Research Article

Decentralized State Estimation Algorithm of Centralized Equivalent Precision for Formation Flying Spacecrafts Based on Junction Tree

Mengyuan Dai,¹ Hua Mu,¹ Meiping Wu,¹ and Zhiwen Xian^{1,2}

¹College of Mechatronics Engineering and Automation, National University of Defense Technology, No. 109, Deya Street, Changsha, Hunan 410073, China

²The Australian Centre for Field Robotics, University of Sydney, Rose Street Building J04, Sydney, NSW 2006, Australia

Correspondence should be addressed to Hua Mu; muhua_nudt@hotmail.com

Received 29 April 2015; Revised 28 July 2015; Accepted 9 August 2015

Academic Editor: Paolo Tortora

Copyright © 2015 Mengyuan Dai 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.

As centralized state estimation algorithms for formation flying spacecraft would suffer from high computational burdens when the scale of the formation increases, it is necessary to develop decentralized algorithms. To the state of the art, most decentralized algorithms for formation flying are derived from centralized EKF by simplification and decoupling, rendering suboptimal estimations. In this paper, typical decentralized state estimation algorithms are reviewed, and a new scheme for decentralized algorithms is proposed. In the new solution, the system is modeled as a dynamic Bayesian network (DBN). A probabilistic graphical method named junction tree (JT) is used to analyze the hidden distributed structure of the DBNs. Inference on JT is a decentralized form of centralized Bayesian estimation (BE), which is a modularized three-step procedure of receiving messages, collecting evidences, and generating messages. As KF is a special case of BE, the new solution based on JT is equivalent in precision to centralized KF in theory. A cooperative navigation example of a three-satellite formation is used to test the decentralized algorithms. Simulation results indicate that JT has the best precision among all current decentralized algorithms.

1. Introduction

Formation flying has been recognized as an enabling technology for many future space missions [1], for example, the synthetic aperture radar (SAR) [2], and the stellar interferometry [3]. Compared with conventional space missions based on a single spacecraft, formation flying has several advantages: lower price, robustness after spacecraft failures, modularity, and good flexibility for reconfigurations [4]. The concept of formation flying is driven by the idea that a fleet of small spacecrafts could form a virtual spacecraft of unlimited scale which may serve as a substitution of a monolithic spacecraft [4]. As the size of the formation becomes larger, the guidance, navigation, and control (GNC) task becomes more and more onerous due to the increased size of state and measurement data [5]. Thus it becomes a necessity to develop decentralized algorithms to balance the computational loads among spacecrafts.

This paper takes the navigation task as an example to study the decentralized state estimation algorithms. There are generally two kinds of sensors used for formation navigation: the proprioceptive sensors and the exteroceptive sensors [6]. The first kind of sensors is used for single platform measurements to estimate the state of local spacecraft, for example, GPS receivers [4, 5, 7]. The second kind of sensors is used for interformation measurements to estimate the states of local spacecraft and its neighbors, for example, optical devices [8, 9] and radio frequency (RF) devices [10, 11]. With interformation measurements and communications, the navigation resources at different spacecrafts may be shared among the formation. This kind of navigation is also known as cooperative navigation [12]. As a typical navigation application for formation flying spacecrafts, this paper studies formation flying missions using RF range augmented GPS sensors, which are introduced by Park [4] and Ferguson and How [5]. Though the sensors are specially appointed in this

paper, the technique used in designing decentralized algorithms may inspire other cooperative navigation missions.

To the state of the art, almost all the decentralized estimation algorithms for formation flying are based on distributing centralized EKF through simplifying the measurement models or decoupling the interformation measurements through approximations. In fact, as interformation measurements make the states of spacecrafts strongly coupled, the simplification and decoupling will result in suboptimal estimations, which means that the precision of these algorithms is inferior to EKF. From the perspective of probabilistic graphical models in artificial intelligence, Lauritzen introduced a method for modeling and local computing of exact mean and covariance in dynamic systems based on JT [13]. The most remarkable advantage of the JT algorithm to other decentralized algorithms is that the JT algorithm has equivalent precision to centralized EKF [14]. Thus it is possible to develop a new decentralized cooperative navigation algorithm for formation flying spacecrafts.

This paper is arranged in 5 sections. In Section 2, the model for cooperative navigation is introduced. A brief introduction of existing decentralized solutions is provided in Section 3. And Section 4 introduces the new decentralized solution as a series of local computations and messages passing on JT. Simulation results in Section 5 show that the new solution has centralized equivalent precision which is better than the decentralized algorithms reviewed.

2. Model of Cooperative Navigation for Formation Flying

This paper takes the formation flying mission based on RF range augmented GPS as an example. The GPS pseudorange is used for position estimate at local spacecraft, while the RF range devices are used for relative motion measurements between spacecrafts. As the dynamic model and the measurement model are both nonlinear, some partial derivatives are needed for linearization. The partials used in the numerical simulation of this research are listed in Appendix A.

2.1. Dynamic Model. The state of each spacecraft is defined as the Keplerian orbital elements. Let $\mathbf{x}_k^i$ be the Keplerian orbital elements of satellite i at time step k , $\mathbf{x} = (a, e, i', \Omega, \omega, M)^T$, where a is the semimajor axis, e is the eccentricity, i' is the inclination angle, Ω is the right ascension of ascending node, ω is the argument of perigee, and M is the mean anomaly at reference time. Without considering system inputs such as orbit control, the dynamic model of satellite i is given by

$$\mathbf{x}_{k+1}^i = f(\mathbf{x}_k^i) + \mathbf{w}_k^i, \quad (1)$$

where $\mathbf{w}_k^i$ is the noise and $\mathbf{w}_k^i \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k^i)$, $\mathcal{N}(\mathbf{0}, \mathbf{Q}_k^i)$ is a Gaussian distribution of mean $\mathbf{0}$ and variance $\mathbf{Q}_k^i$.

Linearize (1) at $\hat{\mathbf{x}}_k^i$ using first order Taylor expansion, and we have

$$\mathbf{x}_{k+1}^i = f(\hat{\mathbf{x}}_k^i) + \mathbf{F}_k^i \cdot (\mathbf{x}_k^i - \hat{\mathbf{x}}_k^i) + \mathbf{w}_k^i, \quad (2)$$

where $\mathbf{F}_k^i$ is the state transition matrix and

$$\mathbf{F}_k^i = \left. \frac{\partial f}{\partial \mathbf{x}_k^i} \right|_{\mathbf{x}_k^i = \hat{\mathbf{x}}_k^i}. \quad (3)$$

2.2. Measurement Model. There are two kinds of measurements in the formation flying example: single platform measurement based on GPS pseudorange and interformation measurement based on RF range.

2.2.1. Single Platform Measurement. For simplicity, the time delay for signal transmission and errors such as hardware delay, ionosphere delay, multipath, relativistic effect, and clock error are not considered. Let $\mathbf{r}_{k+1}^s$ be the position of observed GPS satellite given by the broadcast ephemeris, and let $\mathbf{r}_{k+1}^i$ be the position of spacecraft i at time step $k + 1$. The single platform measurement based on GPS pseudorange, z_{k+1}^i , is given by

$$z_{k+1}^i = |\mathbf{r}_{k+1}^i - \mathbf{r}_{k+1}^s| + v_{k+1}^i, \quad (4)$$

where $|\mathbf{r}|$ means the length of vector $\mathbf{r}$ and v_{k+1}^i is the measurement noise, $v_{k+1}^i \sim \mathcal{N}(0, R_{k+1}^i)$.

In general, the single platform measurement function may be written as

$$z_{k+1}^i = h_1(\mathbf{x}_{k+1}^i) + v_{k+1}^i. \quad (5)$$

Linearize (5) at $\hat{\mathbf{x}}_{k+1}^i$ using first order Taylor expansion, and we have

$$z_{k+1}^i = h_1(\hat{\mathbf{x}}_{k+1}^i) + \mathbf{H}_{1,k+1}^i \cdot (\mathbf{x}_{k+1}^i - \hat{\mathbf{x}}_{k+1}^i) + v_{k+1}^i, \quad (6)$$

where $\mathbf{H}_{1,k+1}^i$ is the measurement matrix and

$$\mathbf{H}_{1,k+1}^i = \left. \frac{\partial h_1}{\partial \mathbf{x}_{k+1}^i} \right|_{\mathbf{x}_{k+1}^i = \hat{\mathbf{x}}_{k+1}^i}. \quad (7)$$

2.2.2. Interformation Measurement. Similar to single platform measurement, let $\mathbf{r}_{k+1}^i$ and $\mathbf{r}_{k+1}^j$ be the position of spacecraft i and j at time step $k + 1$, respectively. The interformation measurement between spacecrafts i and j , z_{k+1}^{ij} , is given by

$$z_{k+1}^{ij} = |\mathbf{r}_{k+1}^i - \mathbf{r}_{k+1}^j| + v_{k+1}^{ij}, \quad (8)$$

where v_{k+1}^{ij} is the measurement noise, $v_{k+1}^{ij} \sim \mathcal{N}(0, R_{k+1}^{ij})$.

In general, the single platform measurement function may be written as

$$z_{k+1}^{ij} = h_2(\mathbf{x}_{k+1}^i, \mathbf{x}_{k+1}^j) + v_{k+1}^{ij}. \quad (9)$$

Linearize (9) at $\hat{\mathbf{x}}_{k+1}^i$ and $\hat{\mathbf{x}}_{k+1}^j$ using first order Taylor expansion, and we have

$$\begin{aligned} z_{k+1}^{ij} = & h_2(\hat{\mathbf{x}}_{k+1}^i, \hat{\mathbf{x}}_{k+1}^j) + \mathbf{H}_{2,k+1}^i \cdot (\mathbf{x}_{k+1}^i - \hat{\mathbf{x}}_{k+1}^i) \\ & + \mathbf{H}_{2,k+1}^j \cdot (\mathbf{x}_{k+1}^j - \hat{\mathbf{x}}_{k+1}^j) + v_{k+1}^{ij}, \end{aligned} \quad (10)$$

where $\mathbf{H}_{2,k+1}^i$ and $\mathbf{H}_{2,k+1}^j$ are the measurement matrix and

$$\begin{aligned} \mathbf{H}_{2,k+1}^i &= \left. \frac{\partial h_2}{\partial \mathbf{x}_{k+1}^i} \right|_{\mathbf{x}_{k+1}^i = \hat{\mathbf{x}}_{k+1}^i}, \\ \mathbf{H}_{2,k+1}^j &= \left. \frac{\partial h_2}{\partial \mathbf{x}_{k+1}^j} \right|_{\mathbf{x}_{k+1}^j = \hat{\mathbf{x}}_{k+1}^j}. \end{aligned} \quad (11)$$

3. Review of Decentralized Algorithms

To the best knowledge of the authors, most current decentralized solutions are based on centralized EKF. Generally, there are two types of decentralized algorithms for formation flying spacecraft: the full order extended Kalman filter (FOEKF) and the reduced order extended Kalman filter (ROEKF) [5]. In FOEKF, each spacecraft runs a full order estimator which estimates the full states of the system, but only based on local measurements related to the very spacecraft. In ROEKF, each spacecraft runs a reduced order estimator which only processes the state of itself. The reduction of the order of the filter is based on decoupling the measurement models. As FOEKF and ROEKF all require simplifications and approximations of the EKF model, they all suffer from suboptimal precision which is inferior to centralized EKF. To compensate the precision loss, a direct method is to use iterations, which leads to the iterated cascade extended Kalman filter (ICEKF) [5].

3.1. Centralized EKF. Consider a formation of n spacecrafts. Let $\mathbf{X}_k$ be the state of the whole formation at time step k , $\mathbf{X}_k = [(\mathbf{x}_k^1)^T (\mathbf{x}_k^2)^T \cdots (\mathbf{x}_k^n)^T]^T$. Without considering the system input, the linearized model of cooperative navigation could be rewritten as follows.

Dynamic Model.

$$\mathbf{X}_{k+1} = \Phi_k \mathbf{X}_k + \mathbf{W}_k, \quad \mathbf{W}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k). \quad (12)$$

Measurement Model.

$$\mathbf{Z}_{k+1} = \mathbf{H}_{k+1} \mathbf{X}_{k+1} + \mathbf{V}_{k+1}, \quad \mathbf{V}_{k+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{k+1}), \quad (13)$$

where Φ is the state transition matrix, $\mathbf{W}$ is the noise of the dynamic model, $\mathbf{Z}$ is the vector of measurements, $\mathbf{H}$ is the measurement matrix, and $\mathbf{V}$ is the noise of measurements. The procedure of EKF could be described as a sequence of time update and measurement update.

Time Update.

$$\hat{\mathbf{X}}_{k+1}^- = \Phi_k \hat{\mathbf{X}}_k^+, \quad (14)$$

$$\mathbf{P}_{k+1}^- = \Phi_k \mathbf{P}_k^+ \Phi_k^T + \mathbf{Q}_k. \quad (15)$$

Measurement Update.

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1})^{-1}, \quad (16)$$

$$\hat{\mathbf{X}}_{k+1}^+ = \hat{\mathbf{X}}_{k+1}^- + \mathbf{K}_{k+1} (\mathbf{Z}_{k+1} - \mathbf{H}_{k+1} \hat{\mathbf{X}}_{k+1}^-), \quad (17)$$

$$\mathbf{P}_{k+1}^+ = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \mathbf{P}_{k+1}^-, \quad (18)$$

where mean $\hat{\mathbf{X}}_k^+$ and covariance $\mathbf{P}_k^+$ are the best estimates of system state at time step k .

