Scalable Recursive Distributed Collaborative State Estimation for Aided Inertial Navigation

Roland Jung[1] and Stephan Weiss[2]

***Abstract*— This paper presents a novel approach to recover outdated cross-covariance between correlated agents at the moment they perform joint observations. This allows to render Collaborative State Estimation (CSE) fully distributed, with communication only required for the moment of joint observation and most importantly, it significantly reduces the maintenance effort in case of high frequent propagation sensors. These properties make the approach suitable to a wide range of multi-robot applications. In our evaluation on a Quaternion-based Error-State Extended Kalman Filter (Q-ESEKF) using an Inertial Measurement Unit (IMU) as propagation sensor at a rate of 200 Hz, we showed a significant speedup against our previous approach for maintaining a couple of interdependence. We compared the approach in total against four different approaches on both, a simulation and on a real-world dataset for Micro Aerial Vehicles (MAVs).**
Video: `https://youtu.be/xkljfwbhMP0`

I. INTRODUCTION

Teams of collaborating agents have potential in many applications. For coordinating teams, an accurate localization estimate of agents is a prerequisite [1].

Collaborative State Estimation (CSE) aims to fuse information provided by different sensors from all agents in a statistically optimal fashion. It allows to treat the team as one system at the cost of strong coupling, maintenance, communication and computational effort.

Rendering CSE exact can (i) significantly improve the estimation performance of individual agents, (ii) provide redundancy in case of sensor failures, jamming or spoofing, and (iii) agents with less accurate sensors benefit from agents with more accurate ones [2], [3]. Therefore it is often used in the field of collaborative localization (CL).

A naive centralized implementation leads to computational complexity of $\mathcal{O}(N^2)$ and communication complexity of $\mathcal{O}(N)$ per sensor measurement per agent, where N is the number of agents. The total computational complexity per time step is $\mathcal{O}(N^4)$ [4]. To render CSE distributed on agents is challenging as it can either degrade the performance, lead to inconsistent estimates, requires extensive bookkeeping, communication or computation if done naively.

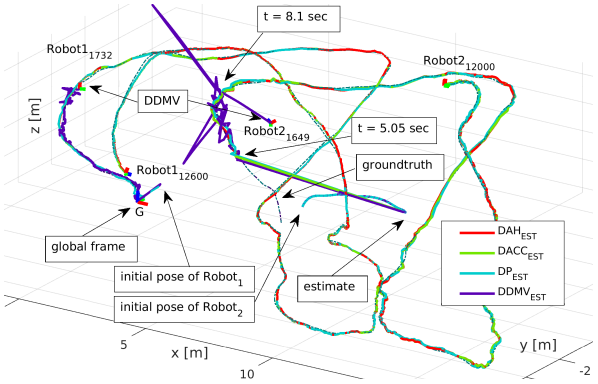


Fig. 1: Scenario S_1: Estimated trajectory comparison using four different CSE fusion techniques: DAH (red, proposed), DACC (green), DP (cyan), and DDMV (purple). A_1 receives global pose measurement, while A_2 receives correction from joint relative position measurements with A_1, starting from $t = 5.05\,\mathrm{s}$ on. Due to wrongly initialized states, A_2 is drifting heavily, but in most cases it can quickly converge towards ground truth despite no direct global information being available. The agents fusing joint observation using DDMV receive strong state correction and A_2 starts diverging. The problem is described in Section VI-A

It is even more challenging if agents use sensors that provide measurements at high rates. In our case we use an Inertial Measurement Unit (IMU) as propagation sensor for a Q-ESEKF with rates typically above 100 Hz. The IMU is often used in a modular fashion in combination with different exteroceptive sensors, e.g. Global Navigation Satellite System (GNSS) module, camera, sonar, pressure, magnetometer, or ultra-wideband (UWB) range measurements, to correct the IMU propagation [5].

Recent work by Luft et al. in [6] and ourselves [7] has shown that communication complexity can be reduced to $\mathcal{O}(1)$ and is just required for joint observation. Yet, the local *maintenance* of interdependencies (i.e. agent cross-covariances) still increases linearly with the number of met agents. This renders the Distributed Approximated Cross-Covariance (DACC) presented in [6] ill-suited for large swarms and long-duration missions with many encounters.

Therefore, the primary motivation for our proposed approach is the aim for $\mathcal{O}(1)$ in both maintenance and communication. We propose a Distributed Approximated History (DAH) approach to restore the cross-correlation between agents at the moment they meet again as an extension to DACC. Our experiments manifest the improvement while accomplishing the same accuracy. More precisely our contributions are:

[1]R. Jung is with the Karl Popper School on Networked Autonomous Aerial Vehicles (KPK-NAV), University of Klagenfurt, Austria (e-mail: roland.jung@ieee.org).

[2]S. Weiss is with the Control of Networked Systems Group, University of Klagenfurt, Austria (e-mail: stephan.weiss@ieee.org).

This work was supported by the EU-H2020 project BUGWRIGHT2 (GA 871260) and the doctoral school KPK-NAV of the University of Klagenfurt.

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org.

Digital Object Identifier follows ASAP

- Development of a novel approach for CSE able to restore outdated cross-covariance between agents at the moment they meet.
- Reduction of maintenance cost from $\mathcal{O}(N)$ in [4], [6], [8] to $\mathcal{O}(1)$ for private observations and to $\mathcal{O}(1)$ for propagation steps (if the buffer size suffices) in DAH, while performing equivalently as DACC.
- Comparison of DAH against two other approaches (i) in a swarm of 20 collaborating agents, focusing on the estimator credibility and timing, and (ii) on an a real-world dataset.

II. Related Work

In the past decades, distributed multi-agent localization has received considerable attention. Here we will focus on distributed Kalman filter approaches due to their computation efficiency and recursive architecture.

Roumeliotis and Bekey investigated in [4] the decoupling of the state propagation equation of the centralized multi-agent EKF. Their Distributed Propagation (DP) approach allows distributed local state propagation, but requires mesh-based connectivity with all agents leading to a $\mathcal{O}(N)$ communication complexity per update per agent. From performance point of view it is equivalent to a centralized architecture and thus regarded as baseline for our evaluations.

Later, a theoretical analysis about the influence of the homogeneous swarm size and sensors employed regarding the localization accuracy and the upper uncertainty bound was presented in [3], [9]. This was relaxed for a heterogeneous swarm with an arbitrary topology in [2].

Aramebel et al. presented in [10] an approach to eliminate the need for full connectivity through maintaining a copy of the entire global state. These approaches have significant issues regarding propagation/evolution as it requires the kinematic motion model of the individual agents.

Kia et al. proposed in [8] an approach where a master agent sends correction messages to all other agents based on the information it received from all of them. Network constraints are relaxed through message passing and relaying. It requires no communication for local state propagation and private updates, while joint updates requires corrections to be received by all agents.

To avoid the maintenance of interdependencies, Carrillo-Arce et al. presented in [11] an approach using Covariance Intersection (CI), introduced by Julier and Uhlmann in [12], to fuse generated pseudo observations from *other* agents.

Zhu and Kia proposed in [13], [14] an Distributed Discorrelated Minimum Variance (DDMV) approach to discorrelate the agents' local beliefs by weighting and underestimating them. That is similarly to CI computing the upper bound of the joint covariance matrix and thus allows ignoring cross-covariances. This reduces both communication and maintenance cost to $\mathcal{O}(1)$, yielding a highly scalable, but pessimistic CSE approach. Unfortunately, this approach did not work as shown in Figure 1 due to inconsistencies.

Later, Zhu and Kia presented in [14] the Estimated Cross-covariance Minimum Variance (ECMV) approach to reduce the conservatism of DDMV by estimating the unknown cross-covariance between agents in a cascaded optimization. In their case study ECMV outperforms DDMV at the cost of $150\times$ longer processing times for joint updates, thus it is not real-time capable.

Recently, another major milestone for CSE in the field of CL was achieved by Luft et al. in [6], proposing reasonable approximations for the state correction of non-participating agents in private and joint updates at the cost of performance.

Later in [7], Jung et al. combined these approximations with a Q-ESEKF formulation to achieve collaboratively aided inertial navigation.

Restoring outdated cross-covarinace at a later point in time is dual problem to handling delayed updates as discussed by Allak et al. in [15]. They proposed an efficient and versatile way of computing covariances by the use of scattering theory without recomputing all intermediate measurements.

III. Problem Formulation

A team of N distributed and communicating agents are equipped with an IMU and running a Q-ESEKF. It provides them with drifting pose estimates, while exteroceptive observations are fused to correct them. Joint observations between agents are provided by a black box, e.g. provided by visual tag recognition (e.g. *AprilTag* [16]) on other agents. The following simplifications are made:

- synchronized system clocks, e.g. by network based synchronization protocols [17],
- the period of exteroceptive sensors is an integer multiple of the IMU period,
- communication range is larger than the sensing distance,
- and exchanged information between agents and sensor measurements arrive without delay.

A. Notation

The mean and covariance of multivariate random variable are defined as $\mathbf{X}_i \sim \mathcal{N}(\hat{\mathbf{x}}_i, \mathbf{\Sigma}_{ii})$. A right subscript specifies the agent's identifier $\{\mathsf{A}_i, i \in 1, \ldots, N\}$. The time indices of state variables are indicated by the right superscript, e.g. $\mathbf{X}^k$, denoting the state at the time $t(k) \equiv t^k$. Names of reference frames are capitalized and calligraphic, e.g. $\mathcal{I}$ for IMU. A coordinate vector ${}^{\mathcal{C}}_{\mathcal{C}}\mathbf{p}_{P_1}$ is read as ${}^{from}_{in}\mathbf{x}_{\,to}$. The operators $\oplus$ and $\ominus$ should emphasize that rotational in SO^3 and translational components in $\mathbb{R}^3$ have to be treated differently. Positions, velocities and biases are additive, e.g. ${}^{\mathcal{G}}\mathbf{p}_{\mathcal{I}} = {}^{\mathcal{G}}\hat{\mathbf{p}}_{\mathcal{I}} + {}^{\mathcal{G}}\tilde{\mathbf{p}}_{\mathcal{I}}$. Rotational errors are right-multiplicative, e.g. ${}^{\mathcal{G}}\mathbf{R}_{\mathcal{I}} = {}^{\mathcal{G}}\hat{\mathbf{R}}_{\mathcal{I}}(\mathbf{I}_3 + \left[{}^{\mathcal{G}}\tilde{\boldsymbol{\theta}}_{\mathcal{I}}\right]_\times) \in \mathrm{SO}^3$.

IV. Quaternion-based Error-State EKF in CSE

The state space representation and the system propagation model of each agent's Q-ESEKF is inspired by Weiss and Siegwart [18], yielding a 16-element state vector:

$$\mathbf{X}_i = \left[{}^{\mathcal{G}}_{\mathcal{G}}\mathbf{p}_{\mathcal{I}}, {}^{\mathcal{G}}_{\mathcal{G}}\mathbf{v}_{\mathcal{I}}, {}^{\mathcal{G}}\mathbf{q}_{\mathcal{I}}, {}_{\mathcal{I}}\mathbf{b}_\omega, {}_{\mathcal{I}}\mathbf{b}_a\right]_i, i \in 1, \ldots, N, \quad (1)$$

with ${}^{\mathcal{G}}\mathbf{p}_{\mathcal{I}}, {}^{\mathcal{G}}_{\mathcal{G}}\mathbf{v}_{\mathcal{I}}$, and ${}^{\mathcal{G}}\mathbf{q}_{\mathcal{I}}$ as the position, velocity and orientation of the IMU $\mathcal{I}$ w.r.t. the global frame $\mathcal{G}$ (or navigation frame). ${}_{\mathcal{I}}\mathbf{b}_\omega$ and ${}_{\mathcal{I}}\mathbf{b}_a$ are the estimated gyroscope and accelerometer biases to correct the related IMU readings.

The corresponding error-state vector ($\widetilde{\mathbf{x}} = \mathbf{x} \ominus \hat{\mathbf{x}}$) using the small angle approximation ${}^{\mathcal{G}}_{\mathcal{I}}\boldsymbol{\theta}_{\mathcal{I}}$ for rotations is

$$\widetilde{\mathbf{X}}_i = \left[{}^{\mathcal{G}}\tilde{\mathbf{p}}_{\mathcal{I}}, {}^{\mathcal{G}}\tilde{\mathbf{v}}_{\mathcal{I}}, {}^{\mathcal{G}}_{\mathcal{I}}\tilde{\boldsymbol{\theta}}_{\mathcal{I}}, {}_{\mathcal{I}}\tilde{\mathbf{b}}_{\omega}, {}_{\mathcal{I}}\tilde{\mathbf{b}}_{a} \right]_i . \tag{2}$$

The error state kinematic used for the IMU propagation and corrections through exteroceptive sensors is well studied and can be found in e.g. [18].

V. Distributed Approximated History for CSE

In the proposed Q-ESEKF, state propagation is performed at a rate of the IMU which is typically above 100 Hz. Not only the state but also all interdependencies due to cross-correlations with met agents have to be propagated at this rate. The more agents are correlated, the more maintenance is required, even after not seeing each other for a long time. To overcome that issue, we propose to keep locally just (i) the most recent factorized cross-covariance between agents, (ii) a timestamp of the event, and (iii) a sliding window buffer $\boldsymbol{\mathcal{B}}$ keeping track of the last corrections. The factorized cross-covariance and timestamp can e.g. be stored in a dictionary Dict accessed via the other agent's ID.

We store the decomposed cross-covariance between the agents A_i and A_j

$$\boldsymbol{\Sigma}^k_{ij} = \mathcal{S}^k_{ij} \left(\mathcal{S}^k_{ji}\right)^{\mathsf{T}}, \tag{3}$$

where the choice of decomposition is open [4], e.g. $\mathcal{S}^k_{ij} = \boldsymbol{\Sigma}^k_{ij}$ and $\mathcal{S}^k_{ji} = \mathbf{I}$.

Compared to other implementations, [4] [8] or [6], we do not maintain all factorized cross-covariances during the propagation step. Instead, we restore an approximated *a priori* cross-covariance $\boldsymbol{\Sigma}^{m+1(-)}_{ij}$ between agents at the moment t^{m+1} they meet again by using the agents' buffer histories. For an exact forward propagation these buffers would require a history reaching to the very last encounter. However, we use a cyclic buffer $\boldsymbol{\mathcal{B}}$ of a static size per agent. If a factorized cross-covariance is about to fall outside the past time horizon, we propagate it with an accumulated product towards current time step. This ensures that the time horizon of the buffer is always sufficient and allows the proposed approach to behave *equal* as [6] in terms of estimation performance (we rely on same assumptions), while reducing the maintenance effort. Due to space limitation we will highlight just the main assumptions. For details and proofs we would like to refer the readers to Luft et al. [6].

In the following we investigate on the correction factors that need to be provided to buffer at the different filter steps: propagation ($\boldsymbol{\Phi}$), private observation ($\boldsymbol{\Upsilon}$), and joint observations ($\boldsymbol{\Lambda}$). The strategy for the buffer maintenance and cross-covariance propagation is described in Section V-C and DAH is summarized in Algorithms 1 to 5.

A. Propagation

The motion of different agents is independent and corrupted by Gaussian noise $\boldsymbol{\nu}_p \sim \mathcal{N}(\mathbf{0}, \mathbf{N})$

$$\mathbf{x}^{k|k-1} = g(\mathbf{x}^{k-1}, \mathbf{u}^k + \boldsymbol{\nu}^k_p), \tag{4}$$

with a smooth motion model g, where $\mathbf{u}$ can either be a control input or a proprioceptive sensor measurement. The motion model needs to be linearized to be used in the Kalman filter propagation step $\boldsymbol{\Phi}^{k|k-1} = \frac{\partial g(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}(\hat{\mathbf{x}}^{k-1}, \mathbf{u}^k)$ with the process noise $\mathbf{Q}^{k|k-1} = \mathbf{G}^k \mathbf{N}^k (\mathbf{G}^k)^{\mathsf{T}}$, $\mathbf{G}^k = \frac{\partial g(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}(\hat{\mathbf{x}}^{k-1}, \mathbf{u}^k)$.

The state covariance matrix on A_i is propagated using

$$\boldsymbol{\Sigma}^{k+1}_{ii} = \boldsymbol{\Phi}^{k+1}_{ii} \boldsymbol{\Sigma}^k_{ii} \boldsymbol{\Phi}^{k+1\,\mathsf{T}}_{ii} + \mathbf{Q}^{k+1}_{ii}. \tag{5}$$

with $\boldsymbol{\Phi}^{k+1}$ as the discretized state transition matrix and $\mathbf{Q}^{k+1}$ the discretized process noise covariance matrix.

As proposed in [4], the distributed cross-covariance propagation can be performed exactly. Typically it is done by multiplying the state transition matrix on the factorized cross-covariances [4], [6], [8]

$$\boldsymbol{\Sigma}^{k+1}_{ij} = \boldsymbol{\Phi}^{k+1}_{ii} \mathcal{S}^k_{ij} (\mathcal{S}^k_{ji})^{\mathsf{T}} (\boldsymbol{\Phi}^{k+1}_{jj})^{\mathsf{T}}. \tag{6}$$

In contrary, we append the state transition matrices $\boldsymbol{\Phi}$ to our sliding buffer (propagation step is advancing the buffer)

$$\boldsymbol{\mathcal{B}}_i(t^{k+1}) = \boldsymbol{\Phi}^{k+1}_{ii}, \tag{7}$$

B. Private and Joint Observation

An observation is described by a measurement function h of a state $\mathbf{x}$ and $\boldsymbol{\nu}_z \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ defining an independent Gaussian noise

$$\mathbf{z}^k = h(\mathbf{x}^k, \boldsymbol{\nu}^k_z). \tag{8}$$

For legibility we will neglect the time index $\{\}^k$. Private and joint observations are technically the same, while later requires, in addition to the local state estimate, estimates from one or multiple other agents. Therefore we can distinguish between *participants* (p) and *non-participants* (others o), resulting in stacked random variable $\mathbf{X}^{\mathsf{T}} = \begin{bmatrix}\mathbf{X}^{\mathsf{T}}_p & \mathbf{X}^{\mathsf{T}}_o\end{bmatrix}$ ($\mathbf{X}_p$ is a joint belief of participants, e.g. constituting of A_i's and A_j's belief $\mathbf{X}^{\mathsf{T}}_p = \begin{bmatrix}\mathbf{X}^{\mathsf{T}}_i & \mathbf{X}^{\mathsf{T}}_j\end{bmatrix}$ and $\mathbf{X}_o$ a joint belief of others.

We assume that the computation of an observation is performed on an interim master agent $\mathsf{A}_i \in p$, e.g. based on the lowest integer ID among participants. The participants might be directly or indirectly correlated with non-participating agents $\mathsf{A}_o \in o := \{1 \dots N | o \notin p\}$. The exact state update according to the Kalman filter can be computed as follows. Assume a joint covariance $\boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\Sigma}_{pp} & \boldsymbol{\Sigma}_{po} \\ \boldsymbol{\Sigma}^{\mathsf{T}}_{po} & \boldsymbol{\Sigma}_{oo}\end{bmatrix}$ and the joint measurement matrix $\mathbf{H} = \begin{bmatrix}\mathbf{H}_p & \mathbf{0}\end{bmatrix}$. The corrected *a posteriori* covariance is

$$\boldsymbol{\Sigma}^{(+)} = \begin{bmatrix} (\mathbf{I} - \mathbf{K}_p \mathbf{H}_p)\boldsymbol{\Sigma}^{(-)}_{pp} & (\mathbf{I} - \mathbf{K}_p \mathbf{H}_p)\boldsymbol{\Sigma}^{(-)}_{po} \\ (\boldsymbol{\Sigma}^{(+)}_{po})^{\mathsf{T}} & -\mathbf{K}_o \mathbf{H}_p \boldsymbol{\Sigma}^{(-)}_{po} + \boldsymbol{\Sigma}^{(-)}_{oo} \end{bmatrix} \tag{9}$$

with Kalman gain

$$\begin{bmatrix}\mathbf{K}_p \\ \mathbf{K}_o\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Sigma}^{(-)}_{pp}\mathbf{H}^{\mathsf{T}}_p \\ \boldsymbol{\Sigma}^{(-)}_{po}\mathbf{H}^{\mathsf{T}}_p\end{bmatrix} \left(\mathbf{H}_p \boldsymbol{\Sigma}^{(-)}_{pp} \mathbf{H}^{\mathsf{T}}_p + \mathbf{R}\right)^{-1}, \tag{10}$$

and $\mathbf{R}$ being the measurement noise covariance. The *a posteriori* mean is

$$\begin{bmatrix}\hat{\mathbf{x}}_p^{(+)}\\ \hat{\mathbf{x}}_o^{(+)}\end{bmatrix} = \begin{bmatrix}\hat{\mathbf{x}}_p^{(-)} \oplus \mathbf{K}_p\mathbf{r}\\ \hat{\mathbf{x}}_o^{(-)} \oplus \mathbf{K}_o\mathbf{r}\end{bmatrix}, \tag{11}$$

with measurement residual $\mathbf{r} = \mathbf{z} \ominus h(\hat{\mathbf{x}})$.

Like [6], we aim for constant communication complexity. Thus we approximate non-participants' beliefs by setting them to $\breve{\mathbf{X}}_o^{(+)} \sim \mathcal{N}\left(\breve{\hat{\mathbf{x}}}_o^{(+)}, \breve{\mathbf{\Sigma}}_{oo}^{(+)}\right) = \mathbf{X}_o^{(-)}$. Consequently, non-participants do not directly benefit from participants' observations. However, the cross-covariance of non-participants can be corrected $\mathbf{\Sigma}_{po}^{(+)} = \mathbf{\Upsilon}_p\mathbf{\Sigma}_{po}^{(-)}$ using

$$\mathbf{\Upsilon}_p = (\mathbf{I} - \mathbf{K}_p\mathbf{H}_p) \tag{12}$$

and allows this approximation to be consistent with respect to the global system (9) as the resulting covariance error is positive semidefinite

$$\tilde{\mathbf{\Sigma}}^{(+)} = \breve{\mathbf{\Sigma}}^{(+)} - \mathbf{\Sigma}^{(+)} = \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{K}_o\mathbf{H}_p\mathbf{\Sigma}_{po}^{(-)}\end{bmatrix} \succeq \mathbf{0}. \tag{13}$$

Summarized: For private observations (case of a single participant) the participant, e.g. A_i, applies the correction factor $\mathbf{\Upsilon}$ (Equation (12)) on the corresponding element of buffer

$$\boldsymbol{\mathcal{B}}_i\left(t^k\right) = \mathbf{\Upsilon}_i^k\boldsymbol{\mathcal{B}}_i\left(t^k\right). \tag{14}$$

For joint observations with multiple participants, each participant applies a correction in relation to gained information $\mathbf{\Lambda}_i^k = \mathbf{\Sigma}_{ii}^{k(+)}\left(\mathbf{\Sigma}_{ii}^{k(-)}\right)^{-1}$ on the corresponding element of the buffer

$$\boldsymbol{\mathcal{B}}_u\left(t^k\right) = \mathbf{\Lambda}_u^k\boldsymbol{\mathcal{B}}_u\left(t^k\right),\ u \in p \tag{15}$$

The factor $\mathbf{\Lambda}_i$, proposed by [6], is reasonable approximation if participants p are strongly directly or indirectly correlated with non-participants o before the joint observation.

C. Buffer maintenance and propagation strategy

In this subsection we describe how the cross-covariances are propagated, once needed for joint observations, and how we prevent cross-covariances to fall out of the past horizon. Figure 2 shows how correction factors from different events are used to propagate previous cross-covariance forward, which is also described in Algorithm 4. To keep cross-covariance in the buffer's horizon, we suggest to perform a sanity check e.g. at the end of each propagation step. The aim is to find factorized cross-covariances by their timestamps, that are exactly at the border of the horizon. In that case, we perform immediately a forward propagation using the entire history, which is described in Algorithm 2. Note that setting the buffer size of $\boldsymbol{\mathcal{B}}$ to 1, DAH emulates DACC. Consequently a smaller buffer increases the chance that a forward propagation is performed in propagation steps (assuming sporadic joint observations). In the best case the buffer size matches the ratio between the rate of the propagation sensor and the rate of joint observations rendering the approach constant in maintenance complexity.

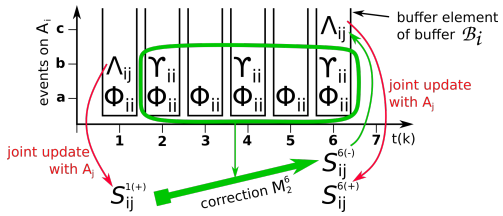


Fig. 2: Decomposed cross-covariance forward propagation scheme using elements with accumulated correction terms from the buffer $\boldsymbol{\mathcal{B}}_i$ on A_i. At $t(k) = 1$, A_i performs a joint observation with the uncorrelated A_j resulting in a correction $\mathbf{\Lambda}_{ij}^1$ (event b at t^1) and a decomposed cross-covariance $\mathcal{S}_{ij}^{1(+)}$. Propagations and private updates result in $\mathbf{\Phi}$ and $\mathbf{\Upsilon}$, respectively. The events are left multiplied in order on the buffer elements. At $t(k) = 6$, the correlated agents perform again a joint observations. Therefore each agent accumulates it's buffer elements to forward propagate $\mathcal{S}_{ij}^{1(+)} \rightarrow \mathcal{S}_{ij}^{6(-)}$ by $\mathbf{M}_2^6$ (Algorithm 4). After that, a new correction factor $\mathbf{\Lambda}_{ij}^6$ and decomposed cross-covarinace $\mathcal{S}_{ij}^{6(+)}$ is obtained.

Algorithm 1: Propagation on A_p

Input : $\hat{\mathbf{x}}_p^{k(-)}, \mathbf{\Sigma}_{pp}^{k(-)}, \boldsymbol{\mathcal{B}}_p, \mathbf{u}^k, \mathbf{N}^k, \mathsf{Dict}_p$

$\mathbf{\Phi}_p^{k+1} = \left[\frac{\partial g(\mathbf{x}_p,\mathbf{u})}{\partial \mathbf{x}_p}(\hat{\mathbf{x}}_p, \mathbf{u})\right]^k$

$\mathbf{G}_p^{k+1} = \left[\frac{\partial g(\mathbf{x}_p,\mathbf{u})}{\partial \mathbf{u}}(\hat{\mathbf{x}}_p, \mathbf{u})\right]^k$

$\mathbf{Q}^{k+1} = \mathbf{G}_p^{k+1}\mathbf{N}^k(\mathbf{G}_p^{k+1})^\mathsf{T}$

$\hat{\mathbf{x}}_p^{k+1} = g(\hat{\mathbf{x}}_p^k, \mathbf{u}^k)$

$\mathbf{\Sigma}_{pp}^{k+1} = \mathbf{\Phi}_p^{k+1}\mathbf{\Sigma}_{pp}^k(\mathbf{\Phi}_p^{k+1})^\mathsf{T} + \mathbf{Q}^{k+1}$

$\boldsymbol{\mathcal{B}}_p\left(t^{k+1}\right) = \mathbf{\Phi}^{k+1}$ (push back)

check_horizon($\boldsymbol{\mathcal{B}}_p, \mathsf{Dict}_p, t^{k+1}$) (Alg. 2)

Algorithm 2: check_horizon on A_p

Input : $\boldsymbol{\mathcal{B}}_p, \mathsf{Dict}, t^k$

$t^{\text{oldest}} = min(\boldsymbol{\mathcal{B}}_p)$

for $\{t^m, \mathcal{S}^{m(-)}, id\}$ **in** Dict **do**

  if $t^m \equiv t^{oldest}$ **then**

    $\mathbf{M}_m^k =$ compute_corr($\boldsymbol{\mathcal{B}}_p, t^m, t^k$) (Alg. 4)

    Dict(id) $= \{\mathbf{M}_m^k\mathcal{S}^{m(-)}, t^k\}$ (forward prop.)

  end

end

Algorithm 3: Private Observation on A_p

Input : $\hat{\mathbf{x}}_p^{k(-)}, \mathbf{\Sigma}_{pp}^{k(-)}, \boldsymbol{\mathcal{B}}_p, \mathbf{z}^k, \mathbf{R}^k$

$\mathbf{H}_p = \left[\frac{\partial h(\mathbf{x}_p)}{\partial \mathbf{x}_p}(\hat{\mathbf{x}}_p)\right]^{k(-)}$

$\mathbf{K}_p = \mathbf{\Sigma}_{pp}^{k(-)}\mathbf{H}_p^\mathsf{T}(\mathbf{H}_p\mathbf{\Sigma}_{pp}^{k(-)}\mathbf{H}_p^\mathsf{T} + \mathbf{R}^k)^{-1}$

$\hat{\mathbf{x}}_p^{k(+)} = \hat{\mathbf{x}}_p^{k(-)} \oplus \mathbf{K}_p\left(\mathbf{z}^k \ominus h(\hat{\mathbf{x}}_p)\right)$

$\mathbf{\Sigma}_{pp}^{k(+)} = (\mathbf{I} - \mathbf{K}_p\mathbf{H}_p)\mathbf{\Sigma}_{pp}^{k(-)}$

$\mathbf{\Upsilon}_p^k = (\mathbf{I} - \mathbf{K}_p\mathbf{H}_p)$

$\boldsymbol{\mathcal{B}}_p\left(t^k\right) = \mathbf{\Upsilon}_p^k\boldsymbol{\mathcal{B}}_p\left(t^k\right)$

Algorithm 4: compute_correction on A_p

Input : $\boldsymbol{\mathcal{B}}_p, t^{m-1}, t^m, \Delta t$

$\mathbf{M}_{m-1}^m = \mathbf{I}$

for $i \leftarrow t^{m-1} + \Delta t$ **to** t^m **by** Δt **do**

  $\mathbf{M}_{m-1}^m = \boldsymbol{\mathcal{B}}_p(i)\mathbf{M}_{m-1}^m$

end

Output: $\mathbf{M}_{m-1}^m$

Algorithm 5: Joint Observation on $\mathsf{A}_{\{i,j\}}$

Input : $\hat{\mathbf{x}}_{\{i,j\}}^{k(-)}, \boldsymbol{\Sigma}_{\{ii,jj\}}^{k(-)}, {}^{i}\mathbf{z}_{j}^{k}, \mathbf{R}^{k}, \mathrm{id}_{\{i,j\}}, \mathsf{Dict}_{\{i,j\}}, \boldsymbol{\mathcal{B}}_{\{i,j\}}$

if *$id_i < id_j$ /* one possibility */* **then**
- /* Interim Master */
- A_i receives $\{\hat{\mathbf{x}}_j^{k(-)}, \boldsymbol{\Sigma}_{jj}^{k(-)}, \mathrm{id}_j, \mathcal{S}_{ji}^{k(-)}\}$ from A_j
- $\{\mathcal{S}_{ij}^{m(-)}, t^m\} = \mathsf{Dict}_i(\mathrm{id}_j)$
- $\mathbf{M}_m^k = \text{compute\_correction}(\boldsymbol{\mathcal{B}}_i, t^m, t^k)$ (Alg. 4)
- $\mathcal{S}_{ij}^{k(-)} = \mathbf{M}_m^k \mathcal{S}_{ij}^{m(-)}$
- $\boldsymbol{\Sigma}_{ij}^{k(-)} = \mathcal{S}_{ij}^{k(-)} (\mathcal{S}_{ji}^{k(-)})^{\mathsf{T}}$
- $\boldsymbol{\Sigma}_{pp}^{k(-)} = \begin{bmatrix} \boldsymbol{\Sigma}_{ii} & \boldsymbol{\Sigma}_{ij} \\ \boldsymbol{\Sigma}_{ij}^{\mathsf{T}} & \boldsymbol{\Sigma}_{jj} \end{bmatrix}^{k(-)}$
- $\mathbf{H}_p = \begin{bmatrix} \frac{\partial h(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) & \frac{\partial h(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_j}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) \end{bmatrix}^{k(-)}$
- $\mathbf{K}_p = \boldsymbol{\Sigma}_{pp}^{k(-)} \mathbf{H}_p^{\mathsf{T}} (\mathbf{H}_p \boldsymbol{\Sigma}_{pp}^{k(-)} \mathbf{H}_p^{\mathsf{T}} + \mathbf{R}^k)^{-1}$
- $\hat{\mathbf{x}}_p^{k(-)} = \begin{bmatrix} (\hat{\mathbf{x}}_i^{k(-)})^{\mathsf{T}} & (\hat{\mathbf{x}}_j^{k(-)})^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$
- $\hat{\mathbf{x}}_p^{k(+)} = \hat{\mathbf{x}}_p^{k(-)} \oplus \mathbf{K}_p \left({}^{i}\mathbf{z}_j^{k} \ominus h(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) \right)$
- $\boldsymbol{\Sigma}_{pp}^{k(+)} = (\mathbf{I} - \mathbf{K}_p \mathbf{H}_p) \boldsymbol{\Sigma}_{pp}^{k(-)}$
- /* Note: split $\boldsymbol{\Sigma}_{pp}^{k(+)}$ and $\hat{\mathbf{x}}_p^{k(+)}$ again */
- $\mathcal{S}_{ij}^{k(+)} = \boldsymbol{\Sigma}_{ij}^{k(+)}$
- $\mathcal{S}_{ji}^{k(+)} = \mathbf{I}$
- A_i sends $\{\hat{\mathbf{x}}_j^{k(+)}, \boldsymbol{\Sigma}_{jj}^{k(+)}, \mathrm{id}_i, \mathcal{S}_{ji}^{k(+)}\}$ to A_j
- $\boldsymbol{\Lambda}_i^k = \boldsymbol{\Sigma}_{ii}^{k(+)} (\boldsymbol{\Sigma}_{ii}^{k(-)})^{-1}$
- $\boldsymbol{\mathcal{B}}_i(t^k) = \boldsymbol{\Lambda}_i^k \boldsymbol{\mathcal{B}}_i(t^k)$
- $\mathsf{Dict}_i(\mathrm{id}_j) = \{\mathcal{S}_{ij}^{k(+)}, t^k\}$

else
- $\{\mathcal{S}_{ji}^{m(-)}, t^m\} = \mathsf{Dict}_j(\mathrm{id}_i)$
- $\mathbf{M}_m^k = \text{compute\_correction}(\boldsymbol{\mathcal{B}}_j, t^m, t^k)$ (Alg. 4)
- $\mathcal{S}_{ji}^{k(-)} = \mathbf{M}_m^k \mathcal{S}_{ji}^{m(-)}$
- A_j sends $\{\hat{\mathbf{x}}_j^{k(-)}, \boldsymbol{\Sigma}_{jj}^{k(-)}, \mathrm{id}_j, \mathcal{S}_{ji}^{k(-)}\}$ to A_i
- A_j receives $\{\hat{\mathbf{x}}_j^{k(+)}, \boldsymbol{\Sigma}_{jj}^{k(+)}, \mathrm{id}_i, \mathcal{S}_{ji}^{k(+)}\}$ from A_i
- $\boldsymbol{\Lambda}_j^k = \boldsymbol{\Sigma}_{jj}^{k(+)} (\boldsymbol{\Sigma}_{jj}^{k(-)})^{-1}$
- $\boldsymbol{\mathcal{B}}_j(t^k) = \boldsymbol{\Lambda}_j^k \boldsymbol{\mathcal{B}}_j(t^k)$
- $\mathsf{Dict}_j(\mathrm{id}_i) = \{\mathcal{S}_{ji}^{k(+)}, t^k\}$

end

VI. Experiments

The experiments are simulated in a MATLAB framework, that allows to load existing datasets or to generate trajectories. The exteroceptive measurements (private or joint observations) are generated based on the ground truth trajectory, the sensors calibration states and noise parameters. The real-world IMU samples provided by the datasets are directly (without modification) used as measurement. Finally, all measurements are sorted chronologically and are locally processed in a multi-instance manager. It is maintaining multiple filter instances, while communication between filter instances is handled locally.

The evaluation is performed on two scenarios $\mathsf{S}_{\{1,2\}}$.

A. Scenario S_1

Following the motivation of our previous work [7], we want to evaluate if all 6-DoF can be recovered by an agent that locally obtains IMU measurements and sporadic joint observations with another agent that has additionally private observations providing absolute position measurements. This time, the interdependencies are not maintained and just approximately restored using DAH. We evaluate the estimation performance using the Machine Hall (MH) sequences (MH4 for A_1 and MH5 for A_2) of the EuRoC dataset [19].

Agent A_1 is obtaining absolute position measurement at a rate of 10 Hz with a standard deviation of $\sigma = 0.1\,\mathrm{m}$. Local relative position measurements between the agents are obtained from $t = 5.05\,\mathrm{s}$ at a rate of 10 Hz with a standard deviation of $\sigma = 0.1\,\mathrm{m}$. Both receive IMU measurements at 200 Hz. The states were initialized wrongly to demonstrate the self-calibration capabilities and emphasize the state convergence.

Figure 1 shows that agent A_2 is drifting heavily, due to wrongly initialized gyroscope and accelerometer biases (${}_{\mathcal{I}}\mathbf{b}_{\omega}$ and ${}_{\mathcal{I}}\mathbf{b}_{a}$). Using DDMV agent A_2 is diverging and A_1 receives significantly wrong corrections from joint updates, that can fortunately be compensated by private ones. Thus, our implementation of DDMV is *not applicable* for Q-ESEKF and not included in the following evaluation.

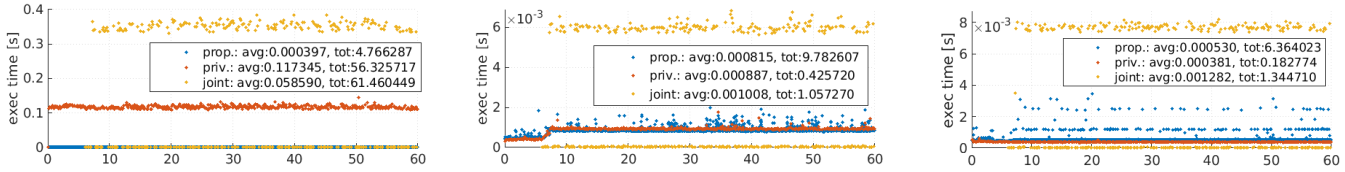
Table I lists the Average Root Mean Square Error (ARMSE) and the mean of the Normalized Estimation Error Squared (NEES) ($\overline{\mathrm{NEES}}$) over the entire trajectory (including the drift phases) of the estimated states for different CSE approaches. No remarkable differences between either fusion techniques are noticeable, while DAH is best scalable. DP has to be considered as ground-truth as it fuses the observations in a statistically optimal way. The NEES for all states should be in average 3; lower than that indicates conservatism, but all states are far from being considered inconsistent. The lower and upper 99.97 % bound are 0.05 and 13.93, respectively. For details regarding estimator credibility and the NEES please refer to [20] [21].

B. Scenario S_2

The main motivation of the proposed approach is to reduce the maintenance effort for estimators with high propagation and update rates. In simulation with 20 agents, arranged in a circle, where each performs a unique take-off (altitude of 20 m, circle (diameter of 10 m), and landing procedure, we compare different CSE fusion techniques. The duration is 60 s. To emphasize the maintenance effort, each agent observes relative position measurements with three other agents (counter clock-wise on the formed circle). Thus, in total each agent knows six agents and all agents are directly or indirectly correlated. A quarter of the agents are provided with absolute position measurements (14 only rely on IMU and joint observations). The absolute position update is received at a rate of 10 Hz, with a standard deviation of $\boldsymbol{\sigma}_{\text{abs}} = 0.3\,\mathrm{m}$, from $t = 0.1\,\mathrm{s}$ on, and with a random message drop rate of 20 %. Local relative position observations are obtained at a rate of 10 Hz, with $\boldsymbol{\sigma}_{\text{rel}} = 0.1\,\mathrm{m}$, and a random message drop rate of 60 %. All receive unbiased and very noisy IMU measurements at 200 Hz with a standard deviation of $\boldsymbol{\sigma}_{\text{acc}} = 0.01\,\mathrm{m/s^2}$ and $\boldsymbol{\sigma}_{\text{gyr}} = 0.01\,\mathrm{rad/s}$ for the accelerometer and gyroscope, respectively. All measurements are generated from the ground truth trajectory. All agent's

		$\mathbf{p}$ [m]		$\mathbf{v}$ [m/s]		$\mathbf{q}$ [deg]		$\mathbf{b}_a$ [m/s^2]		$\mathbf{b}_w$ [rad/s]	
ID	CSE	ARMSE	$\overline{\text{NEES}}$	ARMSE	$\overline{\text{NEES}}$	ARMSE	$\overline{\text{NEES}}$	ARMSE	$\overline{\text{NEES}}$	ARMSE	$\overline{\text{NEES}}$
1	DP	0.046	3.211	0.051	**3.07**	0.19	**1.53**	0.002	0.81	0.004	**0.46**
1	DACC	**0.045**	**3.09**	**0.05**	2.88	0.19	1.45	0.002	**0.87**	0.004	0.44
1	DAH	0.046	**3.09**	0.051	2.88	0.19	1.46	0.002	**0.87**	0.004	0.44
2	DP	0.72	**2.7**	0.3	5.28	1.91	**1.04**	0.001	0.13	0.01	**1.03**
2	DACC	0.72	2.42	0.3	3.86	**1.8**	**1.04**	0.001	**0.23**	0.01	0.95
2	DAH	0.72	2.42	0.3	**3.85**	**1.8**	1.03	0.001	**0.23**	0.01	0.94

TABLE I: Scenario S_1: ARMSE and $\overline{\text{NEES}}$ values of the states estimated in the proposed Q-ESEKF. Real-world IMU measurements from UAVs are used and described in VI-A, using different fusion approaches: DP, DACC, and the proposed DAH. Note: DACC and DAH should perform in this setup equivalently. Due to random noise on measurements a slight deviation is given. Best values in bold.



(a) execution time on A_{20} using DP. (b) execution time on A_{20} using DACC. (c) execution time on A_{20} using DAH.

Fig. 3: Scenario S_2: Execution time plots of A_{20}, showing the execution time for each filter step: propagation (blue), private observation (red), and joint observation (yellow). The legends inform about the average and the accumulated time for each type. Note: As joint observations are performed on an interim master, the execution is rather short for the other participants and lowers the average of joint observations (e.g in Figure 3a). The increasing maintenance effort can be clearly seen for DACC in Figure 3b, which increases at $t \approx 10\,\text{s}$ as joint observations began. Average timings are listed in Table II. Further details are given in VI-B.

	$\mathbf{p}$ [m]		$\mathbf{q}$ [deg]		[ms]	[ms]	[ms]
CSE	AR	AN	AR	AN	$\bar{t}_{\text{prop}}$	$\bar{t}_{\text{priv}}$	$\bar{t}_{\text{joint}}$
DP	**0.076**	**2.41**	**1.54**	0.69	**0.4**	35	58
DACC	0.105	5.19	1.91	1.04	0.83	0.28	**1.4**
DAH	0.106	5.21	1.94	**1.09**	0.53	**0.12**	1.6

TABLE II: Scenario S_2: Shows the average over 20 agents of the (i) ARMSE (AR), (ii) $\overline{\text{NEES}}$ (AN), (iii) average propagation time $\bar{t}_{\text{prop}}$, (iv) average private update time $\bar{t}_{\text{priv}}$, and (v) average joint update time $\bar{t}_{\text{joint}}$. The agents used the proposed Q-ESEKF with different fusion approaches: DP, DACC, and the proposed DAH. Problem formulation in Section VI-B. Best values in bold.

states are initialized correctly with a reasonable uncertainty. The buffer size was set to 100 elements (horizon of 0.5 s).

Figure 3 and Table II show that DAH outperforms DACC in terms of total execution time. Using DAH, the execution time of the estimator was in total 7.87 s, while using DACC, it was 11.26 s which is 43 % slower. It can also be seen, that the processing time of joint observations are in DAH approximately 33 % higher as in DACC. Concluding, DAH keeps constant maintenance complexity for propagation and private filter steps, if the buffer size satisfies the period of joint observations. The estimation performance of DAH seems to be a bit worse than DACC, despite there should be no difference. Figure 4 provides a snapshot at $t = 8.5\,\text{s}$ of the simulation performing DAH-CSE.

VII. Conclusion

Driven by the need to reduce local maintenance effort using fast propagation sensors, this paper introduces a new scheme for distributed Kalman filters, that performs the forward propagation of interdependencies at the moment they are needed. Our experiments manifest that the proposed DAH approach outperforms DACC, while achieving the same filter performance in terms of accuracy and credibility. It can be seen as direct extension to DACC that requires additionally a small amount of statically allocated storage

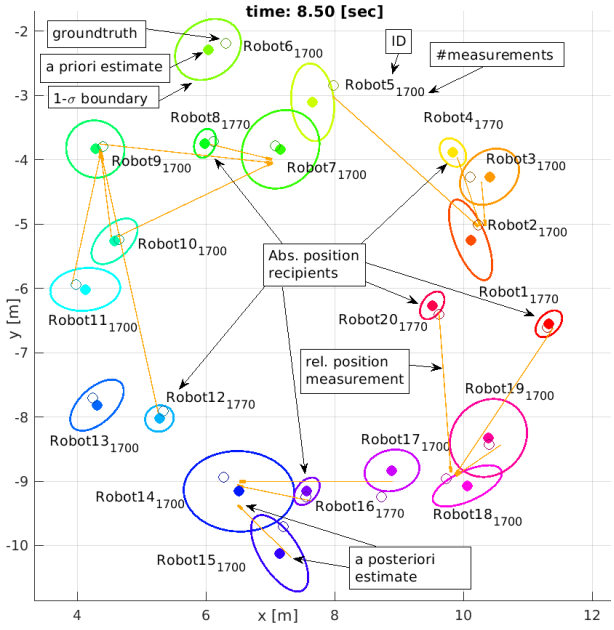


Fig. 4: Scenario S_2: Top view on 20 agents' *a priori* estimates, their 1σ uncertainty bound, as well as the ground truth position at $t = 8.5\,\text{s}$. Some are performing joint observations (indicated by orange lines connecting the corrected poses) that compensate drifting states due to noisy IMU propagation. As only 6 agents are provided by absolute position measurements, these joint observations guarantee that the uncertainty remains bounded. The problem is described in Section VI-B.

per estimator. DAH allows to render CSE distributed with support for generic measurement and propagation models, communication is required only in case of joint observation and no maintenance effort for private observations. By choosing an appropriate buffer size, the maintenance effort in propagation steps can be completely shifted to the moment of joint observations making DAH a fully scalable CSE approach with constant complexity both in communication and maintenance.

REFERENCES

[1] P. Mazdin, M. Barciś, H. Hellwagner, and B. Rinner, "Distributed Task Assignment in Multi-Robot Systems based on Information Utility," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 734–740.

[2] A. I. Mourikis and S. I. Roumeliotis, "Performance analysis of multirobot Cooperative localization," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 666–681, Aug. 2006.

[3] S. I. Roumeliotis and I. M. Rekleitis, "Analysis of multirobot localization uncertainty propagation," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 2, Oct. 2003, pp. 1763–1770 vol.2.

[4] S. I. Roumeliotis and G. A. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[5] K. Hausman, S. Weiss, R. Brockers, L. Matthies, and G. S. Sukhatme, "Self-calibrating multi-sensor fusion with probabilistic measurement validation for seamless sensor switching on a UAV," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, 2016, pp. 4289–4296.

[6] 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*, p. 0278364918760698, Mar. 2018.

[7] R. Jung, C. Brommer, and S. Weiss, "Decentralized Collaborative State Estimation for Aided Inertial Navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 4673–4679, iSSN: 2577-087X.

[8] S. S. Kia, S. Rounds, and S. Martinez, "Cooperative Localization for Mobile Agents: A Recursive Decentralized Algorithm Based on Kalman-Filter Decoupling," *IEEE Control Systems Magazine*, vol. 36, no. 2, pp. 86–101, Apr. 2016.

[9] S. I. Roumeliotis and I. M. Rekleitis, "Propagation of Uncertainty in Cooperative Multirobot Localization: Analysis and Experimental Results," *Autonomous Robots*, vol. 17, no. 1, pp. 41–54, Jul. 2004.

[10] P. O. Arambel, C. Rago, and R. K. Mehra, "Covariance intersection algorithm for distributed spacecraft state estimation," in *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, vol. 6, Jun. 2001, pp. 4398–4403 vol.6.

[11] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis, "Decentralized multi-robot cooperative localization using covariance intersection," Nov. 2013, pp. 1412–1417, iSSN: 2153-0866.

[12] 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, Jun. 1997, pp. 2369–2373 vol.4.

[13] J. Zhu and S. S. Kia, "A Loosely Coupled Cooperative Localization Augmentation to Improve Human Geolocation in Indoor Environments," in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sep. 2018, pp. 206–212.

[14] ——, "Cooperative Localization Under Limited Connectivity," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1523–1530, Dec. 2019.

[15] E. Allak, R. Jung, and S. Weiss, "Covariance Pre-Integration for Delayed Measurements in Multi-Sensor Fusion," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 6642–6649, iSSN: 2153-0866.

[16] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3400–3407, iSSN: 1050-4729.

[17] A. Barciś, M. Barciś, and C. Bettstetter, "Robots that sync and swarm: A proof of concept in ROS 2," in *Proc. IEEE Int'l Symp. on Multi-Robot and Multi-Agent Systems*, 2019, zSCC: 0000003.

[18] S. Weiss and R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," May 2011, pp. 4531–4537.

[19] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, Sep. 2016.

[20] X. R. Li, Z. Zhao, and X.-B. Li, "Evaluation of Estimation Algorithms: Credibility Tests," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 1, pp. 147–163, Jan. 2012.

[21] Z. Chen, C. Heckman, S. Julier, and N. Ahmed, "Weak in the NEES?: Auto-Tuning Kalman Filters with Bayesian Optimization," in *2018*

21st International Conference on Information Fusion (FUSION), Jul. 2018, pp. 1072–1079.