Distributed Estimation in Wireless Networks using
Local Message Passing

Boon Loong Ng, Jamie S. Evans, Stephen V. Hanly and Alex J. Grant

Abstract

We consider the problem of distributed linear minimum mean square error (LMMSE) signal estimation in wireless networks with interfering nodes. A key structural feature of the network we consider is localised interference, which we exploit by developing distributed algorithms to approximate the global LMMSE receiver. We assume that a receiver node can communicate with neighbouring receiver nodes, allowing these nodes to share information in a cooperative manner, and our algorithms incorporate message passing between receiver nodes. For simplicity, we consider a one-dimensional model of a wireless network, called the linear array model. Two distributed algorithms with local message passing are presented for efficient implementation of the global LMMSE estimator. The first is a forward-backward algorithm based on the Kalman smoothing framework, and the second is based on the first order stationary iterative method used to solve systems of linear equations. In both cases, we provide limited extent, distributed versions that exploit the locality of the intercell interference, and converge to the global receiver at rates that do not grow with the overall array size. We compare the complexity and convergence rates of the two algorithms.

Index Terms

Distributed estimation; localised interference; message passing; iterative method; Kalman smoothing.

B. Ng, J. Evans and S. Hanly are with the ARC Special Research Centre for Ultra-Broadband Information Networks (CUBIN), Department of Electrical and Electronic Engineering, University of Melbourne, VIC 3010, Australia. CUBIN is an affiliated program of National ICT Australia. A. Grant is with the Institute for Telecommunications Research, University of South Australia, Mawson Lakes SA 5095 Australia.

This work was presented in part at the Australian Communications Theory Workshop, Newcastle, Australia, February 2004 and at the IEEE Global Telecommunications Conference, Dallas, Texas, USA, November 2004.

This work was supported by the Australian Research Council.
I. INTRODUCTION

We consider a wireless network architecture as illustrated in Fig. 1, in which each receiver node is associated with a number of transmitter nodes, forming a “cell”. The transmitter nodes send signals to their receiver nodes, but their signals are heard at neighbouring receiver nodes as well. Rather than taking a traditional frequency partitioning approach to eliminate (or drastically reduce) intercell interference, we instead assume that there are dedicated communication links between neighbouring receiver nodes, and allow a cooperative sharing of information between adjacent receiver nodes. The fundamental problem we wish to address in the present paper is how to exploit only local communication between the receiver nodes to implement the global linear minimum mean square error (LMMSE) estimation of the data symbols in a distributed way.

For simplicity, we assume that the transmitter nodes within each cell are orthogonal, either through the deployment of a time division multiple access (TDMA) scheme or a frequency division multiple access (FDMA) scheme. Orthogonality of the transmitter nodes within each cell ensures that at any given time, there is effectively only one transmitter node per cell. A simple model for such a wireless network can be expressed as

\[ y_n = \alpha_{n,n} x_n + \sum_{n' \in A_n} \alpha_{n,n'} x_{n'} + z_n \]  

(1)

where \( y_n \) is the received signal at the receiver node at cell \( n \), \( x_n \) is the transmitted signal from the transmitter node in cell \( n \), \( A_n \) is the set of indices of the cells adjacent to cell \( n \), and \( z_n \) is background noise. \( \alpha_{n,m} \) is the path gain from transmitter node \( m \) to receiver node \( n \), where for \( n \neq m \), it represents the gain of an interferer. We assume that \( \{x_n\} \) are independent and identically distributed (i.i.d.) random variables, with zero mean and variance \( P \), and that \( \{z_n\} \) are i.i.d. Gaussian with zero mean and variance \( \sigma^2 \), independent of the \( \{x_n\} \). All random variables may be complex. The performance measure of interest for signal recovery in this paper is the mean square error (MSE), which is defined by

\[ \text{MSE}(n) = E \left[ (x_n - \hat{x}_n)(x_n - \hat{x}_n)^\ast \right] \]

1These dedicated communication links between receiver nodes do not interfere with each other or with the signals from the transmit nodes.
where $\hat{x}_n$ is the estimate for $x_n$ and the superscript * denotes the complex conjugate. Although the model in (1) is simple, it provides the critical feature of localised interference between cells, and is a useful starting point in investigating distributed signal processing algorithms.

The model (1) has its origin in the wireless cellular model introduced by Wyner in [1], where a receiver node is a base station and a transmitter node is a mobile station. Since communications between neighbouring base stations are allowed, (1) can be viewed as a Gaussian multiple access channel [1], [2]. In wireless cellular networks, information symbols ($x_n$s) are from a discrete alphabet, and the problem of recovering the $x_n$s is therefore a detection problem. In this case, minimising the MSE is, in general, not equivalent to minimising the symbol error rate. In general networks with the above structure, optimisation in the maximum a posteriori (MAP) sense has a complexity that grows exponentially with the number of cells in the network [3]. This complexity is reduced to linear complexity in the special case of linear networks [4], but even then, the complexity grows with the alphabet size and may still be prohibitive in practice. It is therefore of interest to consider linear schemes that minimise the MSE [3].

The model (1) is also useful to describe another kind of wireless network, namely a single hop heterogeneous sensor network with two types of nodes. In this application, a receiver node is a cluster head and a transmitter node is a sensor node. If we assume that the sensor nodes transmit analogue signals, such as temperature or pressure measurements, then the recovery of the $x_n$s is effectively an estimation problem. In many practical cases, the $x_n$s are actually correlated (for instance, temperatures of two spatial points in a building) and the independence of signal sources assumption might not hold. On the other hand, it may be inefficient for the sensors to be correlated, and this is related to the issue of spatial sampling. In the present paper, we retain the assumption of independent sources, and leave more general models incorporating correlation between sources for future work. Note that in these models, the MSE is a very natural metric to consider.

Our main contribution in this paper is the design of iterative message passing algorithms that implement the global LMMSE estimation of the data symbols. We show that iterative message passing algorithms can be constructed based on the graph describing the connectivity of the nodes in the network. In this
case, the nodes in the graph directly correspond to the nodes in the network, or more precisely, to the symbols transmitted or received at these physical nodes.

Interference between nodes is a basic feature of the model, and this results in local coupling; as a result, the global estimator requires information from the whole network. A key question we address is whether good approximations to the global estimator can be obtained using only a limited number of message passings, and hence only local information. The answer is generally “yes”, although it does require a redesign of the message passing algorithm: the revised algorithm has nodes computing and message passing in parallel, rather than in series. The parallel processing algorithm is appropriate if we want estimates that can be obtained in a time independent of the network size: it can be terminated at any time, and provides estimates of increasing accuracy as time progresses.

We focus attention in the present paper on one-dimensional models (similar to Wyner’s linear cellular array model [1]) as this provides the simplest setting to address the above questions in an analytical way. Our message passing algorithms are developed in the framework of Kalman smoothing. It is shown in [5] that the Kalman smoother can be interpreted as a linear combination of two Kalman filters: one which runs in the forward direction on the observed data set and the other in the backward direction. We use the same interpretation in this paper and the result is a distributed forward-backward algorithm. The forward-backward algorithm suffers from a delay that grows linearly with the network size. To overcome this delay problem, we present a parallel processing or limited extent version of the algorithm that provides nearly optimal results with only a limited number of message passing steps.

