Decentralized Input and State Estimation for Multi-agent System with Dynamic Topology and Heterogeneous Sensor Network

Zida Wu[1], Ankur Mehta[1]

Abstract— A crucial challenge in decentralized systems is state estimation in the presence of unknown inputs, particularly within heterogeneous sensor networks with dynamic topologies. While numerous consensus algorithms have been introduced, they often require extensive information exchange or multiple communication iterations to ensure estimation accuracy. This paper proposes an efficient algorithm that achieves an unbiased and optimal solution comparable to filters with full information about other agents. This is accomplished through the use of information filter decomposition and the fusion of inputs via covariance intersection. Our method requires only a single communication iteration for exchanging individual estimates between agents, instead of multiple rounds of information exchange, thus preserving agents' privacy by avoiding the sharing of explicit observations and system equations. Furthermore, to address the challenges posed by dynamic communication topologies, we propose two practical strategies to handle issues arising from intermittent observations and incomplete state estimation, thereby enhancing the robustness and accuracy of the estimation process. Experiments and ablation studies conducted in both stationary and dynamic environments demonstrate the superiority of our algorithm over other baselines. Notably, it performs as well as, or even better than, algorithms that have a global view of all neighbors.

I. INTRODUCTION

Centralized systems are prevalent in real-world applications, requiring detailed information synchronization across components. However, scenarios like large-scale spatial multi-agent systems or sensor networks face challenges with the reliability and availability of global information in dynamic settings. Decentralized systems, in comparison, demonstrate significant resilience against individual node failures, offer enhanced scalability, and reduce synchronization demands [1]. For clarity and consistency, this paper uses a 'decentralized' multi-agent system to refer to systems where agents dynamically access information only from their immediate neighbors.

Decentralized estimation focuses on developing a fusion algorithm that gathers data from adjacent sensors to achieve consensus among local nodes, often through a consensus algorithm. These algorithms are divided into consensus-based [2]–[5], gossip-based [6], and diffusion-based [7], [8] methods based on their interaction and fusion techniques, as categorized in recent literature [9]. In consensus-based algorithms, each node communicates with connected neighbors to compute a weighted average of received states, iterating until discrepancies are minimized. Gossip-based algorithms

[1]Department of Electrical and Computer Engineering, University of California Los Angeles, Los Angeles, CA, USA. {zdwu,mehtank} @ucla.edu

similarly collect messages iteratively but randomly and asynchronously from neighborhoods, avoiding data aggregation from all. Diffusion-based methods, distinct in requiring only a single iteration between observations, perform an average of states post one iteration. Those methods require strong requirements of full communication rate, large communication range and bandwidth, and bring in correlated problems. Some algorithms, like max consensus [10], achieve consensus in one iteration by synchronizing states to a node within the neighborhood that owns the minimum variance; or based on pseudo-estimates [11], [12] to achieve global optimal but requires strict full communication rate, large bandwidth, and lacks of local optimal.
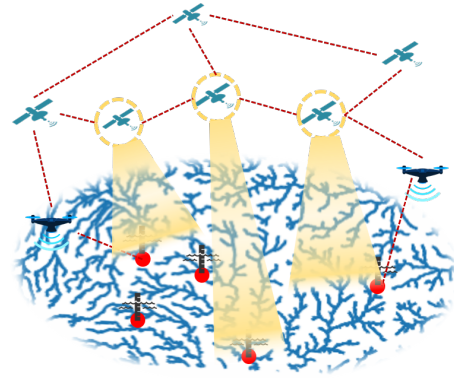

Fig. 1: An example application of a decentralized estimation system is geophysical environment detection, which involves a multi-layered, heterogeneous sensor network with a dynamic topology. Gauges within the river reach offer localized measurements, albeit within a confined spatial range. Satellites afford broader spatial observations but lack real-time monitoring capabilities. Drones complement the observations by providing additional data points.

Beyond state estimation, many scenarios require addressing unknown inputs affecting system dynamics, such as in geophysical detection, environmental monitoring [13] [14], maneuver tracking [15], flight detection [16], and sensor fault detection [17]. These unknown inputs, often resulting from unforeseen events, need to be estimated alongside the system's state. The single-agent unknown input and state estimation has been well-explored in the recursive Kalman filter (RKF) [18]. Decentralized state estimation with unknown inputs generally adopts one of two strategies: input estimation using individual information [19], or assuming agents can access and share system equations and observations with neighbors [20], applying a local recursive Kalman

filter for input estimation. However, the first approach lacks input consensus, while the second poses privacy risks due to the requirement of unrestricted bandwidth for data exchange and potential exposure of agent information.

This paper introduces a decentralized approach for input and state estimation within a sensor network featuring dynamic topology and heterogeneous sensors. Dynamic topology refers to the changing observation agents and neighborhoods over time, while heterogeneous sensors indicate different types of observations among agents. Instead of sharing system equations and observations, nodes only exchange their input and state estimations and covariances with neighbors. Input fusion utilizes covariance intersection [21]–[24], and state fusion, through information filter decomposition [25], achieves performance comparable to full-information filters without the correlation issues that arise if all-to-all communication is allowed. Our method also includes specific operations, such as reviewing observation time windows and state compensation, to enhance estimation in dynamic topologies with a limited sensing range. Our algorithm requires only a one-time information exchange and does not need all-to-all communication or a full communication rate. The messages exchanged avoid exposing system matrices or raw measurements, thereby preserving privacy.

