

Localized Distributed Kalman Filters for Large-Scale Systems^{*}

Yuh-Shyang Wang^{*} Seungil You^{*} Nikolai Matni^{*}

^{*} *Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125, USA (e-mail: {yswang,syou,nmatni}@caltech.edu)*

Abstract: This paper presents a scalable method to design large-scale Kalman-like filters for a class of linear systems. In particular, we consider systems for which both the propagation of dynamics through the plant and the exchange of information between estimators/sensors is subject to delays. Under suitable assumptions on these delays, our proposed Kalman-like filter has the following desirable properties: (1) each local estimator only needs to collect the information within a localized region to estimate its local state, and (2) each local estimator can be designed by solving a local optimization problem using local plant model information. The decomposition of the global problem into local subproblems thus allows for the method to scale to arbitrarily large heterogeneous systems – this is clearly an extremely favorable property for large-scale estimation problems. The effectiveness of our algorithm is demonstrated on a randomized heterogeneous example with 51200 states, in which the traditional Kalman filter cannot be computed within a reasonable amount of time.

Keywords: Kalman filters, State estimation, Optimal estimation, Large-scale systems, Distributed control

1. INTRODUCTION

The celebrated Kalman filter achieves minimum mean square error for linear state estimation problems via an elegant and easily interpretable recursive method. Unfortunately, the Kalman filter is an inherently centralized method, and is neither scalable to compute nor physically implementable for large-scale systems. Specifically, the computation of the traditional Kalman filter involves solving an Algebraic Riccati Equation (ARE) and computing a matrix inverse, and the information in the network is assumed to be collected instantaneously by a central estimator. Even if a centralized estimator can be computed, large-scale estimation problems are nonetheless subject to practical communication delays between sensors and estimators which can degrade the performance of a centralized scheme substantially. These limitations make centralized estimation unappealing in large-scale applications such as weather forecasting (Farrell and Ioannou (2001)), ocean data assimilation (Fukumori and Malanotte-Rizzoli (1995)), biological signal analysis (Long et al. (2006)), and state estimation in the power grid (Huang et al. (2012)).

Various methods have been proposed in the field of distributed Kalman filtering, but many still suffer from scalability issues that limit their application to large-scale

systems. For instance, both the consensus-based algorithm of Olfati-Saber (2007) and the diffusion-based algorithm of Cattivelli and Sayed (2010) require each local sensor to store and use the *global plant model*, and to estimate the *global state* during implementation. This introduces huge computational burden, and is prohibitive for large-scale applications. An exception is the work of Khan and Moura (2008), in which the authors use spatial decomposition, observation fusion, and approximated algorithms on matrix inversion to design a scalable Kalman-like filter. However, the algorithms involve multiple iterations, and the transient behavior of the algorithm is hard to analyze.

In this paper, we propose a scalable method to design large-scale Kalman-like filters based on the notion of *localizability for state estimation*. This notion can be viewed as a generalization of observability, in which the state estimator is restricted to a subspace dictated by spatiotemporal constraints. Intuitively, when information can be shared sufficiently quickly among local estimators, the uncertainty in the state due to local process and sensor noise can be isolated to a localized region. In other words, the closed loop response from process and sensor noise to the estimation error is *localized*. We show that finding such a localized closed loop response, if it exists, can be done in an efficient and scalable manner, and further demonstrate that the resulting Kalman-like estimator can be *designed* and *implemented* in a localized way. Our main technical tool is the duality that exists between Kalman Filtering and Linear Quadratic Regulation (LQR), and use this duality to formulate the localized distributed Kalman filter (LDKF) problem as a localized LQR (LLQR) problem (Wang et al. (2014) and Wang and Matni (2015)).