We also develop a second type of iterative algorithm, called the first order stationary iterative (FOSI) algorithm, which is based on an iterative approach to finding the solution to a system of linear equations. This is possible due to the fact that the global LMMSE estimator is a unique solution to a system of linear equations [6], [7]. While most analyses and discussions of the algorithm in this paper are done in the context of the one-dimensional model, they are also valid for the two-dimensional networks.

The rest of the paper is organised as follows. We present a one-dimensional version of (1), called the linear array, in Section II. The global LMMSE estimator is then presented, followed by some discussions
on implementation issues. In Section III, we develop the Kalman-based distributed algorithms for the linear array. We present a forward-backward algorithm and a limited extent version which we call the KaLiX algorithm. Our second type of algorithm, namely the FOSI algorithm, is presented in Section IV. We compare the performance of the algorithms in terms of convergence rate in Section V. Some analytical results on the convergence properties are also presented. We end with some concluding remarks in Section VI.

The notations used in this paper are as follows. Boldface lower case letters are for vectors, boldface upper case letters are for matrices and $I$ is the identity matrix. The absolute value of a scalar is $|.|$. Finally, We use $\text{tr}(.)$, $(.)^T$, $(.)^H$, $(.)^{-1}$ and $\|.|\|_2$ to denote the trace, the transpose, the complex conjugate transpose, the inverse and the Euclidean norm respectively.

II. LINEAR ARRAY MODEL AND LMMSE ESTIMATOR

A one-dimensional model of (1), termed the linear array model, can be expressed as

$$y_n = \alpha_{n,n-1}x_{n-1} + \alpha_{n,n}x_n + \alpha_{n,n+1}x_{n+1} + z_n, \quad n = 1, \ldots, N.$$ 

where $x_n$ is zero for $n < 1$ or $n > N$ and $N$ is the array size (Fig. 2). An equivalent representation is the following vector form:

$$y = Hx + z$$

where $y = [y_1 \ y_2 \ \ldots \ y_N]^T$, $x = [x_1 \ x_2 \ \ldots \ x_N]^T$, $z = [z_1 \ z_2 \ \ldots \ z_N]^T$. $H$, the channel matrix, is an $N \times N$ tridiagonal matrix, given by

$$H = \begin{bmatrix}
\alpha_{1,1} & \alpha_{1,2} & 0 & 0 & 0 & \ldots & 0 \\
\alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & 0 & 0 & \ldots & 0 \\
0 & \alpha_{3,2} & \alpha_{3,3} & \alpha_{3,4} & 0 & \ldots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & \ldots & \alpha_{N-1,N-2} & \alpha_{N-1,N-1} & \alpha_{N-1,N} \\
0 & 0 & 0 & \ldots & 0 & \alpha_{N,N-1} & \alpha_{N,N}
\end{bmatrix} = [h_1 \ h_2 \ \ldots \ h_N]$$
where \( h_1, h_2, \ldots, h_N \) are the column vectors of \( H \). The band diagonal structure of the channel matrix, \( H \), provides a “local interaction” structure to the estimation problem, which we wish to exploit.

Given full channel state information and prior knowledge of the mean and the covariance of \( x \), the LMMSE estimator of \( x \), given \( y \), can be expressed [8] as

\[
\hat{x} = H^H (HH^H + \sigma_0^2 I)^{-1} y
\]

where \( \sigma_0^2 \equiv \sigma^2/P \), i.e. the inverse of signal-to-noise ratio (SNR). Hence, the LMMSE estimator of \( x_n \) is given by

\[
\hat{x}_n = h_n^H (HH^H + \sigma_0^2 I)^{-1} y, \quad n = 1, \ldots, N.
\]

For the rest of this paper, we shall refer to the global LMMSE estimate (2) as the optimal estimate, as it is optimal in the class of linear estimation schemes.

The implementation of (2) is the subject of interest in this paper. One obvious way to implement (2) is through a centralised scheme in which each receiver node in the network is wire-line connected to a central processor. Our interest in this paper however, is on fully decentralised schemes where (2) is implemented through distributed signal processing and message passing.

Equation (3) suggests that to obtain an optimal estimate of \( x_n \), we would need \( \{y_1, \ldots, y_N\} \), i.e. all information available in the network. However, due to the highly localised intercell interference, as exhibited in the tridiagonal structure of the channel matrix, one expects the importance of the information contributed from a receiver node to the estimation of \( x_n \) to reduce as the distance between them increases.

Consider the problem of estimating \( x_n \) based on \( \{y_{n-j}, \ldots, y_{n+j}\} \), where \( n-j \geq 1 \) and \( n+j \leq N \). We call the receiver nodes \( n-j, \ldots, n+j \) the limited extent set associated with obtaining the estimate \( \hat{x}_n \) (Fig. 3). What can we say about the MSE of \( x_n \) based on the limited extent set as \( j \) increases? Let \( \text{MSE}_{\text{opt}} \) be the MMSE for a transmitter node located in the middle of an array and \( \text{MSE}_{\text{LIX}} \) be the MSE of the middle transmitter node based on a limited extent set. Fig. 4 shows the plots of the ratio of \( \text{MSE}_{\text{LIX}} \) to \( \text{MSE}_{\text{opt}} \) vs the size of the limited extent set of receiver nodes used in LMMSE estimation, averaged over 100000 channel realisations, where \( \alpha_{n,m} \) is assumed to be Rayleigh random variable with \( \mathbb{E}(\alpha_{n,n}) = 1 \) and \( \mathbb{E}(\alpha_{n,m}) = \alpha \) for \( n \neq m \). We assume an array of 31 cells. The graphs in Fig. 4 suggest, not surprisingly,
that in general there are diminishing returns from increasing the limited extent size beyond a certain finite size that depends on the channel parameters. It can be seen that in certain cases, the number of receiver nodes needed is actually very small, except for the interference dominant case where $\alpha$ is significant (e.g. $\alpha > 0.6$) and $\sigma_0^2$ is small. This observation suggests that only a small subset of $y$ is needed for near-optimal performance of the LMMSE estimator and serves as a motivation for developing distributed algorithms that approximate the LMMSE solution by utilising only local information.

For the rest of the paper, we shall demonstrate how the implementation of the global LMMSE estimator (2) can be decentralised, such that only communications between adjacent cells and low-complexity computation for each receiver node are required.

III. Kalman-based Distributed Algorithms

On the linear array, the problem of estimating $x_n$ based on $y$ can be formulated as a Kalman smoothing problem. In this section, we first review preliminary results on Kalman smoothing and then we present a forward-backward algorithm based on the Kalman smoothing framework that implements (3) exactly, albeit at a cost of delay that grows linearly with the size of the array. We then show how the delay problem can be overcome with an approximate distributed algorithm by exploiting the locality of intercell interference, with little loss in performance.

A. Kalman Smoothing

Fraser and Potter [5] showed that the Kalman smoother can be interpreted as a linear combination of two Kalman filters; one which runs in the forward direction on the data set $\{y_1, \ldots, y_n\}$ and the other in the backward direction on $\{y_{n+1}, \ldots, y_N\}$.

Define the state as

$$v_n = [x_{n-1} \ x_n \ x_{n+1}]^T, \quad n = 1, \ldots, N.$$  

Treating the index $n$ as time, the structure of the channel matrix $H$ allows us to model the evolution of $v_n$ according to a state space model, given by

$$v_{n+1} = A^f v_n + b^f x_{n+2}, \quad n = 1, \ldots, N - 1.$$
where

\[ A^f = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad b^f = [0 \ 0 \ 1]^T. \]

One can also write down the state space model in the ‘backward’ direction.

\[ v_{n-1} = A^b v_n + b^b x_{n-2}, \quad n = 2, \ldots, N \]

where

\[ A^b = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad b^b = [1 \ 0 \ 0]^T. \]

The observation model is

\[ y_n = h_n v_n + z_n, \quad n = 1, \ldots, N \]

where

\[ h_1 = [0 \ \alpha_{1,1} \ \alpha_{1,2}]; \quad h_N = [\alpha_{N,N-1} \ \alpha_{N,N} \ 0]; \quad h_n = [\alpha_{n,n-1} \ \alpha_{n,n} \ \alpha_{n,n+1}], \quad i = 2, \ldots, N - 1. \]

1) Forward and Backward Kalman Filters: We denote \( \hat{v}_{i|\{j,\ldots,k\}} \) as the estimate of state \( v_i \) based on \( \{y_j, \ldots, y_k\} \) and \( M_{i|\{j,\ldots,k\}} \) as the covariance of \( \hat{v}_{i|\{j,\ldots,k\}} \). Given the state and observation model, the forward Kalman filter is given by the following set of equations [8]:

**Forward Prediction:**

\[ \hat{v}_{n|\{1,\ldots,n-1\}} = A^f \hat{v}_{n-1|\{1,\ldots,n-1\}} \]

\[ M_{n|\{1,\ldots,n-1\}} = A^f M_{n-1|\{1,\ldots,n-1\}} A^{fT} + b^f P_{n+1} b^{fT} \]

**Forward Correction:**

\[ k_n^f = \frac{M_{n|\{1,\ldots,n-1\}} h_n^H}{\sigma^2 + h_n M_{n|\{1,\ldots,n-1\}} h_n^H} \]

\[ \hat{v}_{n|\{1,\ldots,n\}} = \hat{v}_{n|\{1,\ldots,n-1\}} + k_n^f (y_n - h_n \hat{v}_{n|\{1,\ldots,n-1\}}) \]

\[ M_{n|\{1,\ldots,n\}} = \left( I - k_n^f h_n \right) M_{n|\{1,\ldots,n-1\}} \]
where \( n = 2, \ldots, N \) and \( k_n^f \) is the so-called Kalman gain. \( P_n \) is the variance or power of \( x_n \), which is set zero for \( n < 1 \) or \( n > N \), and a constant, \( P \), otherwise. The iteration is initialised by \( \hat{v}_{1|1} \), which is just the estimate of \( v_1 \) given \( y_1 \) only, and its corresponding MSE is \( M_{1|1} \).

For general \( n \), \( \hat{v}_{n|n} \) and \( M_{n|n} \) can be written down in a straightforward manner as

\[
\hat{v}_{n|n} = \tilde{h}_n^H \left( \tilde{h}_n \tilde{h}_n^H + \sigma_0^2 \right)^{-1} y_n. \tag{9}
\]

The corresponding MSE, a \( 3 \times 3 \) matrix, is

\[
M_{n|n} = P_n \left( I_n - \tilde{h}_n^H \left( \tilde{h}_n \tilde{h}_n^H + \sigma_0^2 \right)^{-1} \tilde{h}_n \right) \tag{10}
\]

where

\[
\tilde{I}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \tilde{I}_n = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad n = 2, \ldots, N - 1.
\]

Similarly, the backward Kalman filter is given by the following set of equations:

**Backward Prediction:**

\[
\hat{v}_{n|\{n+1,\ldots,N\}} = A^b \hat{v}_{n+1|\{n+1,\ldots,N\}} \tag{11}
\]

\[
M_{n|\{n+1,\ldots,N\}} = A^b M_{n+1|\{n+1,\ldots,N\}} A^{bT} + b^b P_{n-1} b^{bT} \tag{12}
\]

**Backward Correction:**

\[
k_n^b = \frac{M_{n|\{n+1,\ldots,N\}} \tilde{h}_n^H}{\sigma^2 + \tilde{h}_n M_{n|\{n+1,\ldots,N\}} \tilde{h}_n^H} \tag{13}
\]

\[
\hat{v}_{n|\{n,\ldots,N\}} = \hat{v}_{n|\{n+1,\ldots,N\}} + k_n^b (y_n - \tilde{h}_n \hat{v}_{n|\{n+1,\ldots,N\}}) \tag{14}
\]

\[
M_{n|\{n,\ldots,N\}} = (I - k_n^b \tilde{h}_n) M_{n|\{n+1,\ldots,N\}} \tag{15}
\]

where \( n = 1, \ldots, N - 1 \) and \( k_n^b \) denotes the Kalman gain in the backward direction. The iteration is initialised by \( \hat{v}_{N\{N\}} \) and \( M_{N\{N\}} \), which can be calculated with (9) and (10).
2) **Optimal Linear Combination of Forward-Backward Kalman Estimates:** The LMMSE estimate of a state, $\hat{v}_n$, can be written as a linear combination of the estimate, $\hat{v}_n|\{1\ldots,n\}$, obtained by forward Kalman filtering and the estimate, $\hat{v}_n|\{n+1\ldots,N\}$, obtained by backward Kalman filtering. Note that $\hat{v}_n|\{1\ldots,n\}$ is a ‘correction’ while $\hat{v}_n|\{n+1\ldots,N\}$ is actually a ‘prediction’. Of course, one can also use $\hat{v}_n|\{1\ldots,n-1\}$ (prediction) for the forward filter and $\hat{v}_n|\{n\ldots,N\}$ (correction) for the backward filter; the choice is arbitrary. It is not difficult to show (see Appendix) that

$$\hat{v}_n = M_n \left(M_n^{-1}|\{1\ldots,n\} \hat{v}_n|\{1\ldots,n\} + M_n^{-1}|\{n+1\ldots,N\} \hat{v}_n|\{n+1\ldots,N\}\right)$$

(16)

$$M_n = \left(M_n^{-1}|\{1\ldots,n\} + M_n^{-1}|\{n+1\ldots,N\} - \frac{1}{P} \tilde{I}_n\right)^{-1}.$$  

(17)

Alternatively, one can also express $\hat{v}_n$ as a linear combination of $\hat{v}_n|\{n\}$, $\hat{v}_n|\{1\ldots,n-1\}$ and $\hat{v}_n|\{n+1\ldots,N\}$:

$$\hat{v}_n = M_n \left(M_n^{-1}|\{1\ldots,n-1\} \hat{v}_n|\{1\ldots,n-1\} + M_n^{-1}|\{n+1\ldots,N\} \hat{v}_n|\{n+1\ldots,N\} + M_n^{-1}|\{n\} \hat{v}_n|\{n\}\right)$$

(18)

$$M_n = \left(M_n^{-1}|\{1\ldots,n-1\} + M_n^{-1}|\{n+1\ldots,N\} + M_n^{-1}|\{n\} - \frac{2}{P} \tilde{I}_n\right)^{-1}.$$  

(19)

Now, only ‘predictions’ are obtained from both the forward and backward filters. Information extracted from $y_n$ is explicitly separated from the forward and backward filters.

