This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3132583, IEEE Open Journal of Signal Processing

Multi Stage Kalman Filter (MSKF) Based Time-Varying Sparse Channel Estimation with Fast Convergence

Parthapratim De, Markku Juntti, *IEEE Fellow*, and Christo K. Thomas
Institute for Infocomm Research, Singapore, Univ of Oulu, Finland, and Qualcomm, Espoo, Finland,
pd4267@yahoo.com,markku.juntti@oulu.fi, kurisumm@eurecom.fr,

***Abstract*— The paper develops novel algorithms for time-varying (TV) sparse channel estimation in Massive multiple-input, multiple-output (MMIMO) systems. This is achieved by employing a novel reduced (non-uniformly spaced tap) delay-line equalizer, which can be related to low/reduced rank filters. This low rank filter is implemented by deriving an innovative TV (Krylov-space based) Multi-Stage Kalman Filter (MSKF), employing appropriate state estimation techniques. MSKF converges very quickly, within few stages/iterations (at each symbol). This is possible because MSKF uses those signal spaces, maximally correlated with the desired signal, rather than the standard principal component (PCA) signal spaces. MSKF is also able to reduce channel tracking errors, encountered by a standard Kalman filter in a high-mobility channel. In addition, MSKF is well suited for large-scale MMIMO systems. This is unlike most existing methods, including recent Bayesian-Belief Propagation, Krylov, fast iterative re-weighted compressed sensing (RCS) and minimum rank minimization methods, which requires more and more iterations to converge, as the scale of MMIMO system increases. A Bayesian Cramer Rao lower bound (BCRLB) for noisy CS (in sparse channel) is also derived, which provides a benchamrk for the performance for novel MSKF and other CS estimators.**

I. Introduction

Massive MIMO (MMIMO) systems are considered for high data rate communications in sparse channels, e. g. digital television (DTV) [1]- [2], echo cancellation, underwater [3], millimeter-wave (mmwave) 5G communications [4]. For example, in terrestrial DTV transmission [2], [1], a typical receiver is expected to handle multipath with delays as long as 18 microseconds, which at high symbol rates, requires adaptive finite impulse response (FIR) linear equalizers with several hundred symbol-spaced taps [5]. In order to alleviate dynamic multipaths, due to propagation effects, flutter from moving objects, e.g., airplanes and changing atmospheric conditions, the equalizer must update its coefficients at high speed. This situation is also witnessed in a high data rate wireless channel, where only the main signal and a few multipath reflected signals are significant, among (maybe) hundreds of channel taps, (in a tapped-delay line model). Advanced sparse channel estimation methods, requiring estimation of only few significant channel tap weights, have been developed for orthogonal frequency division multiplexing (OFDM) [6] and code division multiple access (CDMA) systems, and provide superior performance.

Existing works estimate the significant channel tap locations first [7], after which least-squares (LS) methods (employing training subcarriers only) are used to estimate the significant channel tap weights, but may require a large amount of training data, making them unsuitable for MMIMO. The large number of training symbols/subcarriers required or the resultant pilot contamination/re-use problem in MMIMO [8] necessitated the development of blind/semi-blind sparse time-invariant (ITV) channel/data estimation methods, [9] - [12]. Though [11], [12] perform much better than [9], [10], all of these blind algorithms are data block-based methods (i. e., require a block of received data symbols to be collected, before filtering), and is not adapted for symbol-by-symbol update, for rapidly time-varying (TV) channels. This motivates the development of time (symbol) iterative/update Kalman-like filters. The novel methods here utilize the ideas of reduced-rank, sparse, and multistage Kalman filters (MSKF) jointly to exploit the sparsity in different dimensions (time, space etc).

Recent methods, like the popular sparsity based compressed sensing (CS) and Bayesian methods [13] - [21], yield superior performance in MMIMO mmwave spatially sparse channels, by exploiting the low rank angular structure induced by the multi-ray channel model with narrow angular spread (AS). But these methods have been derived for *single path* (not multipath) channels, and thus do not utilize the temporal sparsity (in multipath lag) domain. Moreover, CS and Bayesian methods are computationally very demanding, and their few iterative versions, like proximal re-weighted CS (RCS) [16], [18] or [20], converge after many symbols, making them unsuitable for use in high mobility TV channels. Moreover, one of the few existing Krylov space spaced TV channel estimators [22] models the time variation by a basis exponential method (BEM), which inevitably introduces approximation error to channel estimates, due to the imperfect model assumed. Here, on the other hand, the novel equalizer is updated by *reduced-rank, TV* novel MSKF and Krylov-Kalman filters, with reduced computational complexities. In particular, the novel multi-stage MSKF performs very well, employs some data censoring and converges quickly, within a few stages/iterations, at each time symbol. Additionally, the novel MSKF is able to *reduce channel tracking errors* of a standard Kalman filter, which occurs in a high-mobility, TV channel (as seen in Fig. 8 [23] and text below it).

A. Contributions

The *main* contributions of this paper are

1) A novel, symbol-iterative MSKF, exhibiting superior performance and rapid stage-wise convergence (at each symbol), is developed for high mobility sparse TV channels. This is achieved by
   a) having a novel TV *reduced (non-uniformly spaced)* equalizer structure, reminiscent of some time-invariant (ITV) reduced equalizers for DTV and echo cancellation. ITV Reduced equalizers have been seen to outperform uniformly symbol-spaced estimators, as evidenced in [1]) (Fig. 3), and in learning

curves (Figure 4, [1] and Figure 3, [2]). Using autoregressive (AR) TV channel models, the reduced equalizer is generalized to exploit any available sparsity in cluster-sparse channels, even with continuous ISI, and with time-varying significant channel tap locations, to cater to real-world channels, encountered in in 4G/5G transmission (Sections II and III).

b) Substantial synthesis and analysis of the reduced equalizer leads to reduced channel vectors and matrices for data estimation in high mobility sparse TV channels. Then the very large number of receive antennas J in MMIMO systems allows the derivation of a *low rank algebraic* reduced equalizer structure (Section III).

c) Next, motivated by some ITV Multi-stage Wiener Filter (MSWF), exhibiting fast stage-wise convergence for some DOA applications [24], a novel TV Multi-stage Kalman Filter (MSKF) is integrated into the reduced rank equalizer structure above (Sections III and IV). Significant derivation of this novel dynamic, sparse MSKF equalizer, with data censoring (as in sensor networks [25]) is developed in Sections III and IV.

d) Derivation of Bayesian Cramer Rao bounds (BCRB) for noisy CS in sparse channel, and its comparison with novel MSKF filter (Section VI).

e) Performance analysis of TV MSKF's much improved (order/iteration)-wise convergence of normalized mean squared error (NRMSE) and analytical comparison and connection with Bayesian, CS and other existing sparse methods, (Section V).

Other contributions include

- 1. Close connection between the ideas of compressed sensing(CS) and reduced rank filters (matrix rank minimization) in sparse estimation (Sec III. B); development of a second novel TV Krylov filter (Sec V).
- 2. Extensive comparative simulations of MSKF with recent Bayesian, CS and Krylov space based sparse estimators, in high mobility MMIMO systems (Section VII). Conclusions are provided in Section VIII.

Notations: Bold upper-case symbols $\mathbf{A}$ denote matrices. Bold lower-case symbols $\mathbf{b}$ denote vectors. $\mathbf{I}_i$ is an identity matrix of size $i \times i$, $\mathbf{0}_{j,k}$ is a $j \times k$-sized zero matrix. Also, $\mathbf{A}(i : j, k : l)$ denotes the ith to jth rows and kth to lth columns of the matrix $\mathbf{A}$.

II. System Model

A single-carrier sparse channel transmission system, with maximum multipath delay spread of up to L symbols, is considered. A novel algorithm is designed to determine the finite impulse response (FIR) equalizer, required to invert this channel in the minimum mean squared error (MMSE) sense [5]. Consider first a SIMO system, with a single transmit antenna and J receive antennas. The TV lth tap channel weight, at symbol n, is $\mathbf{h}(n,l) = [h_1(n,l), h_2(n,l), \ldots, h_J(n,l)]^T$, $l = 0, 1, \cdots, L-1$, ($h_j(n,l)$ is the lth lag channel weight from transmitter to the jth receive antenna at nth symbol). However in a sparse channel, only D, (out of a total of L), channel tap weights, have non-zero values. In many cases, $D << L$.

In next section, we consider generic *cluster-sparse* channels with continuous ISI, (e. g. 3G LTE channel). The $J \times 1$ received signal (on J received antennas) is

$$\begin{aligned}\mathbf{y}(n) &= \sum_{m=0}^{L-1} \mathbf{h}(n,m)s(n-m) + \mathbf{w}(n)\\ &= \sum_{k=0}^{D-1} \mathbf{h}(n,l_k)s(n-l_k) + \mathbf{w}(n);\\ &0 \leq l_k \leq L-1,\ k = 0,1,2,\cdots,D-1, \qquad (1)\end{aligned}$$

where l_k's denotes the kth non-zero weighted channel tap locations. Generally, $l_0 = 0$, [9], [10]. Also, assume that $0 = l_0 < l_1 < l_2 < \cdots < l_{D-1} \leq L-1$; $\mathbf{w}(n)$ is the $J \times 1$ additive white gaussian noise (AWGN). For MIMO systems with $\bar{S}$ transmit antennas and J receive antennas, the channel matrix $\tilde{\mathbf{H}}(n,l)$ is a $J \times \bar{S}$ matrix, given by $\tilde{\mathbf{H}}(n,l) = [\mathbf{h}^{(1)}(n,l)\, \mathbf{h}^{(2)}(n,l) \cdots, \mathbf{h}^{(\bar{S})}(n,l)]$, with $J \times 1$-sized $\mathbf{h}^{(k)}(n,l)$ being the channel from the kth transmit antenna to the J receive antennas, at lth delay and nth symbol.

Three general assumptions are made as follows:

($\mathcal{A}1$) The symbol sequence of each user $s(n)$ is temporally white with zero mean and unit variance, and is statistically uncorrelated with $s(n-m)$ for $m \neq 0$.

($\mathcal{A}2$) The noise sequences $w_j(n)$ are stationary, and temporally and spatially white with zero mean and variance σ_w^2.

($\mathcal{A}3$) The symbol sequences $s(n)$ are statistically uncorrelated with the noise sequences $w_j(n)$.

Remark: In point-to-point MIMO systems, the transmit and receive antennas are co-located. In such a case, the propagation delay is approximately the same for all transmit-receive pairs; thus, significant channel tap locations, should be the same for all transmit-receive pairs [10], i.e., common sparsity support across all receive antennas. However, this assumption may not hold over a large number of receive antennas in MMIMO [15]. Ma *et. al.* proposes a spatial domain BEM (SBEM), with beamforming so that each ray directed to one user cluster. However, [18] derives algorithms for time delay and angle estimation in MMIMO, with the same multipath delays for all receive antennas.

III. Novel Time Varying (TV) Reduced Equalizer for Sparse Channels

A. Extended Channel Model

1) Group or Clustered Sparsity: Here, group sparsity [26] is considered, where the few non-zero (i.e., significant) channel taps occur in clusters/blocks in a structured manner, see Fig. 1 a). Suppose the multipath channel consists of D clusters (instead of D single taps). The total support of the channel S is given by $S = \bigcup_{k=0}^{D-1} S(k)$, $S(k)$ being the support of the kth cluster. The kth cluster consists of $|S_k|$ consecutive multipaths at lags of $l_k, (l_k+1), \cdots, (l_k+|S_k|-1)$; let $\bar{S} = \max_{k=0,1,\cdots,D-1} |S_k|$. Then (1) can be rewritten as

$$\begin{aligned}\mathbf{y}(n) = \sum_{i=0}^{\bar{S}-1} \sum_{k=0}^{D-1} \mathbf{h}(n, l_k+i)s(n-l_k-i) + \mathbf{w}(n),\\ 0 \leq l_k \leq L. \qquad (2)\end{aligned}$$

For a fixed i, the inner summation in (2) is again a D-sparse channel, with sparse multipaths, still separated by $\{(l_{k+1} + i) - (l_k + i) = (l_{k+1} - l_k)\}$ taps, just as in (1). Defining $\mathbf{H}(n, l_k) \triangleq [\mathbf{h}(n, l_k),\ \mathbf{h}(n, l_k+1),\ \cdots, \mathbf{h}(n, l_k+\bar{S}-1)]$, and $\mathbf{s}(n-l_k) \triangleq [s(n-l_k),\ s(n-l_k-1),\ \cdots, s(n-l_k-\bar{S}+1)]^T$, (1) is equivalent to

$$\mathbf{y}(n) = \sum_{k=0}^{D-1} \mathbf{H}(n, l_k)\mathbf{s}(n - l_k) + \mathbf{w}(n). \tag{3}$$

Since the different components of $\mathbf{H}(n, l_k)$ are uncorrelated with each other (and so also for $\mathbf{s}(n)$), equation (3) can be regarded as a $\bar{S}$-user sparse channel model.

Then there is the issue of continuous ISI, as in a 3G LTE Pedestrian B channel. There may be some multipaths, in between the significant multipath clusters, shown in Figure 3, [14]. Their powers may be approximately $50, 70, 170$ dB below that of the main path. Simulation results, for TV 3G LTE channel, in Sec. VII show that the effect of these paths is not much, see (last right-most paragraph, pp. 1428) [11] and Table II, [9].

2) Time-Variation of Significant Tap locations: Assume that the time-evolution of random, TV channel is modeled by a first order auto-regressive (AR(1)) model [23], [5], (with significant channel tap locations $l_k(n)$ as a function of the nth symbol),

$$\mathbf{H}(n, l_k(n)) = \lambda\mathbf{H}(n-1, l_k(n-1)) + \mathbf{V}(n), \tag{4}$$

where $\mathbf{V}(n)$ is the process noise with zero mean and variance $\sigma_v^2\mathbf{I}$. λ represents how fast and how much the time-varying part of channel taps $\mathbf{H}(n, l_k(n))$ varies with respect to the mean of $\mathbf{H}(n, l_k(n))$. Actually, $\lambda = J_0(2\pi f_D T)$, where J_0 and $f_D T$ are the zero-th order Bessel function and Doppler rate respectively; the Doppler frequency f_D corresponds to the vehicular velocity [23].

Substituting (4) into (3),

$$\begin{aligned}\mathbf{y}(n) &= \sum_{k=0}^{D-1} \mathbf{H}(n, l_k(n))\mathbf{s}(n - l_k(n)) + \mathbf{w}(n)\\ &= \sum_{k=0}^{D-1} (\lambda\mathbf{H}(n-1, l_k(n-1)) + \mathbf{V}(n))\\ &\quad \mathbf{s}(n - 1 - (l_k(n) - 1)) + \mathbf{w}(n)\\ &= \lambda \sum_{k=0}^{D-1} \mathbf{H}(n-1, l_k(n-1))\\ &\quad [\mathbf{s}(n - 1 - (l_k(n) - 1))] + \tilde{\mathbf{w}}(n) = \lambda\mathbf{y}(n-1) + \tilde{\mathbf{w}}(n)\end{aligned} \tag{5}$$

where $\tilde{\mathbf{w}}(n) \triangleq \mathbf{w}(n) + \mathbf{V}(n)\sum_{k=0}^{D-1} \mathbf{s}(n-1-(l_k(n)-1))$ is the overall noise . Let $m = n - 1 - (l_k(n) - 1)$.
Now

$$\begin{aligned}E\{\tilde{\mathbf{w}}(n)\mathbf{s}^T(m)\} &= E\{\mathbf{V}(n)\}E\{\mathbf{s}(m)\mathbf{s}^T(m)\}\\ &\quad + E\{\mathbf{w}(n)\}E\{\mathbf{s}^T(m)\} = \mathbf{0},\end{aligned}$$

as noises $E\{\mathbf{V}(n)\} = \mathbf{0}$, $E\{\mathbf{w}(n)\} = \mathbf{0}$, and both are also uncorrelated with signal $\mathbf{s}(m)$.

