

Research Article

Decentralized Cooperative Localization Approach for Autonomous Multirobot Systems

Thumeera R. Wanasinghe, George K. I. Mann, and Raymond G. Gosine

IS Lab, Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, NL, Canada A1B 3X5

Correspondence should be addressed to Thumeera R. Wanasinghe; ruwansiriwat@gmail.com

Received 4 November 2015; Revised 5 February 2016; Accepted 11 February 2016

Academic Editor: Giovanni Muscato

Copyright © 2016 Thumeera R. Wanasinghe et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study proposes the use of a split covariance intersection algorithm (Split-CI) for decentralized multirobot cooperative localization. In the proposed method, each robot maintains a local cubature Kalman filter to estimate its own pose in a predefined coordinate frame. When a robot receives pose information from neighbouring robots, it employs a Split-CI based approach to fuse this received measurement with its local belief. The computational and communicative complexities of the proposed algorithm increase linearly with the number of robots in the multirobot systems (MRS). The proposed method does not require fully connected synchronous communication channels between robots; in fact, it is applicable for MRS with asynchronous and partially connected communication networks. The pose estimation error of the proposed method is bounded. As the proposed method is capable of handling independent and interdependent information of the estimations separately, it does not generate overconfidence state estimations. The performance of the proposed method is compared with several multirobot localization approaches. The simulation and experiment results demonstrate that the proposed algorithm outperforms the single-robot localization algorithms and achieves approximately the same estimation accuracy as the centralized cooperative localization approach, but with reduced computational and communicative cost.

1. Introduction

With the advancement of information and communication technology, robots from different vendors, different domains of operation, and various perception and operational capabilities are unified into a single framework in order to perform a collaborative task. These heterogeneous multirobot systems (MRS) offer greater coverage in exploration and searching tasks, robustness against individual failures, improved productivity through parallel tasking, and implementation of more complex operations than a single robot [1]. Integration of robots from different domains of operation, such as a robot team with unmanned ground vehicles (UGV) and unmanned aerial vehicles (UAV), enhances team perception while increasing accessibility in a cluttered environment [2].

Robots in a heterogeneous MRS may host different exteroceptive and proprioceptive sensory systems resulting in a significant variation in self-localization across teammates. Interrobot observations and flow of information between

teammates can establish a sensor sharing technique such that the localization accuracy of each member improves over the localization approaches that solely depend on the robot's onboard sensors. This sensor sharing technique is known as cooperative localization and was initially developed to improve the localization accuracy of MRS that are navigating in global positioning system- (GPS-) denied areas or areas without preinstalled (known) landmarks [3, 4]. However, it is reported that a minimum of one agent in a cooperative localization team should possess absolute positioning capabilities in order to have a bounded estimation error and uncertainty [5].

Available cooperative localization algorithms can be classified into three groups: centralized/multicentralized approaches, distributed approaches, and decentralized approaches. Although the centralized/multicentralized approaches generate consistent pose estimate with high accuracy, these approaches entail considerably higher computational and communicative cost. Distributed cooperative localization

algorithms attempt to reduce the communication cost. However, the computational cost of most of the distributed algorithms remains as high as the centralized/multicentralized approaches. Decentralized schemes have reduced both the computational and communication cost of multirobot cooperative localization tasks. However, available decentralized cooperative localization (DCL) schemes sometime generate overconfidence pose estimates for robots in an MRS.

Therefore, computing a nonoverconfidence state estimate with bounded error while reducing computation and communication cost is a key requirement for successful implementation of multirobot collaborative missions that rely on cooperative localization. In this paper, we extend our previous work presented in [6] and propose a scalable DCL approach for heterogeneous MRS which is guarantee for nonoverconfidence pose estimate with the bounded estimation error. The proposed method applies the split covariance intersection (Split-CI) algorithm to accurately track independencies and interdependencies between teammates' local pose estimates. The recently developed cubature Kalman filter (CKF) is exploited for sensor fusion. Each robot periodically measures the relative pose of its neighbours. These measurements and associated measurement uncertainties are first transformed into a common reference frame and then communicated to corresponding neighbour robots. Once a robot receives pose measurements from a neighbour, it uses the Split-CI algorithm to fuse received measurements with its local belief. The work presented in this paper offers the following key contributions: (i) a novel DCL approach that combines the properties of the Split-CI algorithm and CKF in order to avoid double counting of the information while generating pose estimation with bounded estimation error, (ii) an extension to the general CKF to accurately calculate and maintain both the independent and the dependent covariances, and (iii) a consistent and debiased algorithm to convert information between two Cartesian coordinate systems. The proposed algorithm has the following properties: (i) per-measurement computational and communication cost of the proposed DCL approach is constant, (ii) there is no requirement for a synchronous or fully connected communication network, (iii) the algorithm is robust against the single point of failure, and (iv) the algorithm is capable of generating a nonoverconfidence pose estimate with the bounded estimation error for each member of the MRS.

2. Background

In 1994, Kurazume et al. proposed the first cooperative localization algorithm [3]. Ever since, numerous approaches for cooperative localization have been reported. These implementations can be categorized into three main groups: (1) centralized/multicentralized cooperative localization, (2) distributed cooperative localization, and (3) decentralized cooperative localization.

Centralized cooperative localization approaches augment each robot's pose into a single state vector and perform state estimation tasks at a central processing unit [3, 4, 8–12]. The computational cost of the centralized algorithms scales to

$\mathcal{O}(|N|^4)$, where $|N|$ is the number of robots in the MRS. Since these implementations require each robot to communicate its high frequency egocentric data and interrobot observations to a central processing unit, communication links should have wider bandwidth. Multicentralized approaches have been reported to improve the robustness of the general centralized approaches against the single point of failure by duplicating the state estimation process on a few or all robots in the team [12, 13]. This causes an increasing per measurement communicative cost linearly with respect to the number of independent processing units. A multicentralized approach that enables cooperative localization for sparsely communicating robot networks requires more communication bandwidth and onboard memory than traditional multicentralized approaches [14].

In the distributed cooperative localization, each robot locally runs a filter to fuse egocentric measurements and propagate the state over time. As a result, there is no requirement for exchanging high-frequency egocentric measurements with teammates or with a central server. This enables the implementation of distributed cooperative localization algorithms using communication channels with limited bandwidth. However, interrobot observations are still fused at a central processor leading to the same computational complexity as the centralized/multicentralized approaches (i.e., $\mathcal{O}(|N|^4)$). Work presented in [15] proposed distributed maximum a posteriori estimation for cooperative localization and achieved reduced computational complexity, that is, $\mathcal{O}(|N|^2)$, and also improved robustness against the single point of failure.

Decentralized algorithms focus on reducing both computational and communicative complexity. In general, each robot in a DCL team locally runs a filter (estimator) to fuse its ego-centric measurements as well as the interrobot observations received from its neighbours with its local belief. Available decentralized algorithms perform the sensor fusion either in a suboptimal manner [6, 16–21] or in an approximate manner [22–25]. Approximate approaches assume that a given robot local pose estimate is independent from the measurements sent by neighbours. This assumption leads to an overconfidence state estimation. Work presented in [17] uses a *dependency-tree* to track the recent interaction between robots. However, this approach maintains only the recent interdependencies of robot pose estimate; it tends to be overconfidence. An interlaced extended Kalman filter-(EKF-) based suboptimal filtering approach is presented in [18, 19] to avoid the possibility of generating an overconfidence state estimation. This approach requires each robot in the MRS to maintain a bank of EKFs representing the interaction between teammates. Although it produces a nonoverconfidence state estimation, this book-keeping approach is unscalable, as the number of EKF runs on a single robot increases exponentially with the number of robots in the MRS. A suboptimal filtering approach called *channel filtering* is presented in [16] which requires a communication network without loops. However, a communication network without loops is an unrealistic assumption for the practical implementation of cooperative localization. An extended information filter is used for implementing the DCL [20],

in which each robot maintains the history of robot-to-robot relative measurement updates. CI-based approaches are also reported for the DCL [21]. However, the general CI algorithm neglects possible independencies between local estimates. This causes more conservative estimates and may produce an estimation error covariance which is larger than that of the best unfused estimate [26].

Available cooperative localization algorithms use four types of interrobot observations: relative range-only [27, 28], relative bearing-only [29, 30], both relative range and relative bearing [31–33], and full relative pose measurement [5, 6, 21, 24, 25, 34]. Full relative pose measurement-based cooperative localization schemes always generate more accurate pose estimate than range-only, bearing-only, and range-and-bearing measurement systems [35]. The lowest estimation accuracy was found with the bearing-only measurement system [35].

3. Problem Statement

To facilitate the mathematical formulation, let N represent the set that contains the unique identification indices of all robots in the MRS. The cardinality of this set, that is, $|N|$, corresponds to the total number of robots in the MRS. Further, it is assumed that the robots are represented by $\{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_{|N|}\}$ where the identification indices of the robots range from 1 to $|N|$. The set N and its cardinality $|N|$ may or may not be known to each agent in the MRS.

It is assumed that the robots are navigating in a flat, two-dimensional (2D) terrain. Each robot in the MRS hosts a communication device to exchange information between teammates; an interoceptive sensory system to measure ego-motion (linear and angular velocities); and an exteroceptive sensory system to measure the relative pose of neighbouring robots. Known data correspondence is assumed for the exteroceptive sensory system. Besides these sensory systems and communication modules, some of the robots in the MRS host a sensory system, such as a differential global positioning system (DGPS), to receive absolute positioning information periodically.

3.1. Robot's Motion Model. Robot navigation in a 2D space is modeled by a general three-degree-of-freedom (3-DOF) discrete-time kinematic model

$$\begin{aligned} \mathbf{x}_{q,k} &= \mathbf{g}(\mathbf{x}_{q,k-1}, \bar{\mathbf{u}}_{q,k-1}) \quad \forall q \in N, \\ \begin{bmatrix} x \\ y \\ \phi \end{bmatrix}_{q,k} &= \begin{bmatrix} x \\ y \\ \phi \end{bmatrix}_{q,k-1} \\ &+ \delta t \begin{bmatrix} \bar{v}_x \cos(\phi) - \bar{v}_y \sin(\phi) \\ \bar{v}_x \sin(\phi) + \bar{v}_y \cos(\phi) \\ \bar{\omega}_z \end{bmatrix}_{q,k-1}, \end{aligned} \quad (1)$$

where $\mathbf{x}_{q,k} \in \mathbb{R}^3$ is the robot's pose with respect to a given coordinate frame (or global coordinate frame) at discrete time k . The nonlinear state propagation function and the

sampling time are represented by $\mathbf{g}(\cdot)$ and δt , respectively. $\bar{\mathbf{u}}_{q,k} \in \mathbb{R}^3$ is the system input and $\bar{\mathbf{u}}_{q,k} = \mathbf{u}_{q,k} + \boldsymbol{\nu}_{q,k}$, where $\mathbf{u}_{q,k} = [v_x \ v_y \ \omega_z]^T$. The parameters v_x and v_y are nominal linear velocities in x - and y -directions, respectively. ω_z is the nominal angular velocity. $\boldsymbol{\nu}_{q,k}$ represents the additive white Gaussian noise term with covariance $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$. For nonholonomic robots, terms associated with linear velocities in y -direction, that is, $\bar{v}_y \sin(\phi)$ and $\bar{v}_y \cos(\phi)$, should be omitted from the system model.

3.2. Interrobot Relative Measurement Model. Consider the scenario where robot \mathcal{R}_q measures relative pose of robot \mathcal{R}_r . This relative pose measurement is modeled as

$$\begin{aligned} \mathbf{y}_{q,k}^{r,q} &= \mathbf{h}(\mathbf{x}_{q,k}, \mathbf{x}_{r,k}) + \mathbf{n}_{q,k}^{r,q}, \\ q &\in N, \quad r \in N_{q,k}, \quad d_{q,k}^{r,q} \leq d_m, \\ \begin{bmatrix} \delta x \\ \delta y \\ \delta \phi \end{bmatrix}_{q,k}^{r,q} &= \mathbf{\Gamma}_{\mathbf{x}_{q,k}}^T (\mathbf{x}_{r,k} - \mathbf{x}_{q,k}) + \mathbf{n}_{q,k}^{r,q} \end{aligned} \quad (2)$$

with

$$\begin{aligned} \mathbf{\Gamma}_{\mathbf{x}_{q,k}} &= \begin{bmatrix} \mathbf{C}(\phi_{q,k}) & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix}, \\ \mathbf{C}(\phi_{q,k}) &= \begin{bmatrix} \cos(\phi_{q,k}) & -\sin(\phi_{q,k}) \\ \sin(\phi_{q,k}) & \cos(\phi_{q,k}) \end{bmatrix}, \end{aligned} \quad (3)$$

where $\mathbf{y}_{q,k}^{r,q} \in \mathbb{R}^3$ is the relative pose of robot \mathcal{R}_r as measured by \mathcal{R}_q ; that is, $\mathbf{y}_{q,k}^{r,q} = [\delta x_{q,k}^{r,q} \ \delta y_{q,k}^{r,q} \ \delta \phi_{q,k}^{r,q}]^T$, where δx , δy , and $\delta \phi$ are x -position, y -position, and the orientation of robot \mathcal{R}_r with respect to robot \mathcal{R}_q . This pose measurement is on the body-fixed coordinate system of robot \mathcal{R}_q . The nonlinear measurement function is represented by $\mathbf{h}(\cdot)$. The measurement noise covariance, $\mathbf{n}_{q,k}^{r,q}$, is assumed to be an additive white Gaussian noise with covariance $\mathbf{R}_{q,k}^{r,q} \in \mathbb{R}^{3 \times 3}$. Parameters $d_{q,k}^{r,q}$ and d_m represent the distance between two robots and the sensing range of robot \mathcal{R}_q , respectively. It is assumed that the communication range of a given robot is greater than or equal to its sensing range. $N_{q,k}$ represents the set that contains the unique identification indices of robots which are within the sensing range of robot \mathcal{R}_q at the discrete time k . The matrix transpose operation is represented by T . At a given time step, there may be no robots, one robot, or multiple robots operating within the sensing range of \mathcal{R}_q . The maximum number of robots that can operate within the sensing range of \mathcal{R}_q is one robot less than the total number of robots in the MRS, that is, $|N| - 1$. These imply that $0 \leq |N_{q,k}| \leq |N| - 1$ for $k = 1, 2, \dots, \infty$. Let

$$\mathcal{Y}_{q,k}^q = \left\{ p \left(\mathbf{z}_{q,k}^{r,q} \right) \mid r \in N_{q,k}, \quad d_{q,k}^{r,q} \leq d_m \right\} \quad (4)$$

represent all relative pose measurements acquired by \mathcal{R}_q at time step k , where $p(\mathbf{z}_{q,k}^{r,q}) = \mathcal{N}(\mathbf{y}_{q,k}^{r,q}, \mathbf{R}_{q,k}^{r,q})$. Symbol $\mathcal{N}(a, b)$ represents that the distribution is Gaussian with mean (actual measurement) a and covariance b . The cardinality of the set $\mathcal{Y}_{q,k}^q$ follows the same property as set $N_{q,k}$; that is, $0 \leq |\mathcal{Y}_{q,k}^q| \leq |N| - 1$ for $k = 1, 2, \dots, \infty$.

4. Decentralized Cooperative Localization Algorithm

4.1. State Propagation. The objective of the state propagation step is to predict current pose and the associated estimation uncertainties of a given robot using both the robot's posterior state density and the odometry reading at the previous time step. In order to avoid the cyclic update (cyclic update is the process that uses the same information more than once), each robot maintains two covariance matrices: total covariance and independent covariance. Once the total and independent covariances are known, dependent covariance can be calculated as

$$P_{q_d,k} = P_{q,k} - P_{q_i,k}, \quad (5)$$

where $P_{q,k}$, $P_{q_i,k}$, and $P_{q_d,k}$ are total covariance, independent covariance, and dependent covariance of R_q 's pose estimation at time step k , respectively.

This study employs CKF for sensor fusion. CKF is a recently developed suboptimal nonlinear filter which uses the spherical-radial cubature rule to solve the multidimensional integral associated with the Bayesian filter under the Gaussian approximation [36]. CKF is a Jacobian-free approach that is always guaranteed to have a positive definite covariance matrix and has demonstrated superior performance than the celebrity EKF and the unscented Kalman filter (UKF) [37–39].

For a system with n state variables, the third-order spherical-radial cubature rule selects $2n$ cubature points in order to compute the standard Gaussian integral, as follows:

$$I_{\mathcal{N}}(\mathbf{f}) = \int_{\mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \hat{\mathbf{P}}) d\mathbf{x} \approx \sum_{i=1}^{2n_x} \frac{1}{2n} \mathbf{f}(\hat{\mathbf{x}} + \mathbf{S}\xi_i), \quad (6)$$

where the square-root factor \mathbf{S} of the covariance $\hat{\mathbf{P}}$ satisfies the equality $\hat{\mathbf{P}} = \mathbf{S}\mathbf{S}^T$. The cubature points ξ_i are given by

$$\xi_i = \begin{cases} \sqrt{n}e_i, & i = 1, 2, \dots, n, \\ -\sqrt{n}e_{i-n}, & i = n+1, n+2, \dots, 2n, \end{cases} \quad (7)$$

where $e_i \in \mathbb{R}^{n_x \times 1}$ represents the i th elementary column vector. In this paper, robot pose and odometry vectors are augmented into a single state vector leading to $n = n_x + n_c$, where n_x is the size of pose vector and n_c is the size of the odometry vector. Further, the general CKF algorithm is extended to accommodate independent covariance calculation capabilities.

The proposed state propagation approach is summarized as follows.

Algorithm 1 (state propagation).

Data. Assume at time k posterior density function of robot's pose estimation $p(\mathbf{x}_{q,k_+}) = \mathcal{N}(\hat{\mathbf{x}}_{q,k_+}, \mathbf{P}_{q,k_+})$, independent covariance matrix \mathbf{P}_{q_i,k_+} , and odometry reading $\bar{\mathbf{u}}_{q,k}$ are known.

Result. Calculate the predictive density function of robot's pose estimation $p(\mathbf{x}_{q,(k+1)_-}) = \mathcal{N}(\hat{\mathbf{x}}_{q,(k+1)_-}, \mathbf{P}_{q,(k+1)_-})$ and associated independent covariance matrix $\mathbf{P}_{q_i,(k+1)_-}$.

- (1) Augment state end odometry reading into single vector:

$$\hat{\mathbf{x}}_{k_+} = [\hat{\mathbf{x}}_{q,k_+}^T \quad \bar{\mathbf{u}}_{q,k}^T]^T. \quad (8)$$

- (2) Compute the corresponding covariance matrix:

$$\mathbf{P}_{k_+} = \begin{bmatrix} \mathbf{P}_{q,k_+} & \mathbf{0}_{n_x \times n_c} \\ \mathbf{0}_{n_c \times n_x} & \mathbf{Q} \end{bmatrix}. \quad (9)$$

- (3) Factorize

$$\mathbf{P}_{k_-} = \mathbf{S}_{k_+} \mathbf{S}_{k_+}^T. \quad (10)$$

- (4) Generate cubature points ($j = 1, 2, \dots, m$):

$$\mathbf{X}_{j,k_+} = \mathbf{S}_{k_+} \xi_j + \hat{\mathbf{x}}_{k_+}, \quad (11)$$

where $m = 2(n_x + n_c)$.

- (5) Propagate each set of cubature points through nonlinear state propagation function given in (1) ($j = 1, 2, \dots, m$):

$$\mathcal{X}_{j,(k+1)_-} = \mathbf{g}(\mathbf{X}_{j(1:n_x),k_+}, \mathbf{X}_{j(n_x+1:n_x+n_c),k_+}). \quad (12)$$

- (6) Predict next state:

$$\hat{\mathbf{x}}_{q,(k+1)_-} = \frac{1}{m} \sum_{j=1}^m \mathcal{X}_{j,(k+1)_-}. \quad (13)$$

- (7) Estimate the predictive error covariance:

$$\mathbf{P}_{q,(k+1)_-} = \frac{1}{m} \sum_{j=1}^m \mathcal{X}_{j,(k+1)_-} \mathcal{X}_{j,(k+1)_-}^T - \hat{\mathbf{x}}_{q,(k+1)_-} \hat{\mathbf{x}}_{q,(k+1)_-}^T. \quad (14)$$

- (8) To calculate independent covariance, construct new block diagonalization covariance matrix as follows:

$$\mathbf{P}_{i,k_+} = \begin{bmatrix} \mathbf{P}_{q_i,k_+} & \mathbf{0}_{n_x \times n_c} \\ \mathbf{0}_{n_c \times n_x} & \mathbf{Q} \end{bmatrix}. \quad (15)$$

- (9) Factorize \mathbf{P}_{i,k_+} , then generate a new set of cubature points, and propagate this new cubature point set through the nonlinear state propagation function (1) (refer to lines (3), (4), and (5) for equations).

(10) Predict new state using independent covariance:

$$\hat{\mathbf{x}}_{q_i,(k+1)_-} = \frac{1}{m} \sum_{j=1}^m \mathcal{X}_{j,(k+1)_-}. \quad (16)$$

(11) Estimate the independent predictive error covariance:

$$\begin{aligned} \mathbf{P}_{q_i,(k+1)_-} &= \frac{1}{m} \sum_{j=1}^m \mathcal{X}_{j,(k+1)_-} \mathcal{X}_{j,(k+1)_-}^T \\ &\quad - \hat{\mathbf{x}}_{q_i,(k+1)_-} \hat{\mathbf{x}}_{q_i,(k+1)_-}^T. \end{aligned} \quad (17)$$

n_x is size of robot's pose vector, n_c is size of robot's odometry vector, $\mathbf{0}_{a \times b}$ is matrix with a rows and b columns and all entries are zeros, k_+ represents $k | k$, and $(k+1)_-$ represents $(k+1) | k$.

The algorithm is initialized with known prior density $p(\mathbf{x}_{q,k_+}) = \mathcal{N}(\hat{\mathbf{x}}_{q,k_+}, \mathbf{P}_{q,k_+})$, independent covariance matrix \mathbf{P}_{q_i,k_+} , and odometry reading $\bar{\mathbf{u}}_{q,k}$ at the previous time step (say, time step k). The algorithm predicts the robot pose for the next time step along with associated total and independent covariances. First, the algorithm augments the estimated pose vector $\hat{\mathbf{x}}_{q,k_+}$ with the odometry vector at time k (line (1)). The associated covariance matrix is then computed by block-diagonalization of the estimation and process covariance matrices (line (2)). In the CKF, a set of the cubature points is used to represent the current estimated pose and associated estimation uncertainties (line (4)). To generate these cubature points, the square-root factor of the covariance matrix is required. Any matrix decomposition approach that preserves the equality given in (10) can be exploited to compute the square-root factor of the covariance matrix (line (3)). The cubature points that represent current state and odometry measurements are evaluated on the nonlinear state propagation function (line (5)), which generates the cubature point distribution for a predicted state. The predicted pose (or state) of the robot is the average of the propagated cubature points (line (6)). Total predictive covariance is then computed from (14). Once the total predictive covariance is calculated, a new block-diagonalized covariance matrix, that is, \mathbf{P}_{i,k_+} , is generated using the independent covariance matrix of time k , that is, \mathbf{P}_{q_i,k_+} , and process covariance matrix, that is, \mathbf{Q} (line (8)). After computing \mathbf{P}_{i,k_+} , its square-root factor is computed (similar to line (3)); then a set of cubature points are generated using the new square-root factor (similar to line (4)); and, finally, the newly generated cubature points are propagated through the nonlinear state propagation function (similar to line (5)) (line (9)). These steps are followed by the computation of prediction for independent propagated state (line (10)) and the associated covariance matrix (line (11)).

4.2. Computing Pose of Neighbours. At a relative pose measurement event each robot acquires the relative pose of its neighbours. These measurements are in the local coordinate system of the observing robot and are required to be transformed to the reference coordinate system prior to executing the sensor fusion at the neighbouring robot's local processor.

