DISTRIBUTED MULTI-TARGET RELATIVE POSE ESTIMATION FOR COOPERATIVE SPACECRAFT SWARM

Kai Matsuka,[*] Elena Sorina Lupu,[†] Yashwanth Kumar Nakka,[‡]
Rebecca Foust [§], Soon-Jo Chung,[¶] and Fred Hadaegh[‖]

Multi-agent relative state estimation is critical in enabling full swarm autonomy. However, relative pose estimation of hundreds to thousands of cooperative agents is challenging due to limited sensing, limited communication, and scalability. We present a distributed algorithm for cooperative multi-agent localization with both limited relative sensing and communication. Each agent locally exchanges the relative measurements and jointly estimates the relative poses of its local neighbors. Because the algorithm only estimates the local neighbors, the number of states does not grow with the total number of agents given the same local sensing and communication graphs, making the algorithm suitable for swarm application. The proposed algorithm is applied to spacecraft swarm localization and verified in simulation and experiments. Experiments are conducted on Caltech's robotic spacecraft simulators, the Multi-Spacecraft Testbed for Autonomy Research (M-STAR), where each spacecraft uses vision-based relative measurements.

INTRODUCTION

Spacecraft swarms have the potential to revolutionize space technology by enabling missions like distributed aperture telescopes, space structure assembly, and cooperative deep space exploration.[1,2] These multi-spacecraft missions have several advantages over monolithic satellite missions, such as robustness to agent loss and improved science return.[3] Relative state estimation is a critical component of formation keeping, path planning, collision avoidance, and rendezvous and docking. However, in order to enable full swarm autonomy in space, relative state estimation algorithms must overcome several challenges that get increasingly difficult as the number of agents increases. Even when the total number of agents is small, multi-agent systems face challenges such as limited relative sensing and communication ranges. In addition, a swarm with a much larger number of agents must use an algorithm architecture that does not grow with the total number of agents in swarm to keep the computation tractable. Existing solutions for multi-agent localization carry the

[*] Graduate Research Assistant, Graduate Aerospace Laboratories of the California Institute of Technology, 1200 E. California Blvd, Pasadena, CA 91125.

[†] Graduate Research Assistant, Graduate Aerospace Laboratories of the California Institute of Technology, 1200 E. California Blvd, Pasadena, CA 91125.

[‡] Graduate Research Assistant, Graduate Aerospace Laboratories of the California Institute of Technology, 1200 E. California Blvd, Pasadena, CA 91125.

[§] Special Student, Graduate Aerospace Laboratories of the California Institute of Technology, 1200 E. California Blvd, Pasadena, CA 91125.

[¶] Bren Professor of Aerospace, Graduate Aerospace Laboratories of the California Institute of Technology, 1200 E. California Blvd, Pasadena, CA 91125.

[‖] Senior Research Scientist and Chief Technologist, Jet Propulsion Laboratory, 4800 Oak Grove Dr, Pasadena, CA 91109.

states for all the agents in the swarm, they are not scalable for swarm applications. Instead, a local estimation algorithm that does not grow with swarm size is desired.

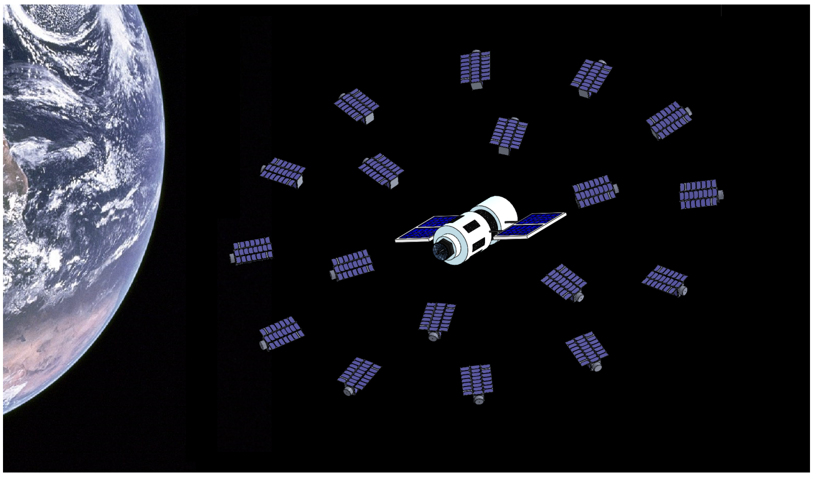


Figure 1. Concept mission with a formation flying swarm of Cubesats inspecting a bigger spacecraft

In this paper, we present the Distributed Pose Estimation (DPE) algorithm for collaborative multi-agent localization of an individual agent with respect to its local neighbors. In the proposed architecture, each agent senses the relative poses of its neighbors with respect to its local frame using vision-based techniques and communicates the relative pose measurements with an associated confidence-level that acts as a prior to the DPE algorithm. The DPE algorithm localizes each agent by fusing relative estimates received from its neighbors. Figure 2 shows an example of the local relative sensing graph given the sensing and communication graph topologies of the whole swarm. The local relative sensing graph can be viewed as a pose graph, over which each agent jointly estimates relative pose states. Hence, the algorithm can utilize the additional cyclic structure in the relative sensing graph, thereby improving the observability.

The DPE algorithm shares measurements with local neighbors, but not their estimated states. Estimated states from different agents do not depend on each other, thereby allowing for more straightforward stability analysis. The DPE is formulated as a nonlinear estimation problem with the measurement equation augmented by relative pose measurements received from the agents in the local graph. In this paper, we use the Extended Kalman Filter (EKF) approach to solve the DPE problem for real-time implementation. The DPE algorithm can also be implemented using other nonlinear estimation algorithms such as an unscented Kalman filter, a particle filter, and an optimal nonlinear observer.[4]

The algorithm is verified using simulations and experiments. The two simulation scenarios consider a planar relative orbital dynamics in a planetary orbit described by the Hill-Clohessy-Wiltshire (HCW) equations, one with four spacecraft and another with sixty spacecraft. The DPE algorithm is compared against an independent EKF run for each neighbor for which direct measurement are observable. Robotic experiments are conducted on the Caltech's robotic spacecraft dynamics simulators, the Multi-Spacecraft Testbed for Autonomy Research (M-STAR).[5] Each spacecraft simulator is equipped with a monocular camera and unique visual markers with known geometry, whose full poses are extracted using a computer vision algorithm run on-board each spacecraft. The M-STAR utilizes a motion capture system, which provides ground truth for algorithmic verification. In sum-

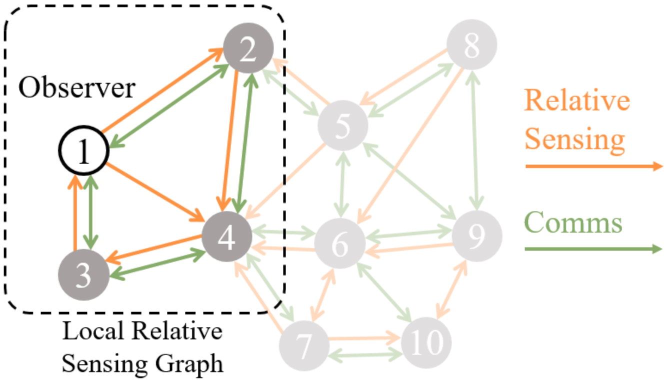


Figure 2. An example of a local relative sensing graph of agent 1 given the communication and sensing topologies.

mary, we present a distributed algorithm for vision-based pose estimation of multiple targets and its experimental validation.