II. Problem Formulation

A. System description

From the view of a single agent i, the system description is as follows:

$$x_k^i = A_{k-1}^i x_{k-1}^i + G_{k-1}^i d_{k-1}^i + w_{k-1}^i \quad (1)$$

$$y_k^i = H_k^i x_k^i + v_k^i \quad (2)$$

where the noise w_k^i and v_k^i are independent, zero-mean, white Gaussian noises with covariance matrices are Q^i and R^i. We primarily focus on homologous states and inputs across agents, meaning system dynamics are consistent. Therefore for simplicity, we assume $A_k^i = A_k$, $G_k^i = G_k$, $Q_k^i = Q_k$, state $x_k^i \in \mathbb{R}^n$, input $d_k^i \in \mathbb{R}^m$, and output $y^i \in \mathbb{R}^{p^i}$ with varying p^i across different observation models. In this system, both state x_k^i and input d_k^i are unknown, with each agent aiming to optimally estimate them.

Agents start with individual input estimation using the initial steps of the recursive Kalman filter (RKF) [16], [18], where they predict the state and covariance according to Eq. (3) and estimate the input via Eq.(4).

$$\hat{x}_{k|k-1}^i = A_{k-1}^i \bar{x}_{k-1|k-1}^i$$
$$\hat{P}_{k|k-1}^i = A_{k-1}^i \bar{P}_{k-1|k-1}^i A_{k-1}^{i^T} + Q_k^i \quad (3)$$

Input estimation:

$$\hat{d}_{k-1}^i = M_k^i \left(y_k^i - H_k^i \hat{x}_{k|k-1}^i \right) \quad (4)$$

where M_k^i is:

$$M_k^i = [(H_k^i G_{k-1}^i)^T \tilde{R}_k^{i^{-1}} (H_k^i G_{k-1}^i)]^{-1} (H_k^i G_{k-1}^i)^T \tilde{R}_k^{i^{-1}} \quad (5)$$

$$\tilde{R}_k^i = H_k^i \hat{P}_{k|k-1}^i H_k^{i^T} + R_k^i \quad (6)$$

The covariance of input estimation is,

$$\hat{S}_k^i(\hat{d}_k^i) = [(H_k^i G_{k-1}^i)^T (\tilde{R}_k^i)^{-1} (H_k^i G_{k-1}^i)]^{-1} \quad (7)$$

The rank condition for input estimation is the same as RKF,

$$rank(H_k^i G_{k-1}^i) = rank(H_{k-1}^i) = m \quad (8)$$

Following individual estimations, agents proceed differently from RKF by performing a pure Kalman filter (KF) update without including input estimation. This choice avoids extra computation as input estimation's inclusion does not alter the fusion result in the final algorithm but increases the computational load. The explanation will be detailed in the state fusion section.

$$\tilde{x}_{k|k}^i = \hat{x}_{k|k-1}^i + L_k^i \left[y_k^i - H_k^i \hat{x}_{k|k-1}^i \right]$$
$$\tilde{P}_{k|k}^{i^{-1}} = \hat{P}_{k|k-1}^{i^{-1}} + H_k^i R_k^{i^{-1}} H_k^i \quad (9)$$
$$L_k^i = \tilde{P}_{k|k}^i H_k^{i^{-1}} R_k^{i^{-1}}$$

Each agent's estimation results in input and state estimations along with their covariances, presented as $\hat{d}_k^i$, $\hat{S}_k^i$, $\hat{x}_{k|k-1}^i$, $\hat{P}_{k|k-1}^i$, $\tilde{x}_{k|k}^i$, and $\tilde{P}_{k|k}^i$, adhering to the unbiased and minimum-variance (MVU) principle. Due to bandwidth and privacy limitations, transmitting direct observation or sharing agents' system dynamics or observation models is infeasible. This paper proposes a fusion algorithm using only these values to achieve an approximate optimal fusion result, bypassing the need for system information exchange. The fusion algorithm comprises two parts: input fusion and state fusion, which are explored in subsequent sections.

III. Input fusion via Covariance Intersection

Within the 1-hop distance of node i, including itself, represented by $\mathcal{N}_i$, communication is confined to $\mathcal{N}_i$. The correlation among received input estimations d_j ($j \in \mathcal{N}^i$) is unknown, preventing optimal fusion akin to fully informed filters [20]. We employ a method known as covariance intersection (CI) for fusion under uncertain correlation. CI, typically used in Kalman filter-based information fusion when source correlations are unclear, offers a way to merge information without complete correlation knowledge, albeit solving its non-linear optimization problems can be computationally intensive. This paper introduces a rapid CI approach [9], [21], [26] to maintain fusion consistency and ensure real-time computational feasibility.

$$\bar{S}_k^{i^{-1}} \bar{d}_{k-1}^i = \sum_j^{\mathcal{N}_i} \omega_j \hat{S}_k^{j^{-1}} \hat{d}_k^j \quad (10)$$

where weights are decided by faster CI which avoids complex matrices manipulation and non-linear optimization.

$$\omega_j = \frac{1/\operatorname{tr} \hat{S}_j}{\sum_{n=1}^{\mathcal{N}_i} 1/\operatorname{tr} \hat{S}_n} \quad (11)$$

$$\bar{S}_k^{\,i} = \left(\sum_{j \in \mathcal{N}_i} w_j \hat{S}_k^{j^{-1}} \right)^{-1} \qquad (12)$$

Given that all input estimations $\hat{d}^j$ are unbiased, the fusion process described in Eq.(10) remains unbiased. It's important to note that this covariance fusion method does not affect future input variance predictions.

In scenarios when moving agents with dynamic communication topology, the target is intermittently observed. If input estimation is constantly employed, abrupt measurements may be misinterpreted as input influences affecting state estimates, which leads to a significant bias. To mitigate this, we propose a specific operation: establishing an observation time window review mechanism, the length of the window can be regarded as a hyperparameter. If the gap between two consecutive measurements exceeds this window, the initial estimation upon receiving new observations will exclude input estimation. After implementing a review through the observation window before input estimation, our algorithm can mitigate this sensitivity issue. However, the time window should be reset to zero if valid input estimates from neighborhoods are available during the input fusion stage. The success of this approach is demonstrated in Fig. 5.

IV. State fusion via information filter decomposition

Rather than iterative state exchange for consensus among agents, state fusion leverages information Kalman filter decomposition [25]. This fusion approach ensures performance equivalent to an agent with complete neighborhood information in both state estimation and covariance matrix.

After input fusion, yielding fused input estimation $\bar{d}_k^i$ and covariance $\bar{S}k^i$, we proceed from an information Kalman filter perspective, incorporating all node data within $\mathcal{N}_i$. The final state estimation relies solely on $\bar{d}_k^i$, $\hat{x}_{k|k-1}^i$, $\hat{P}_{k|k-1}^i$, $\tilde{x}_{k|k}^i$, $\tilde{P}_{k|k}^i$.

In order to avoid symbol chaos of equations, we use superscript $*$ on states and covariance to indicate the state fusion process in contrast to the process before communication. For node i, incorporating input fusion into the prediction process yields:

$$\begin{aligned}
\hat{x}_{k|k-1}^{*^i} &= A_{k-1}^i \bar{x}_{k-1|k-1}^i + G_{k-1}^i \bar{d}_{k-1}^i \\
\hat{P}_{k|k-1}^i &= A_{k-1}^i \bar{P}_{k-1|k-1}^i A_{k-1}^{i^T} + Q_k^i
\end{aligned} \qquad (13)$$

This operation is justified because the input covariance impacts only the Kalman filter's gain matrix, potentially leading to non-unique Kalman gains and additional computational load. Previous research [18], [27] indicates that omitting input covariance injection still yields near-optimal Kalman gain. Moreover, exact state-input correlation would require full distributed agent system knowledge, deviating from our objective. Various consensus algorithms [3], [5], [6], [28] demonstrate that fusion with approximate covariance can achieve satisfactory performance, hence this paper proceeds without detailed proof of covariance fusion's impact.

Assuming a virtual information filter with H_k, y_k, and R_k representing aggregated matrices from $\mathcal{N}i$, we simulate all observations being accessible to node i. Importantly, we show the final fusion result does not depend on external observations or matrices. Maintaining the prediction from Eq.(13), the update process follows:

$$\begin{aligned}
\bar{x}_{k|k}^i &= \hat{x}_{k|k-1}^{*^i} + W_k \left[y_k - H_k \hat{x}_{k|k-1}^{*^i} \right] \\
\tilde{P}_{k|k}^{i^{-1}} &= \hat{P}_{k|k-1}^{*^{i^{-1}}} + H_k R_k^{-1} H_k \\
W_k &= \tilde{P}_{k|k}^i H_k^\top R_k^{-1}
\end{aligned} \qquad (14)$$

Substitute Eq.(13) to Eq.(14),

$$\bar{x}_{k|k}^i = \bar{P}_{k|k}^i [H_k^\top R_k^{-1} y_k + \hat{P}_{k|k-1}^{*^{i^{-1}}} (\hat{x}_{k|k-1}^{*^i} + G_{k-1} \bar{d}_{k-1}^i] \qquad (15)$$

After unpacking the observation vector,

$$H_k^\top R_k^{-1} y_k = \sum_{j \in \mathcal{N}} H_k^{jT} R_k^{j^{-1}} y_k^j \qquad (16)$$

$$H_k^\top R_k^{-1} H_k = \sum_{j \in \mathcal{N}} H_k^{jT} R_k^{j^{-1}} H_k^j \qquad (17)$$

Up to now, the state fusion is derived as:

$$\begin{aligned}
\bar{x}_{k|k}^i = \bar{P}_{k|k}^i \big[\sum_{j \in \mathcal{N}} (\tilde{P}_{k|k}^{j^{-1}} \tilde{x}_{k|k}^j - \hat{P}_{k|k-1}^{j^{-1}} \hat{x}_{k|k-1}^j) \\
+ \hat{P}_{k|k-1}^{*^{i^{-1}}} (\hat{x}_{k|k-1}^i + G_{k-1}^i \bar{d}_{k-1}^i) \big]
\end{aligned} \qquad (18)$$

$$\bar{P}_{k|k}^{i^{-1}} = \hat{P}_{k|k-1}^{*^{i^{-1}}} + \sum_{j \in \mathcal{N}} (\tilde{P}_{k|k}^{j^{-1}} - \hat{P}_{k|k-1}^{j^{-1}}) \qquad (19)$$

Initially, agents estimate $\hat{d}_k^i$, $\hat{S}_k^i$, $\hat{x}_{k|k-1}^i$, $\hat{P}_{k|k-1}^i$, $\tilde{x}_{k|k}^i$, $\tilde{P}_{k|k}^i$ independently. Post-communication, input and state fusion follows Eq.(10), Eq.(12), Eq.(18), and Eq.(19), outlined in Algo.1. As this decomposition shows the exact same as the central estimator, the final fused states also have an unbiased and minimum-variance guarantee.