3.2. FOEKF. Considering a local FOEKF estimator on spacecraft i , it only processes the measurements related to spacecraft i , which means that the measurement $\mathbf{Z}_{k+1}^i$, measurement matrix $\mathbf{H}_{k+1}^i$, and measurement noise $\mathbf{V}_{k+1}^i$ are subsets of $\mathbf{Z}_{k+1}$, $\mathbf{H}_{k+1}$, and $\mathbf{V}_{k+1}$ in EKF, respectively. The measurement update of FOEKF at spacecraft i is as follows:

$$\begin{aligned} \mathbf{K}_{k+1} &= \mathbf{P}_{k+1}^- (\mathbf{H}_{k+1}^i)^T [\mathbf{H}_{k+1}^i \mathbf{P}_{k+1}^- (\mathbf{H}_{k+1}^i)^T + \mathbf{R}_{k+1}^i]^{-1}, \\ \hat{\mathbf{X}}_{k+1}^+ &= \hat{\mathbf{X}}_{k+1}^- + \mathbf{K}_{k+1} (\mathbf{Z}_{k+1}^i - \mathbf{H}_{k+1}^i \hat{\mathbf{X}}_{k+1}^-), \\ \mathbf{P}_{k+1}^+ &= (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}^i) \mathbf{P}_{k+1}^-. \end{aligned} \quad (19)$$

Given that the RF range and Doppler measurements are both nonlinear (the linearization requires the states of other spacecrafts), spacecrafts doing relative measurements must communicate with each other exchanging information of mean and covariance of its own state. Because only part of the measurements is considered at each spacecraft, FOEKF is suboptimal. Meanwhile, as FOEKF runs a full order estimator at each spacecraft, the computational load may not be well balanced.

3.3. ROEKF

3.3.1. Original ROEKF. ROEKF is a further simplification of FOEKF. Taking spacecraft i for example, suppose the states of other spacecrafts are known, and the measurement model between spacecrafts i and j could be simplified as

$$\begin{aligned} z_{k+1}^{ij} &= \mathbf{H}_{2,k+1}^i \mathbf{x}_{k+1}^i + \mathbf{H}_{2,k+1}^j \mathbf{x}_{k+1}^j + v_{k+1}^{ij}, \\ v_{k+1}^{ij} &\sim \mathcal{N}(0, R_{k+1}^{ij}). \end{aligned} \quad (20)$$

Notice that $\mathbf{x}_{k+1}^j$ is known. Let $\bar{z}_{k+1}^{ij} = z_{k+1}^{ij} - \mathbf{H}_{2,k+1}^j \mathbf{x}_{k+1}^j$; the ROEKF measurement model could be rewrite as

$$\bar{\mathbf{Z}}_{k+1}^i = \bar{\mathbf{H}}_{k+1}^i \mathbf{x}_{k+1}^i + \mathbf{V}_{k+1}^i, \quad \mathbf{V}_{k+1}^i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{k+1}^i), \quad (21)$$

where $\bar{\mathbf{Z}}_{k+1}^i$ is the set of all the measurements related with spacecraft i and $\bar{\mathbf{H}}_{k+1}^i$ is the measurement matrix related with $\bar{\mathbf{Z}}_{k+1}^i$.

Thus the local estimator at spacecraft i only needs to estimate its own state, $\hat{\mathbf{x}}_k^i$ and $\mathbf{P}_k^i$. The measurement update of FOEKF at spacecraft i is as follows:

$$\begin{aligned} \mathbf{K}_{k+1} &= \mathbf{P}_{k+1}^{i-} \left(\bar{\mathbf{H}}_{k+1}^i \right)^T \left[\bar{\mathbf{H}}_{k+1}^i \mathbf{P}_{k+1}^{i-} \left(\bar{\mathbf{H}}_{k+1}^i \right)^T + \mathbf{R}_{k+1}^i \right]^{-1}, \\ \hat{\mathbf{x}}_{k+1}^{i+} &= \hat{\mathbf{x}}_{k+1}^{i-} + \mathbf{K}_{k+1} \left(\bar{\mathbf{z}}_{k+1}^i - \bar{\mathbf{H}}_{k+1}^i \hat{\mathbf{x}}_{k+1}^{i-} \right), \\ \mathbf{P}_{k+1}^{i+} &= \left(\mathbf{I} - \mathbf{K}_{k+1} \bar{\mathbf{H}}_{k+1}^i \right) \mathbf{P}_{k+1}^{i-}. \end{aligned} \quad (22)$$

Compared with FOEKF, the computational load is reduced remarkably in ROEKF, because ROEKF only updates the state of local spacecraft. However, as only estimates instead of real states of other spacecrafts could be obtained, the errors of state estimation of other spacecrafts are not considered in ROEKF, rendering a further precision loss than FOEKF. We could use a proper increase to the measurement noise, the $\mathbf{R}$ matrix, to absorb such errors. This leads to the increased $\mathbf{R}$ extended Kalman filter (IREKF) [5].

3.3.2. IREKF. Considering the measurement z_{k+1}^{ij} , the measurement noise caused by using estimates of the state of spacecraft j could be estimated by the following equation:

$$\begin{aligned} E \left[\left(z_{k+1}^{ij} - \mathbf{H}_{2,k+1}^i \mathbf{x}_{k+1}^j \right) \left(z_{k+1}^{ij} - \mathbf{H}_{2,k+1}^i \mathbf{x}_{k+1}^j \right)^T \right] \\ = E \left[\left(\mathbf{H}_{2,k+1}^j \mathbf{x}_{k+1}^j + v_{k+1}^{ij} \right) \left(\mathbf{H}_{2,k+1}^j \mathbf{x}_{k+1}^j + v_{k+1}^{ij} \right)^T \right] \\ = E \left[\mathbf{H}_{2,k+1}^j \mathbf{x}_{k+1}^j \left(\mathbf{H}_{2,k+1}^j \mathbf{x}_{k+1}^j \right)^T \right. \\ \left. + \mathbf{H}_{2,k+1}^j \mathbf{x}_{k+1}^j \left(v_{k+1}^{ij} \right)^T + v_{k+1}^{ij} \left(\mathbf{H}_{2,k+1}^j \mathbf{x}_{k+1}^j \right)^T \right. \\ \left. + v_{k+1}^{ij} \left(v_{k+1}^{ij} \right)^T \right] = E \left[\mathbf{H}_{2,k+1}^j \mathbf{x}_{k+1}^j \left(\mathbf{H}_{2,k+1}^j \mathbf{x}_{k+1}^j \right)^T \right] \\ + E \left[v_{k+1}^{ij} \left(v_{k+1}^{ij} \right)^T \right] = \mathbf{H}_{2,k+1}^j \mathbf{P}_{k+1}^{j+} \left(\mathbf{H}_{2,k+1}^j \right)^T \\ + R_{k+1}^{ij}. \end{aligned} \quad (23)$$

Notice that only estimates of the states of other spacecrafts could be obtained, which means that we could only use $\mathbf{P}_{k+1}^{j-}$ to approximate $\mathbf{P}_{k+1}^{j+}$. Thus the increase to the measurement noise of R_{ij} is given by $\mathbf{H}_{2,k+1}^j \mathbf{P}_{k+1}^{j+} \left(\mathbf{H}_{2,k+1}^j \right)^T$.

3.4. ICEKF. Iteration is a straight forward method to compensate the precision loss caused by approximations. ICEKF could serve as a general scheme to improve FOEKF and ROEKF. In ICEKF, the local estimation at each spacecraft is passed to other spacecrafts for the linearization of measurement functions through a cascade communication chain. Figure 1 gives an example of ICEKF for a three-spacecraft formation compared with EKF. Although ICEKF reduces the computational load while obtaining a better precision than

FIGURE 1: ICEKF compared with centralized EKF [15].

FOEKF or ROEKF, the introduction of iteration will result in an increase in the communicational load.

4. JT Based Decentralized Solution

The decentralized solutions reviewed in the previous section are all based on distributing centralized EKF through simplifications or decoupling, rendering suboptimal estimations with inferior precisions to EKF. This section proposes a general scheme for the design of decentralized algorithms from another point of view: analyzing the hidden decentralized structure of EKF with the aid of probabilistic graphical models.

From the perspective of BE, the dynamic model and the measurement model used in EKF are all conditional Gaussian (CG), which could be interpreted by the probabilistic graphical model of DBN. Previous researches have demonstrated that inference on DBN could be distributed by message passing on a decentralized data structure named JT [16]. By properly splitting and implementing the time update and measurement update of BE on different cliques of JT, it is possible to design a decentralized solution which is centralized-equivalent.

The algorithm of JT contains two parts: symbolic calculation and mathematical calculation. Figure 2 gives the flowchart of JT. Symbolic calculation returns the decentralized structure of JT based on CG distributions defined by the model of cooperative navigation, while mathematical calculation returns the result of inference through message passing on JT. The key elements of the JT algorithm are introduced in the following part of this section.

FIGURE 2: Flowchart of JT.

4.1. Inference Engine. BE is the inference engine on JT. Let $\mathbf{Z}_{1:k}$ be the history measurements from initial time step to current time step k , and let $p(\mathbf{X}_k | \mathbf{Z}_{1:k})$ be the best estimation of current state under history measurements. If the conditional probability between new state and current state, $p(\mathbf{X}_{k+1} | \mathbf{X}_k)$, and the conditional probability between new measurement and new state, $p(\mathbf{Z}_{k+1} | \mathbf{X}_{k+1})$, are given, we could use BE to estimate the most likely state of the next time step. The procedure of BE is as follows.

Time Update.

$$p(\mathbf{X}_{k+1} | \mathbf{Z}_{1:k}) = \int p(\mathbf{X}_{k+1} | \mathbf{X}_k) \cdot p(\mathbf{X}_k | \mathbf{Z}_{1:k}) d\mathbf{X}_k. \quad (24)$$

Measurement Update.

$$p(\mathbf{X}_{k+1} | \mathbf{Z}_{1:k+1}) = \frac{p(\mathbf{X}_{k+1} | \mathbf{Z}_{1:k}) \cdot p(\mathbf{Z}_{k+1} | \mathbf{X}_{k+1})}{\int p(\mathbf{X}_{k+1} | \mathbf{Z}_{1:k}) \cdot p(\mathbf{Z}_{k+1} | \mathbf{X}_{k+1}) d\mathbf{X}_{k+1}}. \quad (25)$$

There are three basic mathematic operations in BE: multiplication, division, and marginalization. Interested readers may refer to Appendix B for details. Notice that $p(\mathbf{X}_k | \mathbf{Z}_{1:k}) = \mathcal{N}(\hat{\mathbf{X}}_k^+, \mathbf{P}_k^+)$; from (12) we have $p(\mathbf{X}_{k+1} | \mathbf{X}_k) = \mathcal{N}(\Phi_k \mathbf{X}_k, \mathbf{Q}_k)$, and from (13) we have $p(\mathbf{Z}_{k+1} | \mathbf{X}_{k+1}) = \mathcal{N}(\mathbf{H}_{k+1} \mathbf{X}_{k+1}, \mathbf{R}_{k+1})$. It can be verified that, by applying the basic operations on CG potentials to BE, BE returns the same result as EKF. The proof is omitted here for simplicity.

4.2. DBN. In the probabilistic graph theory, DBN is used to visualize the relationship of conditional distribution between random variables. Since the dynamic model and the measurement model of EKF each define a CG distribution, the time update and the measurement update of EKF (or BE) may be represented by DBN.

Figure 3 gives the DBN model of the example in Section 3.4, where the random variable $\mathbf{x}_k^i$ ($i = 1, 2, 3$) represents the state of each spacecraft. Notice that measurements are hidden behind the solid lines in DBN, while dynamic models are represented by dashed lines. Let U be the full set of all the random variables in DBN. For the example in Figure 3, $U = \{\mathbf{x}_k^1, \mathbf{x}_k^2, \mathbf{x}_k^3, \mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2, \mathbf{x}_{k+1}^3\}$.

DBN holds the joint distribution of all the variables in U . Inference on DBN means to find the joint distribution of interested variables, which is $p(\mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2, \mathbf{x}_{k+1}^3 | \mathbf{Z}_{1:k+1})$ for this example. To perform the inference, we need to remove the barren variables in DBN, which are $\mathbf{x}_k^1$, $\mathbf{x}_k^2$, and $\mathbf{x}_k^3$. This procedure is often referred to as barren node removal in the probabilistic graphical model.

4.3. Decentralize DBN Based on JT. JT is first introduced by Lauritzen [17] while studying the inference algorithm of the Bayesian network. JT is a tree structure of cliques. A clique C is a subset of the full set U . Each clique is assigned with a potential function, which is a probabilistic distribution of the variables in C . The set of variables shared by two adjacent cliques C_i and C_j is called a separator, denoted by $S_{ij} = C_i \cap C_j$. Measurements are treated as evidences deployed to different cliques. To generate JT from DBN, the sequence of barren node removal must be defined.

FIGURE 3: Example of DBN.

Moreover, an important structural constraint named the running intersection property [16] must be satisfied.

If a variable is contained in cliques C_i and C_j , then it must also be contained in the (unique) path between C_i and C_j .

The Kruskal algorithm and the Prim algorithm are the most common algorithms to generate JT. Interested readers may refer to [18] for details. When a JT is constructed, it must be initialized with potentials and evidences. Here we use two criteria for the initialization of JT.

Criterion 1. The probabilistic distribution defined by a set of random variables is assigned as a potential to the first clique which contains the set. If a clique is assigned with more than one potential, the multiplication of these potentials is treated as the potential for the clique; if there is no distribution assigned to a clique, the potential of this clique is initialized as standard normal.

Criterion 2. A measurement is deployed as an evidence to the first clique which contains all the variables related to the very measurement.

Figure 4 gives a JT based on the DBN model in the previous subsection, where the shaded ellipsoids represent the cliques, the hollow ellipsoids represent the evidences, and the dashed lines indicate the allocation of cliques to spacecrafts.

4.4. Inference on JT. Inference on JT is driven by a series of local computations on cliques and messages passing between adjacent cliques [19]. As the message passing is directional, an arbitrary clique is picked up as the root of JT. The message passing starts from leaves to root, and then it rolls back from root to leaves. If all cliques have received messages from both its parents and children, the inference at this time step is finished. A remarkable advantage of JT is its good modularity, rendering flexibility for reconfigurations and robustness after spacecraft failures. The operations at each clique are in a uniformed manor of three steps: receiving message, collecting evidence, and generating message.

4.4.1. Receiving Message. A message to a clique is defined as a distribution (or potential) passed from an adjacent clique.

If clique C_i have received a message from clique C_j , the potential of C_i is updated by the following equation through multiplication:

$$p^*(C_i) = p(C_i) \cdot p(\text{Msg}), \quad (26)$$

where $p(C_j)$ is the original potential of clique C_i , $p(\text{Msg})$ is the message, and $p^*(C_i)$ is the new potential assigned to C_i .

In fact, the operation of receiving message is a distributed implementation of the time update in centralized BE (or EKF).

4.4.2. Collecting Evidence. Take clique C_i for example. All evidences deployed to C_i are in the form of $p(Z | C_i)$. To collect the evidence of Z means to use the measurement of Z to update the estimate of local state, which is the distribution of state conditioned on Z . The potential of C_i after collecting evidence is updated by

$$p^*(C_i) = \frac{p(C_i) \cdot p(Z | C_i)}{\int (C_i) \cdot p(Z | C_i) dC_i}. \quad (27)$$

The operation of collecting evidence may be seen as a distributed implementation of the measurement update in centralized BE (or EKF).

4.4.3. Generating Message. After updating the potential of a clique based on messages and evidences, the next operation is to generate the message passed to the next clique. The message generated is based on the separator between adjacent cliques. Considering clique C_i generating a message to clique C_j , let $\bar{S}_{ij}^i$ be the complementary set of the separator S_{ij} in set C_i . The message from C_i to C_j is defined as the distribution of S_{ij} , $p(S_{ij})$. To get $p(S_{ij})$, we need to remove the variables in $\bar{S}_{ij}^i$ from C_i through marginalization. The message is given by

$$p(\text{Msg}) = \int p(C_i) d\bar{S}_{ij}^i. \quad (28)$$

Example 1. Consider the example of JT in Figure 4. Select clique C_3 as the root of JT. The operations at each spacecraft return the following results.

(i) Spacecraft 1:

Receiving Message.

$$\begin{aligned} p(\mathbf{x}_k^1, \mathbf{x}_k^2, \mathbf{x}_k^3, \mathbf{x}_{k+1}^1 | \mathbf{Z}_{1:k}) \\ = p(\mathbf{x}_{k+1}^1 | \mathbf{x}_k^1) \cdot p(\mathbf{x}_k^1, \mathbf{x}_k^3, \mathbf{x}_k^2 | \mathbf{Z}_{1:k}). \end{aligned} \quad (29)$$

Generating Message.

$$\begin{aligned} p(\mathbf{x}_k^2, \mathbf{x}_k^3, \mathbf{x}_{k+1}^1 | \mathbf{Z}_{1:k}) \\ = \int p(\mathbf{x}_k^1, \mathbf{x}_k^2, \mathbf{x}_k^3, \mathbf{x}_{k+1}^1 | \mathbf{Z}_{1:k}) d\mathbf{x}_k^1. \end{aligned} \quad (30)$$

FIGURE 4: Example of JT.

(ii) Spacecraft 2:

Receiving Message.

$$p(\mathbf{x}_k^2, \mathbf{x}_k^3, \mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2 | \mathbf{Z}_{1:k})$$

$$= p(\mathbf{x}_{k+1}^2 | \mathbf{x}_k^2) \cdot p(\mathbf{x}_k^2, \mathbf{x}_k^3, \mathbf{x}_{k+1}^1 | \mathbf{Z}_{1:k}). \quad (31)$$

Collecting Evidence.

$$p(\mathbf{x}_k^2, \mathbf{x}_k^3, \mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2 | \mathbf{Z}_{1:k}, z_{k+1}^{12})$$

$$= \frac{p(\mathbf{x}_k^2, \mathbf{x}_k^3, \mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2 | \mathbf{Z}_{1:k}) \cdot p(z_{k+1}^{12} | \mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2)}{p(z_{k+1}^{12})}. \quad (32)$$

Generating Message.

$$p(\mathbf{x}_k^3, \mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2 | \mathbf{Z}_{1:k}, z_{k+1}^{12})$$

$$= \int p(\mathbf{x}_k^2, \mathbf{x}_k^3, \mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2 | \mathbf{Z}_{1:k}, z_{k+1}^{12}) d\mathbf{x}_k^2. \quad (33)$$

(iii) Spacecraft 3:

Receiving Message.

$$p(\mathbf{x}_k^3, \mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2, \mathbf{x}_{k+1}^3 | \mathbf{Z}_{1:k}, z_{k+1}^{12})$$

$$= p(\mathbf{x}_{k+1}^3 | \mathbf{x}_k^3) \cdot p(\mathbf{x}_k^3, \mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2 | \mathbf{Z}_{1:k}, z_{k+1}^{12}). \quad (34)$$

Collecting Evidence.

$$p(\mathbf{x}_k^3, \mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2, \mathbf{x}_{k+1}^3 | \mathbf{Z}_{1:k+1})$$

$$= \frac{p(\mathbf{x}_k^3, \mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2, \mathbf{x}_{k+1}^3 | \mathbf{Z}_{1:k}, z_{k+1}^{12})}{p(z_{k+1}^{13}) \cdot p(z_{k+1}^{23})}$$

$$\cdot p(z_{k+1}^{13} | \mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^3) \cdot p(z_{k+1}^{23} | \mathbf{x}_{k+1}^2, \mathbf{x}_{k+1}^3). \quad (35)$$

Generating Message.

$$p(\mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2, \mathbf{x}_{k+1}^3 | \mathbf{Z}_{1:k+1})$$

$$= \int p(\mathbf{x}_k^3, \mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2, \mathbf{x}_{k+1}^3 | \mathbf{Z}_{1:k+1}) d\mathbf{x}_k^3. \quad (36)$$

The message generated by spacecraft 3, $p(\mathbf{x}_{k+1}^1, \mathbf{x}_{k+1}^2, \mathbf{x}_{k+1}^3 | \mathbf{Z}_{1:k+1})$, is the final result at time step $k + 1$, which is the same as centralized BE.

5. Simulation Results and Discussion

The effectiveness of the decentralized solution proposed is validated by the simulation of a three-satellite formation. The Keplerian elements of the formation satellites are listed in Table 1. For simplicity, only central body gravitation is considered (also known as the two-body problem). In the formation, satellite 1 is selected as the leading spacecraft, with the other two accompanying it. The relative coordinate system (or the Hill frame), $O_s - X_r Y_r Z_r$, is based on satellite 1, where the X axis is from the center of Earth to the center of satellite 1, the Y axis is along the direction of velocity in the orbital plane, and the Z axis is established using the right-hand rule. The relationship between $O_s - X_r Y_r Z_r$ and the Earth-Centered Inertial (ECI) coordinate system, $O_e - X_i Y_i Z_i$, is given in Figure 5. Under the two-body assumption, the relative motion of the formation is given in Figure 6.

5.1. Precision of Decentralized Algorithms. Every spacecraft in the formation is equipped with a GPS receiver as well as RF range and communication devices. All noises are treated as Gaussian. The standard deviation for GPS measurement and RF range are 0.3 m and 0.1 m, respectively. The estimations of the absolute state of satellite 3 using different algorithms are listed as follows.

Centralized EKF is setup as a standard to evaluate the precision of other algorithms. As there are no further simplifications or decoupling, the precision of centralized EKF is better

TABLE 1: Keplerian orbital elements of a satellite formation.

Sat number	a (km)	e	i (°)	Ω (°)	ω (°)	M_0 (°)
1	8400.000000	0.005000	30.000000	120.000000	0.000000	90.000000
2	8400.000000	0.004743	29.996649	119.988151	1.783138	88.250782
3	8400.000000	0.004743	30.003353	120.011847	1.762616	88.250782

FIGURE 5: Relative coordinate system and ECI coordinate system. The relative coordinate system (or the Hill frame) is represented by $O_s - X_r Y_r Z_r$, and the ECI coordinate system is represented by $O_e - X_i Y_i Z_i$.

than all the decentralized solutions reviewed in Section 2. Figure 7 illustrates the error of estimation of the absolute state of satellite 3 in the ECI coordinate system using centralized EKF.

All decentralized algorithms are compared with centralized EKF. For FOEKF, because only local measurements related with each satellite are considered, the simplification of measurement model will result in suboptimal estimations. Figure 8 gives result of absolute state estimation of satellite 3 in the ECI coordinate system using FOEKF. The root-mean-square error (RMSE) tends to fluctuate around 0.05 m, which is inferior to centralized EKF.

The core idea that differs ROEKF from FOEKF is that the states transferred from other spacecrafts are assumed to be precise, and therefore the interformation measurement functions could be decoupled. After decoupling, the measurement function is only related to the state of the local spacecraft. This assumption will result in a further precision loss of ROEKF compared with FOEKF. Figure 9 shows that a notable precision loss is caused by this simplification compared with Figure 8.

Notice that the states transmitted from other spacecrafts are not precise, in other words, with an uncertainty represented by the covariance. In IREKF, such uncertainty is considered as a proper bump-up to the noise of measurements. Figure 10 gives an example of satellite 3 using IREKF. The results are greatly improved in IREKF compared with ROEKF. However, because the covariance matrixes

TABLE 2: RMSE of all algorithms at the end of simulation.

Algorithm	EKF	FOEKF	ROEKF	IREKF	ICEKF-FOEKF	JT
RMSE (m)	0.0248	0.0739	28.1360	0.0690	0.0310	0.0248

transferred from other spacecrafts are generated on local measurements, IREKF is suboptimal. The precision of IREKF is still inferior to EKF. Comparing Figure 10 with Figure 8, it can be seen that IREKF tends to approximate FOEKF.

A common method to compensate the precision loss caused by simplification or linearization is to use iterations. ICEKF is based on the same intuition. Figure 11 gives an example of ICEKF based on FOEKF with 5 iterations per epoch. Without iterations, the algorithm is the same as FOEKF. It can be seen that the result has been improved remarkably compared with FOEKF. However, the result is still slightly inferior to centralized EKF.

Figure 12 gives an example of the decentralized solution based on JT. As there are no decoupling or further assumptions, the result of JT is equivalent to centralized EKF.

In order to provide an intuitive comparison of different algorithms, Table 2 summarizes the RMSE of all algorithms. Because some algorithms fail to reach convergence, the end of simulation is picked up as the epoch when the RMSE of different algorithms is evaluated. It can be seen that JT has the best precision among all the decentralized algorithms discussed in this paper.

5.2. Analysis of Computational Complexity. While analyzing the computational complexity of algorithms, the operation of multiplication of two real numbers is defined as a unit element. Recall the procedure of centralized EKF in Section 3.1. Suppose $\mathbf{x} \in \mathbb{R}^{m \times 1}$ and $\mathbf{Z} \in \mathbb{R}^{d \times 1}$. Notice that (16) requires matrix inversio, and take elementary row transformation as the method for matrix inversion. The computational complexity of (14), (15), (16), (17), and (18) is $O(m^2 n^2)$, $O(m^3 n^3)$, $O(d^3) + O(d^2 mn) + O(dm^2 n^2)$, $O(dmn)$, and $O(dm^2 n^2) + O(m^3 n^3)$, respectively. Thus the total computational complexity of EKF time update and measurement update is $O(m^3 n^3)$ and $O(m^3 n^3) + O(dm^2 n^2) + O(d^2 mn) + O(d^3)$, respectively.

The computational complexity of decentralized algorithms could be estimated using similar methods. For FOEKF, suppose $\mathbf{Z}_{k+1}^i \in \mathbb{R}^{d_i \times 1}$. The computational complexity of FOEKF time update and measurement update at spacecraft i is $O(m^3 n^3)$ and $O(m^3 n^3) + O(d_i m^2 n^2) + O(d_i^2 mn) + O(d_i^3)$, respectively. Notice that, in ROEKF, time update for every spacecraft in the formation is carried out at each local

FIGURE 6: Relative motion between satellites.

FIGURE 7: Error of absolute position estimation for satellite 3 using EKF.

FIGURE 8: Error of absolute position estimation of satellite 3 using FOEKF.

FIGURE 9: Error of absolute position estimation for satellite 3 using ROEKF.

FIGURE 10: Error of absolute position estimation for satellite 3 using IREKF.