Assume that, at time $(k+1)$, robot \mathcal{R}_q measures the relative pose of robot \mathcal{R}_r . This nonlinear coordinate transformation can be modeled as

$$\begin{aligned} \mathbf{y}_{q,k+1}^{r,*} &= \mathbf{f}(\hat{\mathbf{x}}_{q,(k+1)_-}, \mathbf{y}_{q,k+1}^{r,q}) = \hat{\mathbf{x}}_{q,(k+1)_-} \oplus \mathbf{y}_{q,k+1}^{r,k} \\ &= \hat{\mathbf{x}}_{q,(k+1)_-} + \Gamma_{\mathbf{x}_{q,(k+1)_-}} \mathbf{y}_{q,k+1}^{r,k}, \end{aligned} \quad (18)$$

where \oplus is known as the pose composition operator and $\mathbf{y}_{q,k+1}^{r,*}$ is the global pose of \mathcal{R}_r on the reference coordinate frame, as measured by \mathcal{R}_q . The superscript asterisk "*" is used to indicate that the measurement is on the reference coordinate system. Symbol $\Gamma_{\mathbf{x}_{q,(k+1)_-}}$ has the same meaning as (2). Since this Cartesian-to-Cartesian transformation is nonlinear, a cubature point-based approach, as summarized in Algorithm 2, is exploited to achieve consistent and unbiased coordinate transformation (see Appendix).

Algorithm 2 (relative to global conversion).

Data. Assume at time k the predictive density function of a robot's (say, \mathcal{R}_q) pose estimation $p(\mathbf{x}_{q,(k+1)_-}) = \mathcal{N}(\hat{\mathbf{x}}_{q,(k+1)_-}, \mathbf{P}_{q,(k+1)_-})$, independent covariance matrix $\mathbf{P}_{q_i,(k+1)_-}$, and relative pose measurement of a neighbour (say, \mathcal{R}_r) are available.

Result. Calculate global pose measurement of \mathcal{R}_r , that is, $\mathbf{y}_{q,k+1}^{r,*}$, and associated independent and dependent measurement covariances, i.e., $\mathbf{R}_{q_i,k+1}^*$ and $\mathbf{R}_{q_d,k+1}^*$.

(1) Augment the predictive state and relative pose into single vector:

$$\mathbf{y}_{k+1} = \left[\mathbf{x}_{q,(k+1)}^T \left(\mathbf{y}_{q,(k+1)}^{r,*} \right)^T \right]. \quad (19)$$

(2) Construct corresponding covariance matrix:

$$\bar{\mathbf{R}}_{k+1} = \begin{bmatrix} \mathbf{P}_{q,(k+1)_-} & \mathbf{0}_{n_x \times n_x} \\ \mathbf{0}_{n_x \times n_x} & \mathbf{R}_q \end{bmatrix}. \quad (20)$$

(3) Factorize

$$\bar{\mathbf{R}}_{k+1} = \bar{\mathbf{S}}_{k+1} \bar{\mathbf{S}}_{k+1}^T. \quad (21)$$

(4) Generate set of cubature points ($j = 1, 2, \dots, m$):

$$\mathcal{Y}_{j,k+1}^{r,q} = \bar{\mathbf{S}}_{k+1} \boldsymbol{\xi}_j + \mathbf{y}_{k+1}, \quad (22)$$

where $m = 2 * n_x$.

(5) Perform coordinate transform for each set of cubature points ($j = 1, 2, \dots, m$):

$$\mathcal{Y}_{j,k+1}^{r,*} = \mathbf{f} \left(\mathcal{Y}_{j(1:n_x),k+1}^{r,q}, \mathcal{Y}_{j(n_x+1:2n_x),k+1}^{r,q} \right). \quad (23)$$

(6) Compute global pose of neighbour:

$$\mathbf{y}_{q,k+1}^{r,*} = \frac{1}{m} \sum_{j=1}^m \mathcal{Y}_{j,k+1}^{r,*}. \quad (24)$$

(7) Compute total noise (error) covariance:

$$\mathbf{R}_{q,k+1}^{r,*} = \frac{1}{m} \sum_{j=1}^m (\mathcal{Y}_{j,k+1}^{r,*}) (\mathcal{Y}_{j,k+1}^{r,*})^T - (\mathbf{y}_{q,k+1}^{r,*}) (\mathbf{y}_{q,k+1}^{r,*})^T. \quad (25)$$

(8) Construct a block-diagonalized matrix using independent predictive covariance and measurement noise covariance:

$$\bar{\mathbf{R}}_{i,k+1} = \begin{bmatrix} \mathbf{P}_{q_i,(k+1)_-} & \mathbf{0}_{n_x \times n_x} \\ \mathbf{0}_{n_x \times n_x} & \mathbf{R}_q \end{bmatrix}. \quad (26)$$

(9) Factorize $\bar{\mathbf{R}}_{i,k+1}$ and then generate a new set of cubature points followed by the coordination transformation for each cubature point (refer to lines (3), (4), and (5) for equations).

(10) Compute coordinate transformed measurement using independent covariance:

$$\mathbf{y}_{q_i,k+1}^{r,*} = \frac{1}{m} \sum_{j=1}^m \mathcal{Y}_{j,k+1}^{r,*}. \quad (27)$$

(11) Estimate independent covariance for the pose measurement:

$$\mathbf{R}_{q_i,k+1}^{r,*} = \frac{1}{m} \sum_{j=1}^m (\mathcal{Y}_{j,k+1}^{r,*}) (\mathcal{Y}_{j,k+1}^{r,*})^T - (\mathbf{y}_{q_i,k+1}^{r,*}) (\mathbf{y}_{q_i,k+1}^{r,*})^T. \quad (28)$$

(12) Estimate dependent covariance for the pose measurement:

$$\mathbf{R}_{q_d,k+1}^{r,*} = \mathbf{R}_{q,k+1}^{r,*} - \mathbf{R}_{q_i,k+1}^{r,*}. \quad (29)$$

The algorithm is initialized with a known predictive density of the pose estimation of the observing robot along with the predictive independent covariances. At an inter-robot-relative-pose measurement event, the observing robot augments its predictive pose and relative pose measurement into a single state vector (line (1)). The associated covariance matrix is obtained by block-diagonalization of the predictive total covariance ($\mathbf{P}_{q,k+1}_-$) and noise covariance of the relative pose measurement (\mathbf{R}_q) (line (2)). This block-diagonalized covariance matrix is then factorized and exploited for generating a set of cubature points to represent the state vector (lines (3) and (4)). The generated cubature points are evaluated on the nonlinear Cartesian-to-Cartesian coordinate transformation function, that is, (18), in order to compute the coordinate transformed cubature points (line (5)). This step is followed by the computation of the observed robot pose in the reference coordinate system (line (6)) and associated total noise (error) covariance matrix (line (7)). Once the total noise covariance is calculated, the algorithm constructs

a new block-diagonalized covariance matrix, $\bar{\mathbf{R}}_{i,k+1}$, using the predictive independent covariance matrix and the relative pose measurement noise covariance matrix (line (8)). After computing $\bar{\mathbf{R}}_{i,k+1}$, its square-root factor is computed similar to line (3); then a set of cubature points are generated using the new square-root factor (similar to line (4)); and, finally, newly generated cubature points are transformed from the local coordinate system to the global coordinate system (similar to line (5)) (line (9)). These steps are followed by the computing of coordinate transformed measurement and associated independent noise covariance matrix (lines (10) and (11)). Finally, the dependent covariance of the coordinate transformed measurement is calculated as the difference between total and independent covariances (line (12)).

4.3. Update Local Pose Estimation Using the Pose Sent by Neighbours. When a robot receives a pose measurement from a neighbour, this measurement is fused with the observed robot's local estimate in order to improve its localization accuracy.

Algorithm 3 (state update with the measurement sent by neighbours).

Data. Assume predictive density of robot pose estimation $p(\mathbf{x}_{r,(k+1)_-} = \mathcal{N}(\hat{\mathbf{x}}_{r,(k+1)_-}, \mathbf{P}_{r,(k+1)_-})$, the associated independent covariance matrix $\mathbf{P}_{q_i,(k+1)_-}$, and pose measurements from a neighbour $\mathbf{y}_{q_i,k+1}^{r,*}$ along with the associated independent and dependent covariances are available.

Result. Calculate posterior density of time $k + 1$, that is, $p(\mathbf{x}_{r,(k+1)_+} = \mathcal{N}(\hat{\mathbf{x}}_{r,(k+1)_+}, \mathbf{P}_{r,(k+1)_+})$, and the associated independent covariance matrix $\mathbf{P}_{r_i,(k+1)_+}$.

(1) Calculate the predictive dependant covariance:

$$\mathbf{P}_{q_d,(k+1)_-} = \mathbf{P}_{q,(k+1)_-} - \mathbf{P}_{q_i,(k+1)_-}. \quad (30)$$

(2) Compute the weighted predictive covariance

$$\mathbf{P}_1 = \frac{\mathbf{P}_{q_d,(k+1)_-}}{\alpha} + \mathbf{P}_{q_i,(k+1)_-}. \quad (31)$$

(3) Compute the weighted measurement covariance

$$\mathbf{P}_2 = \frac{\mathbf{R}_{q_d,k+1}^{r,*}}{1 - \alpha} + \mathbf{R}_{q_i,k+1}^{r,*}. \quad (32)$$

(4) **if** measurement gate validated **then**

(5) Compute Kalman gain

$$\mathbf{K} = \mathbf{P}_1 (\mathbf{P}_1 + \mathbf{P}_2)^{-1}. \quad (33)$$

(6) Update robot pose

$$\hat{\mathbf{x}}_{r,(k+1)_+} = \hat{\mathbf{x}}_{r,(k+1)_-} + \mathbf{K} (\mathbf{y}_{q_i,k+1}^{r,*} - \hat{\mathbf{x}}_{r,(k+1)_-}). \quad (34)$$

(7) Update total covariance

$$\mathbf{P}_{r,(k+1)_+} = (\mathbf{I}_{n_x} - \mathbf{K}) \mathbf{P}_1. \quad (35)$$

(8) Update independent covariance

$$\begin{aligned} \mathbf{P}_{r_i,(k+1)_+} = & (\mathbf{I}_{n_x} - \mathbf{K}) \mathbf{P}_{r_i,(k+1)_-} (\mathbf{I}_{n_x} - \mathbf{K})^T \\ & + \mathbf{K} \mathbf{R}_{q_i,k+1}^{r,*} \mathbf{K}^T. \end{aligned} \quad (36)$$

(9) **else**

(10) Assign predictive state and covariances into posterior state and covariances

$$\begin{aligned} \widehat{\mathbf{x}}_{r_i,(k+1)_+} & \leftarrow \widehat{\mathbf{x}}_{r_i,(k+1)_-} \\ \mathbf{P}_{r_i,(k+1)_+} & \leftarrow \mathbf{P}_{r_i,(k+1)_-} \\ \mathbf{P}_{r_i,(k+1)_+} & \leftarrow \mathbf{P}_{r_i,(k+1)_-}. \end{aligned} \quad (37)$$

(11) **end if**

\mathbf{I}_{n_x} is identity matrix of ($n_x \times n_x$) and α is weighting coefficient and belongs to the interval $[0, 1]$.

Algorithm 3 summarizes the steps involved in this measurement update. The algorithm is initialized by calculating the observed robot's predictive dependent covariance (line (1)). The weighted predictive covariance and weighted measurement covariance are then calculated as given in (31) and (32), respectively (lines (2) and (3)). Coefficient α belongs to the interval $[0, 1]$ and can be determined such that the trace or determinant of the updated total covariance is minimized. Detection and elimination of outliers are important for preventing the divergence of state estimates. This requirement is fulfilled by employing an ellipsoidal measurement validating gate [40] (line (4)). As the pose measurements from neighbours are in the reference coordinate frame, the measurement model of this sensor fusion becomes linear. Therefore, the linear Kalman filter can be exploited for sensor fusion. In this measurement update, the measurement matrix \mathbf{H} of the traditional Kalman filter becomes an identity matrix, (\mathbf{I}_{n_x}), of $n_x \times n_x$. Using the multiplicative property of the identity matrix (i.e., $\mathbf{I}_m \mathbf{A} = \mathbf{A} \mathbf{I}_n = \mathbf{A}$, where \mathbf{A} is $m \times n$) the Kalman gains, the updated robot pose, and the associated total and independent covariance matrices can be computed from (33), (34), (35), and (36), respectively (lines (5), (6), (7), and (8)). For outliers, measurements are discarded and the predictive pose and the associated total and independent covariance matrices are directly assigned to the corresponding posterior quantities (lines (9) and (10)).