Related Work

Extensive work has been done in the general field of distributed estimation. Several algorithms have been developed to effectively fuse measurements from distributed systems like the Kalman Consensus Filter,[6] the Information-Weighted Consensus Filter,[7] or the Logarithmic Consensus.[8] These algorithms can be applied to the multi-agent localization problem. Several algorithms have also been developed specifically for multi-agent localization. For spacecraft and robotic applications, the λ-estimator estimates the relative translational states of a linear system switching sensing topology,[9] and another algorithm estimates the relative pose of stationary agents without a common reference frame.[10] In wireless sensor network applications, some algorithms[11,12] estimate the sensor locations with the knowledge of some agents in the network. One paper discusses the conditions of observability for attitude states for multi-agent system with a relative sensing graph.[13] Another discusses observability of multi-agent localization in the sense of graph connectivity.[14]

The algorithms discussed above have been successfully tested with some small-scale multi-agent applications, but they assume that the states to be estimated by each agent are the same. A challenge arises when these algorithms are applied to large-scale swarm localization problems where the size of the states to be estimated grows with the number of agents. Large-scale estimation problems can be made tractable by limiting each agent to estimate only locally. As long as the size and complexity of a local graph pertaining to each agent remains approximately the same, the computational load for each agent does not change as more agents are added to the swarm. Estimation of local states in a large-scale problem is studied in the context of static estimation for networked power systems.[15,16] Similar to the existing literature, the algorithm proposed in this paper also considers local relative states to achieve a tractable algorithm complexity for swarm applications. In contrast, our problem involves dynamics and the states to be estimated are the local relative poses of neighbor agents.

Another important aspect of multi-agent localization is dealing with sensing and communication networks simultaneously. This is particularly relevant for spacecraft formation flying missions where difficult sensing and power constraints exist. In general, the limitations for communication and relative sensing graph may be different, and may occur simultaneously for swarm systems.

Much of the existing work focuses on either sensing or communication issues and not on the intersection of the two.[9,10,13]Some prior work[14] discusses the observability of the states when a combination of relative sensing and communication networks are considered. While this addresses the fundamental observability of the agent states given combined sensing and communication networks, it is limited to a case where the communication has an infinite number of information exchange at each time step. In this paper, we modify this notion of observability[14] to a case where the number of information exchanges at each time step is also limited. Based on this observability, we create a rule to select the size of the set of agents that each agent estimates.

The main contribution of this paper are as follows. First, we present a distributed, multi-agent 3-DOF localization algorithm for both limited sensing and communication networks. Estimated states are local so the computation does not grow with the swarm size as long as the size of the local sensing neighbor graph is fixed. Second, we provide a sufficient condition for when each agent can observe its neighbors' states. Based on this condition, we provide a rule to select which agents are included in the local sensing neighbor graph. Finally we validate the proposed DPE algorithm via simulation and experimentation using robotic spacecraft simulators with vision-based pose extraction.

The outline of the rest of the paper is as follows. We first review preliminaries pertaining to the graph theory notations, and the relative orbital dynamics. Next, we propose the Distributed Pose Estimation (DPE) Algorithm. Then we describe the simulations and the robotic experiments results, finally followed by the conclusion section.

PRELIMINARIES

Let $G_s = (\mathcal{V}, \mathcal{E}_s)$ denote a directed graph that describes the *relative sensing graph*, where $\mathcal{V} = \{1, \ldots, N\}$ denotes the set of agents and $\mathcal{E}_s$ denotes a set of edges. An edge $(i, j) \in \mathcal{E}_s$ when i-th agent measures the relative pose of j-th agent. Similarly, let $G_c = (\mathcal{V}, \mathcal{E}_c)$ denote the *communication graph*, a directed graph for the communication topology. We say $(i, j) \in \mathcal{E}_c$ if there is an information flow from i-th agent to j-th agent. Note that the measurement graph and the communication graph may be different.[14] An agent j is a neighbor of i in the graph $\mathcal{X}$ if $(i, j) \in \mathcal{E}(\mathcal{X})$. The set of neighbors of i in graph $\mathcal{X}$ is defined as $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}(\mathcal{X})\}$. An extended set of neighbors is defined as $\bar{\mathcal{N}_i} = \mathcal{N}_i \cup \{i\}$. A *subgraph* of a graph $\mathcal{X}$ is a graph $\mathcal{Y}$ such that $\mathcal{V}(\mathcal{Y}) \subseteq \mathcal{V}(\mathcal{X})$, $\mathcal{E}(\mathcal{Y}) \subseteq \mathcal{E}(\mathcal{X})$. Let $I : \mathbb{Z} \to \mathcal{V}$ be a mapping from an index of a set of vertices to the element at that index.

The relative orbital dynamics of formation-flying spacecraft are described using a local-vertical, local-horizontal (LVLH) attached to each spacecraft. The LVLH frame defined at the i-th spacecraft, denoted L_i, is defined as follows in this paper: the x direction, R, is along the position vector to the spacecraft; the z direction, N, is along the angular momentum vector of the spacecraft's orbit; and the y direction, T, completes the right-handed rule. The origin of the LVLH frame, defined for each spacecraft, coincides with the center of gravity of the spacecraft. The body-fixed frame of i-th spacecraft is denoted B_i, and its origin coincides with the center of gravity of the i-th spacecraft. For two formation flying spacecraft with a separation distance much smaller than its orbit radius, the attached LVLH frames have negligibly small attitude difference. This is because the radial and angular momentum vectors for the two spacecraft are much bigger than their differences. For the rest of the paper, we assume that each pair of agents has a small intersatellite distance compared to the orbital radius so that the relative attitude between the LVLH frames attached to different agents are assumed to be aligned.

Relative Orbital Dynamics: 3DOF System

This section reviews the equation of motion for the relative orbital dynamics. This paper considers that spacecraft are flying in formation in a circular orbit around a planet. For spacecraft in a circular orbit, the linearized relative orbital motion simplifies to the HCW equations assuming no perturbations and a short distance from the origin of the relative frame. The relative state vector of the j-th agent with respect to the i-th agent is denoted $x_{j/i}$ and is defined as

$$x_{j/i}{}^T = \left[\begin{array}{cccc} p_{j/i}^{L_i}{}^T & v_{j/i}^{L_i}{}^T & \theta_{j/L_i} & \omega_{j/L_i} \end{array} \right] \tag{1}$$

where $p_{j/i}^{L_i}, v_{j/i}^{L_i} \in \mathbb{R}^2$ are the position and velocity vectors of agent j with respect to agent i and $\theta_{j/L_i} \in [0, 2\pi)$, $\omega_{j/L_i} \in \mathbb{R}$ are the attitude and angular rate of agent j with respect to LVLH frame of agent i. Since all spacecraft are in the PRO with respect to all the other spacecraft, the relative orbit dynamics can be conveniently defined with respect to L_i. The absolute state vector of the i-th agent $x_i \in [0, 2\pi) \times \mathbb{R}$ is defined as

$$x_i{}^T = \left[\begin{array}{cc} \theta_{i/L_i} & \omega_{i/L_i} \end{array} \right] \tag{2}$$

where $\theta_{i/L_i} \in [0, 2\pi)$ and $\omega_{i/L_i} \in \mathbb{R}$ are the attitude and angular rate of B_i with respect to L_i. Recall that L_i denotes these vectors are expressed in L_i frame. Since we assumed that the orientation of the LVLH frame for each spacecraft L_i are approximately equal to each other, we will denote this superscript L_i with L from here on. While there exist nonlinear, higher fidelity dynamics models that include J_2 and air drag,[17] we focus on the hcw dynamics to illustrate the distributed aspects of the algorithm.

The relative states $x_{j/i}$ can be propagated via a linear time-invariant state transition matrix $x_{j/i}(t_{k+1}) = A_{rel} x_{j/i}(t_k)$, where

$$A_{rel} = \left[\begin{array}{cc} A_1 & 0_{4\times2} \\ 0_{2\times4} & A_2 \end{array} \right] \tag{3}$$

where $A_1 = \Phi_{\text{HCW}}(\Delta t)$ is the state transition matrix given by the solution to the HCW equations and A_2 is the state transition matrix for the double integrator system.