Then $l_k(n-1) = l_k(n) - 1$, i. e., significant channel tap locations are shifted by 1, which is already accommodated in AR(1) model (4) above. This expression for the time variation of $l_k(n)$ is obtained, within the limitations of the assumed AR(1) model, and may differ for other TV channel models. The novel channel model is illustrated in Fig. 1 a) and b).

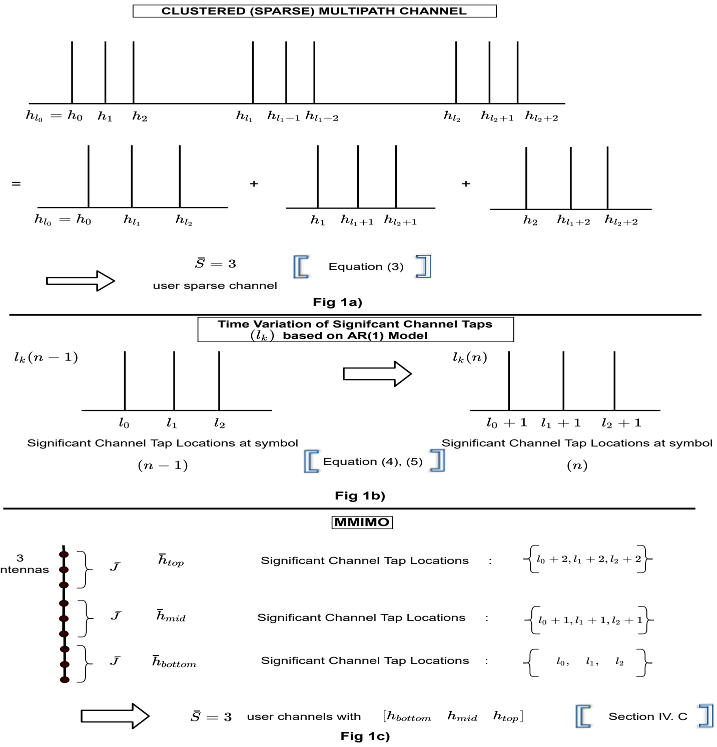

Fig. 1
A) CLUSTERED SPARSE CHANNEL MODEL AS MULTI-USER SPARSE CHANNEL (EQUATION (3)), B) TIME-VARYING SIGNIFICANT CHANNEL TAP LOCATIONS (EQUATIONS (4) AND (5)), C) MMIMO CHANNEL MODEL (SEC IV. C).

B. Reduced TV Equalizer

Motivation

There exists a close connection between the ideas of compressed sensing (CS) and matrix rank minimization/reduced rank filters, in beam formed sparse (angular and temporal) channel estimation. [21], [20], [18]. [21] investigates single path mmwave channel sparsity in angular/space (Direction of Arrival (DOA)) domain, where the low-rank algebraic structure of the channel matrix is exploited by employing a reduced rank method, followed by a CS sparse method. [20] solves the same problem (in MMIMO) using CS methods only. On the other hand, [18] compares the performance of individual CS and reduced rank methods in MMIMO mmwave multipath channels. In this paper, a novel reduced rank filtering method is used for sparse multipath (non-beamformed) channel estimation. This is facilitated by having very large J in MMIMO, which makes Assumption ($\mathcal{A}4$) (below) more likely to be satisfied.

A novel model of a reduced equalizer, adequate for data estimation in a sparse channel, is introduced in this section. A sparse equalizer means that only few of its taps (in a tapped-delay line model of linear FIR equalizer) have significant weights. These significant, *non-uniformly spaced*, FIR equalizer tap locations are seen to be related to the auto-correlation matrix of $\mathbf{y}(n)$ [11]. Define the ith lag auto-correlation matrix, at symbol n, $\mathbf{R}(n,i)\underline{\triangleq}E[\mathbf{y}(n)\mathbf{y}^H(n-i)]$. After evaluating $\mathbf{R}(n,i)$ for the sparse channel in (1), it can be seen that the TV $\mathbf{R}(n,i) \neq \mathbf{0}$, only for the lags i's, [11],

$$\begin{aligned} &i = l_0 - l_0 = 0; i = l_k - l_0 = l_k,\ (k = 1, 2, \cdots, D-1), \\ &i = l_k - l_m,\ (m < k,\ k = 2, 3, \cdots, D-1). \end{aligned} \tag{6}$$

Then, an algorithm for selecting non-uniformly spaced equalizer tap delays $\{m_p\}_{p=1}^N$'s, is enumerated in Table I, by employing *Assumptions* $(\mathcal{A}1) - (\mathcal{A}3)$, and by extending [11].

Now the noiseless reduced data vector $\mathbf{y}_{red}(n) = \mathbf{y}_{red}(n) = [\mathbf{y}^T(n-m_1)\mathbf{y}^T(n-m_2)\cdots\mathbf{y}^T(n-m_N)]^T$, $m_N \leq M$. (defined in (56), Table I) can be written as $\mathbf{y}_{red}(n) = \mathbf{H}_{red}(n)\mathbf{s}_{red}(n)$ ($\mathbf{H}_{red}(n)$ and $\mathbf{s}_{red}(n)$ are the corresponding reduced channel matrix and transmitted data vectors respectively). Obtaining general expressions of the novel reduced channel matrix, (for generic sparse channels, which have many different combinations of $\{l_j\}_{j=0}^{D-1}$'s, even for the same sparsity level D), may not always be possible [11], [12]. The reduced channel matrix and transmitted data vector can *only* be illustrated fully for specific channels, e. g., Example I Channel (equations (14), (47), (48) and (49) in [11]). Extending it to the reduced, TV Example I channel matrix $\mathbf{H}_{red}(n)$ here is given by (for $M = 23$) (equation (7), on top of next page) of size $12J \times 21\bar{S}$. If we had taken equalizer taps at all lags, the "full" channel matrix $\mathbf{H}_{full}(n)$ will be of size $MJ \times (M+L) = 23J \times (23+12)\bar{S} = 23J \times 35\bar{S}$, Example I Channel, [11]. An assumption made is

$(\mathcal{A}4)$ Channel matrix $\mathbf{H}_{full}(n)/\mathbf{H}_{red}(n)$ is of full column rank.

In MMIMO, with very large number of receive antennas J, the channel matrix is a very "tall" matrix, which makes Assumption $(\mathcal{A}4)$ more likely to be valid. Then $\text{rank}(\mathbf{H}_{full}(n)) = 35\bar{S}$ always, irrespective of the sparsity structure in channel. But $\text{rank}(\mathbf{H}_{red}(n)) = 21\bar{S}$ for Example I Channel and has reduced rank, which depends on each specific sparse channel. This is unlike the full channel matrix used traditionally in data estimation; thereby indicating that the reduced equalizer methodology has transformed sparse channel estimation problem into that of TV reduced-rank filtering.

C. State Space Representation and Innovations in Reduced Filter

Here, the state is the channel $\mathbf{H}(n, l_k(n))$, for which the measurement equation is (using (3)),

$$\mathbf{y}(n) = \mathbf{C}(n)\tilde{\mathbf{H}}(n) + \mathbf{w}(n), \tag{8}$$

with measurement matrix $\mathbf{C}(n) = [\mathbf{I}_J \otimes \mathbf{s}^T(n-l_0), \mathbf{I}_J \otimes \mathbf{s}^T(n-l_1), \cdots, \mathbf{I}_J \otimes \mathbf{s}^T(n-l_{D-1})]$ ($\otimes$: Kronecker product). Defining $J\bar{S} \times 1$-sized $\bar{\mathbf{h}}(n, l_k) = [\mathbf{h}^T(n, l_k), \mathbf{h}^T(n, l_k+1), \cdots, \mathbf{h}^T(n, l_k+\bar{S}-1))]^T$, $\tilde{\mathbf{H}}(n)\underline{\triangleq}[\bar{\mathbf{h}}^T(n, l_0)\cdots\bar{\mathbf{h}}^T(n, l_{D-1})]^T$. However, since we don't know the significant channel $\{l_k\}$'s *a priori*, one starts with assuming that all channel taps are present, in novel MSKF algorithm and its simulations. The only information we have is the non-uniformly-spaced reduced equalizer lags, i. e. the $\mathbf{y}_{red}(n)$ vector (which is deduced from Algorithm I). The TV channel's dynamic state equation

$$\tilde{\mathbf{H}}(n) = \text{diag}(\lambda)\tilde{\mathbf{H}}(n-1) + \mathbf{V}(n) \tag{9}$$

follows from (4). Since the reduced equalizer is $\mathbf{y}_{red}(n)$=$[\mathbf{y}^T(n-m_1)\,\mathbf{y}^T(n-m_2)\cdots\mathbf{y}^T(n-m_N)]^T$, $\tilde{\mathbf{H}}(n)$ needs to be updated only at symbols $\{(n-m_N), \cdots, (n-m_2), (n-m_1), \cdots, n\}$.

Note This may be viewed as some form of data censoring in sensor networks [25].

Now, $\mathbf{y}(n)$, in (8) is used in block-based channel estimation method [11], i. e., a block of received data symbols is collected before $\tilde{\mathbf{H}}(n)$ is estimated, i. e., the estimate is not iteratively updated from one symbol to the next, which is the objective of this paper. For our novel, reduced Kalman filter, the innovations (used to derive an time-iterative algorithm) is $\tilde{\mathbf{y}}(n) = \tilde{\mathbf{y}}(n-m_0)[m_0 = 0]\underline{\triangleq}\mathbf{y}(n) - \hat{\mathbf{y}}(n|n-m_1)$. $\hat{\mathbf{y}}(n|n-m_1)$ is the MMSE estimate of $\mathbf{y}(n)$, based on non-uniformly spaced past data $\{\mathbf{y}(n-m_j)\}_{j=1}^N$'s. Similarly, channel estimate $\hat{\tilde{\mathbf{H}}}(n|n)$ uses the current data $\tilde{\mathbf{y}}(n)$; the a-priori channel estimate $\hat{\tilde{\mathbf{H}}}(n|n-m_1)$ uses past data $\{\mathbf{y}(n-m_j)\}_{j=1}^N$'s; a priori estimate error is denoted by $\mathbf{H}^e(n|n-m_1)\underline{\triangleq}\tilde{\mathbf{H}}(n) - \hat{\tilde{\mathbf{H}}}(n|n-m_1)$.

Next, the innovations has to be expressed in terms of Kalman state matrices (in (8) and (9)), [unlike time-invariant Wiener MSWF [24], [11]]. From (8), $\tilde{\mathbf{y}}(n)$ and its auto-correlation matrix are

$$\begin{aligned} &\hat{\mathbf{y}}(n|n-m_1) = \mathbf{C}(n)\hat{\tilde{\mathbf{H}}}(n|n-m_1), \\ &\tilde{\mathbf{y}}(n) = \mathbf{y}(n) - \hat{\mathbf{y}}(n|n-m_1) = \mathbf{C}(n)\mathbf{H}^e(n|n-m_1) + \mathbf{w}(n), \\ &\mathbf{R}_{\tilde{\mathbf{y}}(n)} = \mathbf{C}(n)\mathbf{R}_{\mathbf{H}^e}(n|n-m_1)\mathbf{C}^H(n) + \sigma_w^2\mathbf{I}_J, \end{aligned} \tag{10}$$

where $\mathbf{R}_{\mathbf{H}^e}(n|n-m_1)$ is the a priori channel error correlation matrix. Next, a time-update of the state, $\tilde{\mathbf{H}}(n)$, is obtained,

Lemma 1:

$$\hat{\tilde{\mathbf{H}}}(n|n) = \hat{\tilde{\mathbf{H}}}(n|n-m_1) + \hat{\tilde{\mathbf{H}}}(n|\tilde{\mathbf{y}}(n)). \tag{11}$$

Proof: See Appendix A.

From (11), we have,

$$(\tilde{\mathbf{H}}(n) - \hat{\tilde{\mathbf{H}}}(n|n)) = (\tilde{\mathbf{H}}(n) - \hat{\tilde{\mathbf{H}}}(n|n-m_1) - \hat{\tilde{\mathbf{H}}}(n|\tilde{\mathbf{y}}(n)) \tag{12}$$

$$\mathbf{H}^e(n|n) = \mathbf{H}^e(n|n-m_1) - \hat{\tilde{\mathbf{H}}}(n|\tilde{\mathbf{y}}(n)) \tag{13}$$

Using (13), the a-posteriori channel estimate error auto-correlation matrix $\mathbf{R}_{\mathbf{H}^e}(n|n) = E\{\mathbf{H}^e(n|n)\mathbf{H}^{eH}(n|n))\} = \mathbf{R}_{\mathbf{H}^e}(n|n-m_1) - 2E\{\mathbf{H}^e(n|n-m_1)\hat{\tilde{\mathbf{H}}}^H(n|\tilde{\mathbf{y}}(n))\} + E\{\hat{\tilde{\mathbf{H}}}(n|\tilde{\mathbf{y}}(n))\hat{\tilde{\mathbf{H}}}^H(n|\tilde{\mathbf{y}}(n))\}$. Using orthogonality principle of MMSE estimation, the a-priori channel estimate error $\mathbf{H}^e(n|n-m_1)$ is orthogonal to the a-priori estimate $\hat{\tilde{\mathbf{H}}}(n|n-m_1)$; also, $\hat{\tilde{\mathbf{H}}}(n|\tilde{\mathbf{y}}(n))$ is orthogonal to $\hat{\tilde{\mathbf{H}}}(n|n-m_1)$ [27]. Then

$$\begin{aligned} \mathbf{R}_{\mathbf{H}^e}(n|n) = \mathbf{R}_{\mathbf{H}^e}&(n|n-m_1) \\ &- E\{\hat{\tilde{\mathbf{H}}}(n|\tilde{\mathbf{y}}(n))\hat{\tilde{\mathbf{H}}}^H(n|\tilde{\mathbf{y}}(n))\}. \end{aligned} \tag{14}$$

Since number of receive antennas at MMIMO base-station, J is very large, the state $\mathbf{H}^e(n|n)$, of size $(L+1)J$, has a very large dimension. Thus direct application of the

$$\mathbf{H}_{red}(n)=\begin{bmatrix}\mathbf{H}_0(n,0) & \mathbf{0}_{J,1} & \mathbf{H}(n,4) & \mathbf{0}_{J,3} & \mathbf{H}(n,11) & \mathbf{0}_{J,14} & \\ \mathbf{0}_{J,1} & \mathbf{H}(n,0) & \mathbf{0}_{J,1} & \mathbf{H}(n,4) & \mathbf{0}_{J,4} & \mathbf{H}(n,11) & \mathbf{0}_{J,12} \\ \mathbf{0}_{J,2} & \mathbf{H}(n,0) & \mathbf{0}_{J,1} & \mathbf{H}(n,4) & \mathbf{0}_{J,4} & \mathbf{H}(n,11) & \mathbf{0}_{J,11} \\ \vdots & & & & & & \vdots \\ \mathbf{0}_{J,5} & \mathbf{H}(n,0) & \mathbf{0}_{J,2} & \mathbf{H}(n,4) & \mathbf{0}_{J,5} & \mathbf{H}(n,11) & \mathbf{0}_{J,6} \\ \vdots & & & & & & \vdots \\ \mathbf{0}_{J,11} & \mathbf{H}(n,0) & \mathbf{0}_{J,3} & \mathbf{H}(n,4) & \mathbf{0}_{J,4} & & \mathbf{H}(n,11)\end{bmatrix}, \tag{7}$$

Kalman filter may be computationally prohibitive. In such cases, Krylov based methods [27] - [29] become relevant.

Remark The equivalence between the reduced equalizer and sparsity promoting Bayesian estimator [30], [31] is shown in [12] (Section V), by considering (sparse) channel's prior probability density function (pdf) as $f((\mathbf{H}(n,l_k)_{(i,j)}) = [(\mathbf{H}(n,l_k))_{(i,j)}]^{-1/2}$, (i. e., magnitude of a channel tap will have a low value with high probability, and a large value with low probability [30]). Then equations (26)-(34) in [12] show that this (prior) pdf transforms the above sparsity promoting Bayesian estimator into a reduced rank filter.

IV. Novel, Fast-Converging, Multi-Stage Kalman Filter (MSKF)

A computationally efficient, reduced rank Multistage Wiener Filter (MSWF) (for time-invariant (ITV) systems) has been developed in [24], which converges to some Krylov based methods. It involves a reduction in the dimensionality of the observed data to obtain a MMSE filter, which is as close as possible to what can be attained if all the observed data were used in the estimation process. [24], and its variants [32], have been successfully used in CDMA data estimation, and recently in the author's semiblind estimation of time-invariant sparse channels [11]. The novel Multistage Kalman Filter (MSKF) here, is inspired from such considerations, and involves substantial extension to TV state estimation. This is achieved by utilizing innovations data $\tilde{\mathbf{y}}(n)$ to estimate the desired signal $\tilde{\mathbf{H}}(n|\tilde{\mathbf{y}}(n))$ (i. e., second term in RHS of (11)), by employing a novel, fast-converging, stage-by-stage Kalman filter structure, referred to as MSKF.

A. Full Kalman Filter

The the top-level/0th stage data, $\mathbf{z}_0(n)\overset{\Delta}{=}\tilde{\mathbf{y}}(n)$, uses the full (not multi-stage) Wiener filter's weights $\mathbf{w}_{\mathbf{z}_0}(n)$ [5] to estimate (0th order) desired signal $\mathbf{D}_0(n) = \tilde{\mathbf{H}}(n|\tilde{\mathbf{y}}(n))$. at each symbol n, by

$$\begin{aligned}\mathbf{w}_{\mathbf{z}_0}(n) &= (\mathbf{R}_{\mathbf{z}_0})^{-1}\mathbf{R}_{\mathbf{z}_0,\tilde{\mathbf{H}}(n)},\\ \hat{\mathbf{D}}_0(n) &= \hat{\tilde{\mathbf{H}}}(n|\tilde{\mathbf{y}}(n)) = \mathbf{w}^H_{\mathbf{z}_0}(n)\mathbf{z}_0(n).\end{aligned} \tag{15}$$

Then the aposteriori channel estimate $\hat{\tilde{\mathbf{H}}}(n|\tilde{\mathbf{y}}(n))$ and channel error correlation matrix $\mathbf{R}_{\mathbf{H}^e}(n|n)$ are updated iteratively by (11) and (14) respectively.

B. MSKF Derivation

The top-level Kalman filter weight $\mathbf{w}_{\mathbf{z}_0}(n)$, in (15), is implemented in a multi-stage fashion (MSWF) here, leading to faster stage-wise convergence at reduced complexity. For ease of presentation, the derivation of vector MSWF (V-MSWF) (obtained by extending scalar MSWF [24]), is provided in Technical Report [33]. The main steps of V-MSWF algorithm are shown in Table II. Table II's equations, (58) - (62), are then directly applied to TV state space model, (9), (10), to derive the MSKF's novel state estimator, in terms of state space matrices.

The block diagram of novel MSKF, with $N = 3$ stages, is shown in Fig 2. First, the $J \times J(M+1)\bar{S}$-sized (0th stage) cross-correlation, $\mathbf{R}_{\tilde{\mathbf{y}}(n),\mathbf{H}^e(n|n-m_1)}$, and its normalized version, $\mathbf{C}_1$, are defined as,

$$\mathbf{R}_{\tilde{\mathbf{y}}(n),\mathbf{H}^e(n|n-m_1)} \overset{\Delta}{=} E\{\tilde{\mathbf{y}}(n)\mathbf{H}^{eH}(n|n-m_1)\}, \tag{16}$$

$$\begin{aligned}\mathbf{\Delta}_1 &= [\mathbf{R}^H_{\tilde{\mathbf{y}}(n),\mathbf{H}^e(n|n-m_1)}\mathbf{R}_{\tilde{\mathbf{y}}(n),\mathbf{H}^e(n|n-m_1)}]^{1/2},\\ \mathbf{C}_1 &\overset{\Delta}{=} [\mathbf{R}_{\tilde{\mathbf{y}}(n),\mathbf{H}^e(n|n-m_1)}][\mathbf{\Delta}_1]^{-1}.\end{aligned} \tag{17}$$

In the time-invariant (ITV) case [24], [11], the expectation operator in (16) is implemented by time averaging over some received symbols. But in TV channel, the different received symbols are generated by different channel conditions $\tilde{\mathbf{H}}(n)$, which vary with symbol number n, making time averaging unsuitable in this situation. To circumvent this problem, $\mathbf{R}_{\tilde{\mathbf{y}}(n),\mathbf{H}^e(n|n-m_1)}$ has to be computed in terms of state matrices available at time n only. Substituting (10) in (16),

$$\begin{aligned}&\mathbf{R}_{\tilde{\mathbf{y}}(n),\mathbf{H}^e(n|n-m_1)} = E\{\tilde{\mathbf{y}}(n)\mathbf{H}^{eH}(n|n-m_1)\}\\ &= E\{(\mathbf{C}(n)\mathbf{H}^e(n|n-m_1) + \mathbf{w}(n))\mathbf{H}^{eH}(n|n-m_1)\}\\ &= \mathbf{C}(n)\mathbf{R}_{\mathbf{H}^e}(n|n-m_1) + E\{\mathbf{w}(n)\mathbf{H}^{eH}(n|n-m_1)\}\end{aligned} \tag{18}$$

Now, the 2nd term on RHS of (18), (noise term), is

$$\begin{aligned}E\{\mathbf{w}(n)\mathbf{H}^{eH}(n|n-m_1)\} = E\{\mathbf{w}(n)(\tilde{\mathbf{H}}(n)-\\ \hat{\tilde{\mathbf{H}}}(n|n-m_1))^H\}\end{aligned} \tag{19}$$

The first term in (19), $E\{\mathbf{w}(n)\tilde{\mathbf{H}}^H(n)\} = E\{\mathbf{w}(n)(diag(\lambda^{m_1})\tilde{\mathbf{H}}(n-m_1) + \mathbf{V}(n))^H\} = \mathbf{0}$, since measurement noise $\mathbf{w}(n)$ is uncorrelated with process noise $\mathbf{V}(n)$; $\mathbf{w}(n)$ is also uncorrelated with $\tilde{\mathbf{H}}(n-m_1)$, (which depends on $\mathbf{V}(n-m_1), \cdots, \mathbf{V}(n-m_1-k)$'s etc). Similarly, the second term in (19), $E\{\mathbf{w}(n)\hat{\tilde{\mathbf{H}}}^H(n|n-m_1)\} = \mathbf{0}$, since $\hat{\tilde{\mathbf{H}}}(n|n-m_1)$ is estimated by $\{\tilde{\mathbf{y}}(n-j)\}_{j=m_N}^{j=m_1}$'s, which contain measurement noises $\{\mathbf{w}(n-j)\}_{j=m_N}^{j=m_1}$'s, all of which are uncorrelated with white noise $\mathbf{w}(n)$ at symbol n. Then (17) can be computed by

$$\mathbf{C}_1 = [\mathbf{C}(n)\mathbf{R}_{\mathbf{H}^e}(n|n-m_1)][\mathbf{\Delta}_1]^{-1}. \tag{20}$$

Equation (20) is a key equation. From (8), $\mathbf{C}(n)$ is known at time n. Moreover, $\mathbf{R}_{\mathbf{H}^e}(n|n-m_1)$ is iteratively updated from its past value at $(n - m_1 - m_2)$th symbol, by (14) and (38) below, and is thus available (at present time n) for computing $\mathbf{C}_1$ by (20) as the product of $\mathbf{C}(n)$ (measurement matrix in state-space representation) and $\mathbf{R}_{\mathbf{H}^e}(n|n - m_1)$. This avoids explicit time averaging in (16), required in time-invariant (ITV) filter [11]. Equation (20) also gives the 0th order Wiener filter weights, in (15), as

$$\begin{aligned}\mathbf{w}_{\mathbf{z}_0}(n) = [\mathbf{C}(n)\mathbf{R}_{\mathbf{H}^e}(n|n - m_1)\cdot \\ \mathbf{C}^H(n) + \sigma_w^2\mathbf{I}_J]^{-1}[\mathbf{C}(n)\mathbf{R}_{\mathbf{H}^e}(n|n - m_1)]\end{aligned} \tag{21}$$

Again (21) avoids explicit time-averaging and is computed from state matrices, available at the present nth symbol. In the next stage, the (1st order) $J\bar{S}(M+1) \times 1$ desired signal vector $\mathbf{d}_1(n)$ and blocking matrix $\mathbf{B}_1$ are formed by

$$\mathbf{d}_1(n) = \mathbf{C}_1^H\tilde{\mathbf{y}}(n),\ \mathbf{B}_1 = [\mathbf{I} - \mathbf{C}_1\mathbf{C}_1^H]. \tag{22}$$

It can easily be shown that when $\mathbf{B}_1$ operates on any signal, it removes the component of $\mathbf{C}_1$ present in that signal, i. e., $[\mathbf{C}_1^H\mathbf{B}_1] = \mathbf{0}$. Defining the 1st order signal,

$$\tilde{\mathbf{y}}_1(n) \overset{\Delta}{=} \mathbf{B}_1\tilde{\mathbf{y}}(n) = \mathbf{B}_1(\mathbf{C}(n)\mathbf{H}^e(n|n - m_1) + \mathbf{w}(n)), \tag{23}$$

Then the 1st order normalized cross-correlation is given by

$$\begin{aligned}\mathbf{C}_2 &\overset{\Delta}{=} = \mathbf{R}_{\tilde{\mathbf{y}}_1(n),\mathbf{d}_1(n)}[\boldsymbol{\Delta}_2]^{-1}, \\ \boldsymbol{\Delta}_2 &= (\mathbf{R}^H_{\tilde{\mathbf{y}}_1(n),\mathbf{d}_1(n)}\mathbf{R}_{\tilde{\mathbf{y}}_1(n),\mathbf{d}_1(n)})^{1/2}, \\ \Longrightarrow\ \mathbf{C}_2 &= \mathbf{B}_1\mathbf{R}_{\tilde{\mathbf{y}}(n)}\mathbf{C}_1[\boldsymbol{\Delta}_2]^{-1},\end{aligned} \tag{24}$$

by using (22) and (23). From (20),

$$\begin{aligned}&\mathbf{C}_2 = \mathbf{B}_1(\mathbf{C}(n)\mathbf{R}_{\mathbf{H}^e}(n|n - m_1)\mathbf{C}^H(n) + \sigma_w^2\mathbf{I}_J)\mathbf{C}_1[\boldsymbol{\Delta}_2]^{-1}, \\ &= [\mathbf{I} - \mathbf{C}_1\mathbf{C}_1^H][\mathbf{C}(n)\mathbf{R}_{\mathbf{H}^e}(n|n - m_1)\mathbf{C}^H(n) + \sigma_w^2\mathbf{I}_J]\cdot \\ &[\mathbf{C}(n)\mathbf{R}_{\mathbf{H}^e}(n|n - m_1)][\boldsymbol{\Delta}_1]^{-1}[\boldsymbol{\Delta}_2]^{-1}.\end{aligned} \tag{25}$$

Again, explicit time averaging (over a number of received data symbols) is avoided in (25).
To obtain the equations for any generic order (stage), as the order is changed from ith to $(i + 1)$th, and using (23), $\mathbf{R}_{\tilde{\mathbf{y}}_1(n)} = \mathbf{B}_1\mathbf{R}_{\tilde{\mathbf{y}}(n)}\mathbf{B}_1$, since $\mathbf{B}_1$ is a Hermitian matrix. Again defining 2nd order signals, $\mathbf{B}_2 = [\mathbf{I} - \mathbf{C}_2\mathbf{C}_2^H]$, desired signal $\mathbf{d}_2(n) = \mathbf{C}_2^H\tilde{\mathbf{y}}_1(n)$, $\tilde{\mathbf{y}}_2(n) = \mathbf{B}_2\tilde{\mathbf{y}}_1(n)$, the 2nd order cross-correlation is obtained as

$$\begin{aligned}\mathbf{R}_{\tilde{\mathbf{y}}_2(n),\mathbf{d}_2(n)} &= \mathbf{B}_2E\{\tilde{\mathbf{y}}_1(n)\tilde{\mathbf{y}}_1^H(n)\}\mathbf{C}_2 = \mathbf{B}_2\mathbf{R}_{\tilde{\mathbf{y}}_1(n)}\mathbf{C}_2 \\ &= \mathbf{B}_2(\mathbf{B}_1\mathbf{R}_{\tilde{\mathbf{y}}(n)}\mathbf{B}_1)\mathbf{C}_2.\end{aligned} \tag{26}$$

Then for the generic ith order, it can be shown that

$$\begin{aligned}\tilde{\mathbf{y}}_i(n) &= \mathbf{B}_i\tilde{\mathbf{y}}_{i-1}(n),\ \mathbf{d}_i(n) = \mathbf{C}_i^H\tilde{\mathbf{y}}_{i-1}(n), \\ \mathbf{R}_{\tilde{\mathbf{y}}_i(n),\mathbf{d}_i(n)} &= \mathbf{B}_i\mathbf{R}_{\tilde{\mathbf{y}}_{i-1}(n)}\mathbf{C}_i, \\ \boldsymbol{\Delta}_{i+1} &= (\mathbf{R}^H_{\tilde{\mathbf{y}}_i(n),\mathbf{d}_i(n)}\mathbf{R}_{\tilde{\mathbf{y}}_i(n),\mathbf{d}_i(n)})^{1/2}, \\ \mathbf{C}_{i+1} &\overset{\Delta}{=} \mathbf{R}_{\tilde{\mathbf{y}}_i(n),\mathbf{d}_i(n)}[\boldsymbol{\Delta}_{i+1}]^{-1}.\end{aligned} \tag{27}$$

Also, $\mathbf{R}_{\tilde{\mathbf{y}}_i(n)} = (\prod_{j=1}^{j=i}\mathbf{B}_j)\mathbf{R}_{\tilde{\mathbf{y}}(n)}(\prod_{j=1}^{j=i}\mathbf{B}_j)$. The normalized cross-correlations $\mathbf{C}_i$'s, blocking matrices $\mathbf{B}_i$'s, and ith order desired signal $\mathbf{d}_i(n)$ and data $\tilde{\mathbf{y}}_i(n)$'s have been generated as the order i is increased from $1, 2, \cdots, N$ (up-recursions).

