $\sum_{j=1, j\neq l}^{B} d_j \leq |\Lambda_l| \leq (B-1)U$. These computations should be repeated proportional to the number of bursts B (lines $10-11$) and the number of over-estimated supports in each burst (lines $18-19$).

Algorithm 1: Proposed BFLS Algorithm

Inputs: $\mathbf{y}_{\mathrm{cs}}$, $\mathbf{\Psi}$, B, U, s_l, $\eta_{\mathrm{th}} \leq 1$.
Output: Estimated burst-sparse channel, $\hat{\mathbf{h}}_{BF}$
Step 1 (Form the Burst Areas)
$\mathcal{S}_l = \{s_l + j|\ 0 \leq j \leq (U-1), j \in \mathbb{Z}\}$, $l = 1, 2, \ldots, B$
$\Omega_U = \bigcup_{l=1}^{B} \mathcal{S}_l$;
Initialize: $\hat{\mathbf{h}}_{BF} = \mathbf{0}_{N\times 1}, \Omega_U^1 = \Omega_U, l = 1.$
Step 2 (Estimation and Refinement)
while $l \leq B$ **do**
  Initialize: $i = 0$, $\|\mathbf{r}_{-1}\|_2 = 0$.
  $\hat{\mathbf{h}}_{\mathrm{temp}} = \mathbf{\Psi}^{\dagger}[\Omega_U^l]\mathbf{y}_{\mathrm{cs}}$;
  $\mathbf{r}_i = \mathbf{y}_{\mathrm{cs}} - \mathbf{\Psi}[\Omega_U^l]\hat{\mathbf{h}}_{\mathrm{temp}}$;
  $\Lambda_l = \Omega_U^l \backslash \mathcal{S}_l$;
  repeat
    $i = i + 1$;
    $\Sigma_i = \{s_l + (U-1), s_l + (U-2), \ldots, s_l + (U-i)\}$;
    $\mathcal{I}_i = \mathcal{S}_l \backslash \Sigma_i$;
    $\Omega_{\mathrm{temp}} = \Lambda_l \cup \mathcal{I}_i$;
    $\hat{\mathbf{h}}_i = \mathbf{\Psi}^{\dagger}[\Omega_{\mathrm{temp}}]\mathbf{y}_{\mathrm{cs}}$;
    $\mathbf{r}_i = \mathbf{y}_{\mathrm{cs}} - \mathbf{\Psi}[\Omega_{\mathrm{temp}}]\hat{\mathbf{h}}_i$;
  until $(\|\mathbf{r}_i\|_2 - \|\mathbf{r}_{i-1}\|_2) > \eta_{\mathrm{th}}$;
  $\mathcal{S}_l = \mathcal{I}_{i-1}$;
  $l = l + 1$;
  $\Omega_U^l = \bigcup_{k=1}^{B} \mathcal{S}_k$;
end
$\hat{\mathbf{h}}_{BF}[\Omega_U^{B+1}] = \hat{\mathbf{h}}_{i-1}$;

IV. Location of Bursts in the Channel

As is seen in Algorithm 1, the BFLS approach requires the starting locations of each burst to form the channel support. Since the positions of the users in the cell typically change, the location of the bursts in the channel vector can also vary greatly and consequently the user may not know these locations in advance. If the BFLS algorithm is implemented with invalid values of s_l, the performance can degrade significantly. In this section, we first introduce an approach that estimates the starting location of each individual burst. Then, the performance of this procedure is analyzed when an error occurs, and we will discuss how to recognize and also correct such errors. Finally, by combing these ideas together, we elaborate an algorithm, referred to as starting point estimation (SPE) algorithm to find the location of the bursts in the channel vector.

A. Starting Point Estimation Algorithm

In general, forming the burst areas is not possible when their starting locations are unknown. However, if we compute the maximum correlation between the received signal and the columns of the sensing matrix, we can hope to identify the channel support within a certain burst. Therefore, the main idea behind the proposed algorithm is to use burst areas identified in this manner and then to search for neighboring channel coefficients that fall within the same burst.

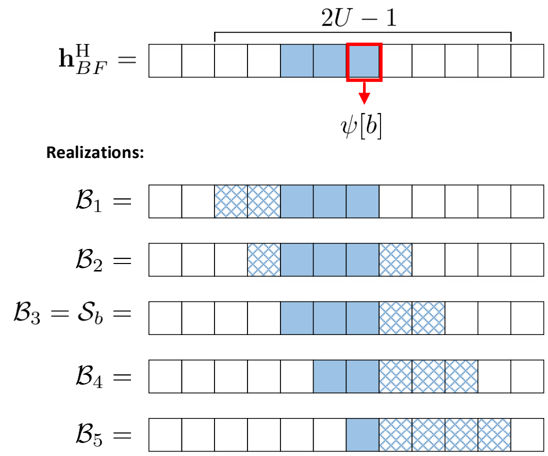

Fig. 2. A simple example of burst realizations with $N = 12$ antennas, $B = 1$ bursts, $d_b = 3$, and sparsity upper bound $U = 5$.

