A Distributed Kalman-like Observer with Dynamic Inversion-Based Correction for Multi-Agent Estimation

Nicola De Carli, Dimos V. Dimarogonas

***Abstract*— We propose a distributed Kalman-like observer for cooperative state estimation in multi-agent systems. By replacing the process covariance matrix with a forgetting factor, the observer enables distributed information matrix propagation while maintaining stability. The correction term is computed dynamically in a distributed manner, avoiding centralized matrix inversion. Unlike existing methods, our approach preserves inter-agent coupling and only requires joint observability, offering flexibility in sensing configurations. Stability guarantees are provided, and numerical simulations in a cooperative localization scenario show the effectiveness of the method in state estimation.**

I. Introduction

Cooperative state estimation is fundamental in multi-agent systems such as robotic networks and distributed monitoring, enabling agents to estimate their states using local and relative measurements. Key challenges include scalability, communication efficiency, and ensuring stability. While distributed Kalman filters (DKFs) are widely used, especially in robotics and power systems [1]–[3], existing variants often lack formal stability guarantees and discard useful inter-agent information to enable distributed computation.

In this work, we propose a distributed observer inspired by Kalman-like observers, which, in the Riccati equation, substitute the process covariance matrix with a forgetting factor. This formulation enables the distributed propagation of the information matrix dynamics, enabling each agent to compute its correction term by dynamically solving a linear equation in a distributed manner which only requires communication between neighboring agents. Our observer guarantees uniform global exponential convergence under joint observability assumptions and a proper time-scale separation between the matrix dynamic inversion and the system dynamics.

We validate the proposed method through cooperative localization simulations.

II. System Modeling

Agents dynamic model: Consider a multi-agent system composed of N agents with decoupled linear dynamics:

$$\dot{\boldsymbol{x}}_i = \boldsymbol{A}_i(t)\boldsymbol{x}_i + \boldsymbol{B}_i(t)\boldsymbol{u}_i \quad \forall i \in \{1, ..., N\} \tag{1}$$

with state $\boldsymbol{x}_i \in \mathbb{R}^{d_i}$, input $\boldsymbol{u}_i \in \mathbb{R}^{m_i}$, where we made explicit the time dependency of the matrices $\boldsymbol{A}_i \in \mathbb{R}^{d_i \times d_i}$ and $\boldsymbol{B}_i \in \mathbb{R}^{d_i \times m_i}$.

Agents interaction model: agents interact through two distinct graphs: a *static directed sensing graph* $\mathcal{G}_s = (\mathcal{V}, \mathcal{E}_s)$, where a directed edge $(i, j) \in \mathcal{E}_s$ indicates that agent i can sense agent j, and an *undirected communication graph* $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$, which allows bidirectional information exchange.

Measurement model: Each agent can collect:

- **Private measurements** of its own state:

$$\boldsymbol{y}_i^p = \delta_i \boldsymbol{H}_i^p(t)\boldsymbol{x}_i \qquad \delta_i \in \{0, 1\}$$

- **Relative measurements** of neighboring agents' states:

$$\boldsymbol{y}_{ij}^r = \boldsymbol{H}_{iji}^r(t)\boldsymbol{x}_i + \boldsymbol{H}_{ijj}^r(t)\boldsymbol{x}_j, \quad \text{for } (i,j) \in \mathcal{E}_s$$

A common special case is when relative outputs depend only on the relative state:

$$\boldsymbol{y}_{ij}^r = \boldsymbol{H}_{ij}^r(t)(\boldsymbol{x}_j - \boldsymbol{x}_i)$$

The collective quantities are defined as:

$$\begin{aligned}
\boldsymbol{x} &= \begin{bmatrix} \boldsymbol{x}_1^\top & \cdots & \boldsymbol{x}_N^\top \end{bmatrix}^\top \in \mathbb{R}^{Nd} \\
\boldsymbol{u} &= \begin{bmatrix} \boldsymbol{u}_1^\top & \cdots & \boldsymbol{u}_N^\top \end{bmatrix}^\top \in \mathbb{R}^{Nm} \\
\boldsymbol{y}^p &= \begin{bmatrix} \boldsymbol{y}_1^{p\top} & \cdots & \boldsymbol{y}_N^{p\top} \end{bmatrix}^\top \in \mathbb{R}^{Nq_p} \\
\boldsymbol{y}^r &= \begin{bmatrix} \boldsymbol{y}_1^{r\top} & \cdots & \boldsymbol{y}_M^{r\top} \end{bmatrix}^\top \in \mathbb{R}^{Mq_r}
\end{aligned}$$

where $M = |\mathcal{E}_s|$ is the number of relative measurements.