^{*} This research was in part supported by NSF NetSE, AFOSR, the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office, and from MURIs “Scalable, Data-Driven, and Provably-Correct Analysis of Networks” (ONR) and “Tools for the Analysis and Design of Complex Multi-Scale Networks” (ARO). The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

The rest of this paper is structured as follows. In Section 2, we review the traditional Kalman filter and emphasize its limitations in the context of large-scale systems. In Section 3, we reformulate the Kalman filter problem in terms of the *closed loop transfer matrices*. We introduce the idea of localizability for state estimation, and formulate the LDKF problem in Section 4. Specifically, LDKF estimator can be designed and implemented in a localized and parallel way using techniques developed to solve the LLQR problem, as described in Wang et al. (2014). To demonstrate the effectiveness of our method, we synthesize the LDKF estimator for a system with 51200 states in Section 5. Finally, conclusions are given in Section 6.

2. TRADITIONAL KALMAN FILTER

This section introduces the system model and the traditional Kalman filter. We then explain the limitations of traditional Kalman filter on large-scale systems, which motivate our work.

2.1 System Model

Consider a discrete time linear time invariant (LTI) system with dynamics given by

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] + w[k] \\ y[k] &= Cx[k] + v[k] \end{aligned} \quad (1)$$

where x is the state, u the control input, y the sensor measurements, w the process noise, and v the sensor noise. Our goal is to design a state estimator \hat{x} based on the measurement y and a pre-specified control input u . In particular, we are interested in the case when the system matrices (A, B, C) are high-dimensional yet suitably sparse. Our approach is to exploit the sparsity of (A, B, C) to derive a scalable algorithm for state estimator design.

We adopt to the common setting for Kalman filter (cf. Anderson and Moore (2012)). Let $\mathbb{E}(\cdot)$ be the expectation operator and δ_{ij} the Kronecker delta function. We assume that the process noise w and sensor noise v are independent zero mean additive white Gaussian noise (AWGN), with covariance matrices given by $\mathbb{E}(w[i]w[j]^T) = \delta_{ij}W$, $\mathbb{E}(v[i]v[j]^T) = \delta_{ij}V$, and $\mathbb{E}(w[i]v[j]^T) = 0$. We assume that w and v are uncorrelated to keep the formulas simple, while the method described in this paper still works when w and v are correlated. The initial condition $x[0]$ is also assumed to be a Gaussian random vector with mean x_0 and variance Σ_0 , and $x[0]$ is uncorrelated with $w[k]$ and $v[k]$ for all k .

2.2 Traditional Kalman Filter

Let $\hat{x}[k|s]$ denote the estimate of the state $x[k]$ given the collected information up to time s , i.e. the measurements $y[t]$ and control inputs $u[t]$ from $t = 1, \dots, s$. The Kalman filter for the LTI system (1) is specified by

$$\hat{x}[k|k] = \hat{x}[k|k-1] + K(y[k] - C\hat{x}[k|k-1]) \quad (2)$$

$$\hat{x}[k+1|k] = A\hat{x}[k|k] + Bu[k] \quad (3)$$

with initial condition given by $\hat{x}[0|-1] = x_0$. The matrix K in (2) is known as the Kalman gain, which can be found

by solving an ARE. Let Σ be the solution to the discrete time ARE

$$\Sigma = A\Sigma A^T + W - A\Sigma C^T(C\Sigma C^T + V)^{-1}C\Sigma A^T. \quad (4)$$

The Kalman gain in (2) can then be computed as

$$K = \Sigma C^T(C\Sigma C^T + V)^{-1}. \quad (5)$$

The Kalman filter is optimal in the sense of minimum mean square error. Let $\tilde{x}[k|k-1] = x[k] - \hat{x}[k|k-1]$ be the estimation error before $y[k]$ is measured. The Kalman filter algorithm in (2) - (3) minimizes the mean square error

$$\mathbb{E}\left(\frac{1}{N}\sum_{k=1}^N \tilde{x}[k|k-1]^T \tilde{x}[k|k-1]\right) \quad (6)$$

for $N \rightarrow \infty$. Similarly, let $\tilde{x}[k|k] = x[k] - \hat{x}[k|k]$ be the estimation error after $y[k]$ is measured. The mean square error of $\tilde{x}[k|k]$ is also minimized.