From the computed varying-order signals, a reduced-rank multistage estimation algorithm is derived. This requires 0th order desired signal $\mathbf{D}_0(n) = \tilde{\mathbf{H}}(n|\tilde{\mathbf{y}}(n))$, (in the outer loop in block-diagram of Fig. 2), to be estimated from the 1st order data $\mathbf{z}_1(n) \triangleq [\mathbf{d}_1^H(n)\ \tilde{\mathbf{y}}_1^H(n)]^H$ (see (9), Table II), which is in the 1st inner loop of Fig. 2. The MMSE filter $\mathbf{w}_{\mathbf{z}_1}$, in (59) (Table II), is employed for this purpose. This process is continued in a nested fashion, to generate the $(i + 1)$th (order) inner loop from the ith loop in Fig. 2. Thus at the $(i + 1)$th stage, $\mathbf{d}_i(n)$ has to be estimated by $\mathbf{z}_{i+1}(n) = [\mathbf{d}_{i+1}^H(n)\ \tilde{\mathbf{y}}_{i+1}^H(n)]^H$. Generalizing (59),

$$\mathbf{w}_{\mathbf{z}_{i+1}} = \begin{bmatrix}\mathbf{I}_{DJ} & -\mathbf{w}_{i+2}^H\end{bmatrix}^H(\mathbf{E}_{i+1}^{-1}\boldsymbol{\Delta}_{i+1}), \tag{28}$$

where $\mathbf{w}_{i+2} = \mathbf{R}^{-1}_{\tilde{\mathbf{y}}_{i+1}}\mathbf{R}_{\tilde{\mathbf{y}}_{i+1},\mathbf{d}_{i+1}}$ are the Wiener tap weights for estimating $\mathbf{d}_{i+1}(n)$ from $\tilde{\mathbf{y}}_{i+1}(n)$. Extending (62) to the ith order (and after some algebra),

$$\begin{aligned}\hat{\mathbf{d}}_i(n) &= \mathbf{w}^H_{\mathbf{z}_{i+1}}\mathbf{z}_{i+1}(n) = \tilde{\mathbf{w}}^H_{i+1}\boldsymbol{\epsilon}_{i+1}(n) \\ \boldsymbol{\epsilon}_{i+1}(n) &\overset{\Delta}{=} [\mathbf{d}_{i+1}(n) - \hat{\mathbf{d}}_{i+1}(n)] = [\mathbf{d}_{i+1}(n) - \mathbf{w}^H_{i+2}\tilde{\mathbf{y}}_{i+1}(n)], \\ \tilde{\mathbf{w}}_{i+1} &= \mathbf{E}^{-1}_{i+1}\boldsymbol{\Delta}_{i+1}.\end{aligned} \tag{29}$$

In (29), estimation error $\boldsymbol{\epsilon}_{i+1}(n) = [\mathbf{d}_{i+1}(n) - \mathbf{w}^H_{i+2}\tilde{\mathbf{y}}_{i+1}(n)]$ is error between $(i+1)$th order desired signal $\mathbf{d}_{i+1}(n)$ and its estimate $\hat{\mathbf{d}}_{i+1}(n) = \mathbf{w}^H_{i+2}\tilde{\mathbf{y}}_{i+1}(n)$ (obtained by using $(i + 1)$th stage data $\tilde{\mathbf{y}}_{i+1}(n)$). The application of weight $\mathbf{w}_{i+1}$ to this $\boldsymbol{\epsilon}_{i+1}(n)$), in (29), provides the estimate of the lower (ith) order desired signal, i. e. $\hat{\mathbf{d}}_i(n)$, leading to the down-recursion in (32).

Initializing $\boldsymbol{\epsilon}_N(n)$ by $\boldsymbol{\epsilon}_N(n) = \mathbf{y}_{N-1}(n) = \mathbf{d}_N(n)$, the error energy is

$$\begin{aligned}\mathbf{E}_N &\overset{\Delta}{=} E\{\boldsymbol{\epsilon}_N(n)\boldsymbol{\epsilon}_N^H(n)\} = E\{\mathbf{d}_N(n)\mathbf{d}_N^H(n)\} \\ &= \mathbf{C}_N^H\mathbf{R}_{\tilde{\mathbf{y}}_{N-1}}(n)\mathbf{C}_N.\end{aligned} \tag{30}$$

Again, no explicit time averaging (over a number of data symbols) is involved. Also,

$$\begin{aligned}\boldsymbol{\Delta}_N &\overset{\Delta}{=} E\{\tilde{\mathbf{y}}_{N-1}(n)\mathbf{d}^H_{N-1}(n)\} \\ &= \mathbf{B}_{N-1}\mathbf{R}_{\tilde{\mathbf{y}}_{N-2}}(n)\tilde{\mathbf{R}}_{\tilde{\mathbf{y}}_{N-2}(n),\mathbf{d}_{N-2}(n)}\end{aligned} \tag{31}$$

Using (28)-(31), the order down-recursions, for $j = N, N-1, \cdots, 1$, are given by

$$\mathbf{w}_j = [\boldsymbol{E}_j]^{-1}\boldsymbol{\Delta}_j,\ \hat{\mathbf{d}}_{j-1}(n) = \mathbf{w}_j^H\boldsymbol{\epsilon}_j(n), \tag{32}$$

$$\boldsymbol{\epsilon}_{j-1}(n) = \mathbf{d}_{j-1}(n) - \hat{\mathbf{d}}_{j-1}(n) = \mathbf{d}_{j-1}(n) - \mathbf{w}_j^H\boldsymbol{\epsilon}_j(n), \tag{33}$$

$$\begin{aligned}\mathbf{E}_{j-1} &= E\{\boldsymbol{\epsilon}_{j-1}(n)\boldsymbol{\epsilon}^H_{j-1}(n)\} = E\{\mathbf{d}_{j-1}(n)\mathbf{d}^H_{j-1}(n)\} \\ &- \mathbf{w}_j^H\mathbf{E}_j\mathbf{w}_j = E\{\mathbf{d}_{j-1}(n)\mathbf{d}^H_{j-1}(n)\} - \mathbf{w}_j^H\boldsymbol{\Delta}_j.\end{aligned} \tag{34}$$

Again, in order to avoid using the expectation operator in (34), we have from (27),

$$\begin{aligned}&E\{\mathbf{d}_{j-1}(n)\mathbf{d}^H_{j-1}(n)\} = \mathbf{C}^H_{j-1}E\{\tilde{\mathbf{y}}_{j-2}(n)\tilde{\mathbf{y}}^H_{j-2}(n)\}\mathbf{C}_{j-1} \\ &= \mathbf{C}^H_{j-1}\mathbf{R}_{\tilde{\mathbf{y}}_{j-2}(n)}\mathbf{C}_{j-1} \\ &\Longrightarrow\ \mathbf{E}_{j-1} = \mathbf{C}^H_{j-1}\mathbf{R}_{\tilde{\mathbf{y}}_{j-2}(n)}\mathbf{C}_{j-1} - \mathbf{w}_j^H\boldsymbol{\Delta}_j.\end{aligned} \tag{35}$$

Thus (35) is implemented from precomputed quantities at previous time, employing only matrix multiplications, without any explicit time averaging of received data symbols.

1) Time-Updates: Using (32)-(35) and (13), desired signal $\mathbf{D}_0(n) = \hat{\tilde{\mathbf{H}}}(n|\tilde{\mathbf{y}}(n))$ channel estimate and the channel estimate error $\mathbf{H}^e(n|n)$ are given by,

$$\hat{\tilde{\mathbf{H}}}(n|\mathbf{y}(n)) = \tilde{\mathbf{w}}_1^H \boldsymbol{\epsilon}_1(n);$$
$$\implies \mathbf{H}^e(n|n) = \mathbf{H}^e(n|n-m_1) - \tilde{\mathbf{w}}_1^H \boldsymbol{\epsilon}_1(n). \tag{36}$$

First, the apriori channel error correlation matrix is iteratively predicted by

$$\mathbf{R}_{H^e}(n|n-m_1) = \mathrm{diag}(\lambda^{m_1})\mathbf{R}_{H^e}(n-m_1|n-m_1)$$
$$\mathrm{diag}(\lambda^{m_1})^H + \mathbf{R}_v(n-m_1) \tag{37}$$

Then, using (14), aposteriori $\mathbf{R}_{H^e}(n|n)$ is updated by

$$\mathbf{R}_{H^e}(n|n) = \mathbf{R}_{H^e}(n|n-m_1) - E\{\hat{\tilde{\mathbf{H}}}(n|\tilde{\mathbf{y}}(n))\hat{\tilde{\mathbf{H}}}^H(n|\tilde{\mathbf{y}}(n))\}$$
$$= \mathbf{R}_{H^e}(n|n-m_1) - \tilde{\mathbf{w}}_1^H \mathbf{E}_1(n)\tilde{\mathbf{w}}_1. \tag{38}$$

Remark:

This nested filter structure is possible because

1. By the orthogonality principle of MMSE estimation, the estimation error $\boldsymbol{\epsilon}_{i+1}(n)$
$= [\mathbf{d}_{i+1}(n) - \mathbf{w}_{i+2}^H \tilde{\mathbf{y}}_{i+1}(n)]$ is orthogonal to the data used in estimation, i. e. $\tilde{\mathbf{y}}_{i+1}(n)$.

2. Again by construction, $\tilde{\mathbf{y}}_{i+1}(n)$ is orthogonal to $\mathbf{d}_i(n)$, since

$$\mathbf{R}_{\tilde{\mathbf{y}}_{i+1}(n),\mathbf{d}_i(n)} = \mathbf{B}_{i+1}[E\{\tilde{\mathbf{y}}_i(n)\mathbf{d}_i^H(n)\}] = \mathbf{B}_{i+1}[\mathbf{C}_{i+1}\boldsymbol{\Delta}_{i+1}]$$
$$= [\mathbf{I} - \mathbf{C}_{i+1}\mathbf{C}_{i+1}^H][\mathbf{C}_{i+1}\boldsymbol{\Delta}_{i+1}] = \mathbf{0}. \tag{39}$$

3. Since $\mathbf{d}_i(n)$ and $\boldsymbol{\epsilon}_{i+1}(n)$ are both uncorrelated with $\tilde{\mathbf{y}}_{i+1}(n)$, $\mathbf{d}_i(n)$ and $\boldsymbol{\epsilon}_{i+1}(n)$ may be correlated, which incidentally is Δ_{i+1} Hence, $\boldsymbol{\epsilon}_{i+1}(n)$ can be used for estimating the desired signal $\mathbf{d}_i(n)$, [i. e., $\hat{\mathbf{d}}_i(n) = \tilde{\mathbf{w}}_{i+1}^H \boldsymbol{\epsilon}_{i+1}(n)$ in equation (29) above], leading to the novel nested MSKF filter. The novel MSKF algorithm is then fully tabulated in

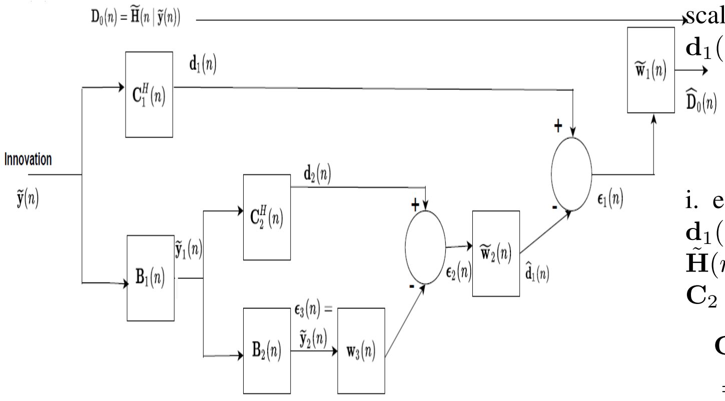

Fig. 2

BLOCK DIAGRAM OF MSKF (N = 3).

C. Some Issues

1) MMIMO case: Next, re-visit the discussion in "Remark" (Section II) about the common sparsity support (over all receive antennas) assumption being violated in MMIMO, since the received signal is delayed at the J different receive antennas, with overall distance between them increasing for large J. Following ([15], pp. 106, and Table I) with distance between 2 consecutive antennas $d = \frac{C}{2f_c}$, (C-velocity of light), the maximum distance (between the farthest antennas in a linear array) $d_{max} = (J-1)d)$ is very large, for large J in MMIMO. Then for high bandwidth (BW) communication systems, if $\frac{d_{max}}{C} > \frac{10}{BW}$, significant channel tap locations l_k vary *spatially* or, are different, across the farthest antennas, see [15]. As an illustrative example, in Fig. 1 c), $J = 3\bar{J}$, and say over the bottom $\bar{J}$ receive antennas, the significant channel taps locations are l_0, l_1, l_2 , and over the next (upper) $\bar{J}$ antennas, the locations are l_0+1, l_1+1, l_2+1, while over the top-most $\bar{J}$ antennas, they are at l_0+2, l_1+2, l_2+2. Thus, for the mid antenna group, channel location vector $\mathbf{h}_{mid} = [0, 0, l_0+1, 0, \cdots, 0, l_1+1, 0, \cdots, 0, l_2+1, 0, \cdots, 0]^T$. Then $\mathbf{y}(n)$, in (3), can be considered as a $\bar{S} = 3$-user system, with channels $\begin{bmatrix}\mathbf{h}_{bottom} & \mathbf{h}_{mid} & \mathbf{h}_{top}\end{bmatrix}$, as in Fig. 1 c) (Simulations in Sec. VII).

Note However, the common sparsity assumption might still be valid for MMIMO systems with compact arrays such as those proposed for futuristic THz frequencies [35].

D. Kalman Krylov Filter (KKL)

In KKL, the Wiener filter (15) is implemented using a Arnoldi-Krylov-Householder method [29], [28], expected to have superior numerical properties than [27]. For ease of presentation, the KKL agorithm is shown in Table IV.