For non-all-to-all and dynamic communication topologies, a diffusion operation post-Eq.(18) helps correct state estimation errors:

$$\bar{x}_{k|k}^i = \bar{x}_{k|k}^i + \epsilon M_i \sum_{j \in \mathcal{N}} (\hat{x}_{k|k-1}^{*^j} - \hat{x}_{k|k-1}^{*^i}) \qquad (20)$$

with $M_i = \bar{P}_{k|k}^i$ and ϵ set to 0.05. This state adjustment, averaging predictions, differs from consensus or diffusion KF, which requires further information exchange post-fusion. Our decomposed algorithm naturally zeros out such an adjustment in all-to-all communication networks, offering advantages over consensus-based filters. The ablation study of the state compensation is presented in Table I.

Algorithm 1 Decentralized input and state estimation on node i based on information filter

input : Given system models A_k^i, G_k^i, H_k^i, Q_k^i, R_k^i, dynamic 1-hop neighborhoods $\mathcal{N}^i$.

output: The input fusion $\bar{d}_k^i$, state fusion $\bar{x}_{k|k}^i$, and their respective covariance $\bar{S}_k^{\,i}$ and $\bar{P}_{k|k}^i$.

Initialize $x_0^i = x_0$.

  for *timestep* $k = 1, 2, ...$ **do**

    1. **Individual estimation**

    All agents independently perform state estimation as Eq.(9). Check if the measurement gap is less than a time window and perform input estimation as Eq.(4).

    2. **Communication**

    Exchange and collect information with available nodes within neighborhoods $\mathcal{N}^i$.

    $\bar{d}_k^i$, $\bar{S}_k^i$, $\hat{x}_{k|k-1}^i$, $\hat{P}_{k|k-1}^i$, $\tilde{x}_{k|k}^i$, $\tilde{P}_{k|k}^i$

    3. **Input fusion**

    Perform input fusion as Eq.(10) and Eq.(12).

    4. **State fusion**

    Perform state fusion as Eq.(18) and Eq.(19), add state compensation as Eq.(20).
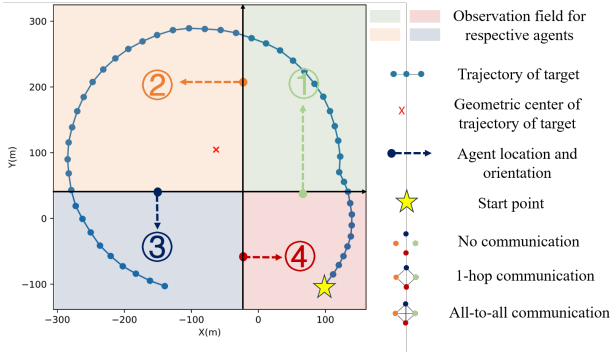
end

Fig. 2: Scenario 1: Four static agents are located in the four quadrants. Two of them can observe only the y-coordinate of the target, while another two are limited to observing only the x-coordinate. Each agent is tasked with observing the target exclusively when it enters its respective quadrant. The communication is established according to three distinct typologies.

V. SIMULATION AND EXPERIMENT RESULT

A. Experiment Setup

In this section, we describe two challenging experiments inspired by a canonical scenario, where a target moves around a base (e.g., the coordinate origin), with its movement influenced by internal unknown inputs, such as sensor faults. Consequently, the trajectory's geometric center deviates from the coordinate origin, as depicted in Fig.2. The target's path is segmented into four quadrants, each monitored by a sensor that detects only one dimension of the target's position. All experiment results shown in this paper are carried on under 20 different seeds and averaged. Table II indicates the
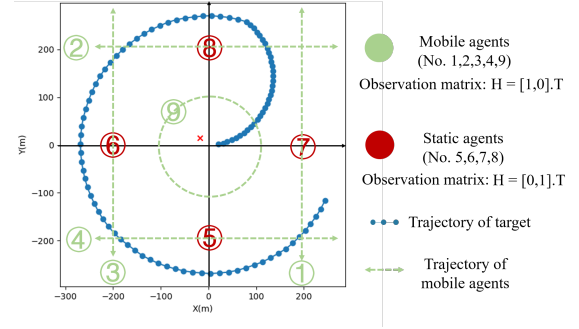

Fig. 3: Scenario 2: 4 static (red) agents and 5 mobile (green) agents compose a dynamic system. Observation of the target by an agent is contingent on sharing the same quadrant with the target, with each agent limited to observing a single target dimension. The trajectories of the mobile agents are delineated by green dashed lines. Each agent possesses a pre-defined communication range, enabling information exchange solely when the communication ranges of two agents intersect.

average of statistics of the first experiment. Fig. 9 shows the MAE, RMSE as well as variance from all seeds of the second dynamic topologies. Table I showcases the average statistics of state compensation in both two experiments. In the first example, agents are static to observe a moving target. Each of the agents has a different observation model. Specifically, Sensor No.1 and No.3 can measure the x position in its quadrant, $H_1 = H_3 = [1, 0]$, while Sensor No.2 and No.4 will measure the y position, $H_2 = H_4 = [0, 1]$. These sensors ensure at any moment, the target is tracked by only one sensor for a singular dimension. During this experiment, observations from a central agent are absent akin to environmental monitoring practices [13].

In the second example, the sensor network consists of two groups: dynamic and static sensors. Most settings are similar to the first example. In particular, the communication topology is fully dynamic, as each agent has its own communication field. A communication link is established only when agents are within range to communicate. The communication range in this paper is 120, the detail is seen in Fig. 3.

The system matrices describe a dynamic model that makes the target move in a circular trajectory.

$$A_k = \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{bmatrix} \qquad (21)$$

where $\theta_k = \omega + \theta_0$,ω is the angular velocity. k is the timestep, θ_0 is the initial angle.

$$G_k = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad B_k = \begin{bmatrix} d(t) \end{bmatrix} \qquad (22)$$

where $d(t)$ is the internal unknown input that cannot be predicted and observed directly, like the internal sensor fault in a sensor network. To mimic the changes of inputs over

time. The function of $d(t)$ is:

$$d(t) = \begin{cases} 0, & \text{if } k < 10 \\ 10, & \text{if } k \geq 10 \end{cases} \quad (23)$$

$$Q_i = [0.2, 0.2]^T; R_i = [2]^T; i \in \{1, 2, 3, 4.., N\} \quad (24)$$

Our algorithm is denoted as DISKF (Decentralized Input and state Kalman filter), and the baselines are [20] which assume agents can have a global view of system equations of neighborhoods, as it seems like a locally centralized filter, so we call it C-DRKF; L-DRKF [19], [29] which do a state consensus but with individual input estimation; KCF [6] which is the representative of consensus algorithms which will take a weighted average of exchanged states; DiffKF [7] which is the representative of diffusion-based algorithms which will average states of neighborhoods after each agent performs a consensus of states; GDKF [10] is a gossip-based algorithm which will randomly and asynchronously collect message from neighborhoods. For fairness, all algorithms are only allowed to communicate once between two consecutive timesteps.

B. Result

Given the page constraints, we only provide a summary of the 4-agents case with 1-hop communication and the dynamic scenarios with 9 agents.

Firstly, we illustrate input estimation using 1-hop communication in Fig.4, whereas some baselines like KCF, DiffKF, and GDKF lack detailed input estimation in their documentation. For fairness, we adopted the RKF's input estimation method [18] for comparison with these algorithms, noting that this adaptation only serves for comparison but didn't affect state estimation; the original algorithms' state estimations were preserved.

In 1-hop limited communication where only one sensor observes the target at any time, no agent can have constant information about the target, thus leading to missing data in Fig.4. Algorithms without specialized input designs, like KCF, DiffKF, and GDKF, exhibit significant bias However, filters with comprehensive information of neighbors may also face challenges, as indicated by the bias in input estimation for nodes 3 and 4 when the target enters quadrants 3 and 4. This occurs because before entering quadrant 3, node 3's state and covariance evolve fully depending on the prediction model which makes state error significantly increases, whereas node 2's estimation adjusts upon the target's entry into quadrant 1. Based on Eq. (4), the input estimation depends on the observation, current states, and its covariance. Then if using node 2's observation but with node 3's state, the estimator will misinterpret the innovation as the influence of unknown input. This is a similar issue we discussed in the observation time window chapter. However, without state and input fusion, even the C-DRKF with full observation and corresponding matrices still cannot solve this issue. Our fusion method, considering both input and state along with their covariances, maintains stable input estimation.
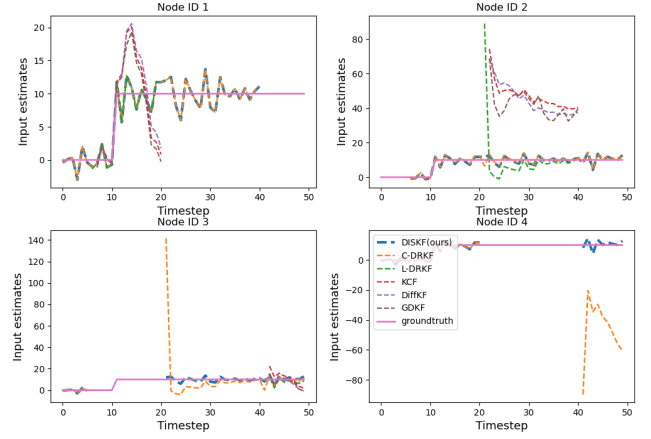

Fig. 4: The input estimation with 1-hop communication. The index of nodes corresponds to Fig.2. Missing estimation means the corresponding algorithm can't recover the input from the data they observe or exchange.

State estimations are depicted in Fig.6 and Fig.7, demonstrating that a sensor's singular dimensional observation leads to asynchronous x and y estimation trends. Algorithms lacking input estimation invariably present bias in state estimations. While Kalman filters can naturally compensate for sudden disturbances over time, continuous or lasting inputs can still introduce bias during estimation.

Estimated trajectories in Fig.8 showcase our algorithm's robustness in scenarios marked by incomplete observations and unknown internal inputs. Despite limited information, our algorithm compares favorably or even outperforms others.

Additionally, we examine input and state estimations across three different topologies in Fig.2, using agent node 2 as an example. Fig.10 illustrates that isolated estimations by each agent yield comparable performance among algorithms with and without input estimation. However, under all-to-all communication, our algorithm (blue line) mirrors the performance of the fully informed C-DRKF filter, aligning with our theoretical expectations, which can also shown in Table II.

TABLE I: Evaluation of State Compensation Effect

Env.	Comm.	State Comp.	MAE (input)	RMSE(input)	MAE(state)	RMSE(state)
Stationary	All-to-all	Yes	4.63	6.35	1.91	2.66
		No	4.63	6.35	1.91	2.66
	1-hop	Yes	**15.76**	**30.05**	**4.67**	**6.37**
		No	26.75	55.63	7.38	15.36
Dynamic	All-to-all	Yes	1.56	4.01	7.40	11.08
		No	1.56	4.01	7.40	11.08
	1-hop	Yes	**10.37**	**19.34**	**8.82**	**14.22**
		No	16.14	26.47	9.11	14.76

Table II displays statistical data for scenarios involving 4 stationary agents, evaluated by Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics. The results demonstrate enhanced estimation accuracy with increased communication, with our algorithms surpassing all baseline models consistently. Figure 9 showcases statistical outcomes for a dynamic system, highlighting dynamic
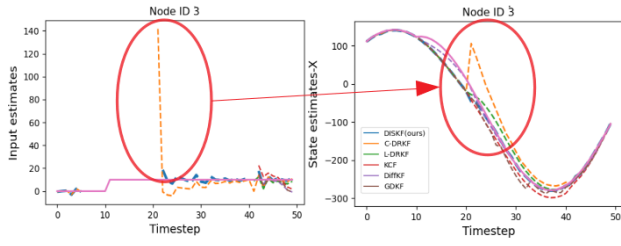
(a) Abrupt measurements induce huge bias in input estimates (left) and then influence state (right).
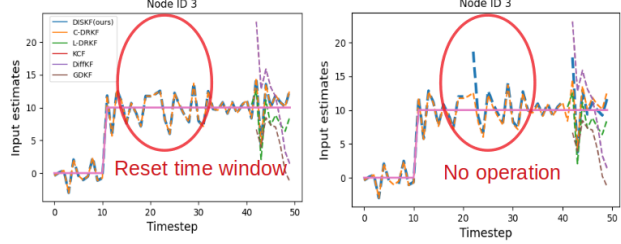
(b) Reset time window counting during input fusion improves estimations

Fig. 5: Observation time window helps deal with intermittent observations in dynamic topology. (a) the sensitivity issue is mitigated by implementing a review through the observation window before estimating inputs. (b) the time window counting should be reset when valid input estimations are available from neighborhoods.

topology with a 120 communication range and all-to-all communication. This figure reveals that our algorithm not only excels in performance within the dynamic topology but also matches the performance of C-DRKF in scenarios of all-to-all communication. The C-DRKF, requiring a global perspective of other agents, represents global optimality in all-to-all communication settings. Our algorithm, requiring only suitable information exchange yet achieving comparable performance, is remarkable. In addition, an in-depth analysis explores the impact of communication range, as illustrated in Fig. 11. This figure confirms that our algorithm consistently outperforms across different communication ranges. Unlike the C-DRKF algorithm, which experiences unstable estimation due to intermittent observation issues in dynamic topologies, our algorithm effectively manages this issue, aligning with theoretical predictions.
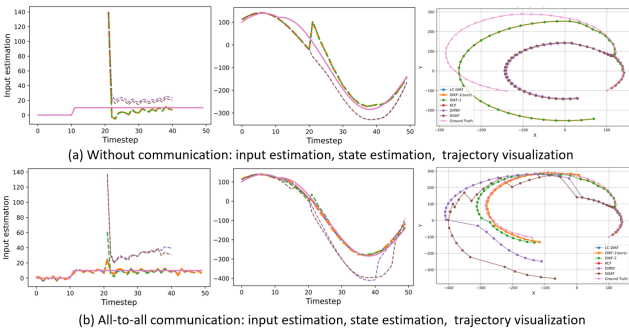

(a) Without communication: input estimation, state estimation, trajectory visualization

(b) All-to-all communication: input estimation, state estimation, trajectory visualization

Fig. 10: Supplementary results of communication without communication and all-to-all communication of node 2.
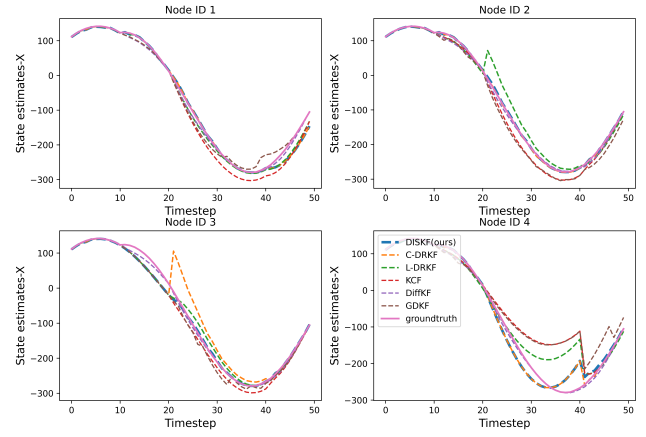

Fig. 6: The x coordinate of position estimation of the target with 1-hop communication. 1) Our algorithm (blue line) can converge back to groundtruth very quickly once one of the nodes within 1-hop has the observation of the target. 2) even with limited information exchange, our algorithm is still very close to filtering (blue line) with full information.
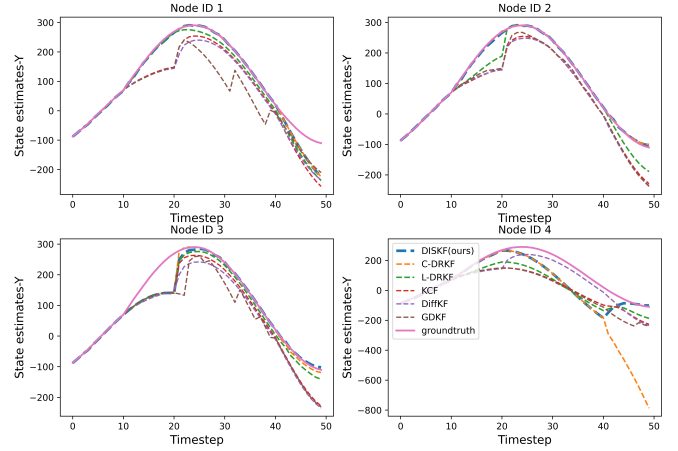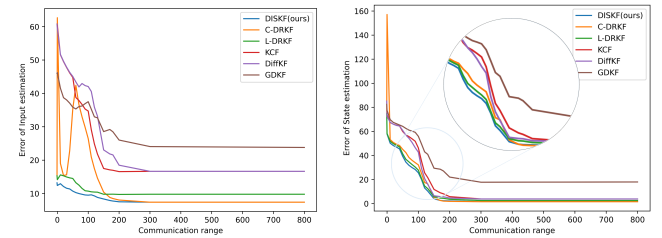

Fig. 7: The y coordinate of position estimation of the target with 1-hop communication.