We define the following matrices:

- $\boldsymbol{A}(t) = \text{blkdiag}(\boldsymbol{A}_1(t), \ldots, \boldsymbol{A}_N(t))$
- $\boldsymbol{B}(t) = \text{blkdiag}(\boldsymbol{B}_1(t), \ldots, \boldsymbol{B}_N(t))$
- $\boldsymbol{H}^p(t) = \text{blkdiag}(\boldsymbol{H}_1^p(t), \ldots, \boldsymbol{H}_N^p(t))$
- $\boldsymbol{\Delta} = \text{blkdiag}(\delta_1 \boldsymbol{I}_d, \ldots, \delta_N \boldsymbol{I}_d)$
- $\boldsymbol{H}^r(t)$: sparse matrix reflecting the sensing graph's incidence pattern

In the case where all relative outputs follow the form $\boldsymbol{y}_{ij}^r = \boldsymbol{H}_{ij}^r(t)(\boldsymbol{x}_j - \boldsymbol{x}_i)$, the output can be written as:

$$\boldsymbol{y}^r = \text{blkdiag}(\boldsymbol{H}_1^r(t), \ldots, \boldsymbol{H}_M^r(t))\boldsymbol{E}_d^\top \boldsymbol{x}$$

where $\boldsymbol{E}_d = \boldsymbol{E} \otimes \boldsymbol{I}_d$ and $\boldsymbol{E}$ is the incidence matrix of the sensing graph.

Compact system model:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(t)\boldsymbol{x} + \boldsymbol{B}(t)\boldsymbol{u}$$

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}^p \\ \boldsymbol{y}^r \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{H}^p(t)\boldsymbol{\Delta} \\ \boldsymbol{H}^r(t) \end{bmatrix}}_{\boldsymbol{H}(t)} \boldsymbol{x}$$

This work was supported in part by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg (KAW) Foundation, the Horizon Europe EIC project SymAware (101070802), the ERC LEAFHOUND Project, the Swedish Research Council (VR), and Digital Futures.

Nicola De Carli and Dimos V. Dimarogonas are with the Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden. E-mail: {ndc,dimos}@kth.se

III. Distributed Kalman-Like Observer

Kalman-Bucy Filter (Centralized).: The continuous-time Kalman filter, known as the Kalman-Bucy filter, is given by:

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}} &= \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{P}\boldsymbol{H}^\top \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}) \\ \dot{\boldsymbol{P}} &= \boldsymbol{A}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^\top + \boldsymbol{Q} - \boldsymbol{P}\boldsymbol{H}^\top \boldsymbol{R}^{-1}\boldsymbol{H}\boldsymbol{P} \end{aligned} \tag{2}$$

where $\boldsymbol{P}$ is the error covariance, $\boldsymbol{R}$ is the measurement noise covariance (block-diagonal), and $\boldsymbol{Q}$ is the process noise covariance (also block-diagonal).

Challenges in Distributed Implementation.: Distributed computation of the Kalman filter is challenging because:

- Certain terms (e.g., $\boldsymbol{A}\boldsymbol{P}$, $\boldsymbol{H}^\top \boldsymbol{R}^{-1}\boldsymbol{H}$) can preserve sparsity if $\boldsymbol{P}(0)$ has a structured form (e.g., block-diagonal or Laplacian-like).
- The product $\boldsymbol{P}\boldsymbol{H}^\top \boldsymbol{R}^{-1}\boldsymbol{H}\boldsymbol{P}$ destroys sparsity, making $\boldsymbol{P}(t)$ fully dense over time.

Information Filter Reformulation.: To mitigate densification, we use the information form with $\boldsymbol{S} = \boldsymbol{P}^{-1}$:

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}} &= \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{S}^{-1}\boldsymbol{H}^\top \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}) \\ \dot{\boldsymbol{S}} &= -\boldsymbol{A}^\top \boldsymbol{S} - \boldsymbol{S}\boldsymbol{A} - \boldsymbol{S}\boldsymbol{Q}\boldsymbol{S} + \boldsymbol{H}^\top \boldsymbol{R}^{-1}\boldsymbol{H} \end{aligned} \tag{3}$$

This form is closer to be sparse, but:

- The term $\boldsymbol{S}\boldsymbol{Q}\boldsymbol{S}$ is not computable in a distributed way.
- Inversion of $\boldsymbol{S}$ is also non-distributable, even if $\boldsymbol{S}$ is sparse.

Kalman-Like Observers with Forgetting Factor.: To avoid modeling process noise explicitly, we set $\boldsymbol{Q} = \boldsymbol{0}$ and introduce a forgetting factor $\gamma > 0$:

$$\dot{\boldsymbol{S}} = -\left(\boldsymbol{A} + \tfrac{\gamma}{2}I\right)^\top \boldsymbol{S} - \boldsymbol{S}\left(\boldsymbol{A} + \tfrac{\gamma}{2}I\right) + \boldsymbol{H}^\top \boldsymbol{R}^{-1}\boldsymbol{H} \tag{4}$$

This ensures exponential decay of older information and preserves sparsity over time. When using relative measurements, the resulting matrix $\dot{\boldsymbol{S}}$ includes a matrix-weighted Laplacian structure.