4.4. Update Local Pose Estimation Using the Measurement Acquired by the Absolute Positioning System. It is assumed that some of the robots in the MRS host DGPS sensor in order to measure global position information. This position measurement at time $k + 1$ is modeled as

$$\mathbf{y}_{q_i,k+1}^A = \mathbf{H}_{q_i,k+1}^A \mathbf{x}_{q_i,k+1} + \mathbf{v}_{K+1}^A, \quad (38)$$

where $\mathbf{H}_{q_i,k+1}^A = [\mathbf{I}_2 \quad \mathbf{0}_{2 \times 1}]$ is the measurement vector and \mathbf{v}_{K+1}^A is the additive white Gaussian noise term with covariance

$\mathbf{R}^A \in \mathbb{R}^{2 \times 2}$. This measurement is linear and independent from the robot's pose estimate. Thus, this measurement can be fused with the current state estimation using the general linear Kalman filter measurement update steps followed by

$$\begin{aligned} \mathbf{P}_{q_i,(k+1)_+} = & (\mathbf{I} - \mathbf{K} \mathbf{H}_{q_i,k+1}^A) \mathbf{P}_{q_i,(k+1)_-} (\mathbf{I} - \mathbf{K} \mathbf{H}_{q_i,k+1}^A)^T \\ & + \mathbf{K} \mathbf{R}^A \mathbf{K}^T. \end{aligned} \quad (39)$$

This equation computes the updated independent covariance matrix at the event of DGPS measurement update.

4.5. Sensor Fusion Architecture. This study assumes that each agent in the MRS initially knows its pose with respect to a given reference coordinate frame. The recursive state estimation framework of the proposed DCL algorithm is outlined in Algorithm 4 and is graphically illustrated in Figure 1.

Algorithm 4 (Split-CI based cooperative localization algorithm).

- (1) Initialize known $\mathbf{x}_{q_i^o}$ and $\mathbf{P}_{q_i^o}$
- (2) Set initial independent covariance: $\mathbf{P}_{q_i^o} \leftarrow \mathbf{P}$.
- (3) **for** $k \in (1, \dots, \infty)$ **do**
- (4) Read ego-motion sensor $\bar{\mathbf{u}}_{q_i,k}$
- (5) Propagate state: Algorithm 1
- (6) **if** $|\mathcal{Y}_{q_i,k+1}^q| > 0$ **then**
- (7) **for** $\forall r \in \mathbf{N}_{q_i,k+1}$ **do**
- (8) Read $\mathbf{y}_{q_i,k+1}^{r,q}$
- (9) Transform relative pose measurement to reference coordinate frame: Algorithm 2
- (10) Transmit ($\mathbf{y}_{q_i,k+1}^{r,*}$, $\mathbf{R}_{q_i,k+1}^{r,*}$, $\mathbf{R}_{q_d}^{r,*}$)
- (11) **end for**
- (12) **end if**
- (13) **if** pose measurement receives from neighbours **then**
- (14) **for** $\forall r \in \bar{\mathbf{N}}_{q_i,k+1}$ **do**
- (15) Collect ($\mathbf{y}_{r,k+1}^{q_i,*}$, $\mathbf{R}_{r,k+1}^{q_i,*}$, $\mathbf{R}_{r_d}^{q_i,*}$) from \mathcal{R}_r
- (16) Perform Split-CI-based measurement update: Algorithm 3
- (17) Enable recursive update

$$\begin{aligned} \widehat{\mathbf{x}}_{q_i,(k+1)_-} & \leftarrow \widehat{\mathbf{x}}_{q_i,(k+1)_+} \\ \mathbf{P}_{q_i,(k+1)_-} & \leftarrow \mathbf{P}_{q_i,(k+1)_+} \\ \mathbf{P}_{q_i,(k+1)_-} & \leftarrow \mathbf{P}_{q_i,(k+1)_+} \end{aligned} \quad (40)$$

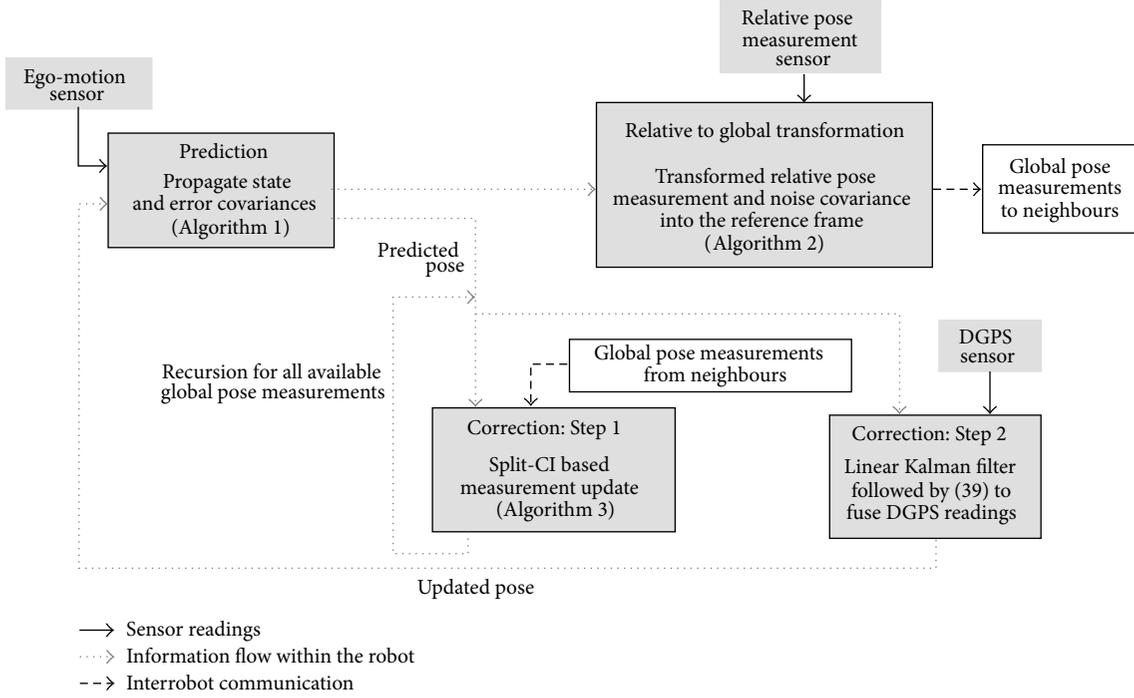


FIGURE 1: Sensor fusion architecture of the proposed decentralized multirobot cooperative localization scheme.

- (18) **end for**
- (19) Set independent covariance to zero: $\mathbf{P}_{q_i,(k+1)_-} \leftarrow [\mathbf{0}_{3 \times 3}]$
- (20) **end if**
- (21) **if** DGPS measurement available **then**
- (22) Read $\mathbf{y}_{q,k+1}^A$
- (23) **if** measurement gate validated **then**
- (24) Compute $\hat{\mathbf{x}}_{q_i,(k+1)_+}$, $\mathbf{P}_{q_i,(k+1)_+}$, and $\mathbf{P}_{q_i,(k+1)_+}$ as detailed in Section 4.4
- (25) **else**
- (26) Assign predictive quantities to corresponding posterior quantities
- $$\begin{aligned} \hat{\mathbf{x}}_{q_i,(k+1)_+} &\leftarrow \hat{\mathbf{x}}_{q_i,(k+1)_-} \\ \mathbf{P}_{q_i,(k+1)_+} &\leftarrow \mathbf{P}_{q_i,(k+1)_-} \\ \mathbf{P}_{q_i,(k+1)_+} &\leftarrow \mathbf{P}_{q_i,(k+1)_-} \end{aligned} \quad (41)$$
- (27) **end if**
- (28) **else**
- (29) Assign predictive quantities to corresponding posterior quantities: (41)
- (30) **end if**
- (31) **end for**

$\bar{\mathbf{N}}_{q,k+1}$ is the set containing unique identification indices of robots that communicate global pose measurements to \mathcal{R}_q at time $k + 1$.

The proposed DCL algorithm has four main steps.

Step 1 (propagate state (lines (4)–(5) in Algorithm 4)). At each time step, the robot acquires its ego-motion sensor reading (odometry). This measurement is fused with the previous time step's posterior estimate in order to compute the predicted pose and the associated total and independent error covariance matrices as detailed in Algorithm 1.

Step 2 (measure neighbours' pose (lines (6)–(12) in Algorithm 4)). At an interrobot relative pose measurement event, first, the robot reads its exteroceptive sensors and collects the relative poses of its neighbours. Then, each relative pose measurement is transformed into the reference coordinate frame as outlined in Algorithm 2. Finally, the transformed global pose measurements and the associated independent and dependent covariance matrices are transmitted to the corresponding neighbouring robots.

Step 3 (update with pose measurements sent by neighbours (lines (13)–(20) in Algorithm 4)). At a given time step, a robot may receive pose measurements from one (or more) neighbour(s). First, the received pose measurement is fused with the local estimation using the Split-CI measurement update structure that is detailed in Algorithm 3. In order to enable the recursion for available pose measurements from multiple neighbours, the updated pose and associated total and independent covariances are assigned back to the corresponding predictive parameter (line (17)). The recursion

is then continued until all the received pose measurements are considered. Work presented in [41] provides a complete theoretical analysis and simulation-based validation for the consistency of the Split-CI-based filtering. However, the simulation study presented in [21] surfaced that the estimated states using the Split-CI based DCL algorithm sometimes diverge. This may occur because the resulting pose estimation might be correlated partially or fully to subsequent pose measurements received from neighbours. To overcome this issue, the proposed DCL algorithm directly assigns the known-independent covariance component to the correlated component (line (19)). In other words, this study set the independent covariance component to zero after interrobot measurement update event, which is not included in the standard Split-CIF algorithm described in [42].

Step 4 (update with absolute position measurement (lines (21)–(30) in Algorithm 4)). The final step of the proposed DCL algorithm is to update the robot’s local pose with the position measurement acquired from an absolute positioning system. When a new position measurement is available, it is evaluated through an ellipsoidal validation gate to identify whether the acquired measurement is a valid measurement or an outlier (line (23)). If it is a valid measurement then the measurement is fused with the local estimation (line (24)). Otherwise, the predictive quantities are directly assigned to the corresponding posterior quantities (line (26)). For the time steps where no absolute position measurements are available, the predictive quantities are directly assigned to the corresponding posterior quantities (line (29)).

5. Simulation Results

5.1. Setup. The performance of the proposed DCL algorithm was evaluated using a publicly available multirobot localization and mapping dataset [43]. This 2D indoor dataset was generated from five robots (which we refer to as \mathcal{R}_1 , \mathcal{R}_2 , \mathcal{R}_3 , \mathcal{R}_4 , and \mathcal{R}_5) that navigated in a 15 m \times 8 m indoor space. Although this dataset consists of odometry readings, ground truth measurements, and range and bearing measurements to neighbours and landmarks we only used the odometry readings and ground truth measurements of each robot in order to evaluate the proposed DCL algorithm. This simulation study assumed that all five robots would be equipped with lightweight sensory systems to uniquely identify and measure the relative pose of their neighbours. Further, it was assumed that only two members of the robot team (i.e., \mathcal{R}_1 and \mathcal{R}_2) would be capable of acquiring DGPS measurements periodically. Interrobot measurements and DGPS measurements were synthesized from the ground truth data. Simulation parameters and sensor characteristics related to this simulation setup are summarized in Tables 1 and 2, respectively.

5.2. Results. Figures 2 and 3 illustrate the mean estimation error and the associated $3\text{-}\sigma$ error boundaries of the proposed DCL algorithm for 20 Monte-Carlo runs. Figure 2 corresponds to a robot with absolute position measuring

TABLE 1: Simulation parameters.

Symbol	Parameter description	Value
$ N $	Number of robots in the team	5
t	Total time period of the dataset	1500 s
d_m	Maximum sensing range	10 m
$d \times l$	Size of the navigation arena	15 m \times 8 m
MC	Number of Monte-Carlo runs	20

TABLE 2: Sensor characteristics.

Sensor type	Measure	Update rate	Noise σ
Odometry	Linear velocity	50 Hz	$\sqrt{5.075}v_{x,q,k}$
	Angular velocity	50 Hz	$\sqrt{0.345} \text{ rads}^{-1}$
Relative pose	Relative x -position	10 Hz	0.1 m
	Relative y -position	10 Hz	0.1 m
	Relative orientation	10 Hz	1 deg
DGPS	Global x -position	10 Hz	0.1 m
	Global y -position	10 Hz	0.1 m

Noise parameters for velocities were extracted from [7].

capabilities and Figure 3 corresponds to a robot without absolute position measuring capabilities. From these results, it can be seen that the estimation errors of the proposed DCL algorithm are always inside the corresponding $3\text{-}\sigma$ error boundaries. This observation verifies that the proposed DCL algorithm is capable of avoiding the cyclic update and generating nonoverconfidence state estimations. Additionally, it is clear that the robots with absolute position measuring capabilities can achieve a more accurate pose estimation than the robots without such capabilities (note that the y -axis of Figures 2 and 3 is presented in two different scales). Further, the results confirm that the estimation error of the proposed DCL algorithm is bounded.