V. PERFORMANCE ANALYSIS: COMPARISON OF ORDER-WISE CONVERGENCE SPEEDS

In [24], it is shown that time-invariant (ITV) MSWF filter converges to a N dimensional subspace, that has the largest correlations between the eigenvectors of $\mathbf{R}_{\tilde{\mathbf{y}}(n)}$ and the desired signal $\mathbf{D}_0(n) = \tilde{\mathbf{H}}(n|\tilde{\mathbf{y}}(n))$, (equation (76), [24]). Suppose at the 1st stage, the Kalman filter weight (with $\tilde{\mathbf{y}}(n) = \mathbf{z}_0(n)$ in (15)) is collinear with the cross-correlation vector $\mathbf{C}_1$, i. e. let, $\mathbf{w}_{\tilde{\mathbf{y}}(n)} = k\mathbf{C}_1$, where k is a scalar constant. Then after just 1 stage/iteration, (by (22)), $\mathbf{d}_1(n)$ becomes

$$\mathbf{d}_1(n) = \mathbf{C}_1^H \tilde{\mathbf{y}}(n) = (1/k)\mathbf{w}_{\tilde{\mathbf{y}}(n)}^H \tilde{\mathbf{y}}(n)$$
$$= (1/k)\hat{\tilde{\mathbf{H}}}(n|\tilde{\mathbf{y}}(n)) = (1/k)\mathbf{D}_0(n), \tag{40}$$

i. e., the final channel estimate. Thus, just after 1 stage, $\mathbf{d}_1(n)$ (in MSKF) gives the optimal estimate of desired signal $\tilde{\mathbf{H}}(n|\tilde{\mathbf{y}}(n))$, for each symbol n. Then using (40), 2nd stage $\mathbf{C}_2$ is

$$\mathbf{C}_2 = \mathbf{R}_{\tilde{\mathbf{y}}_1(n),\mathbf{d}_1(n)}[\boldsymbol{\Delta}_2]^{-1} = E\{\tilde{\mathbf{y}}_1(n)\mathbf{d}_1^H(n)\}[\boldsymbol{\Delta}_2]^{-1}$$
$$= (1/k)(E\{\tilde{\mathbf{y}}_1(n)\mathbf{D}_0^H(n)\})[\boldsymbol{\Delta}_2]^{-1} = (\mathbf{0})[\boldsymbol{\Delta}_2]^{-1}, \tag{41}$$

by (39) (for $i = 0$). Thus, there is *no* further need to compute succeeding stages $\mathbf{C}_j$'s for $j \geq 2$, similar to (pp. 2953, [24]). Again, (14) in MSKF, gives

$$\mathbf{R}_{\mathbf{H}^e}(n|n) = \mathbf{R}_{\mathbf{H}^e}(n|n-m_1) - E\{\hat{\tilde{\mathbf{H}}}(n|\tilde{\mathbf{y}}(n))\hat{\tilde{\mathbf{H}}}^H(n|\tilde{\mathbf{y}}(n))\}$$
$$= [\mathbf{I} - \mathbf{w}_{\tilde{\mathbf{y}}(n)}^H \mathbf{R}_{\tilde{\mathbf{y}}(n),\mathbf{H}^e(n|n-m_1)}(\mathbf{R}_{\mathbf{H}^e}(n|n-m_1))^{-1}]\cdot$$
$$\mathbf{R}_{\mathbf{H}^e}(n|n-m_1). \tag{42}$$

For a given $\mathbf{R}_{\mathbf{H}^e}(n|n-m_1))$, (42) is minimized, when the 2nd term, $\mathbf{w}_{\tilde{\mathbf{y}}(n)}^H \mathbf{R}_{\tilde{\mathbf{y}}(n),\mathbf{H}^e(n|n-m_1)}(= \mathbf{w}_{\tilde{\mathbf{y}}(n)}^H \mathbf{C}_1\boldsymbol{\Delta}_1)$ is maximized [24]. This happens ȏnly when $\mathbf{w}_{\tilde{\mathbf{y}}(n)}$ and $\mathbf{C}_1(n)$ are in phase with each other, i. e., Kalman filter weight $\mathbf{w}_{\tilde{\mathbf{y}}(n)}$ is collinear with the cross-correlation vector $\mathbf{C}_1(n)$. Then the maximum value (of 2nd term of (42)) is $\mathbf{C}_1^H(n)\mathbf{C}_1(n)\boldsymbol{\Delta}_1$, and this minimizes (42) . Thereby, by attempting to make the normalized cross-correlation collinear with the Kalman filter weight, (at each stage/iteration, and symbol n), the MSKF has a fast stage/iteration number-wise convergence. This also leads to novel MSKF exhibiting approximately same convergence speed (number of iterations), even for large MIMO systems, i. e., larger loading ratio $\mathbf{R} = \frac{\bar{S}}{J}$. This has also been seen in ITV MSWF [34], (where equation (46) and Fig. 4) show that for *any* $\mathbf{R} < 1$, the output SINR increases rapidly (to optimal value) with increasing stage number i. This unique property of novel MSKF, i. e., rapid convergence (even for large-scale systems), is not exhibited in Bayesian MSBL [16], [22] and RCS methods (Simulations Sec VII).

On the other hand, Kalman-Krylov filter (KKL) converges to the dominant signal/eigen subspace of $\mathbf{R}_{\tilde{\mathbf{y}}(n)}$, corresponding to its N largest eigenvalues. Theorem 3.5.1 (pp. 48, [27]) shows the angle (between the KKL and the true eigen subspace) decreases at every stage/iteration. Also, the KKL algorithm steps show that it determines a N dimensional subspace for $\mathbf{R}_{\tilde{\mathbf{y}}(n)}$ (principal component analysis PCA), *rather* than converging to one, which has the largest correlations between of $\mathbf{R}_{\tilde{\mathbf{y}}(n)}$'s eigenvectors and desired signal $\mathbf{D}_0(n)$ (as done by novel MSKF). Simulation results (Sec. VII) show the MSKF converge quickly, within an order of 14; (i. e., channel NRMSE remains almost same, even with number of iterations increasing from 14 to 40). But KKL converges *slowly*, with increasing iteration number. For large systems ($J = 28, \bar{S} = 14$), MSKF performs well, while MSBL, CS (RCS), [22] and PCA methods perform inadequately, i. e., they do not not scale up well.

A. Comparison with Existing Methods

Not much work exists on TV MMIMO channel estimation [4], (pp. 1926). The novelty of MSKF vis-a-vis existing algorithms is

1. The MSKF is compared with re-weighted compressed sensing (RCS) [20], [18] (which is closer to l_0 than the l_1 norm criterion). Starting with the minimization of RCS (in proximal form), it develops an iterative CS algorithm using soft thresholding. This is then applied to sparse beamformed channel estimation; [20] is only for single path (not multipath) channel. However, RCS converges very slowly, after many symbols (see [20], Fig. 3) and also in simulations for our signal model (Section VII, Fig, 1), where it takes as many $200-600$ symbols to converge. This requires the channel to be static over that time period, rendering it unsuitable for high Doppler channels; while MSKF works, even with the channel changing every symbol. This is because, the number of stages (in OMP) is sparsity level $d = DKJ$, which increases rapidly with increasing J, K in Large-Scale MMIMO systems. Thus, its convergence speed is slow.
2. By using additional beam-forming hardware, which slows down MMIMO channel time-variation ([17], pp. 2, and its ref [12]). [4] and [17] estimate high Doppler channels in mmwave communications, but both are developed only for single-tap channels, and do not exploit sparsity in temporal (lag) domain.
3. The uniformly-spaced "Full' equalizer perform worse than a reduced/non-uniformly spaced equalizer (see Figure 3 in [1]) and also in learning curves (Figure 4, [1] and Figure 3, [2]), even for time-invariant (ITV) sparse DTV channels, and in simulations (Section VII) here. [Also, similar results in [11] are due to the non-required taps in "Full" equalizer just adding noise to the estimation process]. Also, the Bayesian filter (used in Expectation step in [4] employs a ("Full" - uniformly spaced all equalizer taps) Kalman filter (see 3. below), unlike the novel reduced re-configurable equalizer here.
4. In addition, KKL and Bayesian MSBL [16] are seen to converge much slowly than novel MSKF, especially for large-scale MMIMO systems (Sec. VII).
5. Bayesian methods [14], (simulated only for slow-varying channels, an AR(1) model, $\lambda = 0.9999$), and [4] can be computationally very demanding for MMIMO systems, since they are not equipped with suitable model order reduction.
6. Channel magnitude, corresponding to smaller λ (high-mobility channels), decreases, with increasing n, as distance between mobile and base-station increases. Thus received signal (for large n) will be more noisy, and leads to channel tracking errors in a standard Kalman filter (see Fig. 8, [23]). This is alleviated by data censoring and reduced equalizer in MSKF.
7. MSKF has also been compared to one of few existing Krylov based TV channel estimator [22]. [22] models the channel time-variation by a basis exponential method (BEM), which inevitably introduces approximation error to channel estimates, due to the imperfect model assumed [17]. Though [22], [36] are developed for high mobility channels, they do not exploit its sparsity. [22] performs worse than novel MSKF and KKL methods (Sec. VII).
8. Moreover, our novel reduced-rank filters are shown to be equivalent to some Bayesian estimator [12], with sparse channel's (prior) pdf of $f((\mathbf{H}(n, l_k)_{(i,j)}) = [(\mathbf{H}(n, l_k))_{(i,j)}]^{-1/2}$, i. e., (magnitude of a channel tap will have a low value with high probability). Then equations (26)-(34) in [12] show that this prior pdf leads one to a reduced rank filter - see Section III. B. above
9. Unlike existing methods, MSKF combines both channel sparsity with a Kalman filter incorporating model-order reduction, leading to fast convergence speed.

B. Comparison of Computational Complexities

First, computational complexity of traditional (full - uniformly spaced all equalizer taps) Kalman filter is evaluated. Since one does not know the significant channel tap locations apriori, $(L+1)J\bar{S} \times 1$-sized $\tilde{\mathbf{H}}(n)$ is used in state-space equations (8) and (10), ($\mathbf{C}_{full}(n)$'s size is $J \times (L+1)J\bar{S}$). In [27], Kalman update $\tilde{\mathbf{H}}(n|\tilde{\mathbf{y}}(n)$ and channel error correlation $\mathbf{R}_{\mathbf{H}^e}(n|n)$, $\mathbf{R}^{-1}_{\tilde{\mathbf{y}}(n)}$, along with intermediate quantities, are computed. Then "Full" Kalman requires computational complexity of $J^3(2(L+1)^2 + L + 1) + ((L+1)J)^3 + J^2 + (2/3)J^{2.376} + (L+1)[J^4 + J^3 + J^2] = 5.963 \times 10^8$ multiplications (for Example I Channel with $L+1 = 12$ taps and $D = 3$ significant taps, $J = 60$ receive antennas). for each symbol n, which over 30 symbols, gives the total complexity of "Full" filter as 18×10^9 complex multiplications. Since sparse channel has only D non-significant taps, $\mathbf{y}_{red}(n)$, has fewer $N \leq M$ equalizer taps. Thus, calculating $\mathbf{C}_1$ in (20), first requires multiplication of $J \times DJ$-sized $\mathbf{C}(n)$ and $DJ \times DJ$-sized $\mathbf{R}_{\mathbf{H}^e}(n|n-m_1)$, to obtain intermediate

$\tilde{\mathbf{U}} \overset{\Delta}{=} (\mathbf{C}(n)\mathbf{R}_{\mathbf{H}^e}(n|n-m_1)$, requiring $J[DJ]^2$ multiplications. Additional $\boldsymbol{\Delta}_1$ (in (17)), Cholesky decomposition based matrix inversion for $[\boldsymbol{\Delta}_1]^{-1}$ [37], gives the total computational complexity of $\mathbf{C}_1$ as $3J^3[D]^2 + (2/3)((DJ)^{2.376})$ multiplications. Then complexity of $\mathbf{d}_1(n)$, $\tilde{\mathbf{y}}_1(n)$, $\mathbf{R}_{\tilde{\mathbf{y}}}(n)$, $\mathbf{E}_N^{-1}$, $\boldsymbol{E}_{N-1}$, $\mathbf{w}_j(n)$, $\boldsymbol{\epsilon}_{N-1}(n)$, using (22), (23), (10), (20) etc. is $(D^2+2D)J^2 + DJ^3 + (1/3)(DJ)^{2.376}) + J^3(D^3 + D^2 + D)$ multiplications. Then computational complexity of 14 stages of MSKF (by which convergence is achieved) 18.84×10^7 for each symbol n. Then with *only* 11 time updates , (over 30 symbols), the overall complexity is 2.0724×10^9 complex multiplications. The KKL method [29], [28], [27] requires order of at least 30 to converge (see Fig. 3); thus its complexity $C = J^3(1 + \sum_{k=1}^{30} k^2) + J^2 + 2J + \sum_{k=1}^{30} \frac{2}{3}k^3 + \sum_{k=1}^{30} k^2 J + (L+1)J^2(J+1)$. For $J = 60$, $C = 2.0458 \times 10^9$ multiplications, (at each n), which over 30 symbols, requires 61.374×10^9 complex multiplications. Table V shows the *scaling* of computational complexity as J is increased.

VI. Bayesian Cramer-Rao Bounds for Noisy Sparse Channels

Bayesian Cramer-Rao bounds for noisy non-blind, stationary compressed sensing have been derived in [39], with measurements $\mathbf{y}(n)$

$$\mathbf{y}(n) = \boldsymbol{\Phi}\mathbf{x}(n) + \mathbf{e}(n),\ \mathbf{x}(n) = \boldsymbol{\Psi}\mathbf{w}(n). \tag{43}$$

$\boldsymbol{\Phi}$ is the measurement matrix, $\mathbf{e}(n)$ is the measurement noise, a 0 mean white noise with variance σ_e^2. $\mathbf{x}(n)$ is a vector that is sparse in the dictionary/domain $\boldsymbol{\Psi}$.

Adapting to our problem of MIMO sparse channel estimation, $\boldsymbol{\Phi} = \mathbf{T}(n)$; $\mathbf{T}(n)$ is the transmitted data matrix; $\mathbf{T}(n)$ (for a M length Massive-SIMO filter with J receive atennas) is formed, such that its (ith block-row, jth block-column) entry is defined by

$$\begin{aligned}&[\mathbf{T}(n)]_{((i-1)J+1:iJ,(j-1)J+1:jJ)} = \mathbf{I}_J \otimes s(n+i-j),\\ &i = 0,1,2,\cdots, M-1,\ j = 0,1,\cdots,L,\end{aligned} \tag{44}$$

($\otimes$ again the Kronecker product, $s(n)$: transmitted signal at nth symbol). The vector $\mathbf{x}(n)$, in (43), here is given by $\mathbf{x}(n) = \mathbf{h}(n) = \mathbf{h}$, the $(L+1)J \times 1$-sized (stationary) channel vector (with $(L+1)$ taps), which is sparse in the lag/time domain (i. e., dictionary $\boldsymbol{\Psi}$). Thus the dictionary matrix $\boldsymbol{\Psi} = \mathbf{I}_{(L+1)J}$ and $\mathbf{w}(n) = \mathbf{h}_{red} \overset{\Delta}{=} [\mathbf{h}^T(0), \mathbf{0}, \cdots, \mathbf{h}^T(l_1), \mathbf{0}, \cdots, \mathbf{h}^T(l_{D-1})]^T$ in the dictionary $\boldsymbol{\Psi}$. $(L+1)J \times 1$-sized reduced channel vector $\mathbf{h}_{red}(n)$ is a sparse vector, having DJ non-zero block entries only, i.e. $\|\mathbf{h}_{red}(n)\|_0 = D$. Corresponding to (43), the measured vector is the "full" received data vector $\mathbf{y}_{full}(n)$, given by

$$\mathbf{y}_{full}(n) = [\mathbf{y}^T(n)\,\mathbf{y}^T(n+1)\,\cdots\,\mathbf{y}^T(n+M-1)]^T. \tag{45}$$

For SIMO systems, the transmitted data matrix $\mathbf{T}(n)$ is of size $MJ \times (L+1)J$ and channel vector $\mathbf{h}$ is of size $JL \times 1$. [For MIMO systems (K transmit antennas), $\mathbf{T}((i-1)J+1 : iJ, (j-1)JK+1 : jJK) = \mathbf{I}_J \otimes [s^{(1)}(i+j-2), s^{(2)}(i+j-2) \cdots, s^{(K)}(i+j-2)]$ is of size $MJ \times J(L+1)K$, channel vector $\mathbf{h}$ is of size $J(L+1)K \times 1$].

