ITM Web of Conferences **80**, 01001 (2025) https://doi.org/10.1051/itmconf/20258001001
ACAAI 2025

Distributed sensor fusion estimation algorithms based on Kalman Filtering

Si Ni*
College of professional studies, Northeastern University,360 Huntington Ave, Boston, The United States

Abstract. Efficient distributed sensor fusion is critical for reliable state estimation in applications such as autonomous vehicles, robotics, and environmental monitoring. This review examines four main distributed Kalman filtering approaches: matrix-weighted fusion, covariance intersection, feedback based optimal fusion, and machine learning–augmented schemes. Core equations for each method are outlined. Communication requirements, computational complexity, and estimation accuracy are systematically compared across diverse network conditions, including synchronous, asynchronous, and lossy environments with packet loss. Practical challenges addressed encompass scalability in large-scale, high-dimensional systems, numerical stability under limited computational precision, and inherent trade-offs between estimation performance and resource consumption. Case studies and extensive simulations demonstrate the real-world efficacy of each method. Finally, key future research directions are highlighted, focusing on edge-optimized architectures, robust algorithms tolerant to significant delays and asynchronous updates, and the integration of essential security and privacy features. This synthesis provides a roadmap for advancing distributed Kalman filters within resource-constrained sensor networks.

1 Introduction

The rapid advancement of sensing technologies across domains such as autonomous vehicles, robotics, smart grids, and environmental monitoring has created an urgent demand for accurate, scalable, and robust state estimation techniques. Multi-sensor systems integrate redundant and complementary data from heterogeneous sources (e.g., LiDAR, radar, cameras, and inertial measurement units) to enhance resilience against failures, reduce uncertainty, and support reliable real-time decisions.

Traditional centralized fusion architectures transmit raw data from all sensors to a central processor. While conceptually straightforward, this model suffers from substantial communication overhead, significant latency, and a critical vulnerability to single-point failures. As the number of sensors increases, the bandwidth required to transmit raw data and the processing load at the central unit become prohibitive. Moreover, a fault in the central processor or communication link can compromise the entire system's functionality.

*Corresponding author: ni.s1@northeastern.edu

In contrast, distributed sensor fusion frameworks perform local estimation at individual sensor nodes and transmit data such as state estimate and covariance matrices to a central fusion module. This architecture significantly reduces communication burden, enhances fault tolerance, and improves scalability, this review focuses on architectures where multiple local estimators operate independently and forward their results to a fusion center for final integration. Such systems represent a practical form of distributed sensing with centralized decision support.

The Kalman filter (KF), introduced by Kalman, remains the foundational algorithm for optimal state estimation in linear Gaussian systems [1]. It updates the system state recursively by combining model-based predictions with incoming sensor measurements. This algorithm has enabled widespread applications in tracking, navigation and process control [2-4].

However, directly applying the standard Kalman filter in distributed multi-sensor systems presents critical challenges. These include the lack of known cross-covariances between local estimates, the high cost of transmitting full covariance matrices, and reduced robustness in the face of node failures or dynamic topologies [5].

To address these issues, various distributed Kalman filtering (DKF) algorithms have emerged [6]. These include consensus-based filters that iteratively propagate local estimates, covariance intersection (CI) techniques that conservatively fuse information without requiring cross-correlation knowledge, and bounded or iterative CI methods that improve filter stability and precision [2, 7]. For nonlinear systems, extensions such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Ensemble Kalman Filter have been adapted to distributed frameworks [8].

In addition to classical DKF frameworks, recent advances have proposed distributed optimal linear fusion methods that incorporate feedback from the fusion center to improve estimation accuracy.

Recent developments further integrate learning-based modules such as neural networks or graph neural networks into distributed filtering structures to compensate for model uncertainty, non-Gaussian noise, and dynamic environments [9]. These hybrid approaches enhance adaptability while preserving algorithmic structure and interpretability.

This review systematically summarizes the theoretical foundations of Kalman filtering, compares major distributed fusion strategies, explores linear and machine learning-based extensions, and highlights key challenges and future directions in scalability, asynchrony, and multi-objective optimization.

2 Methodology

This section outlines the methodology used to identify, evaluate, and categorize distributed Kalman filtering (DKF) algorithms. A simplified schematic of the multi-sensor fusion architecture is shown in Figure 1 to guide understanding of the system-level structure considered in this review.

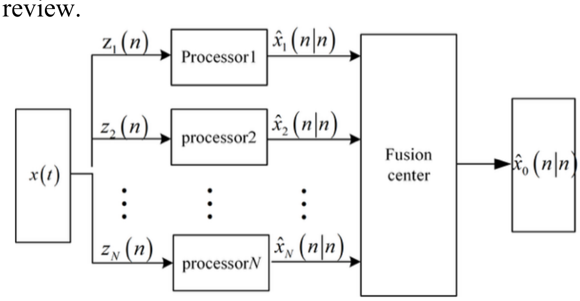

Figure 1. Distributed fusion structure

Considers a linear system with random parameter matrices:

$$x(n+1) = A(n)x(n) + B(n)w(n) \quad (1)$$

$$Z_i(n) = C_i(n)x(n) + v_i(n), i = 1,2, \dots, N \quad (2)$$

where $x(n) \in R^n$ is the system state, $Z_i(n)$ is the $measurement\ from\ sensor\ i, A(n)$ $, B(n)$ and $C_i(n)$ are matrices of appropriate dimensions, process noise $w(n)$ and observation noises $v_i(n)$ are uncorrelated Gaussian white noises with zero mean.

2.1 Local Kalman Filter

At each sensor node's local processor, implement the standard linear Kalman filter to estimate the state from that node's own measurements. The filter operates in two stages at each time step t: (1) Prediction propagate the prior state estimate and error covariance through the system model; (2) Update incorporate the new local measurement to correct the predicted state and covariance. Concretely, denoting the true state by $x(n)$, the local measurement by y(t), and the predicted and updated estimates by $\hat{x}(n|n-1)$ and $\hat{x}(n|n)$ (with covariances$P(n|n-1)$ and$P(n|n)$, the standard filter equations.

Prediction step [5]:

$$P(n|n-1) = \mathrm{A(n)}P_i(n-1|n-1)A(n)^{\mathrm{T}} + B(n)Q_\omega B(n)^{\mathrm{T}} \quad (3)$$

Update step:

$$K_i(n) = P_i(n|n-1)C_i^{\mathrm{T}}(n)(C_i(n)P_i(n|n-1)C_i^T(n) + Q_{v_i})^{-1} \quad (4)$$

$$\hat{x}_i(n|n) = \big(I_n - K_i(n)C_i(n)\big)\hat{x}_i(n|n-1) + K_i(n)Z_i(n) \quad (5)$$

$$P_i(n|n) = (I_n - K_i(n)C_i)P_i(n|n-1) \quad (6)$$

cross-covariance matrix

$$P_{ij}(n|n) = (I_n - K_i(n)C_i(n))P_{ij}(n|n-1)(I_n - K_j(n)C_j(n))^{\mathrm{T}} \quad (7)$$

$$P_{ij}(n|n-1) = A(n)P_{ij}(n-1|n-1)\mathrm{A}^{\mathrm{T}}(n) + D(n)Q_\omega D^T(n) \quad (8)$$

Here, x(n) denotes the system state vector at time step t and A(n) and B(n) the state-transition and control matrices, respectively. The term $w_t \sim \mathcal{N}(0, Q(n))$ represents process noise, while Z(n) is the local measurement vector and $C_i(n)$ the observation matrix. Measurement noise $v(n) \sim \mathcal{N}(0, R(n))$ is assumed Gaussian. The symbols $\hat{x}(n|n-1)$ and $\hat{x}(n|n)$ are the predicted and updated (filtered) state estimates with covariances $P(n|n-1)$ and $P(n|n)$, respectively. Finally, K(n) is the Kalman gain that optimally weights the new measurement in the update step.