(a) Input estimation versus radius

(b) State estimation versus radius

Fig. 11: Algorithm performance versus communication ranges in the dynamic topology scenario. (a) and (b) show the Mean Absolute Error (MAE) of input and state estimation under different communication ranges of each agent.

Table I demonstrates the effectiveness of state compensation, as detailed in Eq.(20). As noted earlier, in the scenario where there is all-to-all communication, state compensation naturally reduces to zero—a unique attribute not seen in
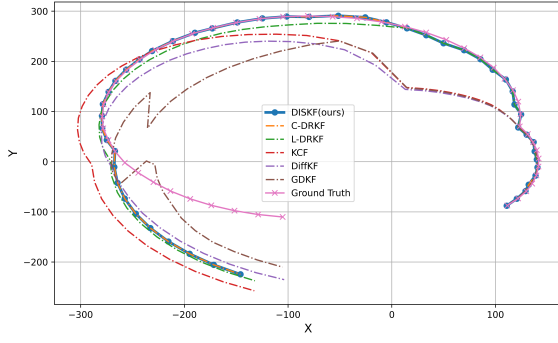
Fig. 8: 2D trajectory visualization of all estimators with 1-hop communication. Our algorithm (blue line) exhibits robustness against unexpected internal input disturbance while maintaining performance parity with the filter that possesses complete neighborhood information.

TABLE II: State and Input Estimation Error in Four Agents Case under Stationary Topology with Heterogeneous Sensors

Comm.	Metric	DISKF (ours)	C-DRKF [20]	L-DRKF [19]	KCF [3]	DiffKF [7]	GDKF [10]
all-to-all	MAE(state)	1.91	1.91	15.66	27.90	27.90	36.20
	RMSE(state)	2.66	2.66	25.05	40.45	40.45	50.15
	MAE(input)	4.63	4.63	6.87	15.39	15.39	14.01
	RMSE(input)	6.35	6.35	9.75	16.51	16.51	15.12
1-hop	MAE(state)	15.76	26.76	26.08	37.40	24.87	40.92
	RMSE(state)	30.05	55.63	41.54	53.41	40.84	58.89
	MAE(input)	4.67	9.73	6.30	15.11	14.53	13.35
	RMSE(input)	6.37	17.34	9.72	16.94	15.47	14.33
None	MAE(state)	63.75	71.01	71.01	66.43	66.43	65.82
	RMSE(state)	92.88	109.31	109.31	90.78	90.78	90.36
	MAE(input)	9.10	21.76	21.76	16.32	16.32	12.65
	RMSE(input)	12.21	32.33	32.33	17.95	17.95	14.50

(a) Input estimation (dynamic)

(b) State estimation (dynamic)

(c) Input estimation (all-to-all)

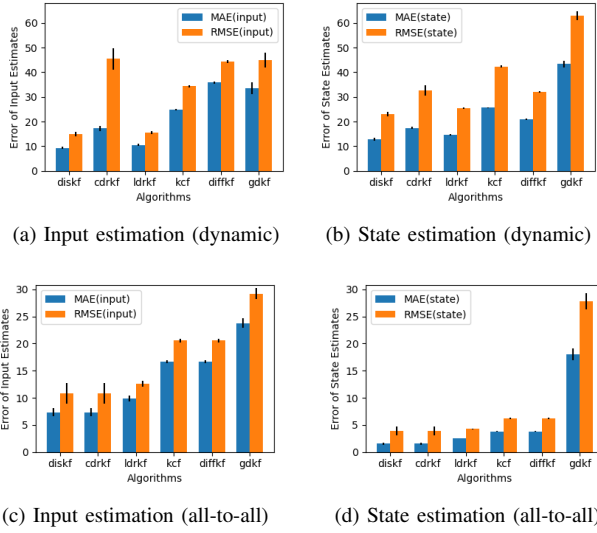(d) State estimation (all-to-all)

Fig. 9: DISKF(ours) shows superiority over all baselines in multi-agent case under dynamic topology with heterogeneous sensors. (a)(b): Input and state estimation error with dynamic communication topology (c): Input and state estimation error with all-to-all communication

dynamic multi-agent systems that consist of heterogeneous sensors with intermittent observations for each agent. This enables us to extend to cooperative tracking, ad-hoc sensor networks, and large spatial-temporal systems like geophysical monitoring, with real-time and privacy demand.

other consensus algorithms, as the statistical data underlines. In a dynamic communication topology, state compensation improves the accuracy of estimations. The relative advantage of state compensation is also revealed in the analysis between C-DRKF and L-DRKF. Unlike C-DRKF, L-DRKF incorporates an additional operation of state compensation (consensus) beyond input estimation. Hence, despite only relying on local information for input estimation, L-DRKF outperforms C-DRKF in certain situations.

VI. CONCLUSION

This paper introduces a decentralized, resilient, and efficient input and state estimation algorithm. Through system decomposition, state compensation, and observation time window review techniques, our algorithm can deal with

REFERENCES

[1] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control Systems Magazine*, vol. 39, no. 3, pp. 40–72, 2019.