For the stationary sparse channel estimation problem, equation (43) simplifies to

$$\begin{aligned}\mathbf{y}_{full}(n) &= \mathbf{T}(n)\boldsymbol{\Psi}\mathbf{h}_{red} + \mathbf{e}(n)\\ &= \mathbf{T}(n)[\mathbf{I}_{(L+1)J}]\mathbf{h}_{red} + \mathbf{e}(n),\end{aligned} \tag{46}$$

from which $\mathbf{h}_{red}$ is solved by various OMP, matching pursuit and other CS algorithms [39]. The pdf of the significant channel tap locations $p(\mathbf{h}(l_j) \neq \mathbf{0}_J) = \mathbf{I}_J$, $j = 0, 1, \cdots, D-1$, and assume that the prior pdf (of the non-significant channel tap location l's) is

$$p(\mathbf{h}(l)) = (2\pi)^{-J/2}[det(\mathbf{I}_J)^{-1/2}]e^{-\frac{[\sum_{j=1}^{J}|h_j(l)|^2]}{\sigma_j^2}}, \tag{47}$$

(σ_j^2: variance of 0-mean $h_j(l)$). For simplicity, we assume that $\sigma_j^2 = \sigma^2$, $j = 1, 2, \cdots, J$. Using the Fisher information matrix (FIM) $\mathbf{J}$, the Bayesian Cramer-Rao (BCRB) for the noisy CS is

$$E\{(\mathbf{h}(n) - \hat{\mathbf{h}}(n))(\mathbf{h}(n) - \hat{\mathbf{h}}(n))^H\} \geq \mathbf{J}^{-1} \tag{48}$$

$$\mathbf{J} = \mathbf{J}_{\mathbf{T}} + \mathbf{J}_p \tag{49}$$

$$[\mathbf{J}_{\mathbf{T}}](i,j) \overset{\Delta}{=} -E_{\mathbf{y}(n),\mathbf{h}(n)}[\frac{\partial^2 log(p(\mathbf{y}|\mathbf{h})}{\partial\mathbf{h}_i\partial\mathbf{h}_j}], \tag{50}$$

$$[\mathbf{J}_p](i,j) \overset{\Delta}{=} -E_{\mathbf{h}(n)}[\frac{\partial^2 log(p(\mathbf{h})}{\partial\mathbf{h}_i\partial\mathbf{h}_j}], \tag{51}$$

where $\mathbf{J}_{\mathbf{T}}$ is the data FIM, $\mathbf{J}_p$ is the prior FIM. After some algebraic manipulations [39], the BCRB is give by

$$\begin{aligned}E\{\|(\mathbf{h} - \hat{\mathbf{h}}(n)\|^2\} \geq \frac{(\sigma_e\sigma)^2)}{L+1}\\ \cdot\frac{D(MJ\sigma^2 + \sigma_e^2) + MJ(L-D+1)\sigma^2}{(\sigma^2 MJ)(MJ\sigma^2 + \sigma_e^2)}\end{aligned} \tag{52}$$

BCRB bounds for $J = 50, K = 1$, Example II channel with channel length $L+1 = 12, D = 3$ are compared against existing CS algorithms in Fig.8. The number of measurements is related to filter length MJ. Initially $M = 23$ for which BCRB is computed. Against all filter length of $M = 23$ for other methods, the novel MSKF selects only 12 equalizer taps for Example I channel (see equation (7) above). This is how novel MSKF exploits MMIMO channel sparsity in lag domain. Thus, BCRB are also shown for $M = 12$ in Fig. 8. BCRB bounds for SNR = 30 dB, $M = 23$ and $M - 12$ are 8.6956×10^{-7} and 1.666×10^{-6} respectively. As in [39], there is substantial performance gap between the practical algorithms and BCR bounds.

Novel MSKF is also compared with an ideal (but unrealizable) "Oracle Estimator" (believed to provide the sparse-est CS solution) in [13], where the significant channel taps are known apriori to be at l_m, $m = 0, 1, 2, \cdots, D-1$. The ith block-row and jth block-column of the associated training data matrix $\mathbf{U}(n)$ is given by

$$\begin{aligned}&[\mathbf{U}(n)]_{((i-1)J+1:iJ,(j-1)J+1:jJ)} = \mathbf{I}_J \otimes s(n+i-j),\\ &i = 0, 1, \ldots, M-1,\ j = l_0, l_1, \cdots, l_{D-1},\end{aligned} \tag{53}$$

the channel estimate is $\hat{\tilde{\mathbf{h}}}_{red} = (\mathbf{U}(n))^{\dagger}\mathbf{y}_{full}(n)$.

VII. Simulation Results

In MMIMO ($J = 50$ receive antennas) systems, the data signals $\{s_i^{(k)}(n)\}$ are binary phase shift keying (BPSK)/quadrature phase shift keying (QPSK) modulated.

Table V: Computational Complexity

J	"Full" Kalman	MSKF	KKL
60	18×10^9	2.0724×10^9	61.374×10^9
100	9.7205×10^{10}	1.0767×10^{10}	2.8408×10^{11}
140	3.0624×10^{11}	2.9465×10^{10}	7.7946×10^{11}

Simulation results are obtained by averaging over 600 trials; for each computer trial, independent and identically distributed complex Gaussian channel coefficients with zero mean and unit variance (Rayleigh fading channel) are generated, with TV component given by parameter λ in (4). The following algorithms are simulated:

1. "Full" Kalman filter, also used in recent [4], [14], TV EM method [38],
2. Reduced-rank KKL filter over varying orders, denoted by "Krylov",
3. Novel Multistage Kalman Filter (MSKF) filter for varying number of iterations, vs symbol number, at different SNRs, and varying λ's,
4. Multi-user Sparse channels,
5. Cluster-Sparse channels,
6. Recent dynamic, Bayesian-Belief Propagation method [16], denoted by "SBL",
7. Large scale MMIMO (large loading ratio $\mathbf{R} = \frac{\bar{S}}{J}$) systems,
8. Iterative re-weighted compressed sensing (RCS) [20],
9. Existing BEM based Krylov TV channel estimation [22], denoted by "Klov-BEM",
10. Oracle Estimator.

The receiver signal-to-noise ratio (SNR) is defined as $\text{SNR} = \frac{\text{E}(||\mathbf{y}(n) - \mathbf{w}(n)||^2)}{\text{E}(||\mathbf{w}(n)||^2)}$, ($\mathbf{w}(n)$: AWGN noise); performance of different estimators measured by normalized MSE (NRMSE)

$$\text{NRMSE} = \frac{1}{500} \sum_{p=1}^{500} \left\{ \frac{\sum_{\ell=0}^{L} ||\mathbf{H}^{(p)}(\ell) - \hat{\mathbf{H}}^{(p)}(\ell)||_F^2}{\sum_{\ell=0}^{L} ||\mathbf{H}^{(p)}(\ell)||_F^2} \right\}. \tag{54}$$

First, following Sec VI. B., we consider a stationary channel, as in [20], to investigate the convergence speed (in symbols) of OMP based RCS, for different values of loading factor $\mathbf{R} = \frac{K}{J}$ and different SNRs of $5, 20, 30$ dBs in Fig 3 a), while Fig 3 b) is the corresponding plot for the novel MSKF. Fig. 3 a) shows RCS to converge very slowly, after as many as $400 - 500$ symbols, making it unsuitable in high Doppler channels. (This has also been witnessed in [20]'s Fig. 3). Also, the performance of RCS degrades substantially, at lower SNRs (it is to be noted that non-stationarity factor λ is not incorporated into RCS [23], as is done in MSKF). Fig 3 b) shows the novel MSKF to perform very well with low NRMSE, starting from $n = 12$ on wards; also there is no channel tracking errors, as this is a stationary channel. Also, MSKF's performance degradation (at low SNR) is much less than that of RCS .

Next, Fig. 4 simulates the channel NRMSE of "Full", (which also limits the performance of Bayesian [4], [14], [38], Sec VI.B.); along with comparative simulations of novel MSKF, KKL, over varying number of iterations, at 30 and 5 dB SNRs, and $\lambda = 0.988$. The plot shows the KKL to *converge slowly*, with its NRMSE decreasing as number of iterations increases from 14 to $22, \cdots, 50$. MSKF sparse channel estimator converges very quickly, within an order of 14; as difference in channel NRMSE (at order of 14 to that at 40) is not significant. The "Full" equalizer also performs inadequately. The NRMSE of MSKF (with 14 iterations) is also less than that of "Full" and KKL (50 iterations). Fig. 4 b) shows results for $\lambda = 0.995$, resulting in lower NRMSEs. Also, the novel MSKF is able to handle the non-stationarity of the channel better than the "Full" filter. For $\lambda = 0.995$, the ratio (of NRMSE at symbol no 26 to that at symbol no 12) is 4.67 for MSKF and 21.33 for "Full"; while it is 10.43 in MSKF and increases rapidly to about 100 in "Full", for a more TV channel ($\lambda = 0.988$). Updating only at symbols $\{(n - m_l)\}$'s, akin to data censoring (in sparse channel) and updating in a reduced subspace, prevents the MSKF from exhibiting larger NRMSE, with increasing n, i. e. channel tracking errors (see Fig. 8 and text below it, [23]) occurring in a TV channel (see Sec VI. A.6). Fig. 5 provides results for multi-user ($\bar{S} = 2$) Example I Channel, and cluster-sparse ($\bar{S} = 3$) channels. Fig. 5 b) also includes the case for space-variant sparse channels, where large antenna arrays make l_j's change by a few lags, over two ends of antennas in MMIMO, (see Section IV. C). As expected in Sec. VI. A. 7,"Klov-BEM" (Krylov-space based method, using BEM, instead of more generic Kalman filter), performs worse than our Kalman-Krylov KKL method (denoted by "Krylov") in Fig. 5 a), b). Fig. 6 a) shows simulation results for 3G LTE Pedestrian B channel (having continuous ISI and some significant multipath clusters, with TV component $\lambda = 0.988/0.995$ incorporated here), illustrating the novel MSKF performs well for such practical channels as well; see Sec. III. A. 1) as to how the MSKF adapts to such general cases. Fig. 6 b) simulates a $J = 50, \bar{S} = 1$, SNR = 10 dB system, where MSKF performs well, similar to the very recent SBL [16]. Fig. 7 shows results for large scale (large $\mathbf{R}$), a) $J = 20, K = 11$ and b) $J = 28, K = 14$, MMIMO systems. Fig. 7 a) shows that MSKF still converges within 14 iterations *even* for this large scale MMIMO systems, i. e., MSKF exhibits almost same convergence speed, (though with a larger NRMSE for a more loaded system), as that for ($J = 50, K = 1$ system in Fig 3). Though it uses a symbol-iterative Kalman filter, recent (sparse PCA based) SBL [16] requires many more iterations. For e. g., $J = 28, K = 14$, at 10 dB SNR, MSKF still converges fast in < 14 iterations; KKL's performance is inferior, (with > 40 iterations). But SBL [16] performs very poorly and does not even converge, even with number of iterations increased to more than 100 (for each symbol n), at both SNR of 10 dB, and higher 35 dB SNR.

This phenomenon of convergence speed of our novel TV MSKF being unaffected, even as the system is scaled up to a large ratio $\mathbf{R}$, has been also witnessed in time-invariant (ITV) MSWF [34], (Fig. 4). [34] also shows that PCA based methods do not scale up well. This unique property makes the novel MSKF ideally suited for TV, large-scale

MMIMO systems. Finally, all sparse estimation methods are compared against Bayesian Cramer Rao bounds (BCRB) for $J = 50, K = 1$ stationary single path and Example I multipath channel, for different filter length M's. As in [39], there is substantial performance gap between the practical algorithms and BCR bounds. To illustrate how the novel MSKF exploits the spatial sparsity only, (by isolating it from the time/lag sparsity), its NRMSE, for single-path TV channel ($\lambda = 0.988$), is also shown in Fig. 8 b), though this effect will be more clearly exhibited when the MSKF filter is integrated into a beam-formed mmwave system.

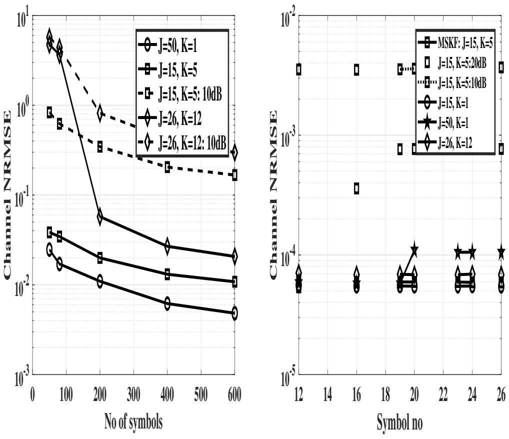

Fig. 3

A) OMP CONVERGENCE, AND B) MSKF FOR STATIONARY CHANNEL

.

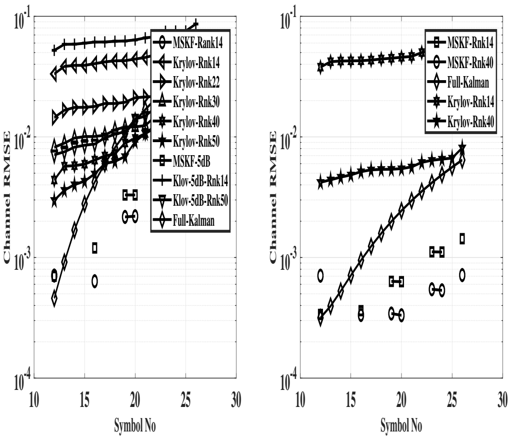

Fig. 4

A) CHANNEL NRMSE VERSUS SYMBOL NO., FOR DIFFERENT ORDERS AND SNRS, FOR $\lambda = 0.988$, B) $\lambda = 0.995$.

VIII. CONCLUSIONS

The paper develops TV, sparse data/channel estimation algorithms in MMIMO, using a novel non-uniformly spaced TV equalizer, which transforms channel/data estimation problem into one of reduced-rank filtering. This is enabled by a novel reduced-rank Multi-Stage Kalman Filter (MSKF).

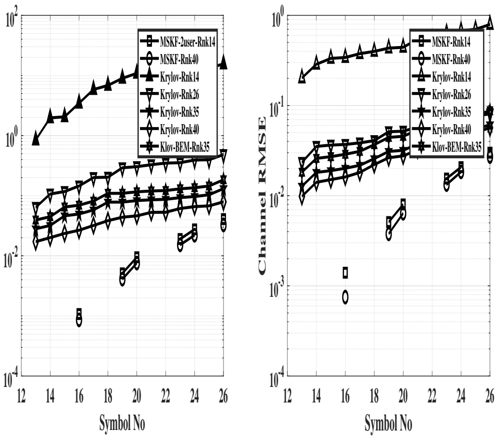

Fig. 5

A) 2 USER CHANNEL NRMSE VERSUS SYMBOL NO., B) CLUSTERED SPARSE CHANNEL NRMSE, $\lambda = 0.988$, FOR DIFFERENT ORDERS.

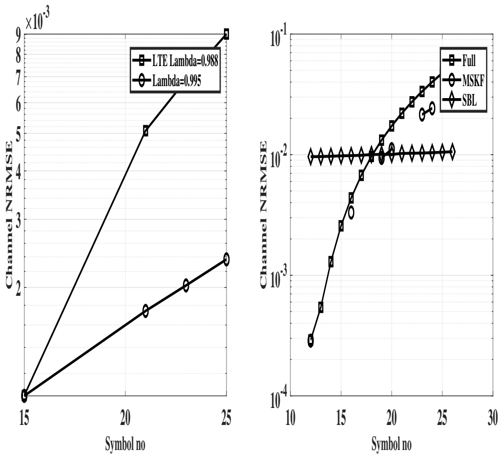

Fig. 6

CHANNEL NRMSE VERSUS SYMBOL NO, A) PRACTICAL 3G LTE CHANNEL, $\lambda = 0.988$, $\lambda = 0.995$, B) COMPARISON OF MSKF WITH SBL, J = 50, K =1, SNR = 10DB, 14 ITERATIONS/STAGES.

