Abstract—GPS navigation in urban environments is prone to error sources such as multipath and signal blockage. However, if we consider several agents, it is highly likely that some agents have better localization capabilities due to different views of the sky, heterogeneity in sensors etc. Collaborative localization (CL) is a way to aid navigation in a multi-agent system. CL algorithms face challenges such as scalability, robustness to noisy sensor data and single point of failure, and operability despite limited inter-agent communication. In this paper, we present a decentralized collaborative localization algorithm which is asynchronous, applicable to sparsely communicating networks, and has minimal information exchange. Moreover, the proposed algorithm takes advantage of the variable visibility of the sky for different agents. We propose a methodology for relaying satellite information between agents to augment the set of visible satellites on each agent with virtual satellites, thereby providing more constraint equations to each agent. The methodology is based on coupling of agents’ GPS measurements with range-only sensors and is applicable to multi-agent systems with these modalities. The proposed method is validated on real world dataset involving an aerial vehicle, ground agents, and several range-only sensors.

I. INTRODUCTION

There has been an increasing number of applications of UAVs in urban environments for tasks such as delivery, transport, photography, and search and rescue, yet GPS-based UAV navigation in urban environments is affected by error sources such as multipath and signal blockage. The United States Federal Aviation Administration (FAA) recorded over 626,000 registered unmanned aircraft systems as of December 2016 and forecasted an annual sale of 5.2 million commercial and hobbyist UAVs in the United States by 2020 [1]. A large number of UAVs available in proximity of each other in the near future opens avenues for cooperation among agents to achieve better localization solutions. Scalability, robustness, low communication overhead and distributed computation are desired characteristics of a collaborative localization framework. A decentralized framework with no fusion center is robust to single point of failure and allows computation to be distributed among agents leading to low communication bandwidth (compared to a fusion center) and scalability (number of agents, and geographical) in the system.

The main contribution of this paper is a decentralized collaborative localization algorithm with deep coupling of agents’ GPS measurements with range-only sensors. In our algorithm, each agent iteratively updates the belief of its own position and cross-correlation information with the estimates it obtains from its teammates. When two agents communicate, they exchange GPS pseudoranges, and inter-agent range information along with their beliefs and cross-correlations. At each interaction, we identify the satellites visible to only the transmitting agent. We, then, estimate pseudoranges to these satellites from the receiving agent using position estimates of the agents and ranging information. We update these pseudorange estimates by a correction term obtained through linearization of the ranging equation every time we perform a prediction of the belief. These pseudorange estimates then augment the set of visible satellites to the receiving agent and are subsequently used in the navigation solution. We experimentally validate our proposed decentralized collaborative localization algorithm in urban settings.

A. Related Work

Early developments in CL were initiated by Kurazume et al. [2] wherein they showed lower accumulated positioning error than dead reckoning. Since then, CL has been exten-
sively researched and numerous studies and developments have followed. Broadly, the approaches may be classified on the basis of system architecture as centralized ([3],[4],[5],[6]) and decentralized ([7],[8],[9],[10]) algorithms. The centralized approaches rely on a fusion center or multiple fusion centers [11] to which all agents communicate information. While centralized algorithms are capable of producing the optimal fusion result, these approaches are susceptible to points of failure and impose high communication bandwidth on the fusion centers. Such frameworks limit scalability in terms of the number of robots as well.

Decentralized CL approaches on the other hand distribute the computation among the agents and prevent a single point of failure. Roumeliotis et al. proposed in [12] a decentralized CL framework showing that the equations of a centralized Kalman estimator can be distributed into reduced dimension CL framework showing that the equations of a centralized algorithms are capable of producing the optimal fusion result, these approaches are susceptible to points of failure and impose high communication bandwidth on the fusion centers. Such frameworks limit scalability in terms of the number of robots as well.

Another category of distributed CL frameworks rely on methods that approximate the centralized approaches. Approximations allow these frameworks to operate within constraints such as sparsely connected networks ([13],[10]), lossy and asynchronous communication [14], as well as lower communication and processing overhead ([7],[10],[15]). However, such approaches require careful consideration to keep track of the cross-correlation terms to ensure consistency of individual estimates within the system. Information reuse or double counting is a problem pertinent to these methods. If an approach treats the neighbors’ poses as deterministic estimates and ignores the cross-correlation terms, it may lead to overly confident estimates that diverge from true agents’ positions ([16],[17]). On the other hand, if we assume no knowledge of the interdependencies but assume the estimates to be maximally correlated, employing methods such as Covariance Intersection Filter (CIF) [18] lead to conservative estimates. A variation of CIF called the Split-CIF [19] separates the covariance into dependent and independent components but leads to overly pessimistic estimates as well. Although CIF and Split-CIF guarantee consistent estimates, these methods require the agents to be able to measure the relative pose of their neighbors. This makes these methods unusable if the agents possess any other inter-agent sensor modalities such as range-only or range and bearing sensors.

Prorok et al. proposed a low-cost decentralized CL algorithm based on particle filter in [14]. Their framework demonstrated capability to handle asynchronous and lossy communications. The results were, however, demonstrated on a low dimensionality case (2D robots) with range and bearing sensor modalities and communication overhead was dependent on the number of particles and particle clusters. Specifically, CL approaches that directly fuse GPS satellite ranging with other sensor modalities have been proposed in [16],[20],[21]. Hybrid Cooperative Particle Filter (HC-PF) proposed in [16], assumes no cross-correlations between agents and instead models the uncertainties in agents positions as additional sensor noise. Thus, the problem of double counting is not explicitly handled and the method is prone to overly-optimistic estimates.