Equations (2) and (3) can be combined into a single equation as

$$\hat{x}[k+1|k] = A\hat{x}[k|k-1] + Bu[k] + L(y[k] - C\hat{x}[k|k-1]) \quad (7)$$

with $L = AK$ is a gain matrix. We refer to (7) as the *delayed form* of state estimation. We can also combine equations (2) and (3) to obtain

$$\hat{x}[k+1|k+1] = (I - KC)(A\hat{x}[k|k] + Bu[k]) + Ky[k+1]. \quad (8)$$

We refer to (8) as the *current form* of state estimation.

2.3 Limitations

Here we point out some limitations of the traditional Kalman filter for large-scale systems.

- (1) The Kalman gain given by (5) is generally dense even when the system matrices (A, B, C) that specify the system dynamics (1) are sparse. This means that the measurements from all sensors need to be shared *instantaneously*, which requires infinite (or impractically fast) communication speed.
- (2) A dense Kalman gain (5) also implies that the measurements from all sensor need to be collected by *every estimator in the network*, which is not scalable to implement.
- (3) To compute the Kalman gain (5), one need to solve a large-scale ARE (4). This is not scalable for large systems.
- (4) When the global plant model (A, B, C) changes locally, one needs to recompute the solution to (4) to resynthesize the global Kalman filter. This is not scalable for incremental design when the physical system expands.

To design a state estimation algorithm for large-scale systems, one must overcome the aforementioned limitations of the traditional Kalman filter. Distributed Kalman filter architectures in Olfati-Saber (2007) or Cattivelli and Sayed (2010) may resolve the first two limitations, but not the latter two. This motivates our development of the LDKF.

3. ALTERNATIVE FORMULATION

In this section, we use *closed loop transfer matrices* to analyze the estimation error dynamics of the Kalman

filter. We then reformulate the traditional Kalman filter problem as an optimization problem over all valid closed loop transfer matrices. This new formulation plays an important role in the derivation of the LDKF.

3.1 Alternative Formulation for the Delayed Form

Consider first the delayed form (7) of the Kalman filter. Taking the z-transform of equation (7), we get

$$(zI - A)\hat{x} = Bu + L(y - C\hat{x}). \quad (9)$$

In the following, we assume that the estimator structure (9) is fixed, but the gain matrix L is unknown and needs to be designed. Although the Kalman filter can be implemented via a static gain L , this is not necessary. We relax the gain L to be a proper transfer matrix in the sequel. This extra freedom will be key in allowing us to incorporate spatiotemporal constraints on the transfer matrices that define the estimator. Combining (9) and (1), we have the estimation error dynamics

$$(zI - A + LC)\tilde{x} = w - Lv. \quad (10)$$

Define $M_w := (zI - A + LC)^{-1}$ to be the closed loop transfer matrix from process noise w to estimation error \tilde{x} , and likewise let $M_v := -M_w L$ be the closed loop transfer matrix from sensor noise v to estimation error \tilde{x} . Equation (10) can then be rewritten as

$$\tilde{x} = M_w w + M_v v. \quad (11)$$

Rather than finding a suitable gain matrix L to optimize the closed loop transfer matrices (M_w, M_v) indirectly, we instead characterize the set of valid transfer matrices (M_w, M_v) , and instead optimize *directly* over those sets. Let $M_w[i]$ denote the i th spectral component of M_w , i.e. $M_w(z) = \sum_{i=0}^{\infty} z^{-i} M_w[i]$, and let \mathcal{RH}_{∞} denote the set of real rational stable proper transfer matrices. The following Proposition gives a characterization of all valid closed loop transfer matrices (M_w, M_v) , given the estimator structure (9).

Proposition 1. The closed loop transfer matrices (M_w, M_v) with finite mean square error can be induced by an estimator with structure (9) if and only if the following two affine constraints hold.

$$M_w(zI - A) - M_v C = I \quad (12)$$

$$M_w, M_v \in \frac{1}{z} \mathcal{RH}_{\infty} \quad (13)$$

Proof. To prove the necessary direction, we show that (12) and (13) must hold for an estimator with structure (9) and suitable gain matrix L . Equation (12) can be verified using the identity $M_w(zI - A + LC) = I$ directly. For (13), note that the transfer matrices M_w and M_v must be stable so that the mean square error of the estimator is finite. Besides, as $M_w = (zI - A + LC)^{-1}$ and $M_v = -M_w L$, the transfer matrices M_w and M_v must be strictly proper, i.e. $M_w[0] = 0, M_v[0] = 0$. Therefore, (13) must hold.