DISTRIBUTED POSE ESTIMATION ALGORITHM

In this section, we present the Distributed Pose Estimation Algorithm (DPE). The DPE algorithm estimates the relative poses of a subset of agents in a swarm, given the relative sensing and communication network topologies. First, each spacecraft makes relative pose measurements of the neighbor agents. Then each spacecraft communicates its relative measurements and associated measurement covariances with its communication neighbors. The agent i receives this information from its communicating neighbor agent $l \in \mathcal{N}_c^i$. Finally, using the relative measurements received by communication, the DPE estimates the relative pose of the spacecraft. This algorithm is summarized in Algorithm 1. The same algorithm is implemented on each agent.

Algorithm 1: DPE Algorithm

```
Result: Estimate $\hat{x}_i^+$ and $P_i^+$
$\hat{x}_i^+ = \hat{x}_i(0)$, $\hat{P}_i^+ = \hat{P}_i(0)$
while true do
    $\hat{P}_i^- = \hat{P}_i^+$, $\hat{x}_i^- = \hat{x}_i^+$
    $\mathcal{M}^i$ = Measure()
    for $j \in \mathcal{N}_c^i$ do
        Broadcast($\mathcal{M}^i$)
        $\bar{\mathcal{M}}^i$ = $\mathcal{M}^i \cup$ Receive($\mathcal{M}^j$)
    end
    $\tilde{\mathcal{M}}^i$ = PruneUnobservable($\bar{\mathcal{M}}^i$)
    $\hat{x}_i^+$, $P_i^+$ = EKF($\tilde{\mathcal{M}}^i$, $\hat{x}_i^-$, $\hat{P}_i^-$)
end
```

We begin by defining a local relative sensing graph $\mathcal{G}_i$ with respect to the agent i. The local relative sensing graph is a subgraph of the relative sensing graph (i.e. $\mathcal{G}_i \subseteq \mathcal{G}_f$). We suppose that a list of vertices for the local relative sensing graph $\mathcal{V}_i$ is given (an approach on how to select $\mathcal{V}_i$ is discussed in detail in later sections), and define the set of neighbor states to be

$$\mathcal{S}_i = \{x_{j/i} \mid j \in \mathcal{V}_i \setminus i\}. \tag{4}$$

The goal of the DPE is to estimate the augmented state vector $x_i \in \mathbb{R}^{nN_i}$, which is a column concatenation of $\mathcal{S}_i$.

Relative Pose Sensing and Measurement Models

Each spacecraft is assumed to carry a relative pose sensor which directly measures the relative pose of neighboring sensors in its field-of-view. The sensor is assumed to be capable of measuring multiple targets simultaneously. For example, a monocular camera and computer vision algorithms with known target geometry can provide such an estimate.[18] We define the planar true pose of j-th body frame with respect to the i-th body frame as

$$T_j^i \triangleq \begin{bmatrix} p_{j/i}^i \\ \theta_{j/i} \end{bmatrix} = \begin{bmatrix} R(\theta_{i/L})^T p_{j/i}^L \\ \theta_{j/L} - \theta_{i/L} \end{bmatrix} \tag{5}$$

The second equality relates the relative pose between agent i and j in terms of the states estimated by agent i. Note that $\theta_{i/L}$ is an absolute measurement of agent i and is not a dependent variable, hence both the absolute and relative measurement equations are linear. Then a relative measurement of j-th agent with respect to i-th agent is given by following

$$y_{j/i} = h_{j/i} + v_{j/i} \tag{6}$$

where $h_{j/i} = T_j^i$ and $v_{j/i}$ is a zero-mean Gaussian noise with covariance $V_{j/i}$. The subscript j/i in $y_{j/i}$ denotes it is a relative measurement associated with the i-th agent observing the j-th agent.

The same relative measurement may be used in DPE algorithms running on multiple agents. Since each agent estimates the relative pose of neighbors with respect to itself, the relative states representing an agent are different for each agent. Therefore, while $y_{j/i}$ represents a measurement, its measurement model may be expressed in different ways, depending on which agent the estimation is done. Specifically, each relative measurement may take three different forms. Suppose that the i-th agent is the reference spacecraft, on which the DPE algorithm is run. The first model is the case where the relative measurement of the j-th agent is made by the i-th agent itself. This results in the trivial case of Equation 6. The second is the relative measurement of the i-th agent made by the j-th agent. Because the agent i has $p^L_{i/j}$ and $\theta_{j/L}$ as states, the relative measurement becomes

$$y_{i/j} = h_{i/j} + v_{i/j} \tag{7}$$

where $h_{i/j} = (T^i_j)^{-1}$ and

$$(T^i_j)^{-1} = \begin{bmatrix} -R(\theta_{j/i})^T p^i_{j/i} \\ -\theta_{j/i} \end{bmatrix} = \begin{bmatrix} -R(\theta_{j/L})^T p^L_{j/i} \\ \theta_{i/L} - \theta_{j/L} \end{bmatrix}. \tag{8}$$

Rotation matrix is defined as

$$R^i_j = R(\theta_{j/i}) = \begin{bmatrix} \cos(\theta_{j/i}) & -\sin(\theta_{j/i}) \\ \sin(\theta_{j/i}) & \cos(\theta_{j/i}) \end{bmatrix} \tag{9}$$

such that $v^i = R^i_j v^j$ for some two dimensional vector v. Finally, when the i-th agent uses the relative measurements between two of its neighbor agents $j, k \in \mathcal{N}_i$ then the relative measurement model becomes

$$y_{k/j} = h_{k/j} + v_{k/j} \tag{10}$$

where $h_{k/j} = (T^i_j)^{-1} \boxplus T^i_k$ and $\boxplus$ denotes the addition of successive planar coordinate transformations defined as

$$(T^i_j)^{-1} \boxplus T^i_k = \begin{bmatrix} R(\theta_{j/i})^T (p^i_{k/i} - p^i_{j/i}) \\ \theta_{k/i} - \theta_{j/i} \end{bmatrix} = \begin{bmatrix} R(\theta_{j/L})^T (p^L_{k/i} - p^L_{j/i}) \\ \theta_{k/L} - \theta_{j/L} \end{bmatrix}. \tag{11}$$

The Jacobians of each form of these measurement equations are written as

$$H_{j/i} = \frac{\partial h_{j/i}}{\partial x_{j/i}} = \begin{bmatrix} R(\theta_{i/L})^T & 0_{2\times2} & 0_{2\times1} & 0_{2\times1} \\ 0_{1\times2} & 0_{1\times2} & 1 & 0 \end{bmatrix} \tag{12}$$

$$H_{i/j} = \frac{\partial h_{i/j}}{\partial x_{j/i}} = \begin{bmatrix} -R(\theta_{j/L})^T & 0_{2\times2} & -\left.\frac{\partial R^T}{\partial \theta}\right|_{\theta_{j/L}} p^L_{j/i} & 0_{2\times1} \\ 0_{1\times2} & 0_{1\times2} & -1 & 0 \end{bmatrix} \tag{13}$$

$$H_{k/j} = \frac{\partial h_{k/j}}{\partial x_{j/i}} = \begin{bmatrix} -R(\theta_{j/L})^T & 0_{2\times2} & \left.\frac{\partial R^T}{\partial \theta}\right|_{\theta_{j/L}} (p^L_{k/i} - p^L_{j/i}) & 0_{2\times1} \\ 0_{1\times2} & 0_{1\times2} & -1 & 0 \end{bmatrix} \tag{14}$$

$$H_{j/k} = \frac{\partial h_{j/k}}{\partial x_{j/i}} = \begin{bmatrix} R(\theta_{j/L})^T & 0_{2\times2} & 0_{2\times1} & 0_{2\times1} \\ 0_{1\times2} & 0_{1\times2} & 1 & 0 \end{bmatrix} \tag{15}$$

Communication and Augmented Sensing

At each communication step, each agent broadcasts all of the relative measurements that it makes and the associated covariance matrices to its neighbors, according to the communication graph topology $\mathcal{G}_c$ defined earlier. For each edge $(i, j) \in \mathcal{E}_m$, the relative sensing information is defined as $(y_{i/j}, V_{i/j}, (i,j))$, which is a collection of the measurement, covariance, and edge element of the graph. Then, define $\mathcal{M}^i$ to be a set of a relative sensing information for all the direct measurements i-th agent makes. That is $\mathcal{M}^i = \{(y_{i_j}, V_{i/j}, (i,j)) \in \mathbb{R}^3 \times \mathbb{R}^{3\times 3} \times \mathcal{E}_s \mid j \in \mathcal{N}_s^i\}$ where subscript s in $\mathcal{N}_s$ denotes that it is a neighbor in the relative sensing graph $\mathcal{G}_s$. At each communication time step, each agent broadcasts this $\mathcal{M}^i$ to its communication neighbors.