**B. Optimal Kalman-based Distributed Algorithm**

The Kalman smoothing algorithm presented in the previous section allows the LMMSE estimate of a particular transmitter node’s signal, $x_n$, given $y$, to be computed in a distributed manner. If we associate $v_n$ as the state of interest for the receiver node in cell $n$, then each receiver node receives a prediction from its direct neighbour on its own state, incorporates the information from its own received signal to produce an estimate of its own state (correction) and then passes it on to its other neighbour in a form of prediction on that neighbour’s state. The passing of information operates in both directions. After information in both directions is obtained, the LMMSE estimate of $v_n$ can be computed according to (16) and $\hat{x}_n$ can then be extracted from $\hat{v}_n$.

The Kalman-based distributed algorithm is summarised as follows:

1) $\tilde{h}_n$ and $\sigma^2$ are estimated by each receiver node $n$;
2) At the receiver nodes located at both ends of the linear array, $\hat{v}_{1\{1\}}$ and $\hat{v}_{N\{N\}}$ are calculated, given $y_1$ and $y_N$, according to (9). $M_{1\{1\}}$ and $M_{N\{N\}}$ are calculated, according to (10);

3) $\hat{x}_{1\{1\}}$ and $\hat{x}_{2\{1\}}$ (second and third element of $\hat{v}_{1\{1\}}$) as well as the corresponding covariances from $M_{1\{1\}}$ are passed from receiver node 1 to receiver node 2, while $\hat{x}_{N-1\{N\}}$ and $\hat{x}_N\{N\}$ (first and second element of $\hat{v}_{N\{N\}}$) and the corresponding covariances from $M_{N\{N\}}$ are passed from receiver node $N$ to receiver node $N-1$. This process effectively realises equations (4), (5), (11) and (12) for $n = 2$ and $n = N - 1$ respectively;

4) At receiver node 2 and receiver node $N - 1$, the forward correction equations: (6), (7), (8), and the backward correction equations: (13), (14), (15), are calculated respectively, given $y_2$ and $y_{N-1}$.

$\hat{v}_{2\{1,2\}}$ and $M_{1\{1,2\}}$ are stored at receiver node 2, while $\hat{v}_{N-1\{N\}}$ and $M_{N-1\{N-1,N\}}$ are stored at receiver node $N - 1$;

5) Step 3 and Step 4 are repeated until each receiver node in the network has determined its own $\hat{v}_n\{1,...,n\}$, $\hat{v}_n\{n+1,...,N\}$, $M_n\{1,...,n\}$ and $M_n\{n+1,...,N\}$;

6) $\hat{v}_n$ and $M_n$ are computed according to (16) and (17) for all $n$.

An important remark is that the distributed algorithm outlined above is sensitive to the size of the network. The time required to propagate information to cover the entire network increases linearly with the size of the array. The delay experienced by each receiver node is also dependent on the location of the receiver node in the array. Specifically, receiver nodes at the edge of the array suffer longer delay than the receiver nodes in the middle of the array.

It is also worth noting that $k_n^f$, $k_n^b$ and $M_{i\{j,...,k\}}$ do not depend on the observation data. This implies that in a slow fading environment where the channel may be approximated as static over multiple consecutive symbols, covariance information can be reused and need not be passed between receiver nodes.

A distributed algorithm is of little use if the convergence is too slow. The speed of convergence is particularly crucial for delay sensitive applications, such as voice in a wireless cellular network or control information in a sensor network. It is also of interest from a practical point of view to ensure that the operation of the receiver and its performance is insensitive to the size of the network. We address this
issue in the following section by considering a different implementation of the Kalman message passing algorithm that enables it to be stopped before the global LMMSE estimate is reached, allowing the delay to be constant.

C. Kalman-based Limited Extent Algorithm (KaLiX)

The fact that only a small subset of y may be needed for near-optimal performance of the LMMSE estimator, as shown previously in Fig. 4, gives rise to the limited extent distributed algorithm which we refer to as the KaLiX algorithm. Full operations summarised below are described from the point of view of the receiver node in cell n. Again, note that the covariances need not be recalculated or passed for each symbol estimation if the channel remains fixed.

1) \( \hat{h}_n \) and \( \sigma^2 \) are estimated;
2) \( \hat{v}_{n|n} \) and \( M_{n|n} \) are calculated based on (9) and (10), which are stored;
3) \( \hat{x}_{n|n} \), \( \hat{x}_{n-1|n} \) and the corresponding covariances from \( M_{n|n} \) are sent to the receiver node to the left, while \( \hat{x}_{n|n} \), \( \hat{x}_{n+1|n} \) and the corresponding covariances, also from \( M_{n|n} \), are sent to the receiver node to the right. At the same time, \( \hat{x}_{n|n-1} \) and \( \hat{x}_{n-1|n-1} \) are received from the receiver node to the left, so are the covariances from \( M_{n-1|n-1} \). Similarly, \( \hat{x}_{n|n+1} \), \( \hat{x}_{n+1|n+1} \), the corresponding covariances from \( M_{n+1|n+1} \) are received from the receiver node to the right (Fig. 5);
4) \( \hat{v}_{n|n-1,n} \) and \( M_{n|n-1,n} \) are calculated based on the forward correction equations: (6), (7), (8). Similarly, \( \hat{v}_{n|n,n+1} \) and \( M_{n|n,n+1} \) are calculated based on the backward correction equations: (13), (14), (15);
5) \( \hat{x}_{n|n,n+1} \), \( \hat{x}_{n-1|n,n+1} \), the corresponding covariances from \( M_{n|n,n+1} \) are passed to the left, and \( \hat{x}_{n|n-1,n} \), \( \hat{x}_{n+1|n-1,n} \), the corresponding covariances from \( M_{n|n-1,n} \) are passed to the right and so on; Step 3 and Step 4 are repeated as many times as required;
6) After j passes, \( \hat{v}_{n|n-j,\ldots,n+j} \) and \( M_{n|n-j,\ldots,n+j} \) can be computed using (18) and (19) with \( \hat{v}_{n|n} \), \( \hat{v}_{n-j,\ldots,n-1} \), \( \hat{v}_{n+1,\ldots,n+j} \), \( M_{n|n} \) \( M_{n|n-j,\ldots,n-1} \) and \( M_{n|n+1,\ldots,n+j} \), where we have used the
identifications:

\[ \hat{v}_n \equiv \hat{v}_n|\{n-j, \ldots, n+j\}; \quad \hat{v}_{n|\{1, \ldots, n-1\}} \equiv \hat{v}_{n|\{n-j, \ldots, n-1\}}; \quad \hat{v}_{n|\{n+1, \ldots, N\}} \equiv \hat{v}_{n|\{n+1, \ldots, n+j\}}; \]

\[ M_n \equiv M_n|\{n-j, \ldots, n+j\}; \quad M_{n|\{1, \ldots, n-1\}} \equiv M_{n|\{n-j, \ldots, n-1\}}; \quad M_{n|\{n+1, \ldots, N\}} \equiv M_{n|\{n+1, \ldots, n+j\}}. \]

Observe that the algorithm can be terminated after an arbitrary number of passes, and the estimate obtained is the LMMSE estimate based on the limited extent set corresponding to the number of passes that have taken place. Ultimately, if the number of passes is large enough for the information from the receiver nodes at the edges of the array to arrive, then the resultant LMMSE estimate is optimal. However, the number of passes required (in one direction) to obtain near-optimal performance can be very few in certain cases (e.g., less than three) as shown in Fig. 4. This is a significant advantage over the global forward-backward algorithm introduced in the previous section, particularly for large networks.