To prove the sufficient direction, we show that the desired closed loop response (M_w, M_v) can be induced by an estimator with structure (9) if (12) and (13) hold. For any solution of (12) - (13), we construct a gain matrix

$L_0 = -M_w^{-1} M_v$ for (9). In this case, the estimation error dynamics (10) become

$$(zI - A - M_w^{-1} M_v C)\tilde{x} = w + M_w^{-1} M_v v. \quad (14)$$

Multiplying M_w to both sides of (14) and substituting (12) into the equation, we can show that the desired closed loop response (M_w, M_v) is achieved. \square

Proposition 1 suggests that we can implement the estimator using the gain matrix $L = -M_w^{-1} M_v$ to achieve the desired closed loop response. Substituting this identity back to (9) and multiplying M_w to both sides of the equation, we get

$$\hat{x} = M_w B u - M_v y. \quad (15)$$

This gives a simpler estimator implementation.

Given this characterization of valid closed loop transfer matrices, we now aim to find an expression for the Kalman Fitter objective function in terms of M_w and M_v . Using the error dynamics (11) and the AWGN assumptions on the noise dynamics, it is straightforward to show that the mean square error can be expressed in terms of the impulse response elements of the closed loop transfer matrices as

$$\text{Trace} \left(\sum_{i=0}^{\infty} [M_w[i] \ M_v[i]] \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} M_w[i]^T \\ M_v[i]^T \end{bmatrix} \right).$$

As traditional Kalman filter achieves minimum mean square error, the closed loop transfer matrices (M_w, M_v) for the Delayed Form of the Kalman filter must be the solution to the following optimization problem

$$\begin{aligned} & \underset{M_w, M_v}{\text{minimize}} \quad \text{Trace} \left(\sum_{i=0}^{\infty} [M_w[i] \ M_v[i]] \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} M_w[i]^T \\ M_v[i]^T \end{bmatrix} \right) \\ & \text{subject to} \quad (12) - (13). \end{aligned} \quad (16)$$

Thus we can view optimization (16) as an alternative formulation for the Delayed Form of the Kalman filter problem.

3.2 Alternative Formulation for the Current Form

Consider the current form of Kalman filter in (8). After some calculations, the closed loop transfer matrices from noise to the estimation error are shown to be

$$\begin{aligned} M_w &= (zI - A + KCA)^{-1} (I - KC) \\ M_v &= -(I - \frac{1}{z}(I - KC)A)^{-1} K. \end{aligned} \quad (17)$$

For the characterization of all valid closed loop transfer matrices, we still have the identity (12). The constraint in (13) changes slightly however, as the transfer matrix from sensor noise to estimation error M_v is only restricted to be proper in the Current Form of the Kalman filter. The estimator equation in (15) remains unchanged. We can then give an alternative formulation for the Current Form of the Kalman filter as

$$\begin{aligned} & \underset{M_w, M_v}{\text{minimize}} \quad \text{Trace} \left(\sum_{i=0}^{\infty} [M_w[i] \ M_v[i]] \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} M_w[i]^T \\ M_v[i]^T \end{bmatrix} \right) \\ & \text{subject to} \quad (12), M_w \in \frac{1}{z} \mathcal{RH}_{\infty}, M_v \in \mathcal{RH}_{\infty}. \end{aligned} \quad (18)$$

In the interest of space, we only discuss the Delayed Form of the Kalman filter (16), although all methods described in this paper work equally well for the Current Form (18).

4. LOCALIZED DISTRIBUTED KALMAN FILTER

In this section, we introduce the notion of *localizability for state estimation* and give the LDKF formulation. We then highlight the key steps to solve LDKF in a scalable manner using LLQR techniques. Finally, we propose two scalable ways to implement the LDKF estimator.