[2] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[3] R. Olfati-Saber and N. F. Sandell, "Distributed tracking in sensor networks with limited sensing range," in *2008 American Control Conference*, pp. 3157–3162, IEEE, 2008.

[4] A. T. Kamal, C. Ding, B. Song, J. A. Farrell, and A. K. Roy-Chowdhury, "A generalized kalman consensus filter for wide-area video networks," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 7863–7869, IEEE, 2011.

[5] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Information weighted consensus," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 2732–2737, IEEE, 2012.

[6] K. Ma, S. Wu, Y. Wei, and W. Zhang, "Gossip-based distributed tracking in networks of heterogeneous agents," *IEEE Communications Letters*, vol. 21, no. 4, pp. 801–804, 2016.

[7] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed kalman filtering and smoothing," *IEEE Transactions on automatic control*, vol. 55, no. 9, pp. 2069–2084, 2010.

[8] V. Vahidpour, A. Rastegarnia, A. Khalili, and S. Sanei, "Partial diffusion kalman filtering for distributed state estimation in multiagent networks," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 12, pp. 3839–3846, 2019.

[9] S. He, H.-S. Shin, S. Xu, and A. Tsourdos, "Distributed estimation over a low-cost sensor network: A review of state-of-the-art," *Information Fusion*, vol. 54, pp. 21–43, 2020.

[10] A. Petitti, D. Di Paola, A. Rizzo, and G. Cicirelli, "Consensus-based distributed estimation for target tracking in heterogeneous sensor networks," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 6648–6653, IEEE, 2011.

[11] F. Govaers and W. Koch, "An exact solution to track-to-track-fusion at arbitrary communication rates," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2718–2729, 2012.

[12] F. Govaers, C.-Y. Chong, S. Mori, and W. Koch, "Comparison of augmented state track fusion methods for non-full-rate communication," in *2015 18th International Conference on Information Fusion (Fusion)*, pp. 862–869, IEEE, 2015.

[13] B. Gorr, A. A. Jaramillo, Z. Wu, W. Cho, K. Cheng, M. Stroud, V. Ravindra, C. David, H. Gao, Y. Sun, A. Mehta, G. Allen, and D. Selva, "Multi-instrument flood monitoring with a distributed, decentralized, dynamic and context-aware satellite sensor web," in *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2023.

[14] A. A. Jaramillo, B. Gorr, V. Ravindra, C. David, M. Stroud, A. Mehta, G. Allen, W. Cho, K. Cheng, H. Gao, *et al.*, "Decentralized market-based observation assignment strategy for dynamic networks in sensor web mission concepts," in *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*, pp. 4748–4751, IEEE, 2023.

[15] Y.-L. Lee and Y.-W. Chen, "Imm estimator based on fuzzy weighted input estimation for tracking a maneuvering target," *Applied mathematical modelling*, vol. 39, no. 19, pp. 5791–5802, 2015.

[16] Z. Wu, Z. Zheng, and A. Mehta, "Joint state and input estimation of agent based on recursive kalman filter given prior knowledge," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 1675–1681, IEEE, 2022.

[17] M. R. Davoodi, K. Khorasani, H. A. Talebi, and H. R. Momeni, "Distributed fault detection and isolation filter design for a network of heterogeneous multiagent systems," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 1061–1069, 2013.

[18] S. Gillijns and B. De Moor, "Unbiased minimum-variance input and state estimation for linear discrete-time systems," *Automatica*, vol. 43, no. 1, pp. 111–116, 2007.

[19] Z. Peng, Y. Zhang, G. Wen, S. Yang, and T. Huang, "Optimal distributed filter for state and unknown input estimations in multi-sensor networks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2023.

[20] C. Liu, Y. Wang, D. Zhou, and X. Shen, "Minimum-variance unbiased unknown input and state estimation for multi-agent systems by distributed cooperative filters," *IEEE Access*, vol. 6, pp. 18128–18141, 2018.

[21] X. He, W. Xue, and H. Fang, "Consistent distributed state estimation with global observability over sensor network," *Automatica*, vol. 92, pp. 162–172, 2018.

[22] T.-K. Chang, K. Chen, and A. Mehta, "Resilient and consistent multirobot cooperative localization with covariance intersection," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 197–208, 2021.

[23] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the 1997 American Control Conference (Cat. No. 97CH36041)*, vol. 4, pp. 2369–2373, IEEE, 1997.

[24] L. Chen, P. O. Arambel, and R. K. Mehra, "Estimation under unknown correlation: Covariance intersection revisited," *IEEE Transactions on Automatic Control*, vol. 47, no. 11, pp. 1879–1882, 2002.

[25] H. F. Durrant-Whyte, B. Rao, and H. Hu, "Toward a fully decentralized architecture for multi-sensor data fusion," in *Proceedings., IEEE International Conference on Robotics and Automation*, pp. 1331–1336, IEEE, 1990.

[26] W. Niehsen, "Information fusion based on fast covariance intersection filtering," in *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.(IEEE Cat. No. 02EX5997)*, vol. 2, pp. 901–904, IEEE, 2002.

[27] P. K. Kitanidis, "Unbiased minimum-variance linear state estimation," *Automatica*, vol. 23, no. 6, pp. 775–778, 1987.

[28] M. Kamgarpour and C. Tomlin, "Convergence properties of a decentralized kalman filter," in *2008 47th IEEE Conference on Decision and Control*, pp. 3205–3210, IEEE, 2008.

[29] A. Emami, R. Araujo, and A. Asvadi, "Distributed simultaneous estimation of states and unknown inputs," *Systems & Control Letters*, vol. 138, p. 104660, 2020.