A desirable property of the KaLiX algorithm is its fast convergence, however, the amount of computation needed is rather high especially if the covariance matrix needs to be updated as well. In the next section, we introduce a type of distributed algorithm based on the first order stationary method for solving a system of linear equations, that achieves reduction in computation complexity per iteration at the price of an inferior convergence rate.

IV. FIRST ORDER STATIONARY ITERATIVE ALGORITHM (FOSI)

We begin by observing that (2) is the unique solution of the following system of linear equations:

\[ (H^H H + \sigma_0^2 I) \hat{x} = H^H y. \]  

(20)

Many iterative methods exist which can be used to approximate the solution to a system of linear equations [9], with less computational complexity than the direct inversion of the matrix in (2). A commonly used iterative method is the first-order stationary iterative (FOSI) method. When used to solve (20), the FOSI method takes the form:

\[ \hat{x}^{(m+1)} = \hat{x}^{(m)} - \tau \left[ (H^H H + \sigma_0^2 I) \hat{x}^{(m)} - H^H y \right] \]  

(21)
where $\hat{x}^{(m)}$ is the estimate of $x$ after $m$ iterations and $\tau$ is a design parameter which is called the step size. The convergence of $\hat{x}^{(m)}$ to $\hat{x}$ is guaranteed if a certain condition on $\tau$ is met, to be characterised later in the paper. For each transmitter node $n$, the iterative algorithm can be written as

$$
\hat{x}_n^{(m+1)} = \hat{x}_n^{(m)} - \tau \left[ (H^H H + \sigma_0^2 I)_n \hat{x}^{(m)} - h_n^H y \right], \quad n = 1, \ldots, N \tag{22}
$$

where $(H^H H + \sigma_0^2 I)_n$ denotes the $n$-th row of $H^H H + \sigma_0^2 I$, a pentadiagonal matrix.

Simple inspection shows that only local information is needed to perform the computation at each iteration. Specifically, the computation of $\hat{x}_n^{(m+1)}$ requires processed information: $\hat{x}_{n-2}^{(m)}$, $\hat{x}_{n-1}^{(m)}$, $\hat{x}_n^{(m)}$, $\hat{x}_{n+1}^{(m)}$ and $\hat{x}_{n+2}^{(m)}$ and the unprocessed information: $y_{n-1}$, $y_n$ and $y_{n+1}$. This is a direct consequence of the localised nature of the intercell interference, manifested in the band structure of the pentadiagonal matrix $H^H H + \sigma_0^2 I$ and the tridiagonal matrix $H$.

Assuming bidirectional communication between cells, we summarise the distributed algorithm to estimate $x_n$ as follows:

1) **Arbitrary initial conditions**, $\hat{x}_{n-2}^{(0)}$, $\hat{x}_{n-1}^{(0)}$, $\hat{x}_n^{(0)}$, $\hat{x}_{n+1}^{(0)}$ and $\hat{x}_{n+2}^{(0)}$ are assumed;

2) Signal $y_n$ is received and stored;

3) $y_n$ is passed to the adjacent cells. At the same time, $y_{n-1}$ and $y_{n+1}$ are received from the adjacent cells and stored;

4) $\hat{x}_n^{(1)}$ is calculated using (22) and stored;

5) $\hat{x}_n^{(1)}$ is passed to adjacent cells and at the same time, $\hat{x}_{n-1}^{(1)}$ and $\hat{x}_{n+1}^{(1)}$ are received from adjacent cells and stored;

6) $\hat{x}_{n-1}^{(1)}$ is passed to the cell to the right while $\hat{x}_{n+1}^{(1)}$ is passed to the left, $\hat{x}_{n-2}^{(1)}$ is received from the cell to the left and $\hat{x}_{n+2}^{(1)}$ is received from the cell to the right;

7) $\hat{x}_n^{(2)}$ is calculated using (22) and stored;

8) Steps 5 to 7 are repeated with $\hat{x}_n^{(2)}$, $\hat{x}_n^{(3)}$ and so on until $\hat{x}^{(m)}$ is sufficiently close to the true LMMSE solution.

In a similar manner to the KaLiX algorithm of Section III-C, this algorithm can be terminated at an arbitrary point in time. The update equation (22) is clearly much simpler than the KaLiX algorithm.
A question of interest is how does the convergence rate of the two algorithms compare. This shall be addressed in the next section.

V. CONVERGENCE ANALYSES OF KA LiX AND FOSI ALGORITHMS

The performance metric of interest is the MSE. We first state the MSE expression for the KaLiX algorithm. Define $H_m$ as the $(2m+1) \times (2m+3)$ limited extent set channel matrix. The MSE of the KaLiX algorithm after $m$ iterations is given by

$$MSE_{KaLiX}(n, m) = \left[ P(I - H_m^H (H_m H_m^H + \sigma_0^2)^{-1} H_m) \right]_{nn}, \quad n = 1, \ldots, N.$$  

To obtain an MSE expression for the FOSI algorithm, we begin by finding the effective linear receiver after $m$ iterations of the algorithm, which is given by

$$c^{(m)} = \tau \sum_{k=0}^{m-1} (I - \tau (H^H H + \sigma_0^2 I))^k H^H.$$  

For any linear receiver $c$, it can be shown that the MSE is

$$MSE(n, c) = \left[ P(I - H^H c H - cH + c(H H^H + \sigma_0^2 I) c^H) \right]_{nn}, \quad n = 1, \ldots, N.$$  

Therefore, the MSE of the FOSI algorithm after $m$ iterations is given by

$$MSE_{FOSI}(n, m) = MSE(n, c^{(m)}).$$  

We now have closed form expressions of the MSE achieved by the distributed algorithms after an arbitrary number of iterations. Based on these expressions, the ratio of MSE achieved after a certain number of message passing steps to the optimal MSE are plotted and are shown in Fig. 6. Note that one iteration of the KaLiX algorithm corresponds to one message passing step between receiver nodes (in both directions) whereas one iteration of the FOSI algorithm requires two message passing steps (again in both directions).

We have assumed that $\alpha_{n,m}$s are Rayleigh random variables with $E(\alpha_{n,n}) = 1$ and $E(\alpha_{n,m}) = \alpha$ for $n \neq m$. The optimal step size $\tau$ for the FOSI algorithm is assumed available and is recalculated for each channel realisation (more details on $\tau$ in Section V-B).
The plots clearly show the superiority of the KaLiX algorithm in terms of convergence over the FOSI algorithm. This advantage, however, is less significant, in the case where the SNR is small ($\sigma_0^2 = 10$) and the use of the FOSI algorithm may be preferable due to its reduced complexity.

Next, we present some analytical results on the convergence properties of the distributed algorithms. Specifically, we investigate how the KaLiX algorithm converges with the size of the network and the channel parameters. As for the FOSI algorithm, we provide a bound on the convergence rate that is independent of the array size. For simplicity and mathematical tractability, all analyses are done with an idealised model where $\alpha_{n,n-1} = \alpha_{n,n+1} = \alpha$ and $\alpha_{n,n} = 1$. Our aim is to extract some intuition from the idealised model, which also holds true in the more general model.