4.1 Localizability for State Estimation

Consider the alternative formulation of Kalman filter (16), with an additional structural constraint \mathcal{S} imposed on the closed loop transfer matrices (M_w, M_v) . This leads to the LDKF optimization problem

$$\begin{aligned} & \underset{M_w, M_v}{\text{minimize}} \quad \text{Trace} \left(\sum_{i=0}^{\infty} [M_w[i] \quad M_v[i]] \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} M_w[i]^T \\ M_v[i]^T \end{bmatrix} \right) \\ & \text{subject to} \quad (12) - (13) \\ & \quad [M_w \quad M_v] \in \mathcal{S}. \end{aligned} \quad (19)$$

In general, we can use the set \mathcal{S} to encode any kind of spatiotemporal constraints on the closed loop response. A useful approach for large-scale systems is to impose sparsity constraints through \mathcal{S} , e.g., constraining certain elements $(M_w[i])_{jk} = 0$ for some i, j, k . As long as \mathcal{S} is a convex set, optimization problem (19) is convex. If problem (19) is feasible with a constraint \mathcal{S} , then we say that the system (1) is \mathcal{S} -localizable for state estimation. In other words, the \mathcal{S} -localizability for state estimation of a system is determined by the feasibility of (19).

Example 1 Assume that A in (1) is a tridiagonal matrix, and B and C are identity matrices. We use \mathcal{S} to impose a tridiagonal sparsity constraint on M_w , and a pentadiagonal sparsity constraint on M_v . Under these constraints, one feasible solution of (19) is given by

$$M_w = z^{-1}I + z^{-2}A, \quad M_v = -z^{-2}A^2. \quad (20)$$

As each column of M_w/M_v represents the closed loop response from a local process/sensor noise to the state estimation error, the sparsity pattern of M_w/M_v in (20) therefore suggests that the effect of each process/sensor noise to the state estimation error is localized.¹

Recall that the estimator achieving the desired closed loop response is given by (15). When a localized closed loop response exists, the implementation (15) is localized and thus scalable. This is indicated by the sparsity pattern of each row of M_w and M_v in (15) – each component of the state estimate \hat{x} can be computed by collecting only some components of the measurement y and the control action u . In addition, imposing a suitable sparsity constraint \mathcal{S} also allows us to solve optimization problem (19) in a parallel and localized way (cf. §4.2). In other words, the sparsity constraints \mathcal{S} in (19) can provide scalability both on estimator implementation and estimator design.

¹ It should be noted that we are *not* localizing the effect of the process/sensor noise on the state vector. Rather, we are localizing the *state estimation error* due to the noise.

Besides the sparsity constraint, we can also use \mathcal{S} to encode a finite impulse response (FIR) constraint or a communication delay constraint. It should be pointed out that the observability of a system is determined by the feasibility of (19) with \mathcal{S} being an FIR constraint. Using this interpretation, we can view observability as a special case of \mathcal{S} -localizability for state estimation.

The communication delay constraint can be encoded as sparsity constraint on each spectral component of (M_w, M_v) in (15). This contains the information whether $\hat{x}_i[t]$ has access to the measurement $y_j[t-\tau]$ for some i, j, τ . In this paper, we require the communication speed to be faster than the speed of noise propagated in the plant, so that a localized closed loop response can possibly exist. Intuitively, when the information can be shared fast enough among local estimators, the effect of local process noise and sensor noise on the state vector becomes predictable, and this effect on the estimation error can possibly be localized.

4.2 Localized Synthesis

Optimization problem (19) is of the form of a LLQR problem, and thus can be solved in a localized and scalable way. The technical details on localized synthesis are almost identical to the one in Wang et al. (2014) and Wang and Matni (2015) – this is not surprising given the duality between Kalman filter design and LQR control. Due to space constraints, we highlight the key steps and refer the reader to the aforementioned references for details.