FIGURE 11: Error of absolute position estimation for satellite 3 using FOEKF-based ICEKF.

FIGURE 12: Error of absolute position estimation for satellite 3 using JT.

filter, and the computational complexity of decoupling at spacecraft i is $O(d_i m)$ spacecraft i . Thus the computational complexity of ROEKF time update and measurement update at spacecraft i is $O(m^3 n)$ and $O(m^3) + O(d_i m^2) + O(d_i^2 m) + O(d_i^3)$, respectively. Because the calculation of the increase to the measurement noise of $\mathbf{R}$ is $O(d_i m^2)$, the computational complexity of IREKF is at the same level of ROEKF.

Because centralized BE is equivalent to centralized EKF, the total computational complexity of JT is the same as centralized EKF. For spacecraft i , suppose that the number of evidences deployed to clique i is d'_i . Thus the computational complexity of JT at spacecraft i is $O(m^3 n^3) + O(d'_i m^2 n^2) + O(d_i'^2 m n) + O(d_i'^3)$. Notice that, in JT, the measurement update of centralized BE is split and implemented in different cliques of JT. Thus $d'_i \leq d_i$. Hence we could sort the local computational complexity of decentralized algorithms as $\text{ROEKF} < \text{IREKF} < \text{JT} \leq \text{FOEKF}$.

6. Conclusions

The decentralized state estimation algorithms for formation flying spacecraft studied in this paper could be categorized as two main groups: decomposition through simplification and decoupling and decomposition through structural analysis of models. In the first category, irrelevant measurements are ignored and interformation measurements are decoupled at each spacecraft, rendering suboptimal estimates with inferior precisions to centralized EKF. All algorithms reviewed in Section 3 belong to the first category. In the second category, states of different spacecrafts are categorized as cliques of JT and measurements are deployed on different cliques. Since there is no simplification, the new solution based on JT is equivalent in precision to centralized EKF in theory. Simulation results indicate that the new solution of JT holds the best precision among all decentralized algorithms discussed in this paper. For any state estimation applications whose

model is in the same form as cooperative navigation, JT may serve as a general scheme for developing decentralized algorithms.

Appendices

A. State Transition Matrix and Measurement Matrix under Two-body Assumptions

A.1. State Transition Matrix. Recall the dynamic model in Section 2. Under two-body assumptions, the state transition matrix $\mathbf{F}_k^i$ is given by

$$\mathbf{F}_k^i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -\frac{3T}{2} \sqrt{\frac{\mu}{(a_k^i)^5}} & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{A.1})$$

where T is the time step and μ is the gravity constant.

A.2. Measurement Matrix. Recall the measurement model in Section 2. While computing the measurement matrices, we need to consider the Keplerian-to-Cartesian partial derivatives, which are partials of positions to Keplerian orbital

elements. Under two-body assumptions, the Keplerian-to-Cartesian partial derivatives are given by

$$\begin{aligned} \frac{\partial \mathbf{r}}{\partial (a, e, M)^T} &= \mathbf{Rot} \cdot \frac{\partial \mathbf{r}^*}{\partial (a, e, M)^T}, \\ \frac{\partial \mathbf{r}}{\partial (i', \Omega, \omega)^T} &= \frac{\partial \mathbf{Rot}}{\partial (i', \Omega, \omega)^T} \cdot \mathbf{r}^*, \end{aligned} \quad (\text{A.2})$$

where

$$\mathbf{Rot} = \mathbf{R}_z(-\Omega) \cdot \mathbf{R}_x(-i') \cdot \mathbf{R}_z(-\omega),$$

$$\mathbf{R}_x(-i') = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i' & -\sin i' \\ 0 & \sin i' & \cos i' \end{bmatrix},$$

$$\mathbf{R}_z(-\Omega) = \begin{bmatrix} \cos \Omega & -\sin \Omega & 0 \\ \sin \Omega & \cos \Omega & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{R}_z(-\omega) = \begin{bmatrix} \cos \omega & -\sin \omega & 0 \\ \sin \omega & \cos \omega & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (\text{A.3})$$

$$\begin{aligned} \frac{\partial \mathbf{r}^*}{\partial (a, e, M)^T} &= \begin{bmatrix} \cos E - e & -a - \frac{a \sin^2 E}{1 - e \cos E} & -\frac{a \sin E}{(1 - e \cos E)} \\ \sqrt{1 - e^2} \sin E & \frac{a \sin E (\cos E - e)}{\sqrt{1 - e^2} (1 - e \cos E)} & \frac{a \sqrt{1 - e^2} \cos E}{1 - e \cos E} \\ 0 & 0 & 0 \end{bmatrix}. \end{aligned}$$

E is solved by the Kepler's equation as follows:

$$E - e \sin E = M,$$

$$\begin{aligned} \frac{\partial \mathbf{Rot}}{\partial (i', \Omega, \omega)^T} &= \begin{bmatrix} \mathbf{R}_z(-\Omega) \frac{\partial \mathbf{R}_x(-i')}{\partial i'} \mathbf{R}_z(-\omega) & \frac{\partial \mathbf{R}_z(-\Omega)}{\partial \Omega} \mathbf{R}_x(-i') \mathbf{R}_z(-\omega) & \mathbf{R}_z(-\Omega) \mathbf{R}_x(-i') \frac{\partial \mathbf{R}_z(-\omega)}{\partial \omega} \end{bmatrix}, \\ \frac{\partial \mathbf{R}_x(-i')}{\partial i'} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin i' & -\cos i' \\ 0 & \cos i' & -\sin i' \end{bmatrix}, \\ \frac{\partial \mathbf{R}_z(-\Omega)}{\partial \Omega} &= \begin{bmatrix} -\sin \Omega & -\cos \Omega & 0 \\ \cos \Omega & -\sin \Omega & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ \frac{\partial \mathbf{R}_z(-\omega)}{\partial \omega} &= \begin{bmatrix} -\sin \omega & -\cos \omega & 0 \\ \cos \omega & -\sin \omega & 0 \\ 0 & 0 & 0 \end{bmatrix}. \end{aligned} \quad (\text{A.4})$$

B. Basic Operations on CG Potentials

B.1. Multiplication. Consider two random vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$, where $p(\mathbf{x}_1) = \mathcal{N}(\hat{\mathbf{x}}_1, \Sigma_{11})$ and

$$\mathbf{x}_2 = \mathbf{F}\mathbf{x}_1 + \mathbf{u}, \quad \mathbf{u} \sim \mathcal{N}(\hat{\mathbf{u}}, \mathbf{U}). \quad (\text{B.1})$$

Equation (B.1) defines the distribution of $\mathbf{x}_2$ conditioned on $\mathbf{x}_1$, which is

$$p(\mathbf{x}_2 | \mathbf{x}_1) = \mathcal{N}(\mathbf{F}\hat{\mathbf{x}}_1 + \hat{\mathbf{u}}, \mathbf{U}). \quad (\text{B.2})$$