2.1.1 Matrix-weighted Fusion

In matrix-weighted fusion method, the fusion center computes the optimal estimate by leveraging local estimates and their error covariances. Each local sensor sends its estimated state and covariance matrix to the fusion center. The fusion process includes two main steps: computing the global covariance and then fusing the estimates accordingly.

Matrix Weighted Fusion Filter:

$$\hat{x}_0(n|n) = \sum_{i=1}^{N} \Omega_i(n)\hat{x}_i(n|n) \tag{9}$$

Optimal Weight Matrix Calculation:

$$\Omega(n) = (e^T P^{-1}(n)e)^{-1} e^T P^{-1}(n) \tag{10}$$

Where $e = [I_n, \dots, I_n]^T, and\ P(t) = diag(P_1(n), \dots, P_N(n)) \in R^{nN \times nN}$

Error Covariance Matrix of the Matrix Weighted Fusion Filter:

$$P_m(n|n) = (e^T P^{-1}(n)e)^{-1} \tag{11}$$

Moreover, it holds that:

$$P_m(n|n) \le P_i(n|n), i = 1,2, \dots, N \tag{12}$$

2.1.2 Covariance Intersection (CI) Variants

$$P_{cI} = [\omega P_1^{-1} + (1-\omega)P_2^{-1}]^{-1} \tag{13}$$

$$\hat{x}_{CI} = P_{cI}[\omega P_2^{-1}\hat{x}_1 + (1-\omega)P_2^{-1}\hat{x}_2] \tag{14}$$

BCI ω for stability; ICI refines it iteratively.

Covariance intersection requires no knowledge of cross-covariances, so it always produces a consistent (non-overconfident) fused estimate even when correlations are unknown. However, it is inherently conservative and its fused covariance is generally larger than that of an optimal centralized filter. Thus, estimation accuracy can suffer, and tuning or iterating the weight ω adds extra computational overhead.

2.1.3 Linear Fusion Filter with Feedback

In linear fusion filter with feedback, the notable contribution is the distributed optimal linear fusion predictor and filter (DOLFPF), which addresses systems with random parameter matrices and correlated observation noises. The innovation of this algorithm lies in its feedback mechanism, where the fusion center's predicted state estimate is sent back to each local sensor node. These nodes then use this global prediction, rather than their own, to compute updated local estimates. The resulting fusion process achieves the same estimation accuracy as a centralized fusion predictor under certain conditions, while retaining the robustness and flexibility of a distributed structure.

$$\begin{aligned}\hat{x}_i(n|n-1) = A(n-1)(I_n - K_i(n-1)C_i) \\ \hat{x}_0(n-1|n-2) \\ +A(n-1)K_i(n-1)Z_i(n-1)\end{aligned} \tag{15}$$

$$\begin{aligned}P_i(n|n-1) = A(n)(I_n - K_i(n-1)C_i(-1) \\ P_0(n-1|n-2)A(n)^T + D(n)Q_\omega D(n)^T\end{aligned} \tag{16}$$

$$P_f(n) = (\sum i = 1^N P_i^{-1}(n))^{-1} \tag{17}$$

$$\hat{x}_f(n) = P_f(n) \sum i = 1^N P_i^{-1}(n)\hat{x}_i(n) \tag{18}$$

This method assumes that all local estimates are unbiased and uncorrelated. The feedback mechanism allows the fusion center to improve overall accuracy by sending back the fused estimate, which local nodes can use in the next round of updates.