**A. Convergence Analysis of KaLiX Algorithm**

The question we address here is how does the MSE achieved by the KaLiX algorithm, which is actual MSE achieved by the LMMSE estimator on the limited extent network, compare with MSE achievable if the network is infinite; specifically, how does the difference between the two MSE quantities vary with size of the limited extent network and the channel parameters.

To further simplify the analysis, we shall drop the two extra transmitter nodes at the ends of the limited extent network (Fig. 3) and approximate the network with a linear array model of size $2j + 1$. It is shown in [10] that the MSE of $x_n$ in a linear array of size $N$ can be written as

$$MSE_N(n) \equiv P \left\{ \frac{1}{N} \sum_{k=1}^{N} f \left( \frac{k}{N + 1} \right) + \frac{1}{N(N-1)} \sum_{k=1}^{N} \left[ 1 - f \left( \frac{k}{N + 1} \right) \right] \right. $$

$$+ \left. \frac{1}{N + 1} \sum_{k=1}^{N} \cos \left( \frac{2 nk \pi}{N + 1} \right) \left[ 1 - f \left( \frac{k}{N + 1} \right) \right] \right\}$$

\hspace{1cm} (23)

where $f(x) = \sigma_0^2 / \left[ (1 + 2\alpha \cos \pi x)^2 + \sigma_0^2 \right]$ and that the MSE of each node in a infinite-sized linear array is

$$MSE_\infty \equiv P \int_0^1 f(\theta) \, d\theta.$$  \hspace{1cm} (24)

Let’s suppose we have a network with an odd number of cells so that the index for the middle node is
\( \frac{N+1}{2} \). Substituting \( n = \frac{N+1}{2} \) in (23) to get the MSE of our centre node, we have

\[
\text{MSE}_N \left( \frac{N+1}{2} \right) \equiv P \left\{ \frac{1}{N} \sum_{k=1}^{N} f \left( \frac{k}{N+1} \right) + \frac{1}{N(N-1)} \sum_{k=1}^{N} \left[ 1 - f \left( \frac{k}{N+1} \right) \right] \right. \\
+ \frac{1}{N+1} \sum_{k=1}^{N} (-1)^k \left[ 1 - f \left( \frac{k}{N+1} \right) \right] \}; \tag{25}
\]

Rearranging terms and using the fact that \( N \) is odd by assumption, we have

\[
\text{MSE}_N \left( \frac{N+1}{2} \right) = P \left\{ \frac{1}{N+1} \sum_{k=1}^{N} f \left( \frac{k}{N+1} \right) - \frac{2}{N(N^2-1)} \sum_{k=1}^{N} f \left( \frac{k}{N+1} \right) \\
- \frac{1}{N+1} \sum_{k=1}^{N} (-1)^k f \left( \frac{k}{N+1} \right) + \frac{2}{N^2-1} \right\}
\]

or equivalently

\[
\text{MSE}_N \left( \frac{N+1}{2} \right) = \frac{1}{N+1} \sum_{k=1}^{N} \left[ 1 - (-1)^k \right] f \left( \frac{k}{N+1} \right) + \frac{2}{N^2-1} \left[ 1 - \frac{1}{N} \sum_{k=1}^{N} f \left( \frac{k}{N+1} \right) \right].
\]

We are interested in determining an upper bound on the error \( |\text{MSE}_\infty - \text{MSE}_N \left( \frac{N+1}{2} \right)| \), where \( \text{MSE}_\infty \) is defined in (24):

\[
\left| \text{MSE}_\infty - \text{MSE}_N \left( \frac{N+1}{2} \right) \right| \leq P \left\{ \left| \int_0^1 f(\theta) \, d\theta - \frac{1}{N+1} \sum_{k=1}^{N} \left[ 1 - (-1)^k \right] f \left( \frac{k}{N+1} \right) \right| \\
+ \frac{2}{N^2-1} \left| 1 - \frac{1}{N} \sum_{k=1}^{N} f \left( \frac{k}{N+1} \right) \right| \right\}.
\]

The second term on the right is bounded above by \( 2/(N^2-1) \) since for \( \theta \in [0, 1] \) we have \( 0 \leq f(\theta) \leq 1 \).

To bound the first term we first observe that

\[
\frac{1}{N+1} \sum_{k=1}^{N} \left[ 1 - (-1)^k \right] f \left( \frac{k}{N+1} \right) = \frac{1}{L} \sum_{j=0}^{L-1} f \left( \frac{j}{L} + \frac{1}{2L} \right)
\]

where \( j = (k-1)/2 \) and \( L = (N+1)/2 \). Now applying a standard bound on the error in a Riemann sum, the first term is upper bounded by

\[
\frac{1}{2L} \max_{\theta \in [0, 1]} f'(\theta) = \frac{M_f}{N+1}
\]

where

\[
M_f = \max_{\theta \in [0, 1]} 4\pi\alpha\sigma_0^2 \sin \pi \theta \left( 1 + 2\alpha \cos \pi \theta \right) \\
\left[ (1 + 2\alpha \cos \pi \theta)^2 + \sigma_0^2 \right].
\]
Combining these results leads to the upper bound of

\[ \left| \text{MSE}_\infty - \text{MSE}_N \left( \frac{N + 1}{2} \right) \right| \leq P \left( \frac{M_f}{N + 1} + \frac{2}{N^2 - 1} \right). \]  

(26)

The error bound essentially decays as $M_f/N$ for large $N$, which provides the rate of the convergence to the infinite array MSE. The plot of $M_f$ vs $\alpha$ and $\sigma_0^2$ (Fig. 7) reveals that in general, the convergence is slower for larger $\alpha$, and that larger $\sigma_0^2$ always results in improved convergence, which is in agreement with simulations.

\section*{B. Convergence Analysis of FOSI Algorithm}

1) Finite-sized Linear Array: We denote $e^{(m)}$ (error vector) as the difference between the LMMSE estimate and the estimate after $m$ iterations, i.e. $e^{(m)} = \hat{x} - \hat{x}^{(m)}$. It can be shown that a recursive relation exists between $e^{(m)}$ and the initial error vector, $e^{(0)}$, given by

\[ e^{(m)} = \left[ I - \tau \left( H^H H + \sigma_0^2 I \right) \right]^m e^{(0)}. \]

We consider the Euclidean norm of the error vector as the metric of interest in the convergence analysis, it can be shown that the error vector norm can be bounded as follows [9]

\[ \| e^{(m)} \|_2 \leq R^m \| e^{(0)} \|_2 \]

where $R$ is the average convergence factor which can be shown [9] to be

\[ R = \max \left\{ |1 - \tau \lambda_{\min}|, |1 - \tau \lambda_{\max}| \right\} \]  

(27)

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and the maximum eigenvalues of $H^H H + \sigma_0^2 I$. The following results can also be established [9]:

- A necessary and sufficient condition for convergence of the iterative method from any arbitrary initial value is $0 < \tau < 2/\lambda_{\max}$.
- The optimal $\tau$, i.e. the $\tau$ that minimises $R$, is given by $\tau_{\text{opt}} = 2/(\lambda_{\min} + \lambda_{\max})$.
- The optimal average convergence factor, i.e. the minimum $R$, is given by

\[ R_{\text{opt}} = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}. \]  