5.3. Comparison. The estimation accuracy of the proposed DCL algorithm is compared with the estimation accuracy that were obtained from the following localization schemes:

- (1) *Single-Robot Localization (SL) Method.* Each robot continually integrates its odometry readings in order to estimate its pose in a given coordinate frame. This method is also known as dead-reckoning. Robots with DGPS measuring capability fuse their DGPS sensor readings with the local estimate in order to improve pose estimation accuracy.
- (2) *DCL Using Naïve Block-Diagonal (NB) Method.* In this method, the pose measurements sent by neighbours are treated as an independent information and are fused directly with the local estimate. In other words, possible correlations between local estimate and pose measurements sent by neighbours are neglected at the sensor fusion step.
- (3) *DCL Using Ellipsoidal Intersection (EI) Algorithm.* The EI algorithm always assumes there exist unknown correlations between each robot’s local pose estimations and uses a set of explicit expressions to

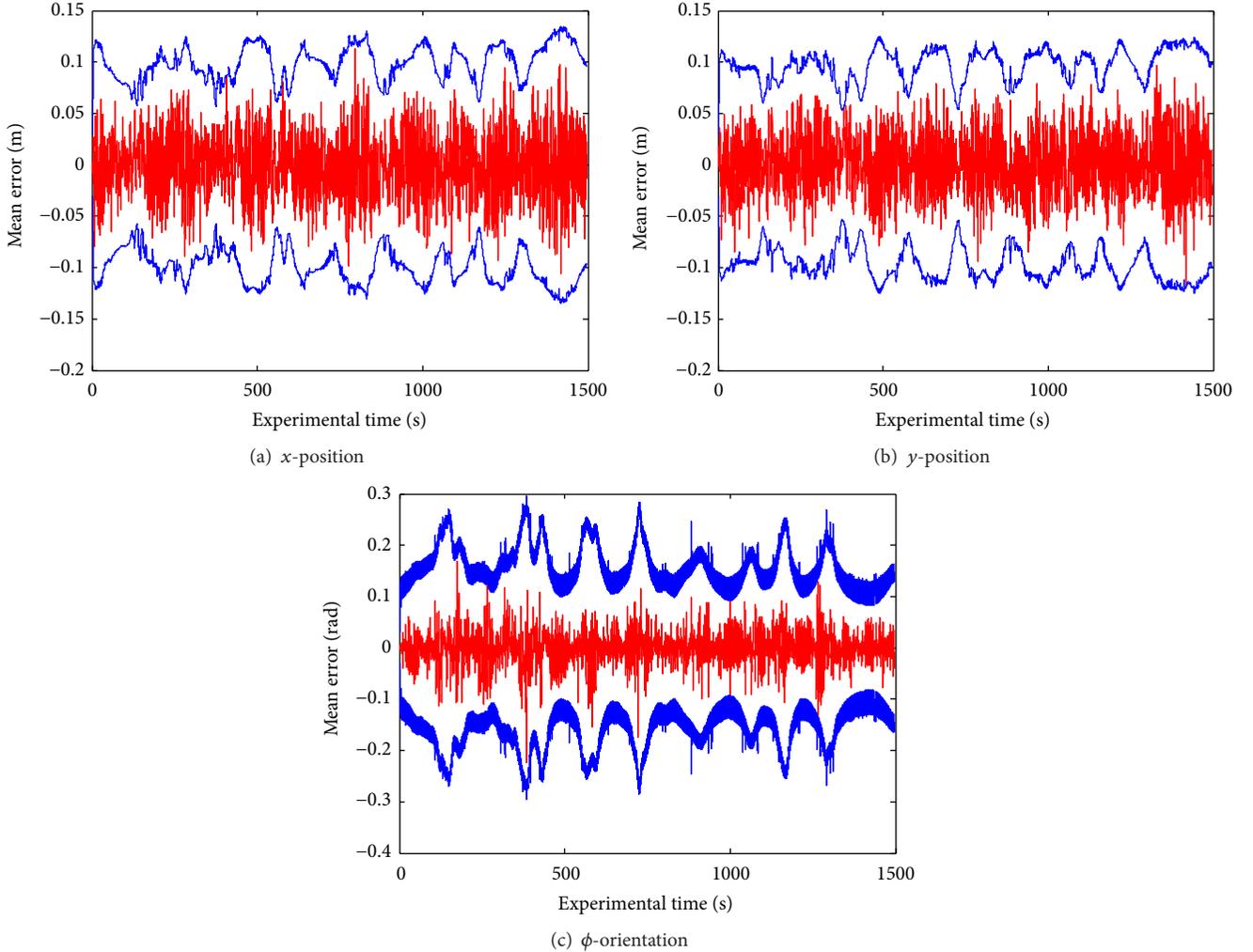


FIGURE 2: Mean estimation error of \mathcal{R}_1 for 20 Monte-Carlo simulations (a robot with absolute position measuring capabilities). In each graph, the solid red line indicates mean estimation error while the solid black lines indicate double-sided $3\text{-}\sigma$ error boundaries.

calculate these unknown correlations, that is, mutual-mean and mutual-covariance. When a robot receives a pose measurement(s) from its neighbour(s) EI algorithm first calculates these unknown correlations. In order to obtain the updated estimation the calculated mutual-mean and mutual-covariance are fused with the robot's local estimates and the pose measurements received from the robot's neighbours [44].

- (4) *DCL Using Covariance Intersection (CI) Algorithm.* Each robot runs a local estimator to estimate its pose using onboard sensors and the pose measurement from neighbours. When a robot receives pose measurements from its neighbours, the covariance intersection algorithm is used to fuse these pose measurements with the robot's local estimate [21].
- (5) *Centralized Cooperative Localization (CCL) Approach.* The pose of each robot is augmented into a single state vector. The ego-centric measurements of robots and interrobot observations are fused using EKF. This is a centralized approach which can accurately track

the correlations between robots' pose estimations. Therefore, the results of this approach will serve as the benchmark for the performance evaluation of the proposed DCL algorithm.

We performed 20 Monte-Carlo simulations per each localization algorithm. Then the RMSE of position and orientation estimation for 20 Monte-Carlo simulations were computed. Finally, the time averaged RMSE values and associated standard deviations were calculated to perform the comparison between different localization schemes.

Consider robots without DGPS measuring capabilities (i.e., \mathcal{R}_3 , \mathcal{R}_4 , and \mathcal{R}_5). The pose estimations of these robots entirely rely on the odometry readings and the interrobot observations. Therefore, the time averaged RMSE and the associated standard deviation values of the pose estimation of these robots provide insight into the performance of each localization algorithm. The time averaged RMSE and the associated standard deviation values of the localization of \mathcal{R}_5 using the single-robot localization scheme were found as

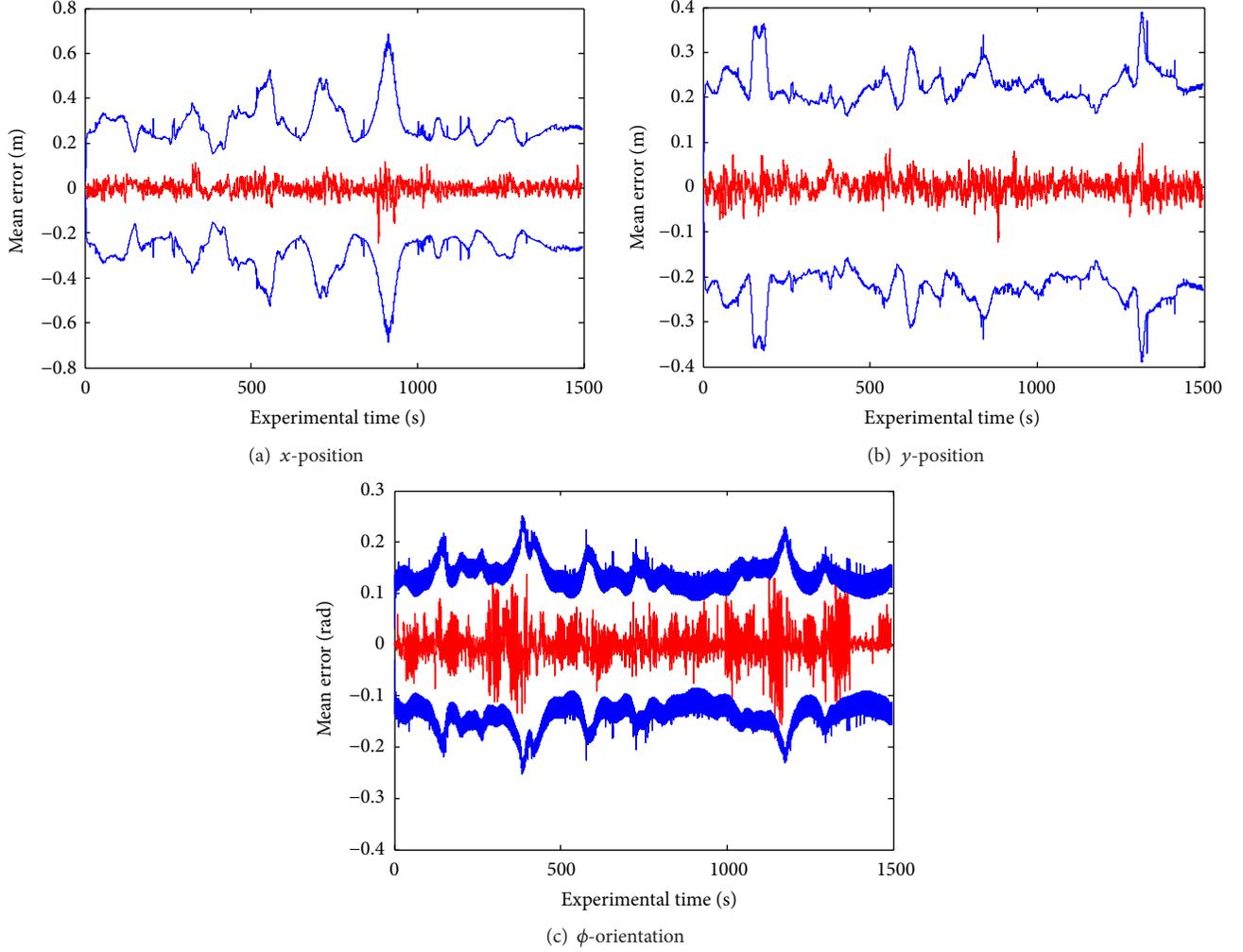


FIGURE 3: Mean estimation error of \mathcal{R}_5 for 20 Monte-Carlo simulations (a robot that does not have absolute position measuring capabilities). In each graph, the solid red line indicates mean estimation error while the solid black lines indicate double-sided 3- σ error boundaries.

3.1762 ± 2.3680 m in x -direction, 5.0073 ± 2.3339 m in y -direction, and 1.1776 ± 0.9015 rad in orientation estimation (the format of the listed estimation errors is mean \pm standard deviation). The time averaged RMSE of the localization of \mathcal{R}_5 using any of the cooperative localization algorithms (NB, EI, CI, Split-CI, and CCL) were less than 10 cm in both x - and y -directions and less than 0.1 rad in orientation estimation. These observations imply that the cooperative localization approaches can significantly improve the accuracy of pose estimation of agents in a MRS.

Time averaged RMSE and the associated standard deviation values of x -position, y -position, and ϕ -orientation estimates of \mathcal{R}_5 using different cooperative localization schemes are compared in Figure 4. This comparison shows that the centralized cooperative localization algorithm outperforms all the other approaches. This was the expected result, as the centralized estimator maintained the joint-state and the associated dense covariance matrix in order to accurately represent the correlation between teammates pose estimates.