2.1.4 Fusion Estimation Based Machine Learning

Fusion estimation based machine learning introduces learned models into the estimation pipeline to address complex, nonlinear, or uncertain environments where traditional models may fall short. Neural networks (NNs) or graph neural networks (GNNs) are commonly used to approximate optimal fusion functions.

$$\hat{x}f(x) = \mathcal{F}\theta(\hat{x}_1(k), \hat{x}_2(k), \dots, \hat{x}_N(k)) \tag{19}$$

A typical implementation may use a multilayer perceptron (MLP):

$$\mathcal{F}_\theta(x) = W^{(2)}\sigma(W^{(1)}x + b^{(1)}) + b^{(2)} \tag{20}$$

Where N is the number of sensor nodes, $P_i(n)$ is the local estimate covariance at sensor i, $\hat{x}i(n)$ is the local state estimate at sensor i, $\mathcal{F}\theta(\cdot)$ is the fusion function parameterized by $\theta = \{W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}\}$, and , $\sigma(\cdot)$ is the activation function such as ReLU or sigmoid.

3 Limitations and Challenges

Distributed Kalman filtering (DKF) provides a robust framework for multi-sensor state estimation. However, several practical challenges limit its deployment in large-scale, real-time, or resource-constrained systems. These challenges can be broadly grouped into two categories: those related to system scalability and numerical reliability, and those arising from incomplete sensing and conflicting design objectives.

3.1 Scalability and Numerical Reliability

A fundamental limitation in DKF arises from the curse of dimensionality. As the state dimension increases, the computational and communication costs of maintaining and transmitting full covariance matrices grow rapidly. In applications such as smart grids, autonomous systems, and high-resolution environmental sensing, this imposes substantial resource demands. While low-rank approximations, partitioned filters, and compressed sensing approaches offer potential relief, they may overlook important cross-covariances, leading to reduced estimation accuracy.

In addition, DKF algorithms especially nonlinear variants such as the EKF and UKF are susceptible to numerical instability. Modeling errors, asynchronous updates, packet loss, and noise accumulation can result in filter divergence. Conservative strategies like Covariance Intersection (CI) and its bounded variants can ensure consistency but may be overly cautious, sacrificing estimation precision and responsiveness.

3.2 Incomplete Information and Multi-Objective Constraints

In many practical settings, sensor observations are incomplete due to intermittent measurements, sensor failures, or limited coverage. These conditions can result in partial observability, where not all components of the system state are inferable at each time step. Solutions such as temporal smoothing, observability-aware graph design, or mobile sensor coordination aim to address this, but often increase latency and system complexity.

DKF must also address multiple conflicting objectives, including estimation accuracy, latency, energy consumption, communication overhead, and robustness to failure. Traditional filtering approaches, which optimize fixed cost functions, often fail to adapt to changing mission requirements or environmental dynamics. Recent advances incorporate adaptive filtering strategies, multi-objective optimization, and reinforcement learning to dynamically balance trade-offs, but achieving theoretical guarantees and real-time performance remains a significant research challenge.

4 Future Prospects and Development Trends

Despite recent advances, distributed Kalman-filtering for sensor networks still faces three broad, interrelated challenges that will shape future research: designing scalable and resource-aware fusion for ever-growing state dimensions, establishing robust fusion under asynchronous and unreliable communications, and integrating learning, security, and privacy into the estimation loop.

4.1 Scalable, Resource-Constrained Fusion for High-Dimensional and Edge Environments

Emerging applications in smart grids, environmental monitoring, and large-scale robotics routinely generate state vectors with thousands or tens of thousands of dimensions. At the same time, these filters must run on tiny IoT devices with limited memory, no floating-point hardware, and strict energy budgets. Future work must therefore marry two goals: exploiting underlying system structure such as low-rank subspaces, sparsity patterns, or graph partitions to factor the global covariance into tractable blocks, and co-designing the filtering algorithms with fixed-point arithmetic and lightweight hardware accelerators. Streaming dimensionality-reduction techniques like randomized sketching or incremental principal-component tracking can compress covariance updates on the fly, while formally quantized Kalman implementations must provide provable stability margins under finite precision. Only by tackling both algorithmic scalability and hardware constraints in concert can distributed fusion systems remain both accurate and deployable at the edge.