(28)
It remains to find explicit bounds on $R_{\text{opt}}$ that enable us to investigate the effect of array size, $N$, on the rate of convergence. The eigenvalues of an $N \times N$ tridiagonal matrix, $\text{diag}(\alpha, 1, \alpha)$ are given by

$$\lambda_k' = 1 + 2\alpha \cos \frac{\pi k}{N+1}, \quad k = 1, \ldots, N$$

and thus the eigenvalues of $(H^2 + \sigma_0^2 I)$ are

$$\lambda_k = (\lambda_k')^2 + \sigma_0^2 = \left(1 + 2\alpha \cos \frac{\pi k}{N+1}\right)^2 + \sigma_0^2, \quad k = 1, \ldots, N. \tag{29}$$

Treating $k$ as a continuous variable and differentiating with respect to $k$, we get

$$\frac{d\lambda_k}{dk} = -\frac{4\alpha \pi}{N+1} \sin \frac{\pi k}{N+1} \left(1 + 2\alpha \cos \frac{\pi k}{N+1}\right).$$

**Case 1:** $0 \leq \alpha \leq 0.5$;

$\frac{d\lambda_k}{dk} \leq 0$ for $1 \leq k \leq N$; i.e. $\lambda_k$ is a decreasing function with respect to $k$ for $0 \leq \alpha \leq 0.5$. In this case, $\lambda_{\text{max}}$ and $\lambda_{\text{min}}$, are found by substituting $k = 1$ and $k = N$ respectively, into (29).

**Case 2:** $0.5 < \alpha \leq 1$;

$\lambda_k$ is a decreasing function for $1 \leq k \leq \lfloor \frac{N+1}{\pi} \cos^{-1} \left(\frac{-1}{2\alpha}\right)\rfloor$ and an increasing function for $\lceil \frac{N+1}{\pi} \cos^{-1} \left(\frac{-1}{2\alpha}\right)\rceil \leq k \leq N$, where $\lfloor \rfloor$ (resp. $\lceil \rceil$) denote the floor (ceiling) rounding operators. From (29), we can see that $\lambda_1$ is always greater than or equal to $\lambda_N$, so in this case, $\lambda_{\text{max}} = \lambda_1$. The value $\lambda_{\text{min}}$ is found by substituting $k = \lfloor \frac{N+1}{\pi} \cos^{-1} \left(\frac{-1}{2\alpha}\right)\rfloor$ or $k = \lceil \frac{N+1}{\pi} \cos^{-1} \left(\frac{-1}{2\alpha}\right)\rceil$, whichever results in lower $\lambda_k$.

We summarise the above findings below, where we make explicit the dependence of these values on the size, $N$, of the cellular array:

$$\lambda_{\text{max}}^{(N)} = \left(1 + 2\alpha \cos \frac{\pi}{N+1}\right)^2 + \sigma_0^2 \tag{30}$$

$$\lambda_{\text{min}}^{(N)} = \begin{cases} 
(1 + 2\alpha \cos \frac{\pi N}{N+1})^2 + \sigma_0^2, & 0 \leq \alpha \leq 0.5 \\
\min \left\{ \lambda_{\text{max}}^{(N)}, \lambda_{\text{min}}^{(N)} \right\}, & 0.5 < \alpha \leq 1 \end{cases}$$

where

\[
\lambda_L^{(N)} = \left(1 + 2\alpha \cos \frac{\pi L}{N+1}\right)^2 + \sigma_0^2; \quad \lambda_U^{(N)} = \left(1 + 2\alpha \cos \frac{\pi U}{N+1}\right)^2 + \sigma_0^2;
\]

\[
L = \left\lfloor \frac{N+1}{\pi} \cos^{-1} \left(-\frac{1}{2\alpha}\right) \right\rfloor; \quad U = \left\lceil \frac{N+1}{\pi} \cos^{-1} \left(-\frac{1}{2\alpha}\right) \right\rceil.
\]

2) Infinite-sized Linear Array: To obtain more insight on the convergence rate, we now take the limit as the array size \( N \) tends to infinity. It is simple to verify the following facts:

i \( \lambda^{(N)}_{\text{max}} \) is increasing in \( N \) and converges to \( \lambda^{(\infty)}_{\text{max}} \), where

\[
\lambda^{(\infty)}_{\text{max}} \equiv (1 + 2\alpha)^2 + \sigma_0^2.
\]

ii \( \lambda^{(N)}_{\text{min}} \) converges to \( \lambda^{(\infty)}_{\text{min}} \), where

\[
\lambda^{(\infty)}_{\text{min}} \equiv \begin{cases} 
(1 - 2\alpha)^2 + \sigma_0^2, & 0 \leq \alpha \leq 0.5 \\
\sigma_0^2, & 0.5 < \alpha \leq 1 
\end{cases}
\]

and \( \lambda^{(N)}_{\text{min}} > \lambda^{(\infty)}_{\text{min}} \) for all \( N \).

iii \( \tau^{(N)}_{\text{opt}} \) converges to \( \tau^{(\infty)}_{\text{opt}} \), where

\[
\tau^{(\infty)}_{\text{opt}} = \begin{cases} 
\frac{1}{1+4\alpha^2+\sigma_0^2}, & 0 \leq \alpha \leq 0.5 \\
\frac{2}{(1+2\alpha)^2+2\sigma_0^2}, & 0.5 < \alpha \leq 1.
\end{cases}
\]

iv \( R^{(N)}_{\text{opt}} \) converges to \( R^{(\infty)}_{\text{opt}} \), where

\[
R^{(\infty)}_{\text{opt}} = \begin{cases} 
\frac{4\alpha}{1+4\alpha^2+\sigma_0^2}, & 0 \leq \alpha \leq 0.5 \\
\frac{(1+2\alpha)^2}{(1+2\alpha)^2+2\sigma_0^2}, & 0.5 < \alpha \leq 1.
\end{cases}
\]

Note that

\[
R^{(N)}_{\text{opt}} = \frac{\lambda^{(N)}_{\text{max}} - \lambda^{(N)}_{\text{min}}}{\lambda^{(N)}_{\text{max}} + \lambda^{(N)}_{\text{min}}}
\]

\[
= \left(1 - \frac{\lambda^{(N)}_{\text{min}}}{\lambda^{(N)}_{\text{max}}}\right) \left(1 + \frac{\lambda^{(N)}_{\text{min}}}{\lambda^{(N)}_{\text{max}}}\right)^{-1}
\]

\[
\leq \left(1 - \frac{\lambda^{(\infty)}_{\text{min}}}{\lambda^{(\infty)}_{\text{max}}}\right) \left(1 + \frac{\lambda^{(\infty)}_{\text{min}}}{\lambda^{(\infty)}_{\text{max}}}\right)^{-1}
\]

\[
= R^{(\infty)}_{\text{opt}}
\]
where the inequality follows from observations i and ii above, and the fact that \((1 - x)/(1 + x)\) is a decreasing function for \(0 \leq x \leq 1\). This implies that with \(\tau \equiv \tau_{opt}^{(N)}\), the convergence rate is lower bounded by the rate corresponding to \(R_{opt}^{(\infty)}\), which is independent of the array size.