The maximum correlation can be obtained as

$$\psi[l] = \mathrm{argmax}_{t=1,2,\ldots,N}|\mathbf{\Psi}^{\mathrm{H}}[t]\mathbf{r}_{\mathrm{temp}}| \tag{13}$$

where $\psi[l]$ is the column index of the sensing matrix which has the maximum correlation with the residual vector, and $\mathbf{r}_{\mathrm{temp}}$ is the residual vector, which is equal to the measurement signal for the first burst ($l = 1$). Consider a particular burst b and its associated column $\psi[b]$ from the sensing matrix. The set of all possible supports for this burst can be described by

$$\mathcal{A} = \mathcal{R} \cup \mathcal{L} \tag{14}$$

where $\mathcal{R} = \{\psi[b] + j|\ 0 \leq j \leq (U-1), j \in \mathbb{Z}\}$ are the elements on the right side of $\psi[b]$, and $\mathcal{L} = \{\psi[b] - j|\ 0 \leq j \leq (U-1), j \in \mathbb{Z}\}$ are those on the left side of $\psi[b]$. The constructed set $\mathcal{A}$ includes all possible realizations of burst b with length U, which we call $\mathcal{B}_i, i = 1, 2, \ldots, U$.

To see the possible realizations, Fig 2 illustrates bursts with length U formed by starting from the elements on the left side of $\psi[b]$. Since $\mathcal{S}_b \subseteq \mathcal{A}$, one of these realizations should be similar to the original burst area $\mathcal{S}_b$. Assume that $\hat{s}_b \in \mathcal{A}$ indicates a realization which is matched to s_b (for example, set $\mathcal{B}_3$ in the realizations of Fig. 2). Clearly the burst area $\hat{\mathcal{S}}_b = \{\hat{s}_b + j|\ 0 \leq j \leq (U-1), j \in \mathbb{Z}\}$ should minimize the norm of the residual vector. Therefore, we can try each $\mathcal{B}_i$ as a potential burst and compute the corresponding residual to find the one with the smallest norm. The number of searches in this stage is equal to U and the process will be performed only once for each burst. When the starting point of the burst b is identified using the above procedure, the burst area $\mathcal{B}_{i^\star}$ related to the minimum residual norm is saved, the new residual vector is computed as $\mathbf{r}_{\mathrm{temp}}$, and the process is then repeated for the other bursts in order to prevent identification of the same burst in subsequent iterations.

The proposed method for estimation of the starting point is based on maximum correlation computation and residual norm minimization for each burst individually. In this approach, the following problem could occur: if the maximum correlation for burst b corresponds to an index that is outside burst b, the resulting realization sets $\mathcal{B}_i$ may not contain burst b, i.e. $\mathcal{S}_b \nsubseteq \mathcal{A}$, and the algorithm may produce an estimated starting point $\hat{s}_b$ that is far from the true value s_b, or possibly even inside the burst. Therefore, an effective mechanism should be introduced in order to recognize and also correct such errors. This mechanism has been designed as follows.

Assume that, due to noise or interference, the index of maximum correlation $\boldsymbol{\psi}[b]$ for burst b corresponds to an angle index that is located outside of burst b. If we form the burst area based on this incorrect $\boldsymbol{\psi}[b]$, estimate the starting point $\hat{s}_b$, and then remove the effect of the estimated burst from the residual vector, then even if the other bursts have been detected correctly, the final residual $\mathbf{r}_{\text{temp}}$ will likely have a norm greater than the noise threshold even after the algorithm is performed B times to find all bursts. Therefore, we can recognize such errors by comparing the residual norm to a threshold; *e.g.,* if $\|\mathbf{r}_{\text{temp}}\|_2^2 > \mu_{\text{th}}$ after we have initially identified B bursts, then we decide that at least one of the identified bursts has been incorrectly located. The threshold parameter μ_{th} can be chosen to be proportional to the noise level, as $\alpha\mathbb{E}\{\|\mathbf{n}\|_2^2\}$ for some $\alpha > 0$. In particular, since

$$\mathbb{E}\{\|\mathbf{n}\|_2^2\} = \mathbb{E}\left\{\sum_{i=1}^{M}|n_i|^2\right\} = \sum_{i=1}^{M}\mathbb{E}\left\{|n_i|^2\right\} = M\sigma_n^2\ , \quad (15)$$