MSKF is obtained by substantial extension of a time-invariant (ITV) Multi-Stage Wiener Filter (MSWF) (seen to perform admirably for some DOA applications) to the TV case, by using suitable state estimation techniques. It is to be noted that the overall MSKF algorithm is non-linear, because of the thresholding involved in Algorithm I. MSKF converges very quickly, within few iterations, because MSKF uses those signal spaces maximally correlated with the desired signal, unlike most existing PCA based KKL, re-weighted CS (RCS) [20], rank minimization [18] and sparsity promoting Bayesian estimators [15], [4], [16], with much reduced calculation load. Moreover, MSKF also reduces channel tracking errors, encountered by a standard Kalman filter, in a high mobility TV channel. A key advantage of novel MSKF is its ability to scale up to large-scale MMIMO systems, with very rapid convergence, in contrast to most existing sparse methods. The paper derives a multi-stage version of the omnipresent Kalman filter, which can be extended to TV 5G mmwave communications, by appropriate

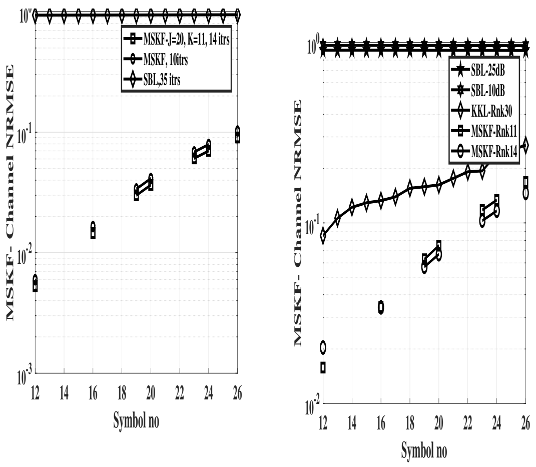

Fig. 7

A) LARGE SCALE SYSTEMS, CHANNEL NRMSE VERSUS SYMBOL NO, $J = 20, K = 11$, SNR = 10 DB, B) $J = 28, K = 14$, MSKF WITH < 14 ITERATIONS, SNR = 10 DB ; SBL: 100 ITERATIONS, SNR = 10 AND 35DB.

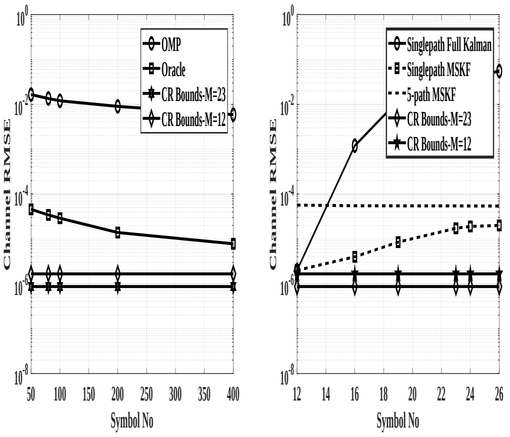

Fig. 8

A) OMP, ORACLE, BAYESIAN CR BOUNDS ($M = 23, 12$), 30 DB, 5 PATH STATIONARY CHANNEL, B) MSKF : 30 DB, CR BOUNDS, 5 PATH STATIONARY CHANNEL, ALSO SINGLE-PATH TV CHANNEL, $\lambda = 0.988$, ILLUSTRATING SPATIAL SPARSITY ONLY.

inclusion of angular estimation [17], [4], and beamforming vectors. Moreover, Bayesian Cramer Rao bounds (BCRB), for noisy CS methods, is derived; novel MSKF and existing CS algorithms.

IX. APPENDIX A : DERIVATION OF *Lemma 1*, EQUATION (11)

Proof: From (56), the reduced data vector $\mathbf{y}_{red}(n) =$ $\mathbf{y}^T(n), \mathbf{y}^T(n-m_1), \mathbf{y}^T(n-m_2), \mathbf{y}^T(n-m_N)\cdots, \mathbf{y}^T(0)]^T$ can be partitioned into

$$\mathbf{y}_{red}(n) = [\mathbf{y}^T(n)|\mathbf{y}^T_{red}(n-m_1)]^T. \tag{55}$$

This is because $\mathbf{y}_{red}(n-m_1) = [\mathbf{y}^T(n-m_1)\,\mathbf{y}^T(n-2m_1)\,\mathbf{y}^T(n-m_1-m_2)\cdots\mathbf{y}^T(n-m_1-m_N)\cdots\mathbf{y}^T(0)]^T$, and components $\{\mathbf{y}(n-2m_1), \mathbf{y}(n-m_1-m_2), \cdots, \mathbf{y}(n-m_1-m_N)\}$'s, (contained in $\mathbf{y}_{red}(n-m_1)$ above), are also all included in $\mathbf{y}_{red}(n)$, [as lags $\{2m_1, m_1+m_2, \cdots, m_1+m_N, \cdots\}$ are included among the $m_p = \sum_l c_{p,l}t_l$'s, which are are calculated in Algorithm I (Steps 2 and 3), in the novel reduced data vector. As in the classical Kalman filter, the space spanned by $\mathbf{y}_{red}(n)$ is equivalent to $\tilde{\mathbf{y}}_{red}(n) = \left[\tilde{\mathbf{y}}^T(n) \quad \mathbf{y}^T_{red}(n-m_1)\right]^T$. Since $\tilde{\mathbf{y}}(n)$ is uncorrelated with $\mathbf{y}^T_{red}(n-m_1)$,

$$\mathbf{R}_{red} = E\{\tilde{\mathbf{y}}_{red}(n)\tilde{\mathbf{y}}^H_{red}(n)\}] = \begin{bmatrix} E(\tilde{\mathbf{y}}(n)\tilde{\mathbf{y}}^H(n)) & \mathbf{0} \\ \mathbf{0} & E\{\mathbf{y}_{red}(n-m_1)\mathbf{y}^H_{red}(n-m_1)\} \end{bmatrix},$$

from which *Lemma 1* (equation (11) follows easily, since (using Wiener filter),

$$\hat{\mathbf{h}}_{red}(n|n) = [(E(\tilde{\mathbf{y}}(n)\mathbf{h}^H_{red}(n))^T, (E(\mathbf{y}_{red}(n-m_1)\mathbf{h}^H_{red}(n))^T] \cdot(\begin{bmatrix} \{E(\tilde{\mathbf{y}}(n)\tilde{\mathbf{y}}^H(n))\}^{-1} & \mathbf{0} \\ \mathbf{0} & \{E\{\mathbf{y}_{red}(n-m_1)\mathbf{y}^H_{red}(n-m_1)\}\}^{-1} \end{bmatrix}) \begin{bmatrix} \tilde{\mathbf{y}}(n) \\ \mathbf{y}_{red}(n-m_1) \end{bmatrix}.$$

Q. E. D.

TABLE I

ALGORITHM I

Step 1.	From computed noisy $\mathbf{R}(n,k)$, find lags k, at which the Frobenius norm $\lVert\mathbf{R}(n,k)\rVert_F$ is above threshold $\gamma(n)$. $\gamma(n)$ calculated as $\gamma(n) = \frac{1}{N+Z}\sum_{j=0}^{M-1}\lVert\mathbf{R}(n,j)\rVert_F$, [9], with $\lVert.\rVert_F$ being the Frobenius norm. Simulations show this threshold to work very well, even in noisy situations, (see discussion in [11]).
Step 2.	For channel with significant taps at l_j, $j = 0,1,2,\cdots,D-1$, $\mathbf{R}(n,k)$ is non-zero at lags of $k = l_j$, $j = 0,1,2,\cdots,D-1$ and at lags of $k = l_j - l_i$, $j = 2,\cdots,D-1$, $1 \le i < j$, for each n. Thus values of auto-correlation lags k_p, determined in Step 1, give us the first few t_p values for $p = 0,1,2,\cdots,D^*$; $(D^* = D - 1 + \binom{D-1}{2})$, where $\binom{D-1}{2}$ is the number of pairs (combinations) of 2 elements from $D-1$ elements), i.e., $t_p = k_p$, $p = 0,1,2,\cdots,D^*$.
Step 3.	Further values of t_p, for $p > D^*$, are obtained from all possible integral combinations of already obtained $t'_p s$, $p = 0,1,2,\cdots,D^*$, (obtained in Step 2), under the constraint that $t_p \le M$, (M : "full " equalizer length), $t_p = \sum_{l=0}^{D^*} c_{p,l}t_l$, $p > D^*$, where $c_{p,l}$ are integer constants.
Step 4.	The values of the multi-channel received signal $\mathbf{y}_i(n-t_p)$'s, thus obtained in Steps 2 and 3, form novel reduced channel equalizer. Then $\mathbf{y}_{red}(n) \overset{\Delta}{=} [\mathbf{y}^T(n-m_1)\,\mathbf{y}^T(n-m_2)\cdots\mathbf{y}^T(n-m_N)]^T,\ m_N \le M. \qquad (56)$

REFERENCES

[1] F. K. J. Lee and P. J. McLane, "Design of Non-uniformly Spaced Tapped-Delay-Line Equalizers for Sparse Multipath Channels, " *IEEE Trans. Commun.*, vol. 52, no. 5, pp. 530-535, Apr 2005.

[2] P. De et. al., "A calculation-efficient algorithm for decision-feedback equalizers," *IEEE Trans. Consumer Electr.*, vol. 45, no. 3, pp. 526-532, Aug.. 1999.

[3] A. Radosevic et. al., "Adaptive OFDM modulation for underwater acoustic communications: design considerations and experimental results," *IEEE Journal of Oceanic Engg.*, vol. 39, no. 2, pp. 357-370, Apr. 2014.

[4] J. Ma, S. Zhang, H. Li, F. Gao, S. Jin, "Sparse Bayesian Learning for the Time-varying Massive MIMO Channels: Acquisition and Tracking," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 1925-1938, March 2019.

[5] S.Haykin, "Adaptive filter theory," Prentice Hall, 2004.

[6] P. De et. al., "Linear Prediction Based Semiblind Channel Estimation for Multiuser OFDM with insufficient guard interval," *IEEE Trans. Wireless Commun.,* vol. 8, no. 12, pp. 5728-5737, Dec. 2009.

TABLE II

Vector Time-invariant (ITV) MSWF

Step 1. 1. Define (0 th stage) normalized cross-correlation:

$$\mathbf{C}_1 = \tilde{\mathbf{R}}_{\tilde{\mathbf{y}}(n),\tilde{\mathbf{H}}(n|\tilde{\mathbf{y}}(n))} \overset{\Delta}{=} E\{\tilde{\mathbf{y}}(n)\tilde{\mathbf{H}}^H(n|\tilde{\mathbf{y}}(n))\}[\boldsymbol{\Delta}_1]^{-1},$$
$$\boldsymbol{\Delta}_1 \overset{\Delta}{=} [\mathbf{R}^H_{\tilde{\mathbf{y}}(n),\tilde{\mathbf{H}}(n|\tilde{\mathbf{y}}(n))}\mathbf{R}_{\tilde{\mathbf{y}}(n),\tilde{\mathbf{H}}(n|\tilde{\mathbf{y}}(n))}]^{1/2}. \quad (57)$$

2. Define blocking matrix as $\mathbf{B}_1 = [\mathbf{I} - \mathbf{C}_1\mathbf{C}_1^H]$.

Step 2 Innovations data $\tilde{\mathbf{y}}(n)$ fed into a filter $\mathbf{T}_1 \overset{\Delta}{=} [\mathbf{C}_1,\ \mathbf{B}_1^H(n)]^H$ to produce 1st order $\mathbf{d}_1(n)$ (on upper branch in inner loop) and data $\tilde{\mathbf{y}}_1(n)$ (on lower branch in inner loop),

$$\begin{aligned}\mathbf{z}_1(n) = \mathbf{T}_1\tilde{\mathbf{y}}(n) &= [(\mathbf{C}_1^H\tilde{\mathbf{y}}(n))^H\ (\mathbf{B}_1(n)\tilde{\mathbf{y}}(n))^H]^H \\ &= [\mathbf{d}_1^H(n)\ \tilde{\mathbf{y}}_1^H(n)]^H. \end{aligned} \quad (58)$$

Step 3 Wiener filter, for estimating (0th order) desired signal $\mathbf{d}_0(n) = \tilde{\mathbf{H}}(n|\tilde{\mathbf{y}}(n))$, (in outer loop), using data $\mathbf{z}_1(n)$ (in inner loop)

$$\begin{aligned}\mathbf{w}_{\mathbf{z}_1} = (\mathbf{R}_{\mathbf{z}_1})^{-1}(\mathbf{R}_{\mathbf{z}_1,\mathbf{H}^e(n|n-m_1)}) = [\mathbf{I} - (\mathbf{R}^{-1}_{\tilde{\mathbf{y}}_1}\mathbf{R}_{\tilde{\mathbf{y}}_1,\mathbf{d}_1}]^T \\ (\mathbf{E}^{-1}\boldsymbol{\Delta}_1))^T = (\mathbf{E}^{-1}\boldsymbol{\Delta}_1)^T[\mathbf{I}_{DJ} - \mathbf{w}_2^T]^T, \end{aligned} \quad (59)$$

where $\mathbf{E} \overset{\Delta}{=} (\boldsymbol{\sigma}^2_{\mathbf{d}_1^2} - \mathbf{R}^H_{\tilde{\mathbf{y}},\mathbf{d}_1}\mathbf{R}^{-1}_{\tilde{\mathbf{y}}_1}\mathbf{R}_{\tilde{\mathbf{y}},\mathbf{d}_1})$ and $\mathbf{w}_2 = \mathbf{R}^{-1}_{\tilde{\mathbf{y}}_1}\mathbf{R}_{\tilde{\mathbf{y}}_1,\mathbf{d}_1}$ are Wiener tap weights for estimating 1st order $\mathbf{d}_1(n)$ from 1st order data $\tilde{\mathbf{y}}_1(n)$.

Step 4. The estimation error, after this stage, is

$$\boldsymbol{\epsilon}_1(n) = [\mathbf{d}_1(n) - \hat{\mathbf{d}}_1(n)] = \mathbf{d}_1(n) - \mathbf{w}_2^H\tilde{\mathbf{y}}_1(n). \quad (60)$$

Step 5. Next, error $\boldsymbol{\epsilon}_1(n)$ used to estimate 0th order $\mathbf{d}_0(n) = \hat{\mathbf{H}}(n|\tilde{\mathbf{y}}(n))$, using Wiener weights $\tilde{\mathbf{w}}_1$,

$$\tilde{\mathbf{w}}_1 = \mathbf{R}^{-1}_{e_1}\boldsymbol{\Delta}_1 = \mathbf{E}^{-1}\boldsymbol{\Delta}_1,\ \Longrightarrow\ \mathbf{w}_{\mathbf{z}_1} = [[\mathbf{I}^H_{DJ}\quad -\mathbf{w}_2^H]]^H\tilde{\mathbf{w}}_1, \quad (61)$$

where $\mathbf{w}_{\mathbf{z}_1}$ is resultant filter of (59).

Step 6.

$$\begin{aligned}\hat{\mathbf{d}}_0(n) = \mathbf{w}^H_{\mathbf{z}_1}\mathbf{z}_1(n) &= \tilde{\mathbf{w}}_1^H[[\mathbf{I}_{DJ}\quad -\mathbf{w}_2^H]]\left[\mathbf{d}_1^T(n)\quad \tilde{\mathbf{y}}_1^T(n)\right]^T \\ &= \tilde{\mathbf{w}}_1^H[\mathbf{d}_1(n) - \mathbf{w}_2^H\tilde{\mathbf{y}}_1(n)] = \tilde{\mathbf{w}}_1^H\boldsymbol{\epsilon}_1(n). \end{aligned} \quad (62)$$

Filter $\mathbf{w}_{\mathbf{z}_1}$'s output estimates 0th order desired signal $\hat{\mathbf{d}}_0(n)$.