Distributed Approximation of $\boldsymbol{S}^{-1}$.: To compute the correction term $\boldsymbol{S}^{-1}\boldsymbol{H}^\top \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}})$ in a distributed way, we use a dynamic approximation based on continuous-time Richardson iteration. Define:

$$\boldsymbol{\xi} := \boldsymbol{S}^{-1}\boldsymbol{H}^\top \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}) \tag{5}$$

We then use:

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}} &= \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \hat{\boldsymbol{\xi}} \\ \mu\dot{\hat{\boldsymbol{\xi}}} &= -(\boldsymbol{S}\hat{\boldsymbol{\xi}} - \boldsymbol{H}^\top \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}})) \\ \dot{\boldsymbol{S}} &= -\left(\boldsymbol{A} + \tfrac{\gamma}{2}I\right)^\top \boldsymbol{S} - \boldsymbol{S}\left(\boldsymbol{A} + \tfrac{\gamma}{2}I\right) + \boldsymbol{H}^\top \boldsymbol{R}^{-1}\boldsymbol{H} \end{aligned} \tag{6}$$

where $\mu > 0$ controls the convergence rate of the approximation.

Distributed Implementation.: Each agent i updates its local variables $\hat{\boldsymbol{x}}_i$, $\hat{\boldsymbol{\xi}}_i$, and matrix blocks $\boldsymbol{S}_{ij}$ using local information and messages from its neighbors:

- Measurements: $\boldsymbol{y}_{ji}^r$
- Neighbor estimates: $\hat{\boldsymbol{x}}_j$, $\hat{\boldsymbol{\xi}}_j$

Theorem 1 (Exponential Stability of the Interconnected System)**.** *Under persistency of excitation assumptions and technical assumptions of boundedness of system matrices and their derivatives, for sufficiently small μ, the estimation error is uniformly exponentially stable.*

IV. Simulation Results

The proposed distributed Kalman-like observer is evaluated in a cooperative localization task with $N = 15$ mobile robots and $A = 4$ anchors with access to absolute position measurements. Robots follow double integrator dynamics and rely on relative position measurements for state estimation. Anchors provide additional absolute position data.

Simulation results show that:

- The estimated correction term tracks the ideal one accurately (Fig. 2).
- The overall estimation error converges to zero (Fig. 3).

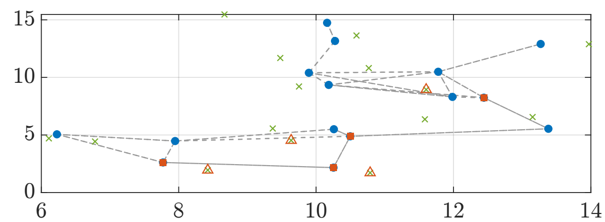

Fig. 1: Initial configuration of the real robots (circle markers) and estimated robots (cross markers). The red circles represent the anchors and their initial estimate is shown with a triangle marker around. The network communication graph is also shown.

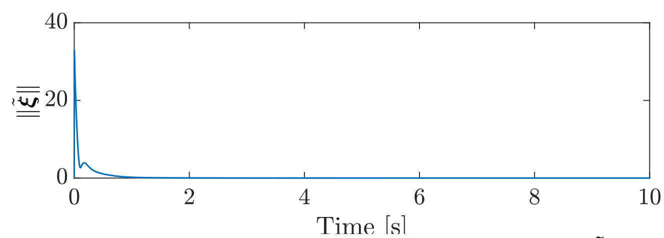

Fig. 2: Norm of the error on the correction term, i.e. $||\tilde{\boldsymbol{\xi}}||$.

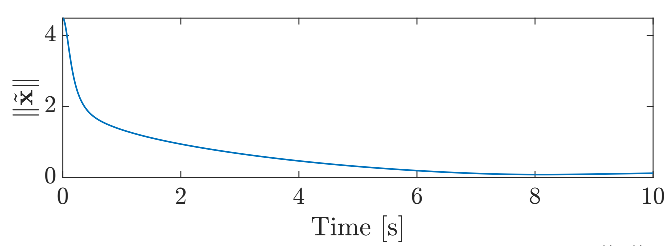

Fig. 3: Norm of the estimation error on the full state, i.e. $||\tilde{\boldsymbol{x}}||$.

References

[1] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, “Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication,” *The International Journal of Robotics Research*, vol. 37, no. 10, pp. 1152–1167, 2018.

[2] R. Jung, L. Luft, and S. Weiss, “Isolated kalman filtering: theory and decoupled estimator design,” *Autonomous Robots*, vol. 49, no. 1, pp. 1–25, 2025.

[3] Y. Sun, M. Fu, B. Wang, H. Zhang, and D. Marelli, “Dynamic state estimation for power networks using distributed map technique,” *Automatica*, vol. 73, pp. 27–37, 2016.