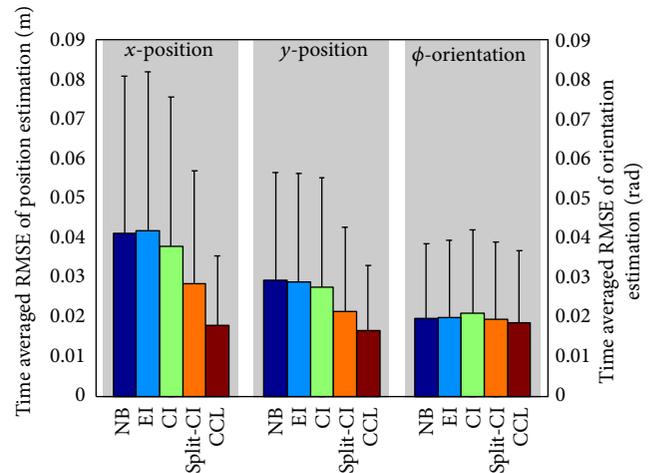


FIGURE 4: Comparison of estimation error of different cooperative localization algorithms. This result is for robot \mathcal{R}_5 (one of the robots without the DGPS measuring capabilities).

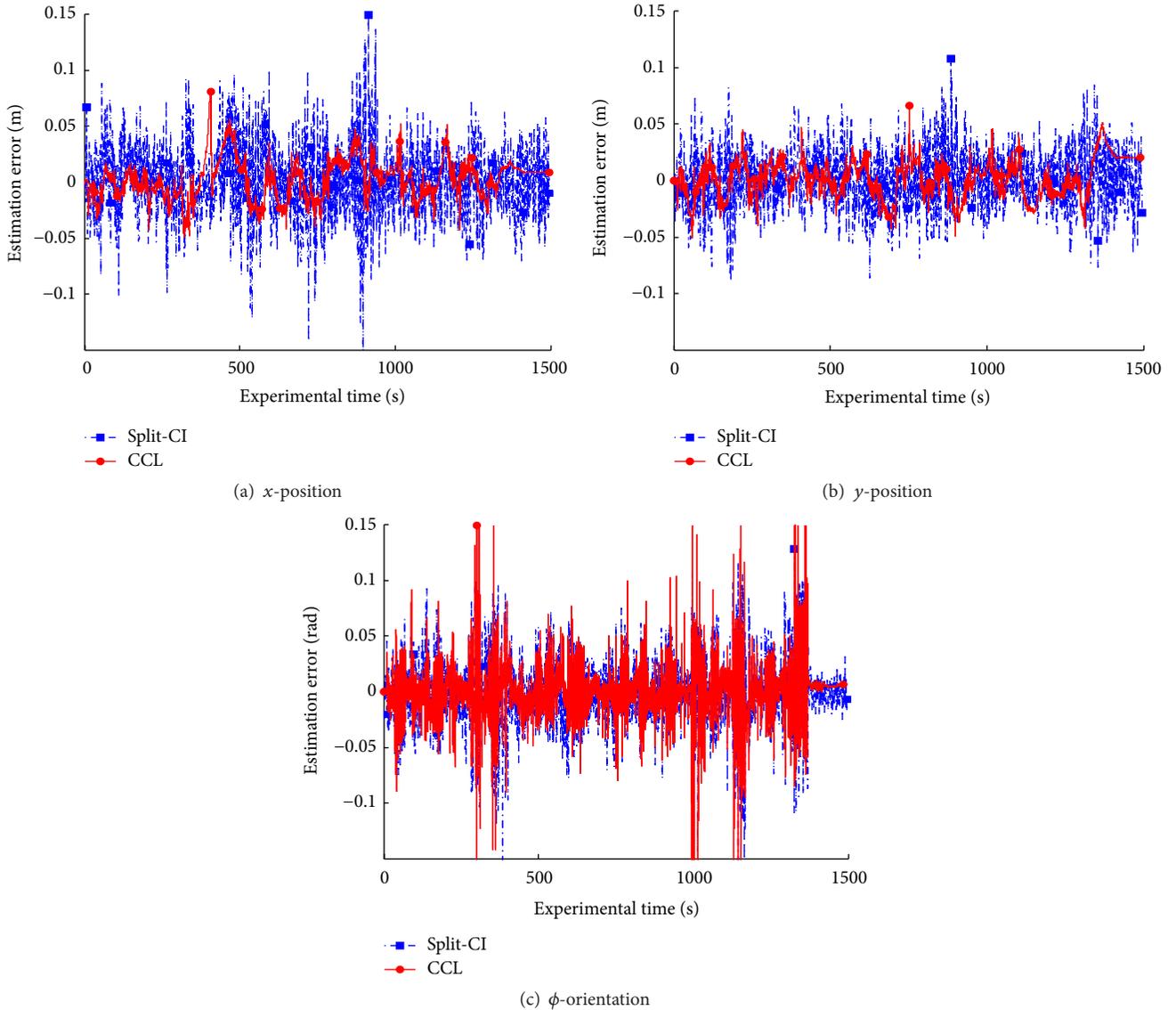


FIGURE 5: Estimation error comparison between the proposed Split-CI based approach and the centralized cooperative localization approach.

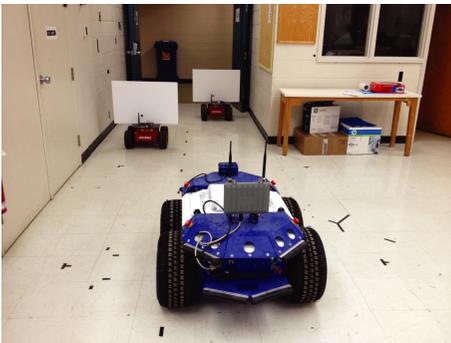


FIGURE 6: Experimental setup.

Although the estimated pose using the proposed Split-CI based DCL algorithm is less accurate than that of the centralized cooperative localization algorithm, it demonstrates

better accuracy over all other DCL approaches that we have evaluated in this paper.

Figure 5 illustrates the estimation error comparison between the proposed Split-CI based DCL algorithm and the centralized cooperative localization algorithm. It indicates that the centralized approach has better accuracy; however, the estimation accuracy obtained from the proposed DCL algorithm is comparable with the estimation accuracy obtained from the centralized approach.

6. Experimental Results

6.1. Setup. The proposed DCL algorithm was experimentally evaluated on a team of three robots (see Figure 6): one Seekur Jr. (we will refer to this as platform *A*) and two Pioneer robots (we will refer to these as platforms *B* and *C*). Each robot is equipped with wheel encoders for odometry.

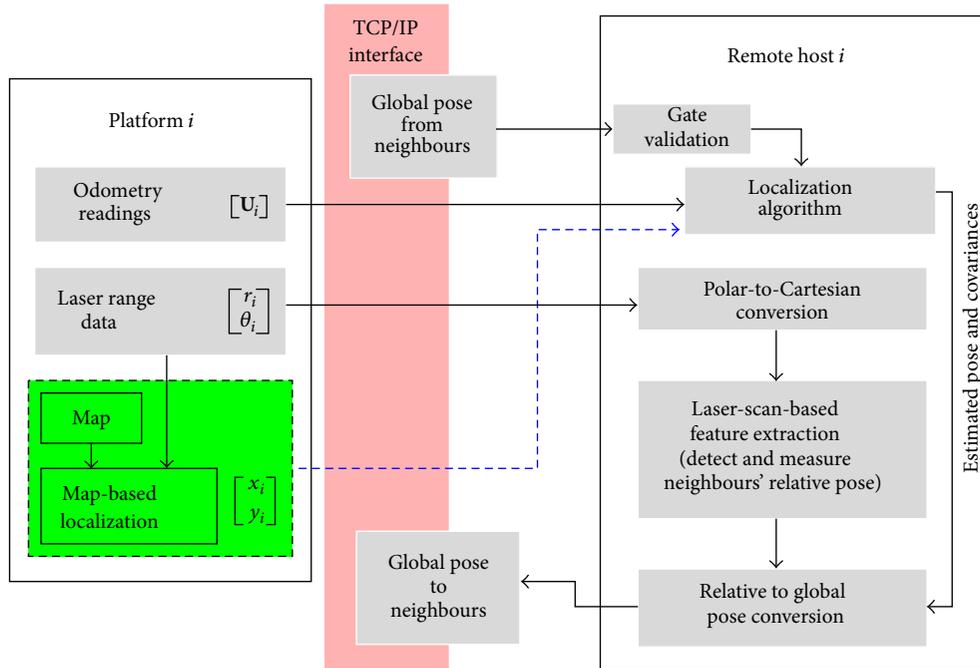


FIGURE 7: System architecture of the experiment setup to validate the proposed DCL scheme. Note that the map-based (scan-matching-based) localization information is available only for platform A.

Additionally, SICK laser scanners were attached to acquire range and bearing measurements for objects around the robot, periodically. Robots were navigated in indoor environment while maintaining triangular formation between them.

6.2. System Architecture. Figure 7 illustrates the system architecture of the experiment setup. Each robot acquires its odometry measurements and laser-scan readings periodically. The acquired measurements are transmitted to a host computer through a TCP/IP interface. Platform A was provided with the map of the navigation space and it performed scan-matching-based localization using this map. The position estimations of this scan-matching-based localization for platform A were considered as absolute position measurements for cooperative localization schemes and were transmitted to the host computer that executes the localization for platform A.

In the host computer, odometry readings were used for state propagation and the global pose measurements and the associated noise covariance from neighbours were used to correct the predicted pose. Note that the pose measurements from neighbours were first evaluated through ellipsoidal measurement validation gate in order to detect and discard outliers. Only platform A used scan-matching-based position calculation data at the sensor fusion. At each host processing unit, the received laser-scan data were first converted to the Cartesian coordinate frame from the polar coordinate system. This gives a set of points that represents the relative positions of objects around the corresponding robot. Laser-scan-based feature extraction algorithm was then exploited to detect and measure the relative pose of neighbouring robots.

The data correspondence problem was addressed through the nearest neighbour data association technique. These relative pose measurements were then converted to global (reference) coordinate frame and then were communicated to the corresponding host.

6.3. Results. Figure 8 illustrates the comparison of pose estimates for platform B that were obtained from three different sensor fusion approaches: the centralized cooperative localization method, the proposed Split-CI based DCL algorithm, and the single-robot localization (dead-reckoning) method. The estimates obtained from the centralized cooperative localization approach serve as the benchmark for evaluating the proposed DCL algorithm. On the other hand, the estimates obtained from the single-robot localization represent the worst case pose estimates for each time step. These results suggest that the proposed Split-CI based DCL algorithm and the centralized cooperative localization algorithm generate approximately the same pose estimates for platform B. Although the two estimates are not identical to one another, the difference between the two estimates did not exceed the double-sided $3\text{-}\sigma$ error boundary, that is, the gray color region of Figure 8, of the proposed DCL algorithm. Pose estimates generated from dead-reckoning diverged from the true state (or the state obtained from the centralized approach) with the increase of experimental time period.

Figure 9 illustrates the comparison of pose uncertainty for three different sensor fusion approaches: the centralized cooperative localization method, the proposed split-CI based DCL algorithm, and single-robot localization method. These results verify that the cooperative localization