we have $\mu_{\text{th}} = \alpha(M\sigma_n^2)$. As we sequentially detect the starting location for each burst, and we add the resulting burst to those that have been previously found, we expect that $\|\mathbf{r}_{\text{temp}}\|_2^2 > \mu_{\text{th}}$ will be true until all B bursts have been detected. At this point, assuming the bursts have all been correctly identified, then $\|\mathbf{r}_{\text{temp}}\|_2^2$ should drop below the threshold μ_{th} and the algorithm will terminate. If not, then we assume an error has occurred and we continue the iterations using the approach described below to fine tune the burst locations and correct the error.

The purpose of each additional iteration beyond the first B is to attempt to find a large correlation at an angle that lies outside the union of the current set of detected bursts. The location of this large correlation is thus a candidate for a potential channel support that was missed by the algorithm and should be included in one of the bursts. Suppose $\psi[l]$ is the angle corresponding to a large correlation at iteration $l > B$. For all possible burst locations $\mathcal{B}_i, i = 1, \cdots, U$, surrounding $\psi[l]$, we find the closest burst from the current set and replace it with $\mathcal{B}_i$. We then compute the resulting residual and we repeat this process for all $i = 1, \cdots, U$, to find the $\mathcal{B}_i$ that results in the smallest residual. If the resulting residual falls below the threshold, the algorithm terminates. If not, the process is repeated for the next iteration $l + 1$ until either the residual finally falls below the threshold, or an upper limit $l_{\max}$ on the iterations is reached.

The starting point estimation algorithm is divided into two combined parts, as detailed in Algorithm 2. In the first part, for iterations l such that $l \leq B$, the maximum correlation computation in line 6 finds a candidate location for the l-th burst, and forms the set $\mathcal{A}$ of all possible starting points in line 9. The candidate bursts $\mathcal{B}_i$ are tentatively added to the total burst support in line 15, and the resulting residual is calculated in line 20. This process is repeated for all possible starting locations, and the one with the minimum residual is found in line 22. The corresponding burst is then labeled as the next "true" burst l and is added to the previously detected bursts in line 25. If the resulting residual is below the threshold, the algorithm terminates at line 31.

Algorithm 2: Proposed SPE Algorithm

1 **Inputs:** $\mathbf{y}_{\text{cs}}$, $\boldsymbol{\Psi}$, B, U, μ_{th}, $l_{\max} > B$.
2 **Output:** Estimation of starting points, $\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_B$.
3 *Initialize:* $\mathbf{r}_{\text{temp}} = \mathbf{y}_{\text{cs}}$, $\boldsymbol{\psi} = \mathbf{0}_{l_{\max}\times 1}$, $\mathcal{S}_0 = \emptyset$, $l = 0$.
4 **repeat**
5 $\quad l = l + 1$;
6 $\quad \boldsymbol{\psi}[l] = \text{argmax}_{t=1,2,\ldots,N}|\boldsymbol{\Psi}^{\text{H}}[t]\mathbf{r}_{\text{temp}}|$;
7 $\quad \mathcal{R} = \{\boldsymbol{\psi}[l] + j|\ 0 \leq j \leq (U-1), j \in \mathbb{Z}\}$;
8 $\quad \mathcal{L} = \{\boldsymbol{\psi}[l] - j|\ 0 \leq j \leq (U-1), j \in \mathbb{Z}\}$;
9 $\quad \mathcal{A} = \mathcal{R} \cup \mathcal{L}$;
10 $\quad$ *Initialize:* $i = 0$.
11 $\quad$ **while** $i < U$ **do**
12 $\quad\quad i = i + 1$;
13 $\quad\quad \mathcal{B}_i = \{\mathcal{A}[i] + j|\ 0 \leq j \leq (U-1), j \in \mathbb{Z}\}$;
14 $\quad\quad$ **if** $l \leq B$ **then**
15 $\quad\quad\quad \Omega_U = \mathcal{B}_i \cup \left(\bigcup_{k=1}^{l}\mathcal{S}_{k-1}\right)$;
16 $\quad\quad$ **else**
17 $\quad\quad\quad k^\star(i) = \text{argmin}_{k=1,2,\ldots,B}|\mathcal{S}_k - \mathcal{B}_i|$;
18 $\quad\quad\quad \Omega_U =$ $\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_{k^\star(i)-1} \cup \mathcal{B}_i \cup \mathcal{S}_{k^\star(i)+1} \cup \ldots \mathcal{S}_B$;
19 $\quad\quad$ **end**
20 $\quad\quad \mathbf{r}_i = \left(\mathbf{I} - \boldsymbol{\Psi}[\Omega_U]\boldsymbol{\Psi}^\dagger[\Omega_U]\right)\mathbf{y}_{\text{cs}}$;
21 $\quad$ **end**
22 $\quad i^\star = \text{argmin}_{i=1,2,\ldots,U}\|\mathbf{r}_i\|_2$;
23 $\quad$ **if** $l \leq B$ **then**
24 $\quad\quad \hat{s}_l = \mathcal{A}[i^\star]$;
25 $\quad\quad \mathcal{S}_l = \mathcal{B}_{i^\star}$;
26 $\quad$ **else**
27 $\quad\quad \hat{s}_{k^\star(i^\star)} = \mathcal{A}[i^\star]$;
28 $\quad\quad \mathcal{S}_{k^\star(i^\star)} = \mathcal{B}_{i^\star}$;
29 $\quad$ **end**
30 $\quad \mathbf{r}_{\text{temp}} = \mathbf{r}_{i^\star}$;
31 **until** $\left(\|\mathbf{r}_{\text{temp}}\|_2^2 \leq \mu_{\text{th}}\ \textit{or}\ l \geq l_{\max}\right)$;

The second part of the algorithm is initiated if $\|\mathbf{r}_{\text{temp}}\|_2^2 > \mu_{\text{th}}$. The idea behind this second part is that, if the residual is still too high, at least one of the bursts has been misplaced and channel components for the user outside the burst still remain.

These components are found by further correlation and burst positioning operations, not to create new bursts, but instead to refine the location of the existing detected bursts. First, a new correlation peak is detected in line 6, and candidate burst regions are found in the vicinity of this new peak. Each of these candidates replaces the closest existing burst in the set (lines $17-18$), and the corresponding residual is computed in line 20. The replacement that leads to the smallest residual in line 22 is the one that is chosen, and the corresponding previously detected burst is replaced with the new one in line 28. This process is repeated until the residual finally falls below the threshold $\mu_{\rm th}$, or the maximum number of iterations $l_{\max}$ is reached.

Once the starting point of the burst are estimated using the SPE algorithm (Algorithm 2), the BFLS algorithm (Algorithm 1) can be implemented directly to precisely determine the burst lengths and to estimate the channel coefficients.

B. Computational Complexity of the SPE Algorithm

The maximum correlation computation (line 6) and the residual update (line 20) are responsible for the bulk of the computational load in each iteration of SPE algorithm. The total complexity of the correlation step is $\mathcal{O}(l_{\max}MN)$ operations. The residual update has a complexity of $\mathcal{O}\left(2MUl+2MU^2l^2+U^3l^3\right)$, where $l\le l_{\max}$ indicates the l-th iteration of the algorithm. This analysis and the computational complexity of BFLS demonstrate that the overall channel estimation complexity for our method grows as $\mathcal{O}(N)$ and $\mathcal{O}(M)$, which is considerably less than the linear programing algorithms used by standard LASSO and Burst LASSO, whose complexity is of order $\mathcal{O}(N^3)$ and $\mathcal{O}(N^3U^3)$ respectively [30].

V. Numerical Results

In this section, we evaluate the performance of the proposed BFLS and SPE algorithms by considering an FDD massive MIMO system with a uniform linear array composed of $N=120$ antennas and $M=45$ reserved pilots (except for the example in Fig. 5). The pilot signals are drawn from a Gaussian distribution with zero mean and variance equal to the transmit power p at the BS side. The channel is assumed to consist of two bursts with different lengths (unless otherwise stated) whose non-zero entries are distributed as complex Gaussian elements. The upper bound parameter is fixed as $U=12$ except for cases where U is variable. Two bursts are assumed in the simulations, with actual lengths of 8 and 10 bins for cases where the bursts are of unequal lengths. When the bursts were assumed to be of the same length, both had 10 non-zero bins. The threshold parameter to halt Algorithm 1 is set as $\eta_{\rm th}=0.8$ and the parameter $\mu_{\rm th}$ in the halting criterion of Algorithm 2 assumes $\alpha=1$.

We compare our algorithms with the following baselines in terms of the normalized MSE (NMSE), defined as

$$\text{NMSE}=\frac{1}{N_c}\sum_{q=1}^{N_c}\frac{\|\mathbf{h}_q^{\rm H}-\hat{\mathbf{h}}_q^{\rm H}\|_2^2}{\|\mathbf{h}_q^{\rm H}\|_2^2} \tag{16}$$

where N_c is the number of simulations, and q refers to the q-th simulation.