Each agent also collects the information broadcast by its neighbors. First define the list of all of the agents that agent i is aware of from direct sensing or from the sensing of agents with which it communicates. Denote this set of agents as $\bar{\mathcal{V}}_i$. This set is obtained by

$$\bar{\mathcal{V}}_i = \bar{\mathcal{N}}_m^i \bigcup_{j\in\mathcal{N}_c} \bar{\mathcal{N}}_m^j \tag{16}$$

Next, define $\bar{\mathcal{E}}_i$ as a set of all the relative sensing edges collected by the i-th agent. That is

$$\bar{\mathcal{E}}_i = \{(m,n) \in \mathcal{E}_s \mid m \in \bar{\mathcal{N}}_c^i, n \in \mathcal{N}_s^m\} \tag{17}$$

Let $\bar{\mathcal{G}}_i$ denote the graph $\bar{\mathcal{G}}_i = (\bar{\mathcal{V}}_i, \bar{\mathcal{E}}_i)$. In general, this graph may not be connected. Let $\mathcal{V}_i \subseteq \bar{\mathcal{V}}_i$ be the set of agents in the local relative sensing graph for i-th agent, selected based on the relative sensing and communication topologies. One of the later subsections discusses how to select $\mathcal{V}_i$ such that all of the states are observable. Once $\mathcal{V}_i$ is given, one can compute an induced sub-graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ from $\bar{\mathcal{G}}_i = (\bar{\mathcal{V}}_i, \bar{\mathcal{E}}_i)$. This sub-graph is defined as a local relative sensing graph. Note that because $\mathcal{V}_i \subseteq \bar{\mathcal{V}}_i \subseteq \mathcal{V}$ and $\mathcal{E}_i \subseteq \bar{\mathcal{V}}_i \subseteq \mathcal{V}$, $\mathcal{G}_i$ is a sub-graph of $\mathcal{G}_s$.

Recall that the measurement equations are given in three forms, Equations 6, 7, and 10. These equations describe any of the the relative measurements in the $\bar{\mathcal{M}}_i$. The augmented measurement model for the augmented measurement vector can be defined as

$$y_i = h_i(x_i) + v_i \tag{18}$$

where h_i stacks Equations 6, 7, and 10 for all of the edges $(j,k) \in \mathcal{E}_i$ and corresponding measurement Jacobian as

$$H_i = \left.\frac{\partial h_i(x_i)}{\partial x_i}\right|_{x_i}. \tag{19}$$

Note that because all of the measurement models only depends only on one or two agent states at a time, the Jacobian H_i will be sparse in the row direction when there are many agents in $\mathcal{V}_i$.

Using the augmented measurement and the augmented state x_i, the dynamical system to estimate for each agent becomes

$$x_i^+ = f_i(x_i) + w_i \tag{20}$$

$$y_i = h_i(x_i) + v_i \tag{21}$$

where $f_i : \mathbb{R}^{nN_i} \to \mathbb{R}^{nN_i}$ and $h : \mathbb{R}^{nN_i} \to \mathbb{R}^{mM_i}$. The process noise term $w_k \in \mathbb{R}^{mM}$ has Gaussian distribution $w_k \sim \mathcal{N}(0, W_k)$. W_k is a block diagonal with W^{ij}. Let F_i be the Jacobian of the possibly nonlinear dynamics

$$F_i = \frac{\partial f_i}{\partial x_i}. \tag{22}$$

Because each agent's equations of motion are decoupled, this augmented Jacobian is a block diagonal collection of the Jacobians with respect to each agent's dynamics.

Estimation Filter

Once the augmented state dynamics and the augmented measurement are well defined, the dynamical system of Equations 20 and 21 are in the standard form. A nonlinear estimation technique of choice may be used to estimate this system. For the purposes of this paper, an EKF is used but any general nonlinear estimator may be used. Note that F_i and H_i are standard Jacobians linearized around the prior estimate, defined as $F_i = \frac{\partial f_i}{\partial x_i}\Big|_{x_i=\bar{x}_i}$ and $H_i = \frac{\partial h_i}{\partial x_i}\Big|_{x_i=\bar{x}_i}$.

$$\bar{x}_i = f_i(\hat{x}_i) \tag{23}$$

$$\bar{P}_i = F_i P_i F_i^T + W_i \tag{24}$$

$$S_i = H_i \bar{P}_i H_i^T + R_i \tag{25}$$

$$K_i = \bar{P}_i H_i^T S_i^{-1} \tag{26}$$

$$\hat{x}_i^+ = \bar{x}_i + K_i(y_i - h_i(\hat{x}_i)) \tag{27}$$

$$P_i^+ = (I - K_i H_i)\bar{P}_i \tag{28}$$

Remark. *When cycles occur in the local graph, the DPE takes advantage of the constraint from relative pose graph, and joint estimation of the relative poses improve the observability and estimation confidence. On the other hand, if any neighbors are not part of a cycle in the local relative sensing graph, the terms for those neighbors in Local Pose Graph EKF (Algorithm 1) can be decoupled into a separate pose EKF for each agent $j \in \mathcal{N}_i$.*

Observability

The DPE requires each agent to know the set of agents $\mathcal{S}_i$ to be included in the local relative sensing graph. One natural criteria for selecting the set is whether the states of those neighbor agents are observable to the agent. If they are not observable, they are pruned from the measurement set. In particular, we consider that the number of communication exchanges at each time step is limited. Suppose that one communication exchange is allowed and assume there is no memory is used to store the measurements from the previous time histories. In this case, any agent $j \notin \bar{\mathcal{V}}_i$ is not observable. This is straightforward as no relative measurements are functions of those agents' states. $\bar{\mathcal{G}}_i$ is not connected in general. The following lemma is used to determine subset of agents that are observable. For this lemma, we first define $\hat{\mathcal{G}}_i = (\bar{\mathcal{V}}_i, \hat{\mathcal{E}}_i)$ as the undirected graph generated from the directed graph, where $\hat{\mathcal{E}}_i = \bar{\mathcal{E}}_i \cup \{(u,v) \in \bar{\mathcal{V}}_i^2 \mid (v,u) \in \bar{\mathcal{E}}_i\}$.

Lemma 1. *Select a set $\mathcal{V}_i$ such that $j \in \mathcal{V}_i$ if and only if there exists a path between i and j in the undirected version of the graph $\hat{\mathcal{G}}_i$. Then the augmented state vector x_i, defined as a concatenation of $x_{j/i}$ for all $j \in \mathcal{V}_i$, is observable. Furthermore, the measurements required to observe these states are represented as $\mathcal{E}_i$ such that $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ is an induced graph of $\bar{\mathcal{G}}_i = (\bar{\mathcal{V}}_i, \bar{\mathcal{E}}_i)$.*

This means that the j-th agent is observable to i-th agent, if the relative sensing graph available to i-th agent is connected. Then the set of observable agents is defined as $\{j | j \in \bar{\mathcal{V}}_i\}$ that are connected to i-th agent on $\bar{\mathcal{G}}_i$. The proof is as follows.

Proof. Recall the augmented measurement equation h_i and the measurement Jacobian H_i are given by Equation 18 and Equation 19. Define $O_i = [H_i; H_i F_i]$. which is part of an observability matrix constructed from a linearized system. We show that O_i is full rank and therefore the augmented states are observable.