TABLE III

MSKF

Step 1. 0th order cross-correlation, between data and desired signal , $\mathbf{R}_{\tilde{\mathbf{y}}(n),\mathbf{H}^e(n|n-m_1)}$, and its normalized cross-correlation $\mathbf{C}_1$ computed:

$$\begin{aligned}\mathbf{R}_{\tilde{\mathbf{y}}(n),\mathbf{H}^e(n|n-m_1)} &\overset{\Delta}{=} E\{\tilde{\mathbf{y}}(n)\mathbf{H}^{eH}(n|n-m_1)\}, \\ \boldsymbol{\Delta}_1 &= [\mathbf{R}^H_{\tilde{\mathbf{y}}(n),\mathbf{H}^e(n|n-m_1)}\mathbf{R}_{\tilde{\mathbf{y}}(n),\mathbf{H}^e(n|n-m_1)}]^{1/2}, \\ \mathbf{C}_1 &= [\mathbf{C}(n)\mathbf{R}_{\mathbf{H}^e}(n|n-m_1)][\boldsymbol{\Delta}_1]^{-1}. \end{aligned}$$

$\mathbf{C}(n)$: Measurement matrix in Kalman state-space representation.

Step 2. 0th innovations data (input to top-correlation level filter), and its correlation matrix: $\tilde{\mathbf{y}}(n) = \mathbf{y}(n) - \hat{\mathbf{y}}(n|n-m_1) = \mathbf{C}(n)\mathbf{H}^e(n|n-m_1) + \mathbf{w}(n)$, $\mathbf{R}_{\tilde{\mathbf{y}}(n)} = \mathbf{C}(n)\mathbf{R}_{\mathbf{H}^e}(n|n-m_1)\mathbf{C}^H(n) + \sigma_w^2\mathbf{I}_J$.

Step 3. Define $\mathbf{B}_1 = [\mathbf{I} - \mathbf{C}_1\mathbf{C}_1^H]$. Innovations data $\tilde{\mathbf{y}}(n)$ fed into a filter $\mathbf{T}_1 = [\mathbf{C}_1,\ \mathbf{B}_1^H(n)]^H$, to generate 1st order desired signal $\mathbf{d}_1(n)$, and 1st order data, $\tilde{\mathbf{y}}_1(n)$ by $\mathbf{d}_1(n) = \mathbf{C}_1^H\tilde{\mathbf{y}}(n)$, $\tilde{\mathbf{y}}_1(n) = \mathbf{B}_1\tilde{\mathbf{y}}(n)$.

Step 4. Normalized cross-correlation (between 1st order $\mathbf{d}_1(n)$ and $\tilde{\mathbf{y}}_1(n)$) $\mathbf{C}_2 = \tilde{\mathbf{R}}_{\tilde{\mathbf{y}}_1(n),\mathbf{d}_1(n)} = E\{\tilde{\mathbf{y}}_1(n)\mathbf{d}_1^H(n)\}[\boldsymbol{\Delta}_2]^{-1} = \mathbf{B}_1\mathbf{R}_{\tilde{\mathbf{y}}}(n)\mathbf{C}_1[\boldsymbol{\Delta}_2]^{-1}$.

Step 5. For generic ith order, (updated from $(i-1)$th order/stage),

$$\begin{aligned}\tilde{\mathbf{y}}_i(n) &= \mathbf{B}_i\tilde{\mathbf{y}}_{i-1}(n),\ \mathbf{d}_i(n) = \mathbf{C}_i^H\tilde{\mathbf{y}}_{i-1}(n), \\ \mathbf{R}_{\tilde{\mathbf{y}}_i(n),\mathbf{d}_i(n)} &= \mathbf{B}_i\mathbf{R}_{\tilde{\mathbf{y}}_{i-1}(n)}\mathbf{C}_i, \\ \boldsymbol{\Delta}_{i+1} &= (\mathbf{R}^H_{\tilde{\mathbf{y}}_i(n),\mathbf{d}_i(n)}\mathbf{R}_{\tilde{\mathbf{y}}_i(n),\mathbf{d}_i(n)})^{1/2} \\ \mathbf{C}_{i+1} \overset{\Delta}{=} \tilde{\mathbf{R}}_{\tilde{\mathbf{y}}_i(n),\mathbf{d}_i(n)} &= \mathbf{R}_{\tilde{\mathbf{y}}_i(n),\mathbf{d}_i(n)}[\boldsymbol{\Delta}_{i+1}]^{-1}, \\ \mathbf{R}_{\tilde{\mathbf{y}}_i(n)} &= (\prod_{j=1}^{j=i}\mathbf{B}_j)\mathbf{R}_{\tilde{\mathbf{y}}(n)}(\prod_{j=1}^{j=i}\mathbf{B}_j). \end{aligned}$$

Step 6. Initialization:

$$\begin{aligned}\mathbf{E}_N &= \mathbf{C}_N^H\mathbf{R}_{\tilde{\mathbf{y}}_{N-1}}(n)\mathbf{C}_N, \\ \boldsymbol{\Delta}_N &= \mathbf{B}_{N-1}\mathbf{R}_{\tilde{\mathbf{y}}_{N-2}}(n)\tilde{\mathbf{R}}_{\tilde{\mathbf{y}}_{N-2}(n),\mathbf{d}_{N-2}(n)}. \end{aligned}$$

Down-recursions, for $j = N, N-1, \cdots, 1$:

$$\begin{aligned}\mathbf{w}_j = [\boldsymbol{E}_j]^{-1}\boldsymbol{\Delta}_j,\ \boldsymbol{\epsilon}_{j-1}(n) &= \mathbf{d}_{j-1}(n) - \hat{\mathbf{d}}_{j-1}(n) \\ &= [\mathbf{d}_{j-1}(n) - \mathbf{w}_j^H\boldsymbol{\epsilon}_j(n)], \\ \mathbf{E}_{j-1} = \mathbf{C}^H_{j-1}\mathbf{R}_{\tilde{\mathbf{y}}_{j-2}(n)}&\mathbf{C}_{j-1} - \mathbf{w}_j^H\boldsymbol{\Delta}_j. \end{aligned}$$

Step 7. Time-Updates: $\mathbf{H}^e(n|n) = \mathbf{H}^e(n|n-m_1) - \tilde{\mathbf{w}}_1^H\boldsymbol{\epsilon}_1(n)$.

Step 8. Prediction

$$\begin{aligned}\mathbf{R}_{\mathbf{H}^e(n|n-m_1)} &= diag(\lambda^{m_1})\mathbf{R}_{\mathbf{H}^e(n-m_1|n-m_1)} \\ &\quad diag(\lambda^{m_1})^H + \mathbf{R}_v(n-m_1), \end{aligned}$$

Update $\mathbf{R}_{\mathbf{H}^e}(n|n) = \mathbf{R}_{\mathbf{H}^e}(n|n-m_1) - \tilde{\mathbf{w}}_1^H\mathbf{E}_1(n)\tilde{\mathbf{w}}_1$.

[7] G. Kutz and D. Raphaeli, "Determination of tap positions for sparse equalizers," *IEEE Trans. Comm.*, vol. 55, no. 9, pp. 1712-1724, Sep. 2008.

[8] E. Larsson et.al, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, pp. 186-195, Feb. 2014.

[9] F. Wan et. al., "Semiblind most significant tap detection for sparse channel estimation of OFDM systems," *IEEE Trans. Circuits and Systems I*, vol. 57, no. 3, pp. 703-713 , March 2011.

[10] F. Wan et. al., "Semiblind sparse channel estimation for MIMO-OFDM systems," *IEEE Trans. Veh. Tech.*, vol. 60, no. 6, pp. 2569-2582, July 2011.

[11] P. De, "Semiblind sparse channel estimation using reduced rank filtering," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1418-1433, March 2018.

[12] P. De, "Fast, Reduced Rank Filtering Based Semiblind MIMO OFDM Sparse Channel Estimation", *IEEE Systems Journal*, vol. 15, no. 1, pp. 1036-1048, March 2021.

[13] W. Bajwa et. al., "Compressed channel sensing: a new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, no. 26, pp. 1058-1076, 2010.

[14] R. Prasad et. al., "Joint Approximately Sparse Channel Estimation and Data Detection in MIMO-OFDM systems using sparse Bayesian learning," *IEEE Trans. Sig. Proc.*, vol. 62, no. 14, pp. 3591-3603, Jul. 2014.

[15] M. Massood, L. H. Afify and T. Al-Naffouri, "Efficient Coordinated Recovery of Sparse Channels in Massive MIMO," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 104-118, Jan. 2016.

[16] Christo K. Thomas, and Dirk Slock. "Low complexity static and dynamic sparse bayesian learning combining BP, VB and EP message passing." *53rd Asilomar Conference on Signals, Systems, and Computers*, 2019.

[17] R. Zhang et. al., "Angular-Domain Selective Channel Tracking and Doppler Compensation for High-Mobility mmwave Massive MIMO," arxiv. 1911: 08683v2 [eess.SP] 28 Nov, 2020.

[18] A. Brighente et. al. "Estimation of Wideband Dynamic mmWave and THz Channels for 5G Systems and Beyond, " *IEEE Jnl. on Selec. Areas Comm.*, vol. 38, no. 9, pp. 2026-2040, Sep. 2020.

[19] Y. Liu et. al. "Uplink-Aided High Mobility Downlink Channel Estimation Over Massive MIMO-OTFS System," " *IEEE Jnl. on Selec. Areas Comm.*, vol. 38, no. 9, pp. 1994-2009, Sep. 2020.

[20] H. Soleimani, et. al., "Mm-Wave channel estimation with accelerated gradient descent algorithms, " *Eurasip Jnl on ireless Comm. and Networking,"* 2019: 272, pp. 1-17.

[21] X. Li et. al., "Millimeter Wave Channel Estimation via Exploiting Joint Sparse and Low-Rank Structures," *IEEE Trans. Wireless Comm*, vo,. 17, no. 2, pp. 1123-1133, Feb 2018.

[22] T. Hrycak et. al., "Low Complexity Equalization for Doubly Selective Channels Modeled by a Basis Expansion," *IEEE Trans. Sig. Proc.*, vol. 58, no. 11, pp. 5706-5719, Nov. 2012.

[23] C. Komninakis et. al., "Multi-Input Multi-Output Fading Channel Tracking and Equalization using Kalman Estimation" *IEEE Trans. Signal Process.*, vol. 50, no. 5, pp. 1065-1076, May 2003.

[24] J. S. Goldstein, I. S. Reed and L.L.Scharf, "A multistage representation

TABLE IV
KKL ALGORITHM

Step 1. Following [29], $J \times 1$ $\mathbf{q}_k$ initialized by $\mathbf{q}_0 = \mathbf{R}_{\tilde{\mathbf{y}}(n)}/norm(\mathbf{R}_{\tilde{\mathbf{y}}(n)})$.

Step 2. Next applying Arnoldi recursion, scalar $\alpha_{i,j}$ calculated by

$$\alpha_{i,j} = \mathbf{q}_i^T \mathbf{R}_{\tilde{\mathbf{y}}(n)} \mathbf{q}_j, \tag{63}$$

Step 3. Then

$$\mathbf{h}_j = \mathbf{R}_{\tilde{\mathbf{y}}(n)} \mathbf{q}_j - \sum_{i=1}^{j} \alpha_{i,j} \mathbf{q}_i,\ \beta_{k+1} = \|\mathbf{h}_k\|,\ \mathbf{q}_{k+1} = \frac{\mathbf{h}_k}{\beta_{k+1}}, \tag{64}$$

Step 4. From [29],

$$[\mathbf{q}_1\, \mathbf{q}_2\, \cdots, \mathbf{q}_N]^T \mathbf{R}_{\tilde{\mathbf{y}}(n)} [\mathbf{q}_1\, \mathbf{q}_2\, \cdots, \mathbf{q}_N] =$$
$$\begin{bmatrix} \alpha_{1,1} & \cdots & \cdots & \alpha_{1,N} \\ \beta_2 & \alpha_{2,2} & \cdots & \alpha_{2,N} \\ \ddots & & & \vdots \\ & & \ddots & \alpha_{N,N} \end{bmatrix} \overset{\Delta}{=} \tilde{\mathbf{H}}_N \epsilon \mathbf{R}^{(N,N)}. \tag{65}$$

Step 5. (15) transformed into one involving tridiagonal $\tilde{\mathbf{H}}_N$, whose solution $\mathbf{Z}$ then used to find the optimal weights. For this, QR decomposition of the Hessenberg matrix $\tilde{\mathbf{H}}_N$ computed

$$\tilde{\mathbf{H}}_N = \mathbf{T}_N \mathbf{R}_N, \tag{66}$$

$\mathbf{T}_N$: unitary matrix and $\mathbf{R}_N$: upper triangular matrix.

Step 6. QR decomposition implemented using a numerically robust Householder transformation [37] and Givens rotation; (detailed algorithm in [29]).

Step 7. Triangular system of equations (66) solved to get $\mathbf{Z}$, which then transformed back into optimal weights $\mathbf{w}_{\mathbf{z}_0}(n)$.

Step 8. Then estimate $\hat{\hat{\mathbf{H}}}(n|\tilde{\mathbf{y}}(n)) = \mathbf{w}_{\mathbf{z}_0}^H(n)\tilde{\mathbf{y}}(n)$ Then the channel estimate and $\mathbf{R}_{\mathbf{H}^e}(n)$ updated by (11) and (14), for next symbol.

of the Wiener filter based on orthogonal projections," *IEEE Trans. Info. Theory,* vol. 44, no. 7, pp. 2943-2959, Nov. 1998.

[25] D. Berbedis and G. Giannakis, "Data Sketching for Large-Scale Kalman filtering," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3688-3701, July 15, 2017.

[26] T. Ballal, T. Al-Naffouri and S. Ahmed, "Low-Complexity Bayesian Estimation of Cluster-Sparse Channels," *IEEE Trans Commun.,* vol. 63, no. 11, Nov. 2016.

[27] M. K. Schneider, "Krylov Subspace Estimation, " *MIT Ph. D. thesis*, 2002, (*SIAM Journal on Scientific Computing (SIAM)*, vol, 22, issue 5, published online June 25, 2006).

[28] Y. Saad, "Iterative Methods for Sparse Linear Systems," *Society for Industrial and Applied Mathematics (SIAM)*, 2003.

[29] R. Plato, "Concise Numerical Mathematics," *American Mathematical Society (AMS)*, vo. 57, 2003.

[30] R. K. Martin et. al., "Exploiting sparsity in adaptive filters," *IEEE Trans. Signal Process.*, vol. 50, no. 8, pp. 1883-1894, Aug. 2002.

[31] A. Bishnu et. al., "Sparse channel estimation for interference limited OFDM systems and its convergence analysis," *IEEE Access,* vol. 5, pp. 17781-17794, 2017.

[32] X. Wang. R. Lamare et. al., " Robust Two-Stage Reduced-Dimension Sparsity-Aware STAP for Airborne Radar With Coprime Arrays," *IEEE Trans. Signal Process.*, vol. 68, pp. 81-96, 202

[33] 'P. De, "Technical Report on MultiStage Kalman Filter (MSKF)," Feb 2021.

[34] M. Honig et. al , "Performance of Reduced-Rank Linear Interference Suppression ," *IEEE Trans. Info. Theory,* vol. 47, no. 5, pp. 1928-1945, Jul 2003.

[35] Personal e-mail correspondence, with Dr T. Ballal (KAUST).

[36] Tareq Y. Al-Naffouri et. al., "A Model Reduction Approach for Channel Estimation under High Mobility Conditions," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. , Apr. 2011.

[37] Golub and Van Loan, "Matrix Computations, " Johns Hopkins University Press, 1996.

[38] Tareq Y. Al-Naffouri, "An EM-Based Forward-Backward Kalman Filter for the Estimation of Time-Variant Channels in OFDM," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. , Jul. 2008.

[39] H. Zayyani, M. B-Zadeh and C. Jutten, "Bayesian Cramer-Rao bound for Noisy Non-Blind and Blind Compressed Sensing," *arXiv: 1005.4316v1 [cs.IT],*" May 24, 2010.

.