1) **Block OMP:** As a conventional block recovery algorithm, block OMP estimates the channel by grouping the supports in the blocks and utilizing greedy and iterative procedures [16], [29]. The algorithm will repeat until the halt conditions are met.
2) **Standard LASSO:** The channel is recovered by solving the convex optimization problem (5). Standard LASSO estimates the channel without exploiting the burst sparsity property [24], [27].
3) **Burst LASSO:** In this approach, standard LASSO is modified by using a lifting transformation to convert the burst sparsity to the block model [13], [25]. In other words, the Burst LASSO approach solves problem (6) to recover the burst-sparse channel.
4) **Aided Burst LASSO:** This corresponds to Burst LASSO assuming that the starting location of each burst is known [13].
5) **Genie Aided LS:** This baseline approach assumes that the location of the channel support is available at the BS side and only requires that the amplitudes be reconstructed using the conventional LS estimator.

Moreover, to analyze the computational complexity of the BFLS algorithms when the number of antennas is increased and to compare it with the baseline approaches, we use the run-time criterion, which is defined as the CPU time for the calculations.

A. Channel Estimation Quality in Terms of NMSE

The Normalized MSE of the estimated channel versus transmit SNR is plotted in Fig. 3 for the various algorithms under consideration. As can be seen, the BFLS algorithm with known burst starting locations outperforms Aided Burst LASSO with exact support information. In addition, the NMSE of the proposed approach is essentially identical to that of the Genie Aided LS lower bound as the SNR increases. As a result, we can expect a high-quality channel estimate from the proposed algorithm if the stating locations of each burst are available. When the starting locations of the bursts are not available at the user side, the SPE algorithm is implemented to find them, and then used in the BFLS algorithm for channel acquisition. The proposed BFLS algorithm shows a significantly improved performance compared with Standard LASSO and its performance is still very close to the Genie Aided lower bound and Aided Burst LASSO with known starting points. The Burst LASSO approach is unable to estimate the channel better than Standard LASSO for the case of different burst lengths. This is because the burst sparse channel cannot be converted to a block sparse model by the lifting transformation in [25]. The Block OMP algorithm also shows poor performance due to block-wise channel recovery.

Fig. 4 illustrates the NMSE versus transmit SNR for the case with equal burst lengths. Similar to the previous case with different lengths, the proposed BFLS algorithm outperforms the other algorithms and its performance approaches the lower bound. From Fig. 3 and Fig. 4, we observe that the Burst

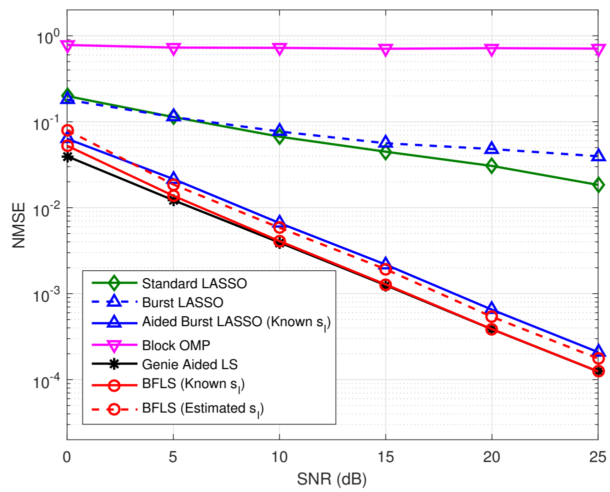

Fig. 3. Normalized MSE versus transmit SNR given $N = 120$, $M = 45$, $B = 2$ and $U = 12$, when the length of the bursts are different.

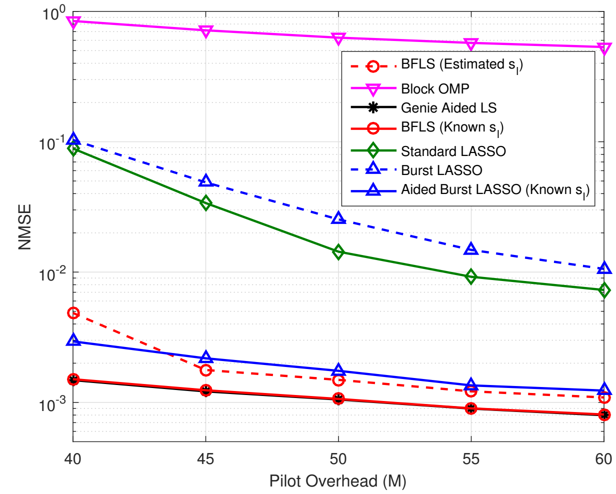

Fig. 5. Normalized MSE versus pilot overhead for transmit SNR = 15dB, $N = 120$, $U = 12$, and two bursts with different lengths.

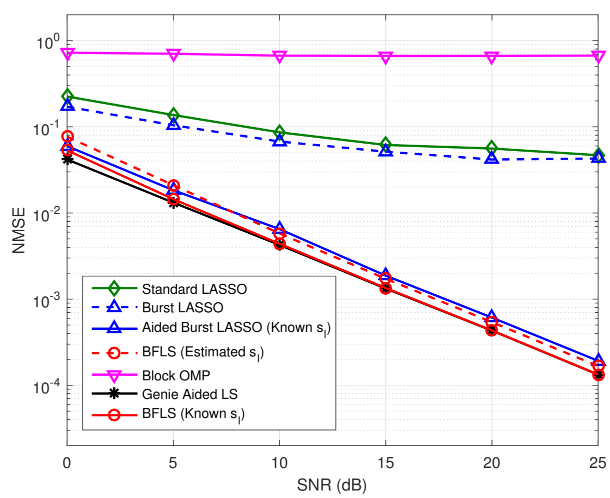

Fig. 4. Normalized MSE in terms of transmit SNR with $N = 120$, $M = 45$, $B = 2$, and $U = 12$, for bursts with the same length.

LASSO approach enhances the estimation quality only when the starting locations are known and the bursts have the same length, while the BFLS algorithm estimates the channel with acceptable quality in both cases whether or not the starting locations of the bursts are known.

Fig. 5 compares the algorithms in terms of pilot overhead when the transmit SNR is 15dB and $U = 12$. The results demonstrate again that the performance of the BFLS algorithm is equal to the lower bound when the starting point is given. We see that the pilot overhead can be reduced by approximately 70% without any degradation in performance, when the exact support information is available at the user side. For unknown support information, the BFLS requires at least $M = 45$ pilots to recover the channel, which corresponds to a 65% pilot and CSI compression (only 5% more pilots are required leading to only a slight decrease in performance). Furthermore, the BFLS algorithm has performance similar to Aided Burst LASSO for $M \geq 45$, using only estimated burst starting points. The Burst LASSO and conventional LASSO algorithms require pilots whose length is up to 50% of the number of antennas for acceptable estimation quality, which leads to a large pilot overhead.

B. Performance of the SPE Algorithm

To illustrate the accuracy of the proposed starting point estimation algorithm, the normalized MSE of the estimated points, defined as $\frac{1}{B}\sum_{l=1}^{B}\mathbb{E}\left\{(s_l-\hat{s}_l)^2/s_l^2\right\}$, in terms of the sparsity upper bound U is plotted in Fig 6. This figure shows how the value of U can affect the performance of the SPE algorithm in different SNR regimes. Because the burst area has a longer length when the upper bound U grows, the residual vectors have almost the same norms and they cannot be easily distinguished. Consequently, as the upper bound U grows away from the actual length of the bursts, the error in the estimated starting location grows correspondingly. Therefore, the only key factor in the performance of the proposed algorithm is the value of the parameter U.

Fig. 7 depicts the effect of the upper bound U on the channel estimation quality for two bursts of 10 non-zero coefficients, i.e. $d_1 = d_2 = 10$. We see from this figure that the BFLS algorithm with unknown starting locations provides performance equal to the lower bound when U is chosen to correspond to the exact sparsity order in each burst. A slight degradation is observed when the difference between U and the length of the bursts grows.

C. Computational Complexity in Terms of Run Time

In Fig. 8, we compare the computational complexity of the algorithms according to their run time for different numbers of transmit antennas. The number of required pilots is taken to be $\frac{2}{5}N$ (60% compression) for each of value of N. From this figure, we see that the LASSO approaches (Burst LASSO, Standard LASSO, and Aided Burst LASSO) require the most

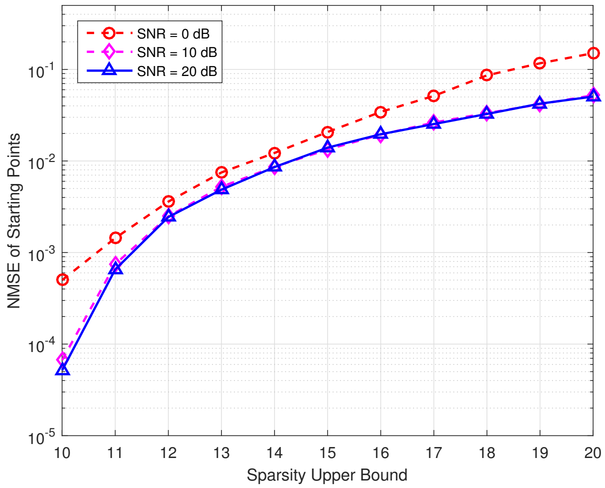

Fig. 6. Normalized MSE of starting points versus U given $N = 120$, $M = 45$, and $B = 2$ for several transmit SNRs.