Let N_i be the size of $\mathcal{V}_i$ and M_i be the size of $\mathcal{E}_i$. Without a loss of generality, sort $\mathcal{V}_i$ such that $j = \{1, \ldots, d, d+1, \ldots, N_i\}$ where an edge exists to j $\forall j \in \{1, \ldots, d\}$, $(i, j) \in \mathcal{E}_i$ or $(j, i) \in \mathcal{E}_i$, and an edge does not exist to j $\forall j \in \{d+1, \ldots, N_i\}$, $(i, j) \notin \mathcal{E}_i$ and $(j, i) \notin \mathcal{E}_i$. Consider the state $x_{j/i}$ for j-th agent, $j = \{1, \ldots, d\}$. For these agents, there exists a direct measurement edge (i, j) or (j, i) in $\mathcal{E}_s$, so denote this as m-th measurement in $\mathcal{E}_i$. Consider the block rows corresponding to m-th measurement on H_i, given by either $H_{j/i}$ (Equation 12) or $H_{i/j}$ (Equation 13). Define $O^{mj} = [H_{j/i}; H_{j/i}A_{rel}]$ or $O^{mj} = [H_{i/j}; H_{i/j}A_{rel}]$, depending on whether $y_{j/i}$ or $y_{i/j}$ are available. Among block rows corresponding to the measurement m, the only non-zero block sub-matrix of O_i is ^{mj} the block corresponding to $x_{j/i}$. It is easy to see that O^{mj} has full column rank, and hence O_i has independent columns for the columns corresponding to $x_{j/i}$. Because this is true for arbitrary $j = \{1, \ldots, d\}$, all the first $6d$ columns are linearly independent.

Next, we show that O_i has independent columns for $(6d+1)$-th to $6N_i$-th column, via induction. For each $j \in \{d+1, \ldots, N_i\}$, there is a path from i to j on $\hat{\mathcal{G}}_i$. Let $(\alpha, \beta) \in \hat{\mathcal{E}}_s$ denote an edge along this path, where α is closer to i than β in the path. For the sake of induction, assume that the states of all agents on the path from i to α are observable. $x_{\alpha/i}$ (i.e. columns of O_i corresponding to $x_{\alpha/i}$) are independent. Because $m, n \in \hat{\mathcal{V}}_i$, we know at least one measurement exists between agents α and β, which are $y_{\alpha/\beta}$ or $y_{\beta/\alpha}$. The measurement model is of the form of Equation 10 and its Jacobians $H_{\alpha/\beta}$ and $H_{\beta/\alpha}$ are given by Equations 14 and 15. Define $O^{m\alpha} = [H_{\alpha/\beta}; H_{\alpha/\beta,\alpha}A_{rel}]$ and $O^{m\beta} = [H_{\beta/\alpha}; H_{\beta/\alpha}A_{rel}]$, which are sub-matrices of O_i corresponding to columns of $x_{\alpha,i}$ and $x_{\beta,i}$ respectively. Whether $H_{\beta/\alpha}$ is taken from Equations 14 or 15, it is easy to see that $O^{m\beta}$ has full column rank. Now for agents $j \in \{d+1, \ldots, N_i\}$, the rows of O_i corresponding to m has two non-zero block matrices $O^{m\alpha}$ and $O^{m\beta}$. By the assumption of the induction, the columns of O_i corresponding to $x_{\alpha,i}$ are already independent. In other words, there exist another rows corresponding to the measurement $\tilde{m}$, $O_{\tilde{m}\alpha}$ is full rank and $O_{\tilde{m}\beta} = 0$. Therefore, following matrix has full column rank.

$$\begin{bmatrix} O_{\tilde{m}\alpha} & 0 \\ O_{m\alpha} & O_{m\beta} \end{bmatrix} \tag{29}$$

This is a sub matrix of O_i. Therefore the columns corresponding to all of the agents on the path from i to β are shown to be linearly independent. We know that the first agent in the path is observable as it is part of $1, \ldots, d$ agent. Therefore, by induction, we conclude that the columns of O_i corresponding to the states of all agents along the path, including j, are linearly independent. This is true for arbitrary $j \in d+1, \ldots, N_i$, therefore this concludes that O_i has a full column rank.

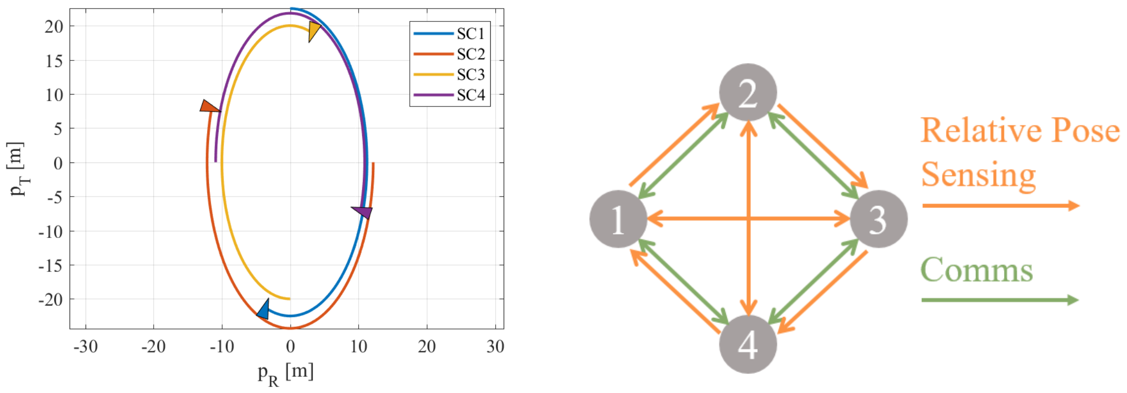
□

SIMULATION RESULTS

The DPE is applied to example scenarios in simulations for formation flying spacecraft. The first example considers four spacecraft in low Earth orbit (LEO) with a 300 km altitude, that are placed in a passive relative orbit (PRO) as shown in Figure 3(a). Figure 3(b) shows the assumed relative sensing and communication network graphs. The relative sensing graph is described by the directed sensor adjacency matrix

$$A_s = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \tag{30}$$

Transnational dynamics of each agent are given by relative orbital mechanics, the HCW equations. The attitude is represented by a double integrator, defined by a constant angular rotation at the same rate as mean motion of the absolute orbit, so that the positive x axis of the each spacecraft's body frame points towards, but not necessarily aligned to, the center of the formation. The relative measurements are obtained as described by Equation 6. The measurement noise for the position is assumed to have a larger uncertainty in the radial direction (same as relative position vector from observer to target) than that in a transverse direction (perpendicular to the radial direction). The simulation is run for 3,000 sec, which is approximately half of an orbit.