The proposed methodology, on the other hand, augments a decentralized CL framework [13] which is flexible to different measurement modalities and explicitly handles the problem of double counting by directly approximating from [12]. This flexibility allows tightly coupling GPS satellite ranging measurements with inter-agent ranging sensor to obtain virtual satellites for neighbouring agents. Further, the underlying algorithm allows for an $O(1)$ communication cost per agent when a relative measurement is obtained.

II. PROBLEM FORMULATION

We consider a network (denoted by $O$) of $N$ agents. For each agent $i \in O$, $S_i^t$ denotes the set of GPS satellites visible to the agent at time $t$. Each agent $i \in O$ has a set of neighboring agents (denoted by $M_i^t$) to which it can communicate some information at time $t$. The state of an agent $i$ at time $t$ is denoted by $X_i^t$ and its dimension is contingent on the motion model used but it always contains $[p_i^t, b_i^t]$. Here $p_i^t = [x_i, y_i, z_i]^T$ is the 3-D position of the agent at time $t$ expressed in the earth-centered earth-fixed (ECEF) frame, and $b_i^t = c \cdot \delta t_i$ where $\delta t_i$ is the GPS receiver clock bias for agent $i$ at time $t$ and $c$ is the speed of light. For an agent $i$, $\Sigma_{ii}^t$ denotes the covariance matrix corresponding to its state $X_i^t$. Together they constitute the belief $b_i^t = \{X_i^t, \Sigma_{ii}^t\}$ of agent $i$ at time $t$. The cross covariance matrix between any two agents $i, j \in O$ at time $t$ is denoted by $\Sigma_{ij}^t$. Initially, the states of the agents may be assumed to be uncorrelated i.e. $\Sigma_{ij}^t = 0 \forall i \neq j$. However, inter-agent interactions lead to non-zero $\Sigma_{ij}$ matrices at subsequent times.

Each agent $i \in O$ performs the following sensor measurements:

(i) **Satellite pseudorange measurement:** For each satellite $s \in S_i^t$, a pseudorange measurement $\rho_{si}^t$ at time $t$ is obtained. This is modeled as

\[
\rho_{si}^t = \|p_i^t - p_s^t\| + b_i^t + \epsilon_{si}
\]

where $\|\|$ denotes euclidean distance, $p_s^t$ is the 3-D coordinate of satellite $s$ in ECEF frame, and $\epsilon_{si}$ is noise. Further, this noise is assumed independent and Gaussian distributed as

\[
\epsilon_{si}^t \sim \mathcal{N}(0, \epsilon_{si}^2)
\]

Here, $\epsilon_{si}^2$ can be obtained as a function of satellite signal to noise ratio, elevation etc. [22].

(ii) **Inter-agent ranging measurement:** Agents possess ranging sensors to estimate distance to neighboring agents with known correspondences. This is modeled as

\[
r_{ij}^t = \|p_i^t - p_j^t\| + \phi_{ij}
\]

where $\phi_{ij}$ is white Gaussian noise and is modeled as

\[
\phi_{ij} \sim \mathcal{N}(0, \phi_{ij}^2)
\]
The term $\psi_{ij}^2$ can be empirically estimated. Note that the clock bias term is eliminated using methods such as round trip time of arrival.

The problem, then, is to obtain a consistent estimate of $\text{bel}_i^t$ and cross covariance matrices $\Sigma_{ij}^t$ for each agent $i \in \mathcal{O}$ at all times $t$ given the pseudorange $\rho_{ji}^t$, inter-agent ranging measurements $r_{ij}^t$, and satellite positions $p_i^t$. Additionally, agents are allowed to communicate some information to a finite set of neighbors $\mathcal{M}_i^t$. As an example, in the proposed algorithm, agents communicate only when relative measurements are obtained, and this communication is limited to only the agents involved in the measurement (i.e. only 2 agents at a time). When two agents $i, j$ interact, they exchange their beliefs ($\text{bel}_i, \text{bel}_j$), decompositions of $\Sigma_{ij}$ matrix (discussed in section III-A), and the sets of visible satellites ($S_i, S_j$).

III. METHODOLOGY

The methodology is derived from the distributed form of a centralized Extended Kalman Filter (EKF) as proposed by Roumeliotis et al. in [12]. Specifically, we follow the framework (called DCL henceforth) proposed by Luft et al. in [13] and reproduced in section III-A. The key contribution of this paper is to add to the DCL framework an algorithm for augmenting the set of satellites visible to an agent with satellites visible to its neighbors. We call this Hybrid-DCL.

A. DCL framework

DCL is a distributed approximation to a general centralized EKF. It follows the same general structure of predicting and updating as an EKF, albeit in a distributed fashion. Each agent maintains an estimate of its belief $\text{bel}_i^t$. Further, $\Sigma_{ij}^t$ is decomposed as

$$\Sigma_{ij}^t = \sigma_{ij}^t (\sigma_{ji}^t)^T$$

(5)

where this can be any possible decomposition (detailed explanