2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Compartmentalized Covariance Intersection: A Novel Filter Architecture for Distributed Localization*

Adam Wiktor and Stephen Rock[1]

Abstract— **This paper introduces the Compartmentalized Covariance Intersection (CCI) algorithm, a consistent technique to fuse measurements in cooperative navigation networks. The algorithm reduces the excess conservatism of standard Covariance Intersection (CI) by assuming that correlation is only present within each measurement stream and not across the different sources. This assumption allows the sources to be compartmentalized and fused with the Kalman equations rather than the CI method, resulting in tighter convergence. The CCI algorithm is applied to a cooperative localization application and is demonstrated to substantially outperform CI, with a covariance that approaches the performance of an ideal centralized estimator in simulations of linear, Gaussian systems. This approach is also demonstrated to be consistent using Monte Carlo simulations. Finally, CCI was used in a laboratory demonstration to perform distributed localization with real-world range measurements between a team of six agents.**

I. Introduction

A. Motivation

Accurate localization is a critical capability for a wide range of robotic systems operating in real-world environments. Regardless of whether a robot is localizing using a Global Navigation Satellite System (GNSS), local navigation beacons, terrain-relative navigation, dead reckoning, or any other measurement source, improving accuracy and reducing uncertainty are both vital. By combining information from various sensors and agents, collaborative multi-robot localization presents the opportunity to significantly improve navigation performance for these systems.

Robots operating in real-world environments often suffer from computation and communication constraints. These limitations can prohibit the use of centralized architectures where a single estimator fuses measurements from all agents. Distributed localization, on the other hand, is better able to overcome the computation and communication limits, but at the expense of lower navigation accuracy. Improving the performance of distributed localization algorithms to approach that of centralized architectures remains an active research area.

When incorporating information from other robots, distributed localization algorithms must account for the well-known problem of measurement correlation [1]–[4]. Correlation violates the fundamental assumptions of most standard estimators, including the Kalman filter and its variants. If not properly handled, this leads to inconsistent estimators and filter overconvergence. Other authors have proposed consistent estimators that account for measurement correlation, including the Covariance Intersection (CI) filter, which is able to properly incorporate measurements with any unknown amount of correlation. However, CI can be excessively conservative and dramatically increase the estimator's uncertainty.

This paper proposes the Compartmentalized Covariance Intersection (CCI) filter, a variant of regular Covariance Intersection. Like the CI filter, CCI is able to consistently fuse information from a network of robots while accounting for measurement correlation. However, by exploiting the structure of the distributed localization problem to compartmentalize correlated measurements, this approach is able to reduce conservatism and improve filter performance. The intent of this paper is to present the theoretical foundation of the CCI approach, demonstrating that the concept of compartmentalizing filters can lead to substantially tighter convergence while maintaining consistency. Other work relaxes the assumptions necessary to guarantee consistency and applies the concept of CCI to a real-world underwater navigation application [5].

B. Prior Work

Distributed localization has been addressed in many related works. One standard approach consists of using a distributed Extended Kalman Filter (EKF) to estimate the states of all vehicles [2]. This method directly tracks the cross-correlation terms between all the vehicles, allowing standard EKF updates to be performed. However, this approach is equivalent to a centralized estimator running on each agent, and thus requires highly reliable communication between the agents to transmit every sensor reading to all other vehicles. Other authors have proposed distributed algorithms to estimate a joint map state to solve the Simultaneous Localization and Mapping problem, but these are not applied specifically to cooperative localization [6]–[8].

In real-world situations, a filter architecture with each agent estimating only its own state can be preferable to a global state estimator. This allows agents to enter or leave the network at any time, and the filter size does not need to scale with the number of robots. Bahr et al. introduce an algorithm that explicitly tracks the cross-correlation terms between vehicles without fully estimating the joint state [4], but this requires a bank of filters exponential in the number of agents. Channel filters are another approach that directly track the cross-correlation between vehicles [9].

*This work was supported in part by the Monterey Bay Aquarium Research Institute

[1]Adam Wiktor and Stephen Rock are with the Department of Aeronautics & Astronautics, Stanford University, Stanford, CA 94305 awiktor@stanford.edu, rock@stanford.edu

These methods maintain a dedicated channel filter for each other robot, but requires that the network take the form of a tree structure. This requirement reduces the information available to each robot, and also requires the filters to be re-initialized each time the network changes (such as when an agent leaves and re-enters).

Other approaches do not explicitly track the cross-correlation terms between vehicles, but then must be careful to avoid over-convergence due to correlated measurements. Some authors propose heuristics to reduce the effects of correlation, but do not guarantee a consistent estimator [3]. Measurement correlation can also be reduced (but not eliminated) by introducing a hierarchy between the vehicles to force information to flow in a single direction and prevent errors from returning to the original vehicle. These hierarchies still do not result in a consistent estimator, and also reduce the information available to each agent resulting in lower convergence [10].

A final class of algorithms attempts to conservatively bound the effects of the correlation. Julier and Uhlmann introduce the Covariance Intersection filter [11], which can consistently incorporate measurements with any unknown amount of correlation. Later works expand the theory of CI and apply it specifically to distributed localization [12], [13]. Mokhtarzadeh et al. use CI to reduce inertial drift for vehicles in GNSS-denied environments [1], [10], [14]. The authors show that CI is effective for propagating information from vehicles with higher-quality navigation sensors to lower-quality peers, but due to the conservative nature of CI there is no benefit when all vehicles have equal state uncertainty [14].

Under certain assumptions, the CI algorithm can be modified to reduce the conservatism and obtain tighter convergence while remaining consistent. Split CI specifically applies in situations where the correlated measurement can be decomposed into correlated and uncorrelated parts [15]–[17], and uses a standard Kalman update for the uncorrelated component and a CI update for the correlated component. Alternatively, if the correlation is known to be below a pre-specified bound, the Bounded Covariance Inflation filter can be used. However, in practice it is often difficult to derive a bound on the correlation and CI is required anyway [10].

One final consideration is the nonlinear nature of many measurement and dynamics models for multi-agent localization. For these systems, extended Kalman filters are not necessarily optimal or consistent. EKF performance often requires good initial state estimates for systems with range or other nonlinear measurements [18]. Some authors propose using a batch nonlinear optimization approach instead of a sequential filter, but this is often too computationally expensive to be done in real-time [19]. Sequential filters like EKFs are able to incorporate each new measurements as it arrives, which is especially important if communication is sporadic and the number of vehicles changes over time. Therefore, sequential filters are commonly used despite their limitations, potentially using nonlinear optimization techniques for the initial state estimate.

II. Algorithm

A. Problem Definition

Consider a set of N robots navigating in an environment. Each robot i is attempting to estimate its state x_i using ego sensor measurements z_i. Ego measurements are influenced only by the state of robot i and are independent of all other vehicles j. Ego measurements could include information from inertial measurement units (IMU), terrain sensors, velocity measurements, odometry, or a global position sensor. Robots are also able to take relative measurements $r_{i,j}$ to other agents, which could be relative position, range, bearing, or elevation. The robots are able to communicate messages to each other, with at least enough bandwidth to transmit their own state estimate, covariance, and relative measurements. The number of robots N may not be known in advance, and robots can enter or leave the communication network at any time. Robots may be heterogeneous with different dynamics and sensors available to each.

The key assumption for this approach is that correlation is only present due to the distributed localization network structure as described in Section II-B, and all ego sensor measurements z are uncorrelated over time and between all the vehicles. Note that this may not strictly be true for some real-world sensors such as GNSS, where atmospheric effects, multi-path, and broadcast ephemeris errors could be correlated depending on the receiver configuration. Other measurements such as terrain sensors, IMUs, and odometry are better modeled by this assumption for vehicles following independent paths. This method therefore is not intended as a general replacement for CI in situations with correlated sensors, but may be applicable to certain cooperative navigation problems.

B. Correlation in Distributed Localization

Even if all sensors are truly uncorrelated, the nature of distributed localization networks still introduces correlation that precludes the use of standard estimation methods like Kalman filters. Error loops occur when information from one robot is incorporated into other vehicles, and then propagates through the network to be reincorporated by the original robot. This is illustrated by the red arrows in Figure 1(a), where measurements propagate from vehicle 1 to 2 to 3, and finally back to 1. Figure 1(b) eliminates the cycle by introducing a hierarchy, but still does not eliminate error loop correlation since there are several paths for a single measurement to propagate from vehicle 1 to 4.

The network in Figure 1(c) fully eliminates correlation due to error loops, but still must account for temporal correlation. Temporal correlation occurs when a filter repeatedly incorporates measurements with a slowly-changing bias [15]. In this case the measurement from vehicle 1 to 2 consists of a relative measurement $r_{2,1}$ projected from vehicle 1's state estimate $\hat{x}_1$. The state estimate $\hat{x}_1$ has an error (bias) that changes slowly as vehicle 1 incorporates ego measurements. If vehicle 2 repeatedly incorporates measurements from vehicle 1, this bias will lead to temporal correlation between the measurements.

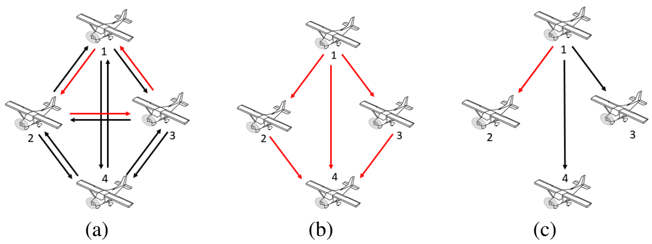

Fig. 1. Sources of correlation in distributed localization networks. Red arrows highlight different flows of information.

C. Compartmentalized Covariance Intersection

The Compartmentalized Covariance Intersection filter is designed to account for both error loops and temporal correlation, while reducing the excess conservatism of standard CI. This is done by exploiting the structure of the problem, isolating the sources of correlation as much as possible and applying CI only when necessary. Instead of a single filter fusing all incoming measurements, CCI splits the estimator into a bank of filters to isolate correlated measurements within each filter. An ego filter onboard each robot i incorporates all ego measurements z_i to generate an estimated ego mean $\hat{x}_i$ and ego covariance P_i. The ego filter, shown in red in Figure 2, is a Kalman filter identical to the single-robot case without distributed localization. Other works [5] extend this approach to non-Gaussian state estimates using a particle ego filter, but the consistency and performance guarantees presented in this paper require the system to be linear and Gaussian.

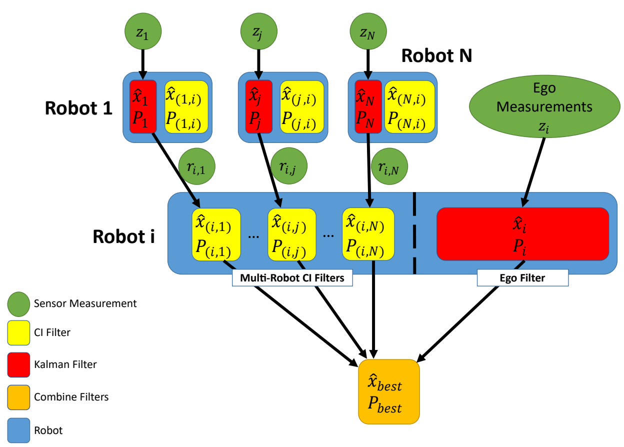

Fig. 2. Compartmentalized Covariance Intersection filter structure.

In addition to the ego filter, the CCI architecture also includes a bank of multi-robot CI filters. Each CI filter on robot i estimates the state of vehicle i based solely on measurements from another robot j, as shown in yellow in Figure 2. The figure shows that information flows solely from the other robot's ego measurements z_j to its ego filter [$\hat{x}_j$, P_j] and then to the corresponding CI filter on robot i with mean $\hat{x}_{(i,j)}$ and covariance $P_{(i,j)}$. This eliminates error loops because information from robot j is isolated in the multi-robot filter and never incorporated into the ego filter on robot i, and therefore never sent on to any other robots. However, CI is still necessary within each multi-robot filter to account for temporal correlation.

All filters in the estimator are propagated independently. The filters are kept separate at all times to avoid mixing correlated information, and are only merged at a higher level to provide a current best estimate of the robot's position (shown in orange in Figure 2). Since the filter architecture eliminates error loops and CI accounts for temporal correlation, each subfilter onboard the robot is independent of one another (again assuming that all ego measurements z are uncorrelated). Therefore, when a best estimate is needed, the filters can be merged using a Kalman Filter update. This is the key insight that provides tighter convergence, since traditional approaches would require the information to be fused using CI.

1) Process Update: In order to propagate the filters independently, it is necessary to account for the fact that measurements in each filter were acquired at different times. Simply applying the standard process update to each filter leads to inconsistency because the process noise is identical for all the filters. Instead, by considering each CI filter as a separate measurement of the state of robot i delayed by Δt timesteps, the method of extrapolated measurements derived by Chagas and Waldmann can be used to consistently propagate the multi-robot CI filters [20]. This method calculates scaling matrices M to conservatively bound the difference between incorporating the measurements at the current timestep instead of at the timestep they were received. In this case, the algorithm requires one scaling matrix $M(t, t_j)$ for each CI filter j to scale the covariance from time t_j when the most recent measurement was received to the current time t. CI filter j also requires a scaling matrix for each other CI filter k to account for the difference between the time of the most recent measurement in each filter, $M(t_j, t_k)$. The equations were derived previously [20], but are reformulated here to apply specifically to the CCI algorithm.

Algorithm 1 shows the process update for the CCI filter from time t to $t+1$. Lines 1 and 2 are the standard Kalman filter process update for the ego filter assuming a dynamics model $x^{t+1} = Fx^t + Bu^t + w$, $w \sim \mathcal{N}(0, Q)$. The rest of the algorithm propagates the multi-robot CI filter means and scaling matrices.

2) Measurement Update: The ego filter is updated for each ego measurement z_i using standard Kalman filter measurement update equations. The CI filter measurement update step for a relative measurement $r_{i,j} = H_{i,j} \left[{x_i^t}^{\intercal} \quad {x_j^t}^{\intercal} \right]^{\intercal} + v_{i,j}$, $v_{i,j} \sim \mathcal{N}(0, R_{i,j})$ is given in Algorithm 2. In order for robot i to fuse a new inter-robot measurement coming from vehicle j at time t, the existing scaling terms M associated with filter j are first incorporated into an equivalent covariance P_{equiv} as shown in lines 1-3. P_{equiv} incorporates all the effects of filter j into a single covariance and is calculated by considering the update that would need to be applied to $P_{\sim j}$ (the best estimate using all measurements except from j) in order to recover P_{best} (the overall best estimate). From the information form of the Kalman filter, $P_{best}^{-1} = P_{\sim j}^{-1} + P_{equiv}^{-1}$. Once P_{equiv} has been calculated,

Algorithm 1 Process_Update()

$\hat{x}_i^{t+1} \leftarrow F\hat{x}_i^t + Bu^t$ ▷ Ego filter process update
$P_i^{t+1} \leftarrow FP_i^tF^\intercal + Q$ ▷ Ego filter process update
$\tilde{H} \leftarrow \left[(P_{(i,1)})^{-1} \cdots (P_{(i,N)})^{-1}\right]^\intercal$
$C_{xy} \leftarrow \left[M(t,t_1)\tilde{H}_1^\intercal \cdots M(t,t_N)\tilde{H}_N^\intercal\right]$
$C_{yy} \leftarrow$ blkdiag($\tilde{H}$) + $\begin{bmatrix} \tilde{H}_1 M(t_1,t_1)\tilde{H}_1^\intercal & \cdots & \tilde{H}_1 M(t_1,t_N)\tilde{H}_N^\intercal \\ \vdots & \ddots & \vdots \\ \tilde{H}_N M(t_N,t_1)\tilde{H}_1^\intercal & \cdots & \tilde{H}_N M(t_N,t_N)\tilde{H}_N^\intercal \end{bmatrix}$
$K \leftarrow C_{xy}(C_{yy})^{-1}$
for $j \in [1,N]$, $j \neq i$ **do** ▷ CI filter process updates
    $\hat{x}_{(i,j)}^{t+1} \leftarrow F\hat{x}_{(i,j)}^t + Bu^t$ ▷ Mean extrapolation
    $M(t+1,t_j) \leftarrow F(I - K\tilde{H})M(t,t_j)$

standard covariance intersection can be used in line 4. The self-scaling term for filter j, $M(t,t_j)$ is reinitialized on line 5. Finally, scaling matrices to capture the time difference between filter j and each other filter k, $M(t_j,t_k)$, are reset as shown in lines 7-9.

Algorithm 2 Multi_Robot_Update($r_{i,j}, R_{i,j}$)

1: $[\hat{x}_{best}, P_{best}] \leftarrow$ Combine_Filters()
2: $[\hat{x}_{\sim j}, P_{\sim j}] \leftarrow$ Combine_Filters(j)
3: $P_{equiv} \leftarrow \left((P_{best})^{-1} - (P_{\sim j})^{-1}\right)^{-1}$
4: $[\hat{x}_{(i,j)}, P_{(i,j)}] \leftarrow$ CI_Update($\hat{x}_{(i,j)}, P_{equiv}, r_{i,j}, R_{i,j}$)
5: $M(t,t_j) \leftarrow P_i$ ▷ Initialize M
6: $t_j \leftarrow t$ ▷ Store measurement time
7: **for** $k \in [1,N]$, $k \neq i,j$ **do**
8:     $M(t_j,t_k) \leftarrow M(t,t_k)$ ▷ Initialize M
9:     $M(t_k,t_j) \leftarrow M(t,t_k)^\intercal$ ▷ Initialize M

3) Merging Filters: The process of merging the ego and multi-robot CI filters is shown in Algorithm 3, again including equations derived by Chagas and Waldmann [20]. The resulting best estimate of position $\hat{x}_{best}, P_{best}$ can be used at a higher level for navigation, but is never re-incorporated into the filters to preserve independence.

The computational complexity of the overall CCI algorithm is driven by the process of combining the filters in Algorithm 3. The most intensive step in Algorithm 3 is inverting C_{yy} in line 8, which is a $(N-1)L\times(N-1)L$ matrix where N is the number of robots and L is the dimension of the state being estimated. Simple matrix inversion algorithms run in cubic time or better, so the overall time complexity is $O((NL)^3)$. This is similar to a centralized Kalman filter estimating the state of all the vehicles, also requiring multiplication and inversion of $NL \times NL$ matrices. A standard decentralized CI filter, on the other hand, would only require manipulation of $L \times L$ matrices.

When inter-robot measurements are received regularly and filters are only propagated a few timesteps between updates, the effect of the scaling matrices is small. In these cases, the scaling matrices may be neglected for a simplified version of CCI. This no longer guarantees consistency, but has the advantage of substantially reducing computational complexity. The process and measurement updates for simplified CCI are identical to standard CI, and combining the filters reduces to a set of N $L \times L$ Kalman updates for an overall complexity of $O(N \cdot L^3)$. The simplified form of CCI was used in other work [5] since the effect of the scaling matrices was dwarfed by the approximations due to nonlinearities and non-Gaussian distributions.

Algorithm 3 $[\hat{x}_{Best}, P_{Best}]$ = Combine_Filters(k (optional))

1: $\tilde{y} \leftarrow [(P_{(i,1)})^{-1}\hat{x}_{(i,1)}, \cdots, (P_{(i,N)})^{-1}\hat{x}_{(i,N)}]^\intercal$
2: $\tilde{H} \leftarrow \left[(P_{(i,1)})^{-1} \cdots (P_{(i,N)})^{-1}\right]^\intercal$
3: **if** $\exists k$ **then** ▷ If an element to exclude, k, was given
4:     $\tilde{y} \leftarrow (\tilde{y}_i)_{i\neq k}$ ▷ If excluding element k, delete it
5:     $\tilde{H} \leftarrow (\tilde{H}_i)_{i\neq k}$
6: $C_{xy} \leftarrow \left[M(t,t_1)\tilde{H}_1^\intercal \cdots M(t,t_N)\tilde{H}_N^\intercal\right]$
7: $C_{yy} \leftarrow$ blkdiag($\tilde{H}$) + $\begin{bmatrix} \tilde{H}_1 M(t_1,t_1)\tilde{H}_1^\intercal & \cdots & \tilde{H}_1 M(t_1,t_N)\tilde{H}_N^\intercal \\ \vdots & \ddots & \vdots \\ \tilde{H}_N M(t_N,t_1)\tilde{H}_1^\intercal & \cdots & \tilde{H}_N M(t_N,t_N)\tilde{H}_N^\intercal \end{bmatrix}$
8: $K \leftarrow C_{xy}C_{yy}^{-1}$
9: $\hat{x}_{Best} \leftarrow \hat{x}_i + K(\tilde{y} - \tilde{H}\hat{x}_i)$ ▷ Compute mean
10: $P_{Best} \leftarrow P_i - KC_{xy}^\intercal$ ▷ Compute covariance

III. Linear Results

A. Simulation Setup

A Matlab simulation was used to demonstrate the performance of compartmentalized covariance intersection with linear dynamics and measurements, as in a standard Kalman filter. The simulation propagates a simple first-order linear system $x_i^{t+1} = x_i^t + u_i\Delta t + w_i,\ w_i \sim \mathcal{N}(0, 0.1m^2I)$ with $\Delta t = 0.1s$ and u_i chosen to drive each robot along a straight-line path from a randomized starting position to a random goal in a 10m by 10m area.

For these experiments, a GNSS-like ego sensor returns a measurement of the 3D position of the ego vehicle plus Gaussian white noise, $z_i = x_i + v_i,\ v_i \sim \mathcal{N}(0, 2m^2I)$. Additionally, an inter-robot sensor provides a relative position measurement to another robot, $r_{i,j} = x_i - x_j + v_{i,j},\ v_{i,j} \sim \mathcal{N}(0, 0.8m^2I)$. The simulation started with each vehicle knowing its own position with high confidence. All experiments were conducted with 6 robots.

B. CCI Performance

The simulations were repeated with varying numbers of robots having access to the GNSS-like sensor. This demonstrates steady-state performance when all vehicles have access to accurate localization; the ability to share information when some vehicles have better-performing sensors; and the ability to bound inertial drift when no robots have access to global navigation sensors. Figure 3 shows the filter performance for three of these cases. The 3-σ covariance bound is plotted, along with errors for a single realization of the filter. All filters are consistent estimators as discussed in Section III-C, so comparing the covariance of the filters is critical whereas the error of a specific simulation is not significant. The filters are estimating the 3D state of the vehicle (x, y, z), but only the x component is plotted since the others are similar.

Figure 3(a) shows a configuration where all vehicles have access to GNSS. The outermost black line represents the single-vehicle case operating with only GNSS and inertial measurements and no inter-vehicle aiding, which is given by the ego filter. The conservative nature of CI offers no improvement for vehicles with identical sensors [14], and therefore the CI filter in red exactly follows the black line. Split CI is able to offer a marginal improvement by separating the uncorrelated relative position measurement form the correlated state estimates. The blue line shows the performance of the ideal centralized estimator, and CCI in green recovers nearly all of the performance of the centralized architecture. The improvement by using CCI over CI is dramatic: while the CI covariance at steady state is 3.4 times larger than the ideal centralized performance, the CCI covariance at steady state is only 1.1 times over ideal. Normalizing the performance by standard CI, where CI has a performance of 0% and the ideal centralized estimator is 100%, CCI has a performance improvement of 95%.

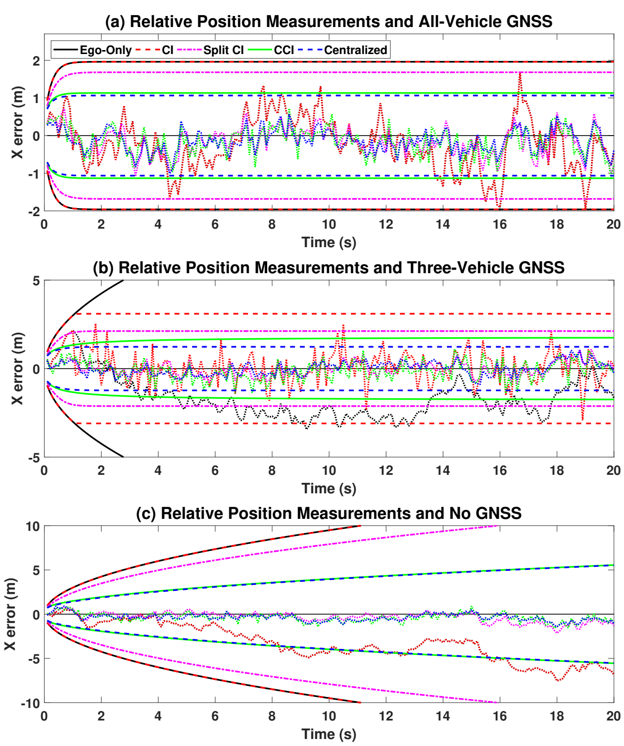

Fig. 3. Filter covariance (3-σ) for 6 robots with relative position measurements.

Similar results are seen in Figure 3(b) for the case where three vehicles with GNSS act as ad-hoc navigational aids to three others, and where none of the vehicles have access to GNSS, 3(c). As before, CCI performance comes closest to the ideal centralized estimator, especially in (c) where the green line is indistinguishable from the blue. The results for all test configurations are summarized in Table I.

TABLE I

COVARIANCE CONVERGENCE IMPROVEMENT

CI = 0%, Centralized = 100%, 20s Average		
# Vehicles with GPS	Split CI	CCI
6	38%	95%
5	68%	93%
4	66%	89%
3	63%	84%
2	59%	75%
1	50%	51%
0	38%	99%

C. CCI Consistency

The argument for CCI consistency rests on the fact that all correlated information is isolated within the individual CI filters. The ego filter on each robot is a Kalman filter processing only the ego inertial and GNSS measurements (identical to a single robot operating in isolation). Since these measurements were assumed to be uncorrelated and the Kalman filter is a consistent estimator for linear systems with zero-mean Gaussian white noise, the ego filter is consistent. Each multi-robot CI filter uses covariance intersection, which has been shown to be consistent for measurements with any unknown amount of correlation [11]. Therefore, each multi-robot filter is consistent. The process update for each filter is consistent because the conservative extrapolation method is used [20]. The only remaining step is merging the filters using a Kalman update. Since the ego measurements were all assumed to be uncorrelated, each CI filter is independent of all other filters and can be merged using the Kalman update. Since this merged estimate is never propagated or incorporated back into a filter, the filters remain independent and therefore consistent.

A Monte Carlo simulation was used to further demonstrate the consistency of the CCI approach. The system was simulated 1000 times for each of the test configurations in Section III-B. Another test configuration with inter-robot measurement updates occurring only every fourth timestep was also simulated in order to demonstrate the consistency of the prediction step. The results from these simulations are shown in Figure 4.

The true estimation error $e = \hat{x} - x$ was normalized by the estimated covariance $\hat{P}$, $\hat{d}^2 = e^T \hat{P}^{-1} e$. The true estimation error normalized by the true covariance P would follow a χ^2 distribution, $d^2 = e^T P^{-1} e \sim \chi^2$. A consistent estimator is defined as one with an estimated covariance greater than or equal to the true covariance, $\hat{P} - P \geq 0$. Therefore, the estimator is consistent if the normalized estimation error $\hat{d}^2$ is less than or equal to the χ^2 distribution, $\hat{d}^2 \leq d^2$. This can be easily seen on a plot of the cumulative distribution function (CDF) if the normalized estimation error is to the left of the χ^2 distribution [21].

The Monte Carlo results are shown in Figure 4. This shows that the CCI filter in green exactly follows the χ^2 distribution and therefore is consistent when using a linear system with relative position measurements. While not shown in Figure 4, the centralized Kalman filter and standard CI also exactly follow the χ^2 CDF. Figure 4 also shows the results when using a naive Kalman filter that does not account for error loops and temporal correlation, demonstrating that it is in fact inconsistent. Finally, a simplified version of the CCI algorithm without accounting for the scaling matrices M is shown in yellow. This shows that while the effect of the process noise correlation is small, it must still be included to guarantee a consistent estimator.

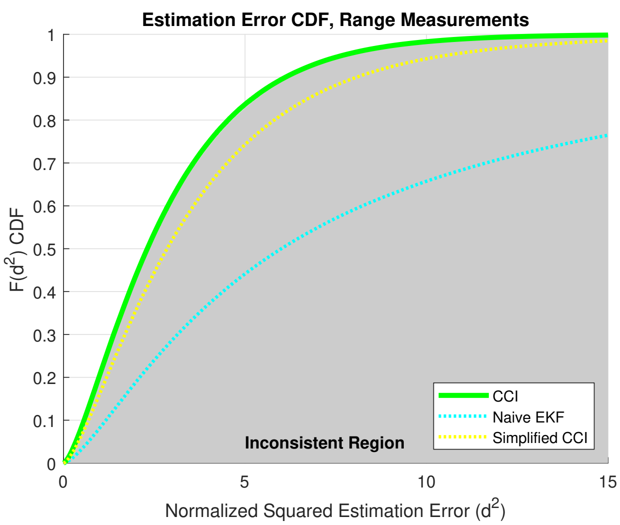

Fig. 4. CCI Monte Carlo consistency results.

IV. Laboratory Demonstration

The focus of this paper is on the theory of the CCI algorithm and the associated convergence and consistency guarantees. This requires similar assumptions to the Kalman filter, specifically a linear system with uncorrelated Gaussian white noise. Other work presents an extension to non-parametric state estimation using particle filters [5]. Simulations using nonlinear range, bearing, and elevation measurements instead of the relative position measurements showed nearly identical results to Section III-B. One additional extension is included here, incorporating range-only measurements with non-Gaussian noise generated using real-world hardware for the inter-robot measurements $r_{i,j}$. Six Pozyx tags were used, with each tag measuring the ranges to the other five tags using an ultra-wideband radio. Range measurements were taken every 0.54 seconds, and were found to have a standard deviation of 0.88m. The Pozyx tags were fitted with reflective markers and carried by hand around a room with OptiTrack cameras capable of measuring the position of the tag with millimeter accuracy. The position and range data was calibrated and processed offline.

The lab data was post-processed with a CCI filter, as well as an single vehicle EKF (ego filter only), CI, split CI, and centralized EKF for comparison. Noise was added to the true position data before being used by the filters as a process update step. Gaussian noise was also added to the true positions before they were used as a GNSS-like sensor for a measurement update, and the range measurements from the Pozyx tags were used without modification. The configurations from Section III-A were repeated, again simulating a scenario with all vehicles having access to the GNSS-like sensor, three vehicles having GNSS, and no vehicles having GNSS. These results are shown in Figure 5. The performance of CCI in this case is not as favorable as the results in Section III-B since the algorithm is poorly conditioned without a fully observable measurement. Despite this limitation, CCI still outperformed CI and split CI (albeit slightly), demonstrating that CCI is effective even when using real-world, nonlinear range measurements.

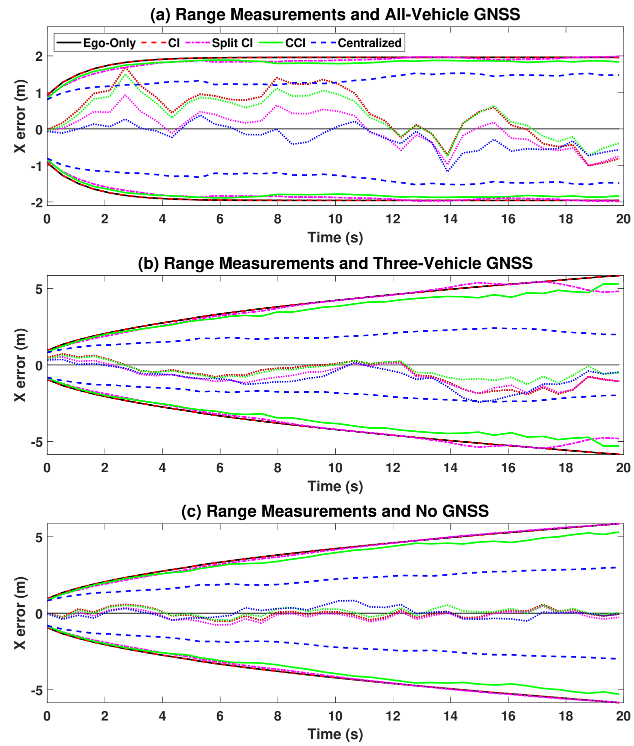

Fig. 5. Filter covariance (3-σ) for lab demonstration of 6 robots with range measurements.