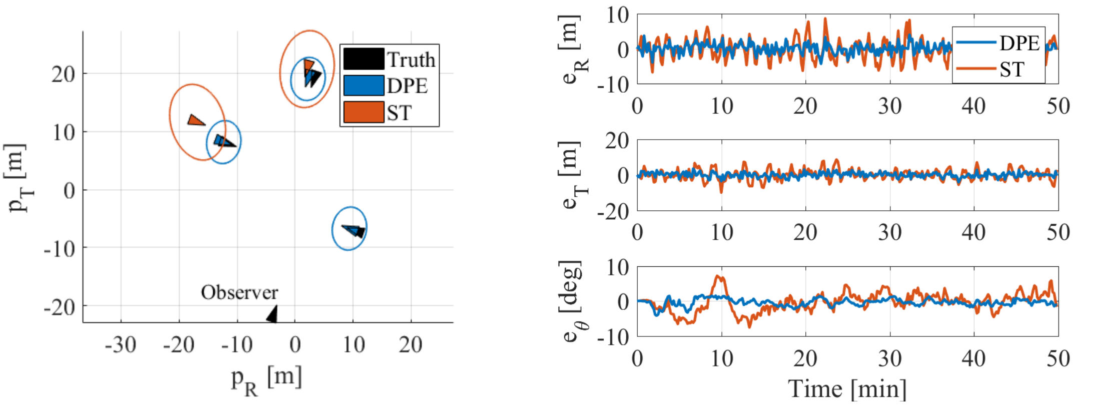


(a) PRO Trajectory

(b) Sensor and Communication Graphs

Figure 3. Example scenario considered in the simulation

As a point of comparison, we also implement a single-target EKF, denoted ST. The ST only uses the direct measurement of one target that the observer is estimating. This is equivalent to the best estimate obtainable when the agents are not cooperating. Each agent may run multiple copies of the ST, one for each target that it observes. The performance of the DPE and the ST are compared in Figures 4(a) and 4(b). In Figure 4(a), the black triangles denote the absolute poses of all of the spacecraft. The blue and orange triangles denote the relative pose estimates made by spacecraft 1 (bottom), which is the observer spacecraft. A position error ellipse corresponding to each position estimate is plotted based on the position covariance. Figure 4(a) shows that the position error covariance for the DPE is much smaller than that of the ST. Especially the uncertainty in the transverse direction is reduced significantly. This improvement is expected, as the DPE estimates multiple relative poses of the neighbor spacecraft and utilizes the structure of the relative pose network over its local graph. Also, the DPE can estimate the states for all three neighboring spacecraft, even though there are only two direct relative measurements available to spacecraft 1. Figure 4(b) shows the estimation error of spacecraft 2 as seen by the spacecraft 1 for both the DPE and the ST. The DPE estimation error is smaller than that of the ST for both position and rotation consistently.



(a) Rel. pose estimate of DPE and ST w.r.t. spacecraft 1 (b) Estimation errors of spacecraft 2 w.r.t. spacecraft 1

Figure 4. Four spacecraft simulation results for the DPE (blue) and ST (orange), with the covariances represented as ellipses and the estimates represented as arrows

The second example considers sixty agents in LEO, where additional spacecraft are placed on PRO trajectories at four radii. Each agent in the outer formation ring observes one neighbor spacecraft in the same ring in clockwise direction and one closest agent in the inner ring. This relative sensing topology is given such that it represents the camera field of view and range constraints. Each agent communicates with one closest neighbors from inner ring, two closest neighbors from same ring, and two closest neighbors from the outer ring. Simulation results are shown in Figures 5(a) and 5(b). Figure 5(a) shows that the DPE can estimate the states of five of its neighbor spacecraft, even though it directly measures the relative poses of only two of them. In Figure 5(b), the statistics of the total estimation error of the DPE and the ST are studied. For each sensing edge in the sensing graph $\mathcal{G}_s$, $i \in \{1, \ldots, m\}$, define the position estimation error as

$$\delta p_i(k) = \hat{p}_i(k) - p_i(k) \tag{31}$$

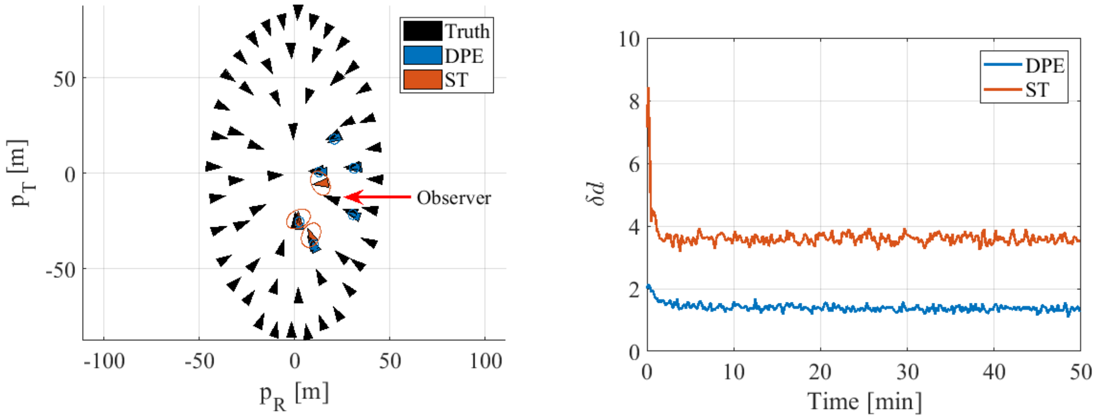
where $\hat{p}_i(k)$ and $p_i(k)$ are the relative pose estimate (from either the DPE or the ST) and the true state at time step k. The averaged error distance is defined as

$$\delta d(k) = \frac{1}{m} \sum_{i=1}^{m} ||\delta p_i(k)||_2. \tag{32}$$

The $\delta d(k)$ is computed for the DPE and the ST are plotted in the Figure 5(b). Figure 5(b) shows that the position estimation error of the DPE is smaller than that of the ST on average.

EXPERIMENTAL RESULTS

The DPE algorithm is also verified in robotic experiments on-board the Caltech's robotic spacecraft simulators, the Multi-Spacecraft Testbed for Autonomy Research (M-STAR).[5] Each spacecraft is equipped with one monocular camera with a high field-of-view (FOV) lens and visual markers (ArUco markers[18]) on each side of the spacecraft simulator as seen in Figure 6. Using the images from the monocular camera on-board each spacecraft, a standard computer vision algorithm[18] combined with additional bias calibration is used to detect and estimate the full pose of the markers, as shown in Figure 7. The spacecraft pose is computed from the pose of the markers by transforming



(a) Rel. pose estimate of DPE and ST w.r.t. spacecraft 1 (b) Averaged error distance for the DPE and the ST

Figure 5. Error distance average for the DPE and the ST

the pose from the marker to the spacecraft body frame and filtering together estimates from multiple markers on-board when multiple markers are visible. The computer vision processing is not discussed in detail since it is not relevant to the contributions of this work. Another vision-based relative pose estimation algorithm (i.e.[19]) can be substituted in place of the ArUco marker based estimate. Two types of experiments were performed, one where the spacecraft are stationary and another where the spacecraft are in motion, following PRO trajectories.