First, note that the objective function and constraints of optimization problem (19) admit a *row-wise decomposition*. Specifically, we can solve for each row of the transfer matrix $[M_w \quad M_v]$ in an independent and parallel way. We refer to the optimization problem that solves for each row of $[M_w \quad M_v]$ as a LDKF subproblem. For each LDKF subproblem, the locality constraint \mathcal{S} further allows us to reduce the dimension of the problem from *global* to *local* scale. Each LDKF subproblem in the reduced dimension can then be solved by a local optimization problem using local plant model information only. This provides a localized, parallel, and scalable estimator design algorithm.

It should be noted that the localized synthesis method is valid for arbitrary noise covariance matrices (W, V) , i.e., even when the noise is globally correlated. In addition, when the system matrices (A, B, C) change locally, we only need to resolve some of the LDKF subproblems and update the estimator locally. It follows that this allows for the incremental addition of new subsystems to the global system without the need for a complete redesign of the estimator.

4.3 Localized Implementation

After solving the LDKF problem, the LDKF estimator can be implemented using the transfer matrix form (15) to achieve the desired closed loop response. Another way to implement the LDKF estimator is given by

$$\begin{aligned}
z\hat{x} &= A\hat{x} + Bu - \beta \\
\beta &= \alpha - \alpha_r \\
\alpha &= zM_v(y - C\hat{x}) \\
\alpha_r &= (zM_w - I)\beta.
\end{aligned} \tag{21}$$

Here, zM_v is proper and $(zM_w - I)$ is strictly proper, so the estimator structure is causal and well-defined. It can be shown that (21) is equivalent to (9) for $L = -M_w^{-1}M_v$. As M_w and M_v are localized transfer matrices, the implementation in (21) is localized and thus scalable. The benefit of (21) over (15) is that (21) is compatible with the form of an extended Kalman filter, which provides a possible approach to extend our methods to nonlinear systems.

5. SIMULATIONS

In this section, we compare the performance of the LDKF with that of the traditional Kalman filter (KF) through simulations. We then compute the LDKF estimator for a randomized heterogeneous system with 51200 states. We note that the computational bottleneck that we faced in computing our large scale example was that we were using a single workstation to compute the estimator (and hence the LDKF subproblems were essentially solved in serial) – in practice, if each local estimator is capable of solving its corresponding LDKF subproblem, our approach scales to systems of *arbitrary size* as all computations can be done in parallel.

5.1 Comparison

We begin with a 20×20 mesh topology representing the interconnection between subsystems, and drop each edge with probability 0.2. The resulting interconnected topology is shown in Fig. 1. The dynamics of each subsystem is

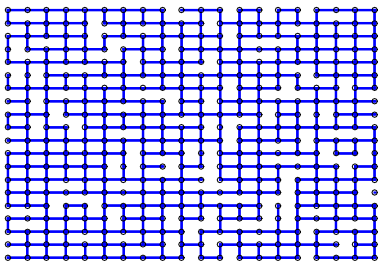


Fig. 1. Interconnected topology for the simulation example

given by the discretized swing equation for power network application. Denote x_i, y_i, w_i, v_i the state, measurement, process noise, and sensor noise for i th subsystem, respectively. The dynamics of each subsystem i is given by

$$\begin{aligned}
x_i[k+1] &= A_{ii}x_i[k] + \sum_{j \in \mathcal{N}_i} A_{ij}x_j[k] + w_i[k] \\
y_i[k] &= C_i x_i[k] + v_i[k],
\end{aligned} \tag{22}$$

where

$$\begin{aligned}
A_{ii} &= \begin{bmatrix} 1 & \Delta t \\ -\frac{k_i}{m_i}\Delta t & 1 - \frac{d_i}{m_i}\Delta t \end{bmatrix} \\
A_{ij} &= \begin{bmatrix} 0 & 0 \\ \frac{k_{ij}}{m_i}\Delta t & 0 \end{bmatrix}, \quad C_i = [1 \ 0].
\end{aligned}$$

The parameters k_{ij}, d_i, m_i^{-1} are randomly generated and uniformly distributed between 0.2 and 1. In addition, we set $\Delta t = 0.2$ and $k_i = \sum_{j \in \mathcal{N}_i} k_{ij}$. The global plant model (1) can be constructed by (22). The process and sensor noise in (1) are zero mean AWGNs with covariance matrices being identity. The initial condition $x[0]$ is zero mean Gaussian with covariance matrix given by $9I$. The instability of the plant is characterized by the spectral radius of the matrix A , which is 1.0387. In this example, the number of states and measurements are 800 and 400, respectively.