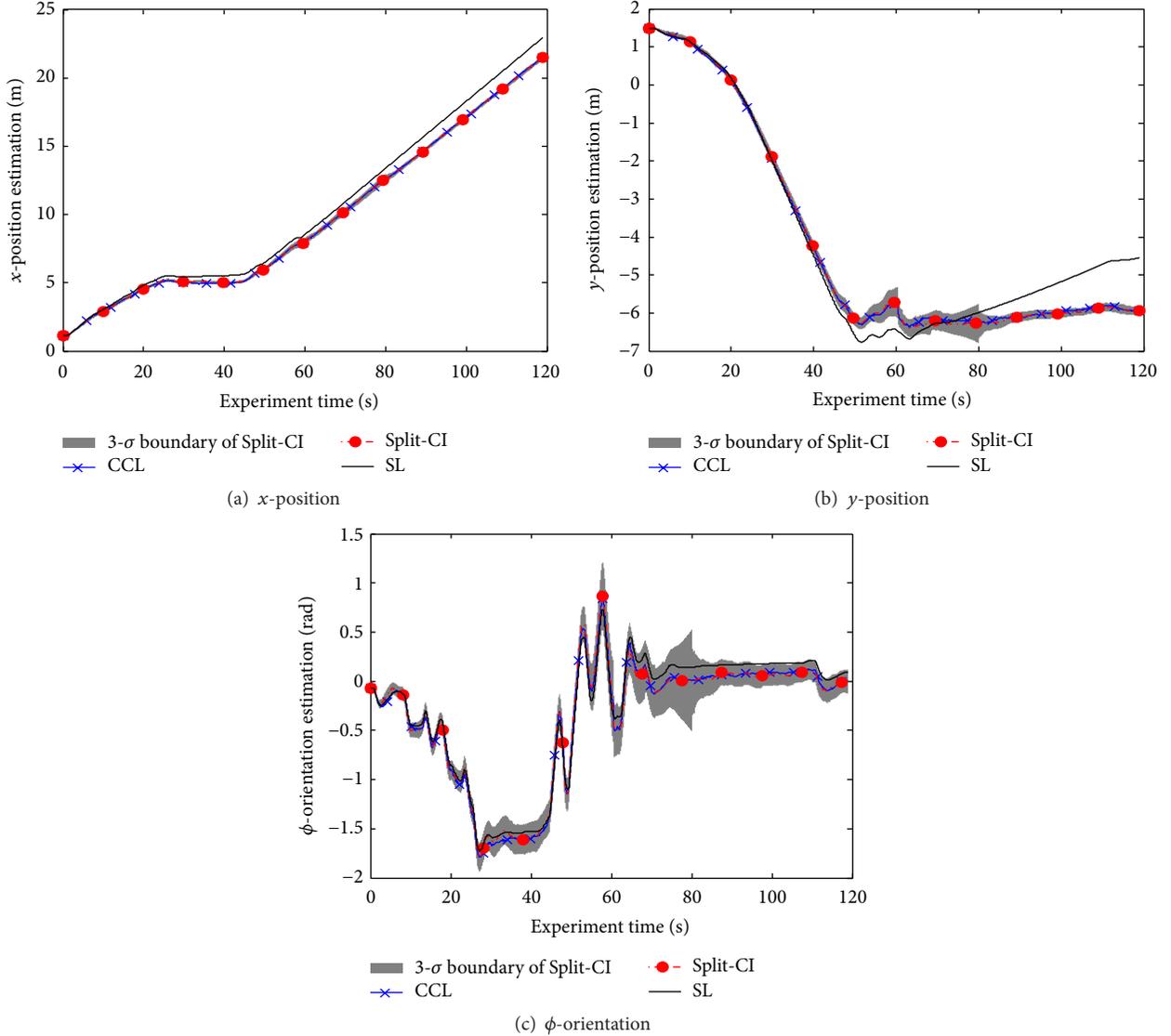


FIGURE 8: Pose estimation comparison of platform B.

approaches have bounded pose estimation uncertainty while the pose estimation uncertainty of the single-robot localization approach increases unboundedly. The lowest pose uncertainty is recorded in the centralized approach (see Figure 9(c)). The pose uncertainty found in the proposed Split-CI based DCL algorithm is slightly greater than that of the centralized approach. This is the expected result as the centralized estimator maintained the joint-state and associated dense covariance matrix in order to accurately represent the correlation between teammates' pose estimates.

7. Complexity

7.1. Computational Complexity. As the pose estimation of the proposed algorithm is decentralized, the computational complexity of the proposed DCL algorithm is increased linearly with the increase of number of neighbouring robots.

In other words, the computational complexity of the proposed DCL algorithm is $\mathcal{O}(|\bar{N}_q|)$, where $|\bar{N}_q|$ is the number of neighbours, per robot per time step. This remains true for all the DCL algorithms while the computational complexity scales $\mathcal{O}(|N|^4)$ for the centralized cooperative localization where $|N|$ is the number of robots in the MRS.

7.2. Communicative Complexity. The proposed DCL algorithm does not require robot to communicate on-board high frequency proprioceptive sensory data with one another or with central processing unit. Only the interrobot measurements are required to exchange between neighbouring robots. These two properties considerably reduce the bandwidth requirement for communication network between robots. In general, communicative complexity of the proposed algorithm remains $\mathcal{O}(|\bar{N}_q|)$ per robot, per interrobot observation event.

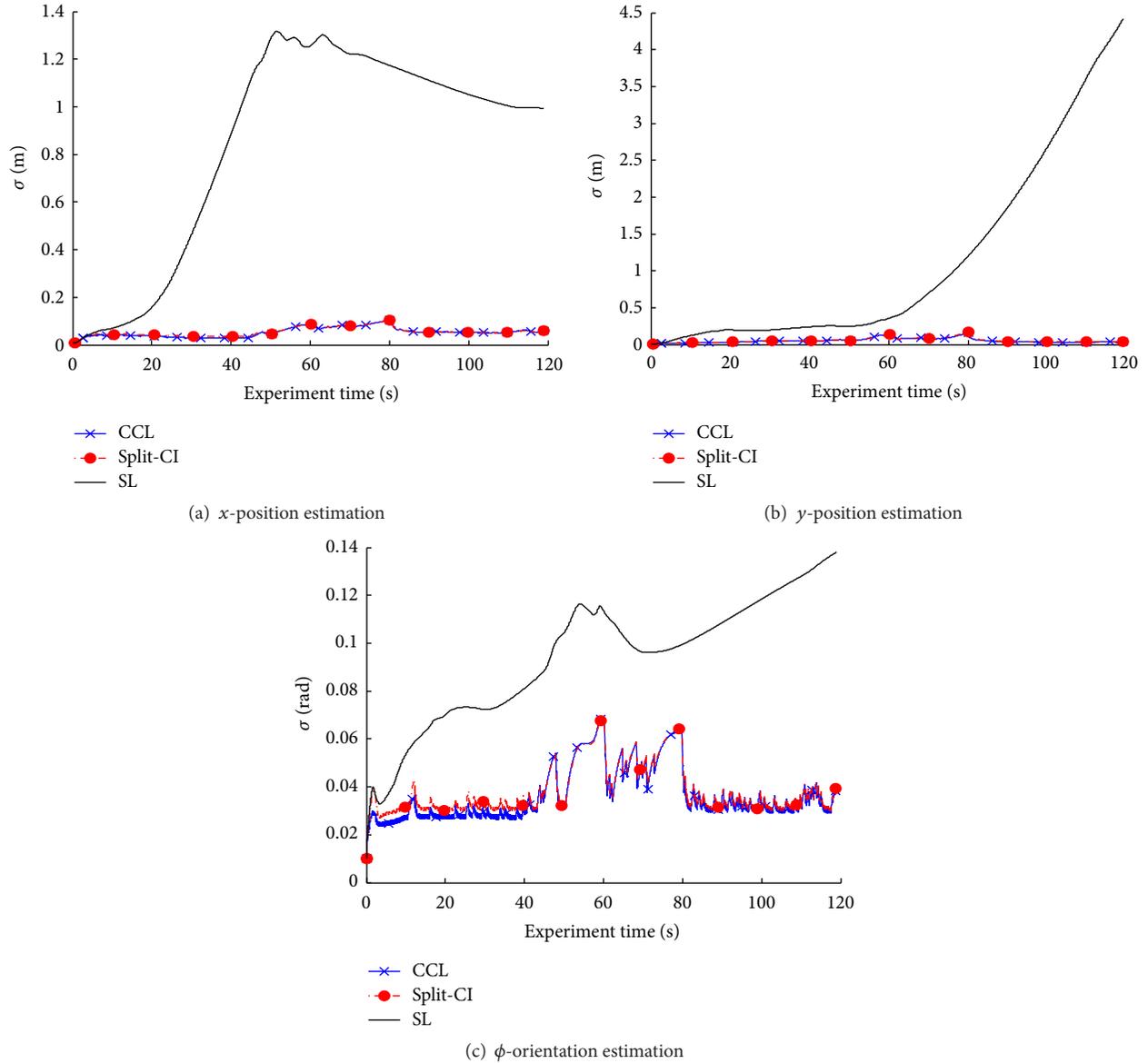


FIGURE 9: Comparison of pose estimation uncertainty (standard deviation) for platform B.

8. Conclusion

This study demonstrates the use of Split-CI algorithm and cubature Kalman filter for decentralized cooperative localization. Both the overall computational (processing) and communicative requirements of the proposed DCL algorithm remain $\mathcal{O}(|\bar{N}_q|)$ per robot per time step, where $|\bar{N}_q|$ is the number of neighbouring robots. This is a considerable reduction compared to state-of-the-art centralized cooperative localization approach in which computational cost scales $\mathcal{O}(|N|^4)$ and requires wider communication bandwidth to exchange high frequency ego-centric measurements between robots and/or central processing unit. Besides the reduced computational and communicative complexity, both the simulation and experiment results demonstrate that the estimation accuracy of the proposed method is comparable

with the centralized cooperative localization. Therefore, the proposed DCL algorithm is more suitable for implementing cooperative localization for a MRS with large number of robots. Additionally, the simulation and experimental results verified that the estimation errors of the proposed DCL scheme are bounded and are not overconfidence. This can be attributed to the modification we added at line (19) in Algorithm 4. The results verified that the cooperative localization approaches outperform the single-robot localization method. Therefore, interrobot observation and flow of information between robots will be the most appropriate approach when implementing localization approach for heterogeneous MRS.

The proposed method can be directly applied to inter-robot relative measurement systems that give either full relative pose or relative position of the neighbours. For the interrobot relative measurement system that measures

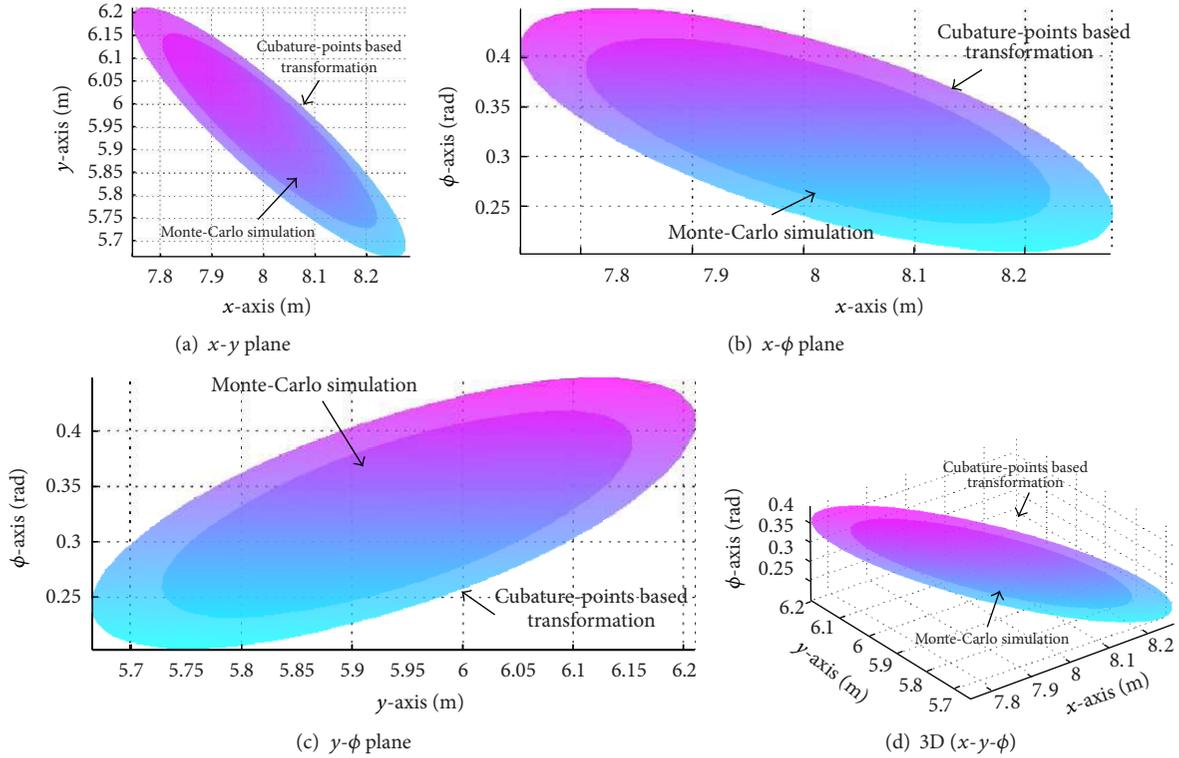


FIGURE 10: Comparison of estimated covariance matrices.

range and bearing of the neighbours, the polar-to-Cartesian conversion can be applied prior to evaluating these measurements on the proposed DCL algorithm. The key limitation of the proposed DCL algorithm is that the proposed method cannot be directly implemented using relative range-only or relative bearing-only measurement systems. However, this limitation can be addressed by implementing a hierarchical filtering approach. In this hierarchical filtering approach, each robot runs a tracking filter per neighbour to track neighbours' relative pose. These tracks are periodically converted into the global coordinate frame and communicated to the corresponding neighbour. Once a robot receives pose measurement from a neighbour, the proposed DCL algorithm can be exploited for sensor fusion. Ongoing work attempts to implement this hierarchical filtering approach and evaluate its performance.

Appendix

Consistent and Debiased Method for Cartesian-to-Cartesian Conversion

Converting a relative pose measurement to a global pose measurement can be defined as a converting of uncertain information from one Cartesian coordinate frame to another Cartesian coordinate frame.

Assume \mathbf{x} is a random variable with mean $\bar{\mathbf{x}}$ and covariance \mathbf{P}_x . Additionally, assume there is another random

variable \mathbf{y} which relates to \mathbf{x} as follows:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}), \quad (\text{A.1})$$

where $\mathbf{f}(\cdot)$ represents nonlinear function. If the objective is to calculate the mean $\bar{\mathbf{y}}$ and covariance \mathbf{P}_y of \mathbf{y} , given $\bar{\mathbf{x}}$, \mathbf{P}_x , and $\mathbf{f}(\cdot)$, the transformed statistics are said to be consistent if the inequality

$$\mathbf{P}_y - E[\{\mathbf{y} - \bar{\mathbf{y}}\}\{\mathbf{y} - \bar{\mathbf{y}}\}^T] \geq \mathbf{0} \quad (\text{A.2})$$

holds [45]. Work presented in this study applies a cubature-point-based approach to perform Cartesian-to-Cartesian coordinate transformation (see Section 4.2 for more details). Here, we present a simulation study to verify that the Cartesian-to-Cartesian conversion algorithm presented in Algorithm 2 holds above inequality.

Consider a robot team with two robots \mathcal{R}_1 , and \mathcal{R}_2 . The global pose of \mathcal{R}_1 and \mathcal{R}_2 is $[5 \ 3 \ 0.6981]^T$ and $[8 \ 6 \ 0.3491]^T$, respectively (the format of the pose vector is $[x \ y \ \phi]^T$, where x and y coordinates are given in m while the orientation ϕ is given in rad). The objective is to find the global pose of \mathcal{R}_2 given the global pose of \mathcal{R}_1 , relative pose of \mathcal{R}_2 with respect to \mathcal{R}_1 , and the associated uncertainties. We compared the statistics we obtained from Algorithm 2 with those calculated by a Monte-Carlo simulation which used 10000 samples. Table 3 and Figure 10 illustrate the comparison of the statistics calculated from two methods. It can be seen that the mean values obtained from the proposed algorithm are approximately overlapped with those

TABLE 3: Comparison of mean global pose.

	True pose	Mean from Monte-Carlo simulation with 10000 samples	Mean from cubature-points based transformation
x (m)	8	8.0127	8.0112
y (m)	6	5.9410	5.9375
ϕ (rad)	0.3491	0.3240	0.3250

calculated by a Monte-Carlo simulation which used 10000 samples. Therefore, the conversion is unbiased. Further, it can be seen that the covariance ellipses of the cubature-point-based approach are always larger than that of the Monte-Carlo simulation. This implies that the proposed Cartesian-to-Cartesian transformation holds the inequality given in (A.2). Additionally, principal axis of covariance ellipse of the proposed approach is approximately overlapped with those of the Monte-Carlo localization. Therefore the proposed coordinate transformation algorithm is consistent.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors would like to thank the Natural Science and Engineering Research Council of Canada (NSERC) and Memorial University of Newfoundland for funding this research project.

References

- [1] M. K. Habib and Y. Baudoin, "Robot-assisted risky intervention, search, rescue and environmental surveillance," *International Journal of Advanced Robotic Systems*, vol. 7, no. 1, pp. 1–8, 2010.
- [2] N. Michael, S. Shen, K. Mohta et al., "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012.
- [3] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '94)*, vol. 2, pp. 1250–1257, May 1994.
- [4] R. Kurazume, S. Hirose, S. Nagata, and N. Sashida, "Study on cooperative positioning system (basic principle and measurement experiment)," in *Proceedings of the 13th IEEE International Conference on Robotics and Automation (ICRA '96)*, vol. 2, pp. 1421–1426, Minneapolis, Minn, USA, April 1996.
- [5] S. I. Roumeliotis and G. A. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.
- [6] T. R. Wanasinghe, G. K. I. Mann, and R. G. Gosine, "Decentralized cooperative localization for heterogeneous multi-robot system using split covariance intersection filter," in *Proceedings of the Canadian Conference on Computer and Robot Vision (CRV '14)*, pp. 167–174, IEEE, Montreal, Canada, May 2014.
- [7] K. Y. K. Leung, *Cooperative localization and mapping in sparsely communicating robot networks [Ph.D. thesis]*, Department of Aerospace Science and Engineering, University of Toronto, 2012.
- [8] I. Rekleitis, G. Dudek, and E. Millios, "Probabilistic cooperative localization and mapping in practice," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '03)*, vol. 2, pp. 1907–1912, Taipei, Taiwan, September 2003.
- [9] I. Rekleitis, G. Dudek, and E. Miliotis, "Experiments in free-space triangulation using cooperative localization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03)*, vol. 2, pp. 1777–1782, IEEE, October 2003.
- [10] N. Trawny and T. Barfoot, "Optimized motion strategies for cooperative localization of mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*, vol. 1, pp. 1027–1032, May 2004.
- [11] S. Tully, G. Kantor, and H. Choset, "Leap-frog path design for multi-robot cooperative localization," in *Field and Service Robotics*, A. Howard, K. Iagnemma, and A. Kelly, Eds., vol. 62 of *Springer Tracts in Advanced Robotics*, pp. 307–317, Springer, Berlin, Germany, 2010.
- [12] E. D. Nerurkar and S. I. Roumeliotis, "Asynchronous multi-centralized cooperative localization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 4352–4359, Taipei, Taiwan, October 2010.
- [13] R. Sharma and C. Taylor, "Cooperative navigation of MAVs in GPS denied areas," in *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI '08)*, pp. 481–486, Seoul, Republic of Korea, August 2008.
- [14] K. Y. K. Leung, T. D. Barfoot, and H. H. T. Liu, "Decentralized localization of sparsely-communicating robot networks: a centralized-equivalent approach," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 62–77, 2010.
- [15] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli, "Distributed maximum a posteriori estimation for multi-robot cooperative localization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 1402–1409, Kobe, Japan, May 2009.
- [16] H. Durrant-Whyte, M. Stevens, and E. Nettleton, "Data fusion in decentralised sensing networks," in *Proceedings of the 4th International Conference on Information Fusion*, pp. 302–307, Montreal, Canada, August 2001.
- [17] A. Howard, M. J. Mataric, and G. Sukhatme, "Putting the 'i' in 'team': an ego-centric approach to cooperative localization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 868–874, Taipei, Taiwan, September 2003.
- [18] S. Panzneri, F. Pascucci, and R. Setola, "Multirobot localisation using interlaced extended Kalman filter," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 2816–2821, IEEE, Beijing, China, October 2006.
- [19] A. Bahr, M. R. Walter, and J. J. Leonard, "Consistent cooperative localization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 3415–3422, Kobe, Japan, May 2009.
- [20] T. Bailey, M. Bryson, H. Mu, J. Vial, L. McCalman, and H. Durrant-Whyte, "Decentralised cooperative localisation for heterogeneous teams of mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 2859–2865, IEEE, Shanghai, China, May 2011.

- [21] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis, "Decentralized multi-robot cooperative localization using covariance intersection," in *Proceedings of the 26th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, pp. 1412–1417, IEEE, Tokyo, Japan, November 2013.
- [22] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "Probabilistic approach to collaborative multi-robot localization," *Autonomous Robots*, vol. 8, no. 3, pp. 325–344, 2000.
- [23] N. E. Özkücur, B. Kurt, and H. L. Akın, "A collaborative multi-robot localization method without robot identification," in *RoboCup 2008: Robot Soccer World Cup XII*, vol. 5399 of *Lecture Notes in Computer Science*, pp. 189–199, Springer, Berlin, Germany, 2009.
- [24] A. Prorok and A. Martinoli, "A reciprocal sampling algorithm for lightweight distributed multi-robot localization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '11)*, pp. 3241–3247, IEEE, San Francisco, Calif, USA, September 2011.
- [25] A. Prorok, A. Bahr, and A. Martinoli, "Low-cost collaborative localization for large-scale multi-robot systems," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '12)*, pp. 4236–4241, St. Paul, Minn, USA, May 2012.
- [26] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*, YBS Publishing, 2011.
- [27] D. Kurth, G. Kantor, and S. Singh, "Experimental results in rangeonly localization with radio," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 974–979, October 2003.
- [28] E. Olson, J. J. Leonard, and S. Teller, "Robust range-only beacon localization," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 949–958, 2006.
- [29] R. Sharma, S. Quebe, R. W. Beard, and C. N. Taylor, "Bearing-only cooperative localization," *Journal of Intelligent & Robotic Systems*, vol. 72, no. 3-4, pp. 429–440, 2013.
- [30] K. E. Bekris, M. Glick, and L. E. Kavraki, "Evaluation of algorithms for bearing-only SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06)*, pp. 1937–1943, IEEE, Orlando, Fla, USA, May 2006.
- [31] O. De Silva, G. Mann, and R. Gosine, "Development of a relative localization scheme for ground-aerial multi-robot systems," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '12)*, pp. 870–875, Vilamoura-Algarve, Portugal, October 2012.
- [32] T. R. Wanasinghe, G. K. I. Mann, and R. G. Gosine, "Pseudo-linear measurement approach for heterogeneous multi-robot relative localization," in *Proceedings of the 16th International Conference on Advanced Robotics (ICAR '13)*, pp. 1–6, IEEE, Montevideo, Uruguay, November 2013.
- [33] M. W. Mehrez, G. K. I. Mann, and R. G. Gosine, "Nonlinear moving horizon state estimation for multi-robot relative localization," in *Proceedings of the IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE '14)*, pp. 1–5, Toronto, Canada, May 2014.
- [34] S. I. Roumeliotis and G. A. Bekey, "Collective localization: a distributed Kalman filter approach to localization of groups of mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, vol. 3, pp. 2958–2965, April 2000.
- [35] I. M. Rekleitis, G. Dudek, and E. E. Milios, "Multi-robot cooperative localization: a study of trade-offs between efficiency and accuracy," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems ((IROS '02)*, vol. 3, pp. 2690–2695, IEEE, October 2002.
- [36] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.
- [37] K. P. B. Chandra, D.-W. Gu, and I. Postlethwaite, "Cubature Kalman filter based localization and mapping," in *Proceedings of the 18th International Federation of Automatic Control (IFAC '11) World Congress*, pp. 2121–2125, Milano, Italy, September 2011.
- [38] Y. Song, Q. Li, Y. Kang, and Y. Song, "CFastSLAM: a new Jacobian free solution to SLAM problem," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '12)*, pp. 3063–3068, St. Paul, Minn, USA, May 2012.
- [39] I. Arasaratnam, S. Haykin, and T. R. Hurd, "Cubature Kalman filtering for continuous-discrete systems: theory and simulations," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 4977–4993, 2010.
- [40] Y. Kosuge and T. Matsuzaki, "The optimum gate shape and threshold for target tracking," in *Proceedings of the SICE Annual Conference*, vol. 2, pp. 2152–2157, Fukui, Japan, August 2003.
- [41] H. Li, F. Nashashibi, and M. Yang, "Split covariance intersection filter: theory and its application to vehicle localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1860–1871, 2013.
- [42] S. Julier and J. K. Uhlmann, "General decentralized data fusion with covariance intersection (CI)," in *Handbook of Data Fusion*, chapter 12, CRC Press, Boca Raton, Fla, USA, 2001.
- [43] K. Y. K. Leung, Y. Halpern, T. D. Barfoot, and H. H. T. Liu, "The UTIAS multi-robot cooperative localization and mapping dataset," *International Journal of Robotics Research*, vol. 30, no. 8, pp. 969–974, 2011.
- [44] J. Sijs, M. Lazar, and P. P. J. V. D. Bosch, "State fusion with unknown correlation: ellipsoidal intersection," in *Proceedings of the American Control Conference (ACC '10)*, pp. 3992–3997, IEEE, Baltimore, Md, USA, June-July 2010.
- [45] S. J. Julier and J. K. Uhlmann, "Consistent debiased method for converting between polar and cartesian coordinate systems," in *Acquisition, Tracking, and Pointing XI*, vol. 3086 of *Proceedings of SPIE*, pp. 110–121, International Society for Optics and Photonics, Orlando, Fla, USA, June 1997.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