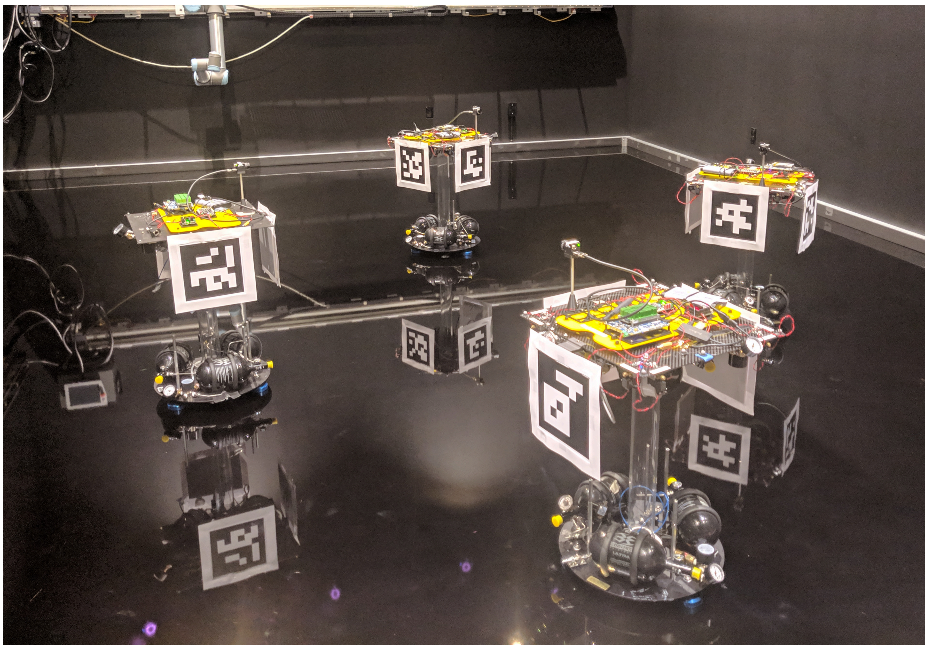


Figure 6. M-STAR with ArUco markers in test environment

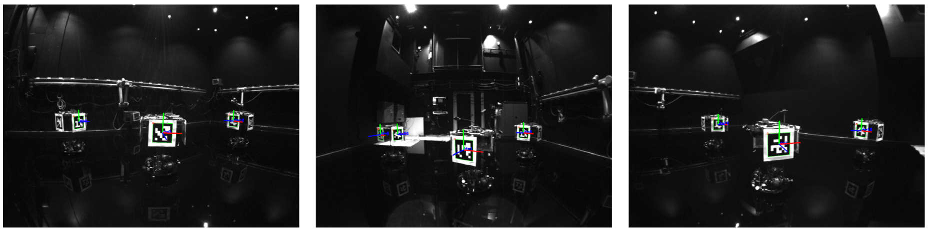
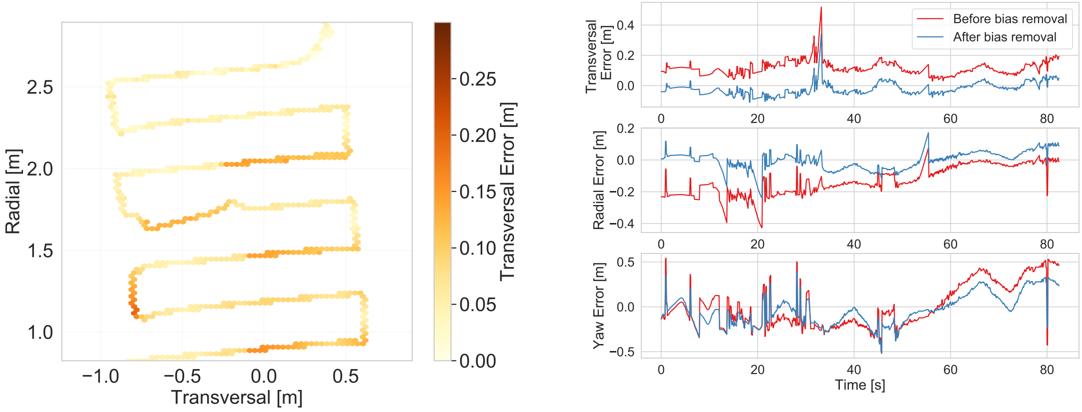


Figure 7. Detected markers as seen from the cameras on-board the spacecraft simulators

Pose Bias Calibration

In the experiments, a monocular camera and ArUco markers are used to extract the relative pose between the camera and the markers. The intrinsic parameters for each camera were estimated using an open-source algorithm.[20] Applying appropriate transformations, one can obtain relative pose estimates between the pair of spacecraft. We call this the relative pose measurement. When the measurement is compared against the ground truth obtained from the motion capture system, the data showed error bias correlated to its relative pose. To achieve accurate pose measurements, we calibrate these error biases with respect to ground truth obtained from the motion capture system.

To determine the biases, pose measurements of the spacecraft simulator are collected from both the camera and the motion capture system. One spacecraft simulator is placed stationary, while the other is moved to determine how the bias changes with distance and placement in the camera field of view. To sample measurements broadly from the possible relative pose space, the spacecraft is moved in a serpentine trajectory (Figure 8(a)). The collected bias estimates are fitted to a multi-dimensional quadratic curve. This bias model was then validated on an independent set of experimental relative pose data. Figure 8(b) shows the error between the actual pose and the estimated pose before and after the model bias correction was applied on the test data. The validated bias model is then used as an input to the DPE algorithm in following experiments.



(a) Serpentine trajectory error before bias removal

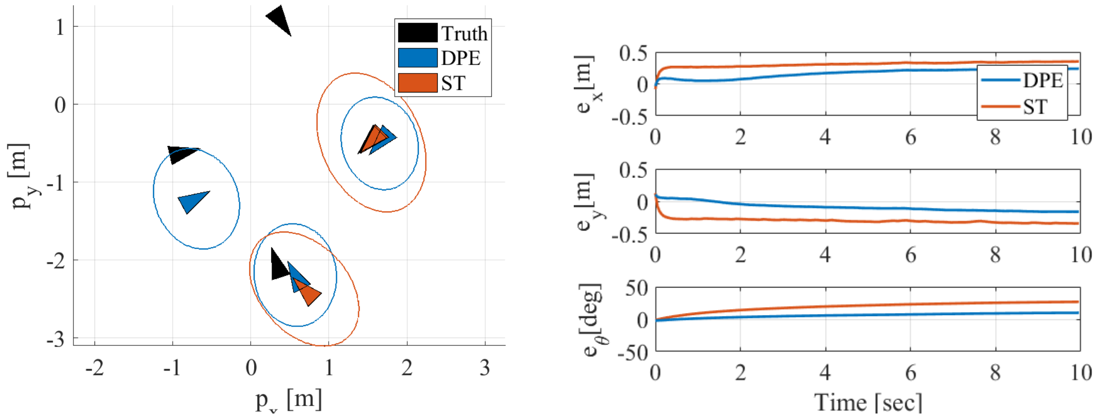
(b) ArUco pose estimate improved by bias extraction

Figure 8. Vision-Based Pose Measurement Bias Analysis

Stationary Experiment

For the stationary experiment, four 3-DOF spacecraft simulators are placed in a stationary formation as shown in Figure 6. This experiment is a step towards the full experiment with the HCW dynamics, discussed in the following subsection. The ground truth for the experiments was obtained from a motion capture system which provides sub-millimeter-level accuracy. The sensor and the communication graphs follow the same scenario as the simulations shown in Figure 3(b). The relative measurements are obtained in a method explained in the previous paragraph. The covariance of the pose estimate was assumed to be a quadratic function of the range.

The experiment results are shown in Figure 9(a) and Figure 9(b). Spacecraft 1 (top) in Figure 9(a) directly observes the poses of spacecraft 2 (right) and 3 (bottom) using the on-board camera. Interestingly, the relative pose estimate for the ST between spacecraft 2 and 1 is slightly better than that of the DPE. Experimental data showed that the vision-based pose measurements had some error that is smaller for closer targets. Because the DPE uses all the available measurements of neighbors to jointly estimate the multiple targets, some measurements that have large error also affect the estimation of spacecraft 2. Since the ST estimates spacecraft 2 only using single measurement that has a small bias, the ST estimate shows smaller error. On the other hand, the ST estimate for spacecraft 3 has a larger error than that of the DPE because the direct measurement from spacecraft 1 of spacecraft 4 is not as accurate at further distances. However, the DPE algorithm is able to decrease the error by fusing additional measurements. Also, the DPE algorithm can still provide an estimate the relative pose of spacecraft 4 (left). In Figure 9(b), the relative state estimation errors of spacecraft 3 as seen by spacecraft 1 show that the DPE outperforms the ST.



(a) Rel. pose estimate w.r.t. spacecraft 1

(b) Rel. pose errors of spacecraft 3 w.r.t. spacecraft 1

Figure 9. Static Experiment Results

Dynamic Experiment

The second experiment considers a formation flying scenario in LEO. Four spacecraft simulators are placed in PRO's, where three spacecraft move around a stationary spacecraft located at the center. Each spacecraft follows the pre-defined position and attitude trajectories using the ground truth given by the motion capture system and on-board thrusters for the control. The phase angles of spacecraft are offset by 120 deg from each other. For the attitude motion, the spacecraft follow a constant slew rate of negative of mean motion, so that the attitude makes one rotation with one