The joint distribution of $\mathbf{x}_1$ and $\mathbf{x}_2$ is given by

$$\begin{aligned} p(\mathbf{x}_1, \mathbf{x}_2) &= p(\mathbf{x}_1) \cdot p(\mathbf{x}_2 | \mathbf{x}_1) \\ &= \mathcal{N}\left(\begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right), \end{aligned} \quad (\text{B.3})$$

where

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix} &= \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \mathbf{F}\hat{\mathbf{x}}_1 + \hat{\mathbf{u}} \end{bmatrix}, \\ \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} &= \begin{bmatrix} \Sigma_{11} & \Sigma_{11}\mathbf{F}^T \\ \mathbf{F}\Sigma_{11} & \mathbf{F}\Sigma_{11}\mathbf{F}^T + \mathbf{U} \end{bmatrix}. \end{aligned} \quad (\text{B.4})$$

B.2. Division. If $p(\mathbf{x}_1, \mathbf{x}_2)$ and $p(\mathbf{x}_2)$ are Gaussian,

$$\begin{aligned} p(\mathbf{x}_1, \mathbf{x}_2) &= \mathcal{N}\left(\begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right), \\ p(\mathbf{x}_2) &= \mathcal{N}(\hat{\mathbf{x}}_2, \Sigma_{22}) \end{aligned} \quad (\text{B.5})$$

and the distribution of $\mathbf{x}_1$ conditioned on $\mathbf{x}_2$ is given by

$$p(\mathbf{x}_1 | \mathbf{x}_2) = \frac{p(\mathbf{x}_1, \mathbf{x}_2)}{p(\mathbf{x}_2)} = \mathcal{N}(\hat{\mathbf{x}}_{1|2}, \Sigma_{1|2}), \quad (\text{B.6})$$

where $\hat{\mathbf{x}}_{1|2} = \hat{\mathbf{x}}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \hat{\mathbf{x}}_2)$ and $\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}^T$.

B.3. Marginalization. If the joint distribution of $\mathbf{x}_1$ and $\mathbf{x}_2$ is Gaussian,

$$p(\mathbf{x}_1, \mathbf{x}_2) = \mathcal{N}\left(\begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right). \quad (\text{B.7})$$

The distribution of each random vector is given by

$$\begin{aligned} p(\mathbf{x}_1) &= \int p(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_2 = \mathcal{N}(\hat{\mathbf{x}}_1, \Sigma_{11}), \\ p(\mathbf{x}_2) &= \int p(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1 = \mathcal{N}(\hat{\mathbf{x}}_2, \Sigma_{22}). \end{aligned} \quad (\text{B.8})$$

Conflict of Interests

The authors declare no conflict of interests.

Authors' Contribution

All authors contributed equally to this work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61203200).

References

- [1] M. L. Psiaki, "Autonomous orbit determination for two spacecraft from relative position measurements," *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 2, pp. 305–312, 1999.
- [2] J. Betbeder, J. Nabucet, E. Pottier, J. Baudry, S. Corgne, and L. Hubert-Moy, "Detection and characterization of hedgerows using TerraSAR-X imagery," *Remote Sensing*, vol. 6, no. 5, pp. 3752–3769, 2014.
- [3] S. Leung and O. Montenbruck, "Real-time navigation of formation-flying spacecraft using global-positioning-system measurements," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 2, pp. 226–235, 2005.
- [4] C.-W. Park, *Precise relative navigation using augmented CDGPS [Ph.D. thesis]*, Department of Mechanical Engineering, Stanford University, Stanford, Calif, USA, 2001.
- [5] P. Ferguson and J. How, "Decentralized estimation algorithms for formation flying spacecraft," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Austin, Tex, USA, 2003.
- [6] H. Mu, T. Bailey, P. Thompson, and H. Durrant-Whyte, "Decentralised solutions to the cooperative multi-platform navigation problem," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 1433–1449, 2011.
- [7] J. D. Wolfe, J. L. Speyers, S. Hwang, Y. J. Lee, and E. Lee, "Estimation of relative satellite position using transformed differential carrier-phase GPS measurements," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1217–1227, 2007.
- [8] B. Guy, D. Thomas, G. E. Ruggi, G. Lukas, T. Björn, and K. Klaus, "The OPTTEL terminal development programme—enabling technologies for future optical crosslink applications," in *Proceedings of the AIAA Space Conference & Exposition*, American Institute of Aeronautics and Astronautics, Ed., Long Beach, Calif, USA, September 2003.
- [9] G. Baister, H. Ch, E. Fischer, and K. Pribil, "Optical communication crosslink terminals for future broadband satellite applications," in *Proceedings of the 18th International Communications Satellite Systems Conference and Exhibit*, American Institute of Aeronautics and Astronautics, 2000.
- [10] M. L. Psiaki, "Absolute orbit and gravity determination using relative position measurements between two satellites," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 5, pp. 1285–1297, 2011.
- [11] P. A. M. Abusali, B. D. Tapley, and B. E. Schutz, "Autonomous navigation of global positioning system satellites using crosslink measurements," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 321–327, 1998.
- [12] H. Mu, *Decentralized algorithms of cooperative navigation for mobile platforms [Ph.D. thesis]*, National University of Defense Technology, Changsha, China, 2010.
- [13] S. L. Lauritzen and F. Jensen, "Stable local computation with conditional gaussian distributions," *Statistics and Computing*, vol. 11, no. 2, pp. 191–203, 2001.

- [14] H. Mu, M. Dai, M. Gao, and X. Pan, "Fully decentralized cooperative localization of a robot team: an efficient and centralized equivalent solution," in *Proceedings of the 6th International Conference on Intelligent Robotics and Applications (ICIRA '13)*, pp. 520–531, Busan, Republic of Korea, September 2013.
- [15] C.-W. Park, P. Ferguson, N. Pohlman, and J. P. How, "Decentralized relative navigation for formation flying spacecraft using augmented CDGPS," in *Proceedings of the 14th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION '01)*, Salt Lake City, Utah, USA, September 2001.
- [16] M. A. Paskin, "Thin junction tree filters for simultaneous localization and mapping," in *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI '03)*, pp. 1157–1164, August 2003.
- [17] S. L. Lauritzen, "Propagation of probabilities, means, and variances in mixed graphical association models," *Journal of the American Statistical Association*, vol. 87, no. 420, pp. 1098–1108, 1992.
- [18] G. A. Ruz and D. T. Pham, "Building Bayesian network classifiers through a Bayesian complexity monitoring system," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 223, no. 3, pp. 743–755, 2009.
- [19] M. A. Paskin and C. E. Guestrin, "Robust probabilistic inference in distributed systems," in *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI '04)*, pp. 436–445, 2004.