Another observation is that the actual eigenvalues are sandwiched between \(\lambda_{\text{min}}^{(\infty)}\) and \(\lambda_{\text{max}}^{(\infty)}\), and substituting \(\tau \equiv \tau_{opt}^{(\infty)}\) in (27) reveals that the average convergence factor is no worse than \(R_{opt}^{(\infty)}\) in this case. With the possible exception of very small array sizes, the degradation in convergence speed as a result of using \(\tau_{opt}^{(\infty)}\) is insignificant.

Using \(R_{opt}^{(\infty)}\) as the convergence factor, we observe that the convergence rate increases with the noise-to-interference ratio. Simulation results confirming this observation are depicted in Fig. 8.

VI. Conclusion

We have provided two distributed algorithms to implement the global LMMSE estimator, to recover the transmitted signals in a wireless network of interfering nodes where neighbouring receivers can communicate with each other. We summarise the properties of the algorithms below, where we have separated the two versions of the Kalman-based algorithm.

*Optimal Kalman-based distributed algorithm*

- applicable in one-dimensional networks modeled by the linear array model;
- always produces the global LMMSE estimate upon completion of the algorithm;
- suffers from delay that grows linearly with the size of the array.

*KaLiX algorithm*

- applicable in one-dimensional networks modeled by the linear array model;
- produces a limited extent LMMSE estimate, with the size of the limited extent set directly proportional to the delay of the algorithm;
- can be terminated after an arbitrary number of iterations and the error in terminating the algorithm after \(N\) steps is \(O(1/N)\);
- converges faster with smaller intercell-interference-to-noise ratio.

*FOSI algorithm*
• has less computational complexity per iteration than the KaLiX algorithm;
• suffers from longer delay than the KaLiX algorithm;
• produces an estimate after a finite delay that cannot be interpreted as an LMMSE estimate;
• has convergence rate that does not deteriorate without bound as the network size increases;
• converges faster with smaller intercell-interference-to-noise ratio.

Finally, we part with the concluding remark that although the algorithms are presented in the context of a one-dimensional network in this paper, the algorithms can be extended to two-dimensional networks. It is easy to see that the FOSI algorithm is directly applicable in a two-dimensional network. As for the Kalman algorithms, the Kalman filter has been shown in [11] to be a special instance of the sum-product algorithm operating on a particular factor graph and the Kalman smoother is no different. Therefore, the Kalman algorithms can be generalised as the sum-product algorithm which can be applied to networks with general topology.

APPENDIX

Consider the problem of estimating \( v_n \) given \( y \). We can write down the reduced set of linear equations as follows:

\[
y = \tilde{H}v_n + w \tag{32}
\]

where \( \tilde{H} \) is the reduced \( H \) matrix \((N \times 3)\), consisting of the columns \( n - 1, n \) and \( n + 1 \) of \( H \). We rewrite (32) as

\[
\begin{bmatrix}
y^l \\
y^r
\end{bmatrix} = \begin{bmatrix}
\tilde{H}^l \\
\tilde{H}^r
\end{bmatrix} \begin{bmatrix}
x_{n-1} \\
x_n \\
x_{n+1}
\end{bmatrix} + \begin{bmatrix}
w^l \\
w^r
\end{bmatrix}
\]
where we have used the following notations:

\[ y^l = [y_1 \ldots y_n]^T \]
\[ y^r = [y_{n+1} \ldots y_N]^T \]
\[ w^l = [\alpha_{1,1} x_1 + \alpha_{1,2} x_2 + z_1 \ldots z_{n-1} z_n]^T \]
\[ w^r = [z_{n+1} \ldots \alpha_{N,N-1} x_{N-1} + \alpha_{N,N} x_N + z_N]^T \]
\[ \tilde{H}^l = \text{rows } 1 \ldots n \text{ of } \tilde{H} \]
\[ \tilde{H}^r = \text{rows } n+1 \ldots N \text{ of } \tilde{H}. \]

The MSE matrix of \( \hat{v}_n \) is

\[
M_n = C_{v_n} - C_{v_n} \tilde{H}^H \left( \tilde{H} C_{v_n} \tilde{H}^H + C_w \right)^{-1} \left( C_{v_n} \tilde{H}^H \right)^H \\
= \left( C_{v_n}^{-1} + \tilde{H}^H C_w \tilde{H} \right)^{-1}
\]  

(33)

where \( C_{v_n} \) is the covariance matrix of \( v_n \) and \( C_w \) is the covariance matrix of \( w \). The equality (33) is due to the matrix inversion lemma stated below [12]:

\[
\left( C^{-1} + B^H A^{-1} B \right)^{-1} = C - C B^H \left( B C B^H + A \right)^{-1} B C.
\]

The LMMSE estimate of \( v_n \) given \( y \) is

\[
\hat{v}_n = C_{v_n} \tilde{H}^H \left( \tilde{H} C_{v_n} \tilde{H}^H + C_w \right)^{-1} y \\
= \left( C_{v_n}^{-1} + \tilde{H}^H C_w^{-1} \tilde{H} \right)^{-1} \tilde{H}^H C_w^{-1} y \\
= M_n \tilde{H}^H C_w^{-1} y
\]  

(34)

where the equality (34) is due to the following matrix identity [12]:

\[
AB^H (C + BAB^H)^{-1} = (A^{-1} + B^H C^{-1} B)^{-1} B^H C^{-1}.
\]
Since \( w^l \) is independent of \( w^r \), (34) can be written as

\[
\hat{v}_n = \left( C_{v_n}^{-1} + \begin{bmatrix} \tilde{H}^H & \tilde{H}^H \end{bmatrix} \right) \begin{bmatrix} \begin{bmatrix} C_{w}^{-1} & 0 \\ 0 & C_{w}^{-1} \end{bmatrix} \begin{bmatrix} \tilde{H}^H \\ \tilde{H}^H \end{bmatrix} \end{bmatrix}^{-1} \begin{bmatrix} y_l' \\ y_r' \end{bmatrix}
\]

where \( \hat{v}_n^l \) and \( \hat{v}_n^r \) are the LMMSE estimates of \( v_n \) given \( y_l \) and \( y_r \) respectively, and \( M_n^l \) and \( M_n^r \) are the corresponding MSE matrices. Therefore,

\[
\hat{v}_n = M_n \left( M_n^l \hat{v}_n^l + M_n^r \hat{v}_n^r \right)
\]

\[
M_n = \left( M_n^l + M_n^r - C_{v_n}^{-1} \right)^{-1}.
\]

(18) and (19) can be proved in a similar way.

REFERENCES


Fig. 1. Wireless network with transmitter nodes “t” and receiver nodes “r”

Fig. 2. Linear Array Model

Fig. 3. Limited extent set of received nodes associated with estimating $x_n$. 
Fig. 4. $\frac{\text{MSE}_{\text{LiX}}}{\text{MSE}_{\text{opt}}}$ of the middle transmitter node vs the number of receiver nodes $(2j + 1)$ used in LMMSE estimation.

(a) $\sigma_0^2 = 0.1$

(b) $\sigma_0^2 = 10$

Fig. 5. Message passing between receiver nodes.
Fig. 6. $\text{MSE}/\text{MSE}_{\text{opt}}$ versus the number of message passing steps ($N = 31$).

Fig. 7. $M_f$ vs $\alpha$ and $\sigma_0^2$
Fig. 8. Comparison of the simulated convergence of the iterative solution and its theoretical upperbound ($N = 50$).