The \mathcal{S} constraint for the LDKF scheme is enforced as follows. Each local process noise is only allowed to affect the estimation error up to its neighboring subsystems, and each sensor noise is allowed to affect the estimation error up to its two-hop neighbors. This means that each subsystem needs to communicate up to its two-hop neighbors during implementation, and use the plant model up to its two-hop neighbors for estimator synthesis. For communication delays, we assume that $\hat{x}_i[t]$ can access $y_j[\tau]$ at time $\tau \leq t - k$ if subsystem (i, j) are k -hop neighbors. As the disturbance takes two steps to propagate to its neighboring subsystems, the communication speed is twice faster than the speed of disturbance propagation. We impose the FIR constraint with length $T = 15$.

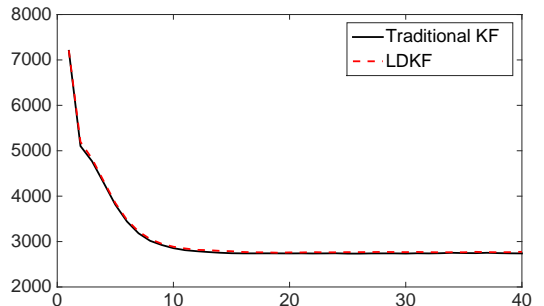


Fig. 2. The vertical axis represents the expected mean square error of the estimation, and the horizontal axis represents the time step. The expectation is simulated over 1000 Monte Carlo trials.

We perform 1000 Monte Carlo trials for both the traditional KF and the LDKF. The expected mean square error (MSE) at each time step is shown in Fig. 2. Recall that the covariance matrix for $x[0]$ is given by $9I$, and the number of states is 800. Theoretically, the expected MSE for the first time step is 7200, which agrees with our simulation. Our simulation shows that the performance of LDKF is almost identical to that of the traditional KF, even when the imposed constraint \mathcal{S} is highly sparse. In this example, the theoretical steady state MSE for the traditional KF is 2739.8, and that of the LDKF is 2767.2. In terms of the MSE, LDKF estimator has 1% performance degradation. However, we get huge benefits in terms of estimator synthesis and implementation.

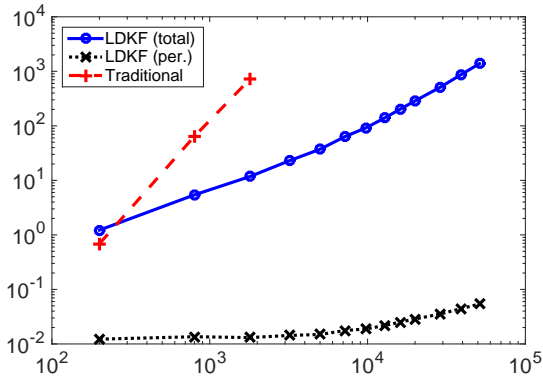


Fig. 3. The horizontal axis represents the number of state, and the vertical axis represents the computation time in seconds.

To further illustrate the advantages of the LDKF, we compare the LDKF with the traditional KF in terms of closed loop performance, estimator synthesis, and estimator implementation. The results are summarized in Table 1. It can be seen that the LDKF has significant advantages over the traditional KF in all aspects, except for the 1% performance degradation in MSE.