4.2 Asynchronous and Delay-Tolerant Fusion in Unreliable Networks

Real-world sensor networks rarely operate in perfect synchrony. Measurements may arrive out of order, clocks drift, links suffer packet loss, and nodes wake only on event triggers. Although event-driven and gossip-based fusion schemes have demonstrated impressive bandwidth savings, a general theoretical framework for asynchronous Kalman filtering remains elusive. Future research must treat timestamps and delays as intrinsic to the filter design characterizing conditions under which out-of-order or skipped updates preserve mean-square stability without full history buffering. Advances in randomized push-pull protocols and adaptive thresholding, which dynamically balance communication costs against estimation fidelity, will be essential. Establishing rigorous convergence and robustness guarantees under arbitrary timing jitter will unlock deployments in underwater sensor arrays, vehicular ad-hoc networks, and other harsh environments where synchronization is infeasible.

4.3 Learning-Enabled, Secure, and Privacy-Preserving Fusion

Classical Kalman filters assume perfect models and Gaussian noise, yet many systems exhibit nonlinearities, heavy-tailed disturbances, or time-varying parameters. Hybrid architectures that embed learned components into the Kalman loop hold great promise. For example, neural networks can predict residual model errors, adjust Kalman gain biases, or approximate expensive matrix inversions. Graph neural networks can learn topology-aware fusion weights that adapt to changing network quality. At the same time, fusion increasingly takes place over public or adversarial networks, demanding resilience to Byzantine nodes and protection of sensor privacy. Techniques such as encrypted state summaries, differentially private noise injection, or secure multiparty computation can safeguard raw measurements while still permitting accurate aggregation. Integrating these learning-based enhancements with formal guarantees of stability, security, and privacy represents one of the most ambitious frontiers in distributed estimation.

5 Conclusion

In this review, established a common framework for distributed Kalman filtering by detailing the system model and the standard local processor Kalman equations. Then surveyed four core algorithmic families, including the novel optimal linear fusion schemes with feedback that achieve the same accuracy as centralized filtering. The ensuing discussion identified four critical challenges scalability to very high dimensions; numerical and communication robustness; and conflicting objectives of accuracy, bandwidth and computation that currently limit real world deployments. Finally, outlined three high impact research directions: the design of structure exploiting filters for resource constrained edge platforms; the development of provably stable fusion methods under asynchronous and unreliable networks; and the integration of machine learning modules with formal security and privacy guarantees. Addressing these open challenges will require both theoretical breakthroughs and practical demonstrations.

References

1. Guo, D., & Wang, X. (2006). Quasi-Monte Carlo filtering in nonlinear dynamic systems. IEEE transactions on signal processing, 54(6), 2087-2098.
2. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. IEEE Communications magazine, 40(8), 102-114.
3. De Freitas, J. F. G., Niranjan, M., Gee, A., & Doucet, A. (1998). Sequential Monte Carlo methods for optimisation of neural network models. Cambridge University Engineering Department, Cambridge, England, Technical Report TR-328.
4. Simon, D. (2006). Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons.
5. Chisci, L., & Zappa, G. (1992). Square-root Kalman filtering of descriptor systems. Systems & control letters, 19(4), 325-334.
6. Olfati-Saber, R. (2005). Distributed Kalman filter with embedded consensus filters. In Proceedings of the 44th IEEE Conference on Decision and Control (pp. 8179-8184). IEEE.
7. Cheng, L., & Fang, L. (2013). Covariance intersection based distributed Kalman filters. IET Control Theory & Applications, 7(12), 1546–1553.
8. Hoteit, A., & Pham, D. T. (2009). The ensemble Kalman filter: Theoretical formulation and practical implementation. Ocean Dynamics, 59(3), 285–300.
9. Rajasekaran, M., & Thiagarajan, S. (2018). A survey on distributed Kalman filtering in sensor networks. IEEE Communications Surveys & Tutorials, 20(3), 2020–2035.