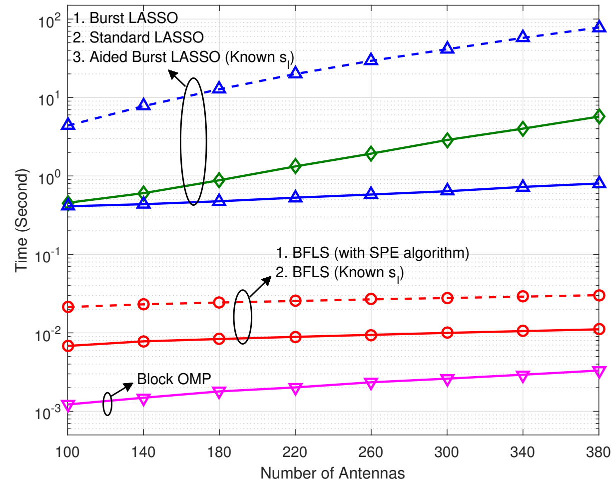

Fig. 8. CPU running time versus number of transmit antennas when $U = 12$, transmit SNR $= 15$dB, and under the configuration: Intel CPU (Xeon) with 2.4 GHz frequency, 4GB RAM.

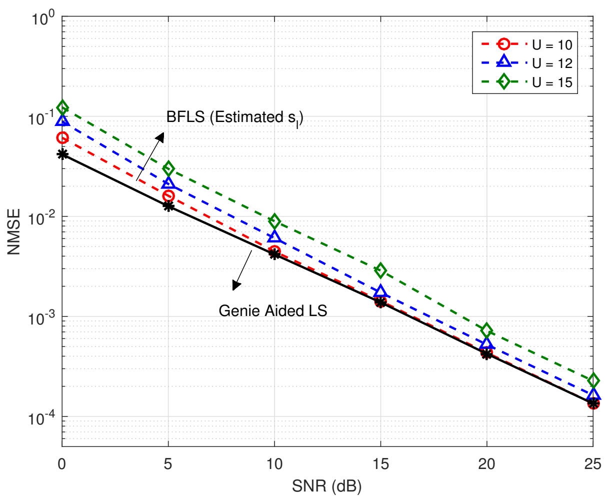

Fig. 7. Normalized MSE of channel estimation in term of transmit SNR for different sparsity upper bounds for bursts with the same lengths.

run time, since they must solve a large-dimensional optimization problem to recover the channel, especially for very large antennas arrays. Due to the large number of variables in the objective function of the Burst LASSO optimization, more computation is needed to estimate the massive MIMO channel. Unlike the LASSO approaches, the proposed BFLS algorithm first reduces the dimension of the problem by determining the locations of the channel elements, which leads to a significant reduction in complexity. For the case with unknown starting locations, the run time increases due to the computation of the SPE algorithm, which tends to dominate the overall computational cost.

VI. Conclusion

In this paper, a novel burst-form LS approach which fully utilizes the burst structure of sparse channels in massive MIMO system has been introduced. In the first scenario considered, the starting location of the bursts is considered to be a variable that varies slowly enough that the user can track it easily and thus is considered to be known. Using the known starting points and an upper bound for the lengths of the bursts, the location of each burst in the channel vector is determined and then the amplitude of the supports are estimated by the LS approach. In addition, a refinement step has been embedded into the proposed algorithm to eliminate over-estimated support sets. We also considered the scenario in which the user cannot provide the starting locations to the BFLS algorithm. In this case, we proposed an algorithm for estimating the starting point locations of each burst. This second algorithm includes a refinement step to fine-tune the estimated starting points in cases where errors are introduced by the first pass of the algorithm. Our simulations verify that the BFLS algorithm can provide very high quality channel estimates for FDD systems with much fewer pilots and with significantly reduced computation compared with competing algorithms, even for the case where the burst starting points must be estimated.

References

[1] T. L. Marzetta, “Massive MIMO: An introduction,” *Bell Labs Technical Journal*, vol. 20, pp. 11–22, 2015.

[2] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, “An overview of massive MIMO: Benefits and challenges,” *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, Oct 2014.

[3] 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.

[4] J. C. Shen, J. Zhang, K. C. Chen, and K. B. Letaief, “High-dimensional CSI acquisition in massive MIMO: Sparsity-inspired approaches,” *IEEE Syst. J.*, vol. 11, no. 1, pp. 32–40, March 2017.