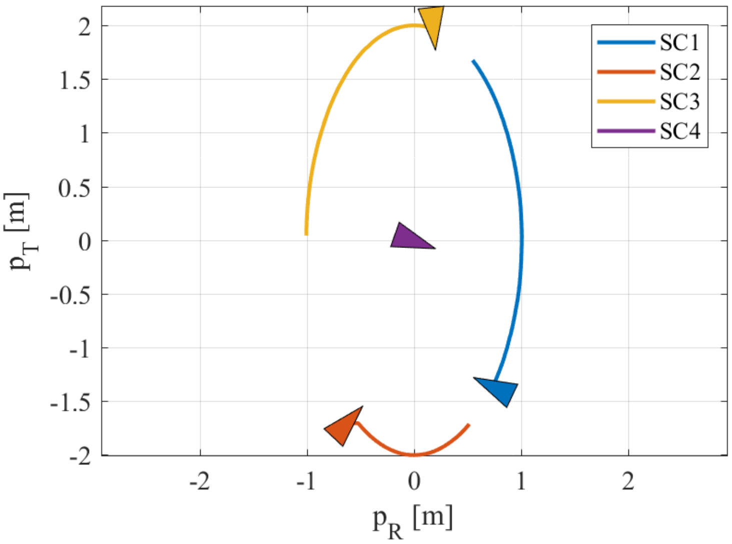


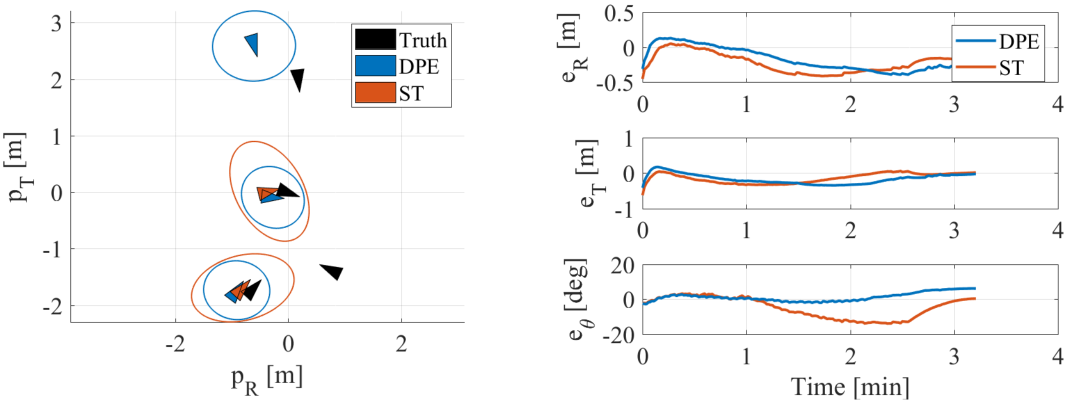
Figure 10. Ground truth trajectories in the dynamic experiment.

orbit. The initial yaw angle for each spacecraft is selected such that when the spacecraft is at the principal axes of the ellipse, the camera optical axis is offset to left by 15 degree. This way, the center spacecraft and the neighbor spacecraft in the clockwise direction are always in the FOV during the entire trajectory. The center spacecraft does not carry a sensor and hence does not make a direct observation of the other spacecraft. The communication network is assumed to be a complete graph. The time scale of the relative orbit trajectories and the time step assumed in the discrete-time state transition matrix of the estimators are adjusted respectively such that the experiment is completed in a shorter time span than an actual relative orbit in LEO. This is done to satisfy experiment constraints like air supply for air-bearing pucks. Figure 10 shows this formation and the ground truth trajectories of the spacecraft during the dynamic experiment.

The estimation algorithms for dynamic experiment are shown in Figure 11(a) and Figure 11(b). Similar to the static experiment, the DPE successfully tracks the relative states in the dynamic experiment.

CONCLUSION

We present a distributed relative pose estimation (DPE) algorithm for multi-agent localization of formation flying spacecraft. The algorithm architecture considers both limited relative sensing and communication graphs, where the communication augments the local sensing capability of each agent, creating a network of relative measurement called local relative sensing graph. The number of states each agent estimates does not grow with the number of agents, as long as its local relative sensing graph does not increase. A rule is presented and justified for selecting the set of agents to be included in the local relative sensing graph such that all agents in the graph are observable. The proposed algorithm is verified with simulations and robotic experiments. Simulations illustrate that the DPE performs well for formation flying mission examples. The DPE is verified in robotic experiments that use state-of-the-art spacecraft simulators.



(a) Rel. pose estimate w.r.t. spacecraft 1 (bottom right)

(b) Rel. pose errors of spacecraft 2 w.r.t. spacecraft 1

Figure 11. Dynamic Experiment Results

ACKNOWLEDGEMENT

This research was supported in part by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The work of Kai Matsuka was supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 1745301. The work of Rebecca Foust was supported by a NASA Space Technology Research Fellowship (Grant No. NNX15AP48H). Also, we would like to thank Aaron Feldman and Jennifer Sun for their support on the robotic experiments.

REFERENCES

[1] F. Y. Hadaegh, S.-J. Chung, and H. M. Manohara, “On development of 100-gram-class spacecraft for swarm applications,” *IEEE Systems Journal*, Vol. 10, No. 2, 2016, pp. 673–684.

[2] S.-J. Chung and F. Y. Hadaegh, “Swarms of Femtosats for Synthetic Aperture Applications,” *Proceedings of the Fourth International Conference on Spacecraft Formation Flying Missions & Technologies*, St-Hubert, Quebec, May 2011.

[3] O. Brown, P. Eremenko, and P. Collopy, “Value-centric design methodologies for fractionated spacecraft: Progress summary from Phase I of the DARPA System F6 Program,” *AIAA Space 2009 Conference & Exposition*, 2009, p. 6540.

[4] A. P. Dani, S.-J. Chung, and S. Hutchinson, “Observer design for stochastic nonlinear systems via contraction-based incremental stability,” *IEEE Transactions on Automatic Control*, Vol. 60, No. 3, 2014, pp. 700–714.

[5] Y. K. Nakka, R. C. Foust, E. S. Lupu, D. B. Elliott, I. S. Crowell, S.-J. Chung, and F. Y. Hadaegh, “Six degree-of-freedom spacecraft dynamics simulator for formation control research,” *2018 AAS/AIAA Astrodynamics Specialist Conference*, AIAA, 2018.

[6] R. Olfati-Saber, “Kalman-consensus filter: Optimality, stability, and performance,” *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, IEEE, 2009, pp. 7036–7042.

[7] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, “Information weighted consensus,” *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, IEEE, 2012, pp. 2732–2737.

[8] S. Bandyopadhyay and S.-J. Chung, “Distributed Bayesian filtering using logarithmic opinion pool for dynamic sensor networks,” *Automatica*, Vol. 97, 2018, pp. 7–17.

[9] B. Açıkmeşe, D. P. Scharf, J. M. Carson III, and F. Y. Hadaegh, “Distributed estimation for spacecraft formations over time-varying sensing topologies,” *IFAC Proceedings Volumes*, Vol. 41, No. 2, 2008, pp. 2123–2130.

[10] R. Aragues, L. Carlone, G. Calafiore, and C. Sagues, "Multi-agent localization from noisy relative pose measurements," *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 364–369.

[11] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal processing magazine*, Vol. 22, No. 4, 2005, pp. 54–69.

[12] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on signal processing*, Vol. 51, No. 8, 2003, pp. 2137–2148.

[13] L. Blackmore and F. Hadaegh, "Necessary and sufficient conditions for attitude estimation in fractionated spacecraft systems," *AIAA Guidance, Navigation, and Control Conference*, 2009, p. 6095.

[14] W. Kang, I. M. Ross, K. Pham, and Q. Gong, "Autonomous observability of networked multisatellite systems," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, 2009, pp. 869–877.

[15] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Transactions on Power Systems*, Vol. 28, No. 2, 2012, pp. 1617–1626.

[16] D. E. Marelli and M. Fu, "Distributed weighted least-squares estimation with fast convergence for large-scale systems," *Automatica*, Vol. 51, 2015, pp. 27–39.

[17] D. Morgan, S.-J. Chung, L. Blackmore, B. Acikmese, D. Bayard, and F. Y. Hadaegh, "Swarm-keeping strategies for spacecraft under J2 and atmospheric drag perturbations," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 5, 2012, pp. 1492–1506.

[18] "ArUco Marker Detection," `https://docs.opencv.org/4.0.0/d9/d6a/group__aruco.html`.

[19] V. Capuano, S. R. Alimo, A. Q. Ho, and S.-J. Chung, "Robust Features Extraction for On-board Monocular-based Spacecraft Pose Acquisition," *AIAA Scitech 2019 Forum*, 2019, p. 2005.

[20] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4304–4311.