V. Conclusion

CCI is a consistent estimator for multi-agent teams localizing collaboratively. The algorithm reduces the excess conservatism of CI by compartmentalizing the correlated information, allowing the use of the Kalman equations for merging information. For linear systems, CCI substantially outperformed both standard CI and split CI, approaching within 95% of the ideal centralized performance. A Monte Carlo test showed that the method is in fact consistent for linear systems. Finally, this paper also demonstrated a linearized variant analagous to an Extended Kalman filter for range measurements, which lacks the performance guarantees but still performed well in a laboratory demonstration. Future work will focus on demonstrating the performance of the CCI architecture in real-world situations with additional non-linearities and measurements with non-Gaussian noise.

REFERENCES

[1] Hamid Mokhtarzadeh and Demoz Gebre-Egziabher. Cooperative inertial navigation. *Navigation*, 61(2):77–94, 2014.

[2] Stergios Roumeliotis and George Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, Oct 2002.

[3] Amanda Prorok and Alcherio Martinoli. A reciprocal sampling algorithm for lightweight distributed multi-robot localization. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3241–3247, Sep. 2011.

[4] Alexander Bahr, Matthew Walter, and John Leonard. Consistent cooperative localization. In *2009 IEEE International Conference on Robotics and Automation*, pages 3415–3422, May 2009.

[5] Adam Wiktor and Stephen Rock. Collaborative multi-robot localization in natural terrain. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4529–4535, 2020.

[6] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1):3–46, 2016.

[7] A. Cunningham, V. Indelman, and F. Dellaert. Ddf-sam 2.0: Consistent distributed smoothing and mapping. In *2013 IEEE International Conference on Robotics and Automation*, pages 5220–5227, 2013.

[8] Martin J. Schuster, Korbinian Schmid, Christoph Brand, and Michael Beetz. Distributed stereo vision-based 6d localization and mapping for multi-robot teams. *Journal of Field Robotics*, 36(2):305–332, 2019.

[9] Alexei Makarenko and Hugh Durrant-Whyte. Decentralized Data Fusion and Control in Active Sensor Networks. In *2004 International Conference on Information Fusion*, 2004.

[10] Hamid Mokhtarzadeh. Correlated-data fusion and cooperative aiding in gnss-stressed or denied environments. Phd, University of Minnesota, Sep 2014.

[11] Simon J. Julier and Jeffrey K. Uhlmann. Non-divergent estimation algorithm in the presence of unknown correlations. *Proceedings of the American Control Conference*, 4:2369–2373, 1997.

[12] Luis C. Carrillo-Arce, Esha D. Nerurkar, Jose L. Gordillo, and Stergios I. Roumeliotis. Decentralized multi-robot cooperative localization using covariance intersection. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1412–1417. IEEE, nov 2013.

[13] Lingji Chen, Pablo O. Arambel, and Raman K. Mehra. Estimation under unknown correlation: Covariance intersection revisited. *IEEE Transactions on Automatic Control*, 47(11):1879–1882, 2002.

[14] Hamid Mokhtarzadeh and Demoz Gebre-Egziabher. Performance of networked dead reckoning navigation system. *IEEE Transactions on Aerospace and Electronic Systems*, 52(5):2539–2553, October 2016.

[15] Hao Li, Fawzi Nashashibi, and Ming Yang. Split covariance intersection filter: Theory and its application to vehicle localization. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1860–1871, Dec 2013.

[16] Hao Li and Fawzi Nashashibi. Cooperative multi-vehicle localization using split covariance intersection filter. *IEEE Intelligent Transportation Systems Magazine*, 5(2):33–44, 2013.

[17] Thumeera Wanasinghe, George Mann, and Raymond Gosine. Decentralized cooperative localization for heterogeneous multi-robot system using split covariance intersection filter. In *2014 Canadian Conference on Computer and Robot Vision*, pages 167–174, May 2014.

[18] Ath Kehagias, J Djugash, and S Singh. Range-only SLAM with Interpolated Range Data. Technical report, Carnegie Mellon University, Pittsburgh, PA, 2006.

[19] Joseph Djugash, Sanjiv Singh, George Kantor, and Wei Zhang. Range-only SLAM for robots operating cooperatively with sensor networks. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2006, pages 2078–2084, 2006.

[20] Ronan A.J. Chagas and Jacques Waldmann. Extrapolation of delayed measurements for fusion in a distributed sensor network. *24th Mediterranean Conference on Control and Automation, MED 2016*, pages 1319–1324, 2016.

[21] Pavel Ivanov, Simo Ali-Loytty, and Robert Piche. Evaluating the consistency of estimation. In *Proceedings of 2014 International Conference on Localization and GNSS, ICL-GNSS 2014*. Institute of Electrical and Electronics Engineers Inc., oct 2014.