[5] L. You, X. Gao, A. L. Swindlehurst, and W. Zhong, “Channel acquisition for massive MIMO-OFDM with adjustable phase shift pilots,” *IEEE Trans. Signal Process.*, vol. 64, no. 6, pp. 1461–1476, March 2016.

[6] P. W. C. Chan, E. S. Lo, R. R. Wang, E. K. S. Au, V. K. N. Lau, R. S. Cheng, W. H. Mow, R. D. Murch, and K. B. Letaief, “The evolution path of 4G networks: FDD or TDD?” *IEEE Commun. Mag.*, vol. 44, no. 12, pp. 42–50, Dec 2006.

[7] E. Bjrnson, E. G. Larsson, and T. L. Marzetta, "Massive MIMO: ten myths and one critical question," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 114–123, February 2016.

[8] B. Lee, J. Choi, J. Y. Seol, D. J. Love, and B. Shim, "Antenna grouping based feedback compression for FDD-based massive MIMO systems," *IEEE Trans. Commun.*, vol. 63, no. 9, pp. 3261–3274, Sept 2015.

[9] J. Choi, D. J. Love, and P. Bidigare, "Downlink training techniques for FDD massive MIMO systems: Open-loop and closed-loop training with memory," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 802–814, Oct 2014.

[10] H. W. Liang, W. H. Chung, and S. Y. Kuo, "FDD-RT: A simple CSI acquisition technique via channel reciprocity for FDD massive MIMO downlink," *IEEE Syst. J.*, vol. 12, no. 1, pp. 714–724, March 2018.

[11] X. Rao and V. K. N. Lau, "Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3261–3271, June 2014.

[12] Z. Gao, L. Dai, W. Dai, B. Shim, and Z. Wang, "Structured compressive sensing-based spatio-temporal joint channel estimation for FDD massive MIMO," *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 601–617, Feb 2016.

[13] A. Liu, V. K. N. Lau, and W. Dai, "Exploiting burst-sparsity in massive MIMO with partial channel support information," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7820–7830, Nov 2016.

[14] Y. Han, J. Lee, and D. J. Love, "Compressed sensing-aided downlink channel training for FDD massive MIMO systems," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 2852–2862, July 2017.

[15] Z. Gao, L. Dai, Z. Wang, and S. Chen, "Spatially common sparsity based adaptive channel estimation and feedback for FDD massive MIMO," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6169–6183, Dec 2015.

[16] C. R. Berger, Z. Wang, J. Huang, and S. Zhou, "Application of compressive sensing to sparse channel estimation," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 164–174, November 2010.

[17] Y. Barbotin, A. Hormati, S. Rangan, and M. Vetterli, "Estimation of sparse MIMO channels with common support," *IEEE Trans. Commun.*, vol. 60, no. 12, pp. 3705–3716, December 2012.

[18] H. Yin, D. Gesbert, M. Filippou, and Y. Liu, "A coordinated approach to channel estimation in large-scale multiple-antenna systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 264–273, February 2013.

[19] M. Masood, L. H. Afify, and T. Y. Al-Naffouri, "Efficient coordinated recovery of sparse channels in massive MIMO," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 104–118, Jan 2015.

[20] C. Qi, G. Yue, L. Wu, Y. Huang, and A. Nallanathan, "Pilot design schemes for sparse channel estimation in OFDM systems," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1493–1505, April 2015.

[21] R. Mohammadian, A. Amini, and B. H. Khalaj, "Deterministic pilot design for sparse channel estimation in MISO/multi-user OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 129–140, Jan 2017.

[22] J. C. Shen, J. Zhang, E. Alsusa, and K. B. Letaief, "Compressed CSI acquisition in FDD massive MIMO: How much training is needed?" *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 4145–4156, June 2016.

[23] C. C. Tseng, J. Y. Wu, and T. S. Lee, "Enhanced compressive downlink CSI recovery for FDD massive MIMO systems using weighted block ℓ_1-minimization," *IEEE Trans. Commun.*, vol. 64, no. 3, pp. 1055–1067, March 2016.

[24] L. Lian, A. Liu, and V. K. N. Lau, "Weighted LASSO for sparse recovery with statistical prior support information," *IEEE Trans. Signal Process.*, vol. 66, no. 6, pp. 1607–1618, March 2018.

[25] A. Liu, V. Lau, and W. Dai, "Joint burst LASSO for sparse channel estimation in multi-user massive MIMO," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[26] A. M. Sayeed, "Deconstructing multiantenna fading channels," *IEEE Trans. Signal Process.*, vol. 50, no. 10, pp. 2563–2579, Oct 2002.

[27] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[28] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[29] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3042–3054, June 2010.

[30] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*. Siam, 1994, vol. 13.