Table 1. Comparison Between Traditional KF and LDKF

		Traditional	LDKF
Closed Loop	Affected region	Global	2-hop
	Affected time	Long	15 steps
	Normalized MSE	1	1.01
Synthesis	Comp. complexity	$O(n^3)$	$O(n)$
	Comp. time (s)	64.4	5.4
	Time per node (s)	64.4	0.014
	Plant model	Global	2-hop
	Redesign	Offline	Real-time
Implementation	Comm. Speed	Inf	2
	Comm. Range	Global	2-hop

5.2 Large-Scale Example

We change the size of the mesh network and compare the computation time between the traditional KF and the LDKF. The result is shown in Fig. 3. For the LDKF, we use parallel for loop in MATLAB with four workers. From Fig. 3, the LDKF for a system with 51200 states can be computed in 23 minutes using a standalone workstation. If the computation is parallelized across all subsystems, as indicated by Fig. 3, the synthesis algorithm can be computed within 0.1 seconds.

6. CONCLUSION

In this paper, we proposed the localized distributed Kalman filter (LDKF) architecture, which is a scalable algorithm for large-scale state estimation in localizable systems. LDKF offers various benefits over traditional and distributed Kalman filters, as the estimator can be both designed and implemented in a localized, parallel, and scalable way. The LDKF algorithm is demonstrated on a randomized heterogeneous system with 51200 states, where the traditional method cannot be computed within a reasonable time period.

It should be noted that the locations of the sensors and the communication allowed between local estimators play an important role on the localizability of a system, which affects the effectiveness of our method significantly. In the future, we plan to adopt to the Regularization for Design framework of Matni and Chandrasekaran (2015), which has been successfully applied for actuator placement in the context of LLQR control by Wang et al. (2015), to *co-design* the location of sensors and the LDKF estimator simultaneously.

ACKNOWLEDGEMENTS

The authors would like to thank John C. Doyle for his enthusiastic support of this work.

REFERENCES

- Anderson, B.D. and Moore, J.B. (2012). *Optimal filtering*. Courier Corporation.
- Cattivelli, F.S. and Sayed, A.H. (2010). Diffusion strategies for distributed Kalman filtering and smoothing. *Automatic Control, IEEE Transactions on*, 55(9), 2069–2084.
- Farrell, B.F. and Ioannou, P.J. (2001). State estimation using a reduced-order Kalman filter. *Journal of the Atmospheric Sciences*, 58(23), 3666–3680.
- Fukumori, I. and Malanotte-Rizzoli, P. (1995). An approximate Kalman filter for ocean data assimilation: An example with an idealized gulf stream model. *Journal of Geophysical Research: Oceans (1978–2012)*, 100(C4), 6777–6793.
- Huang, Y.F., Werner, S., Huang, J., Kashyap, N., and Gupta, V. (2012). State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid. *Signal Processing Magazine, IEEE*, 29(5), 33–43.
- Khan, U.A. and Moura, J.M. (2008). Distributing the Kalman filter for large-scale systems. *Signal Processing, IEEE Transactions on*, 56(10), 4919–4935.
- Long, C., Purdon, P., Temereanca, S., Desai, N., Hamalainen, M., and Brown, E. (2006). Large scale Kalman filtering solutions to the electrophysiological source localization problem—a meg case study. In *Engineering in Medicine and Biology Society, 2006. EMBS’06. 28th Annual International Conference of the IEEE*, 4532–4535. IEEE.
- Matni, N. and Chandrasekaran, V. (2015). Regularization for design. *submitted to IEEE Transactions on Automatic Control*, arXiv:1404.1972. URL <http://arxiv.org/abs/1404.1972v2.pdf>.
- Olfati-Saber, R. (2007). Distributed kalman filtering for sensor networks. In *Decision and Control, 2007 46th IEEE Conference on*, 5492–5498. IEEE.
- Wang, Y.S. and Matni, N. (2015). Localized LQG optimal control for large-scale systems. In *submitted to 2015 54th IEEE Conference on Decision and Control (CDC)*.
- Wang, Y.S., Matni, N., and Doyle, J.C. (2014). Localized LQR optimal control. In *2014 53th IEEE Conference on Decision and Control (CDC)*.
- Wang, Y.S., Matni, N., and Doyle, J.C. (2015). Localized LQR control with actuator regularization. In *submitted to 2015 54th IEEE Conference on Decision and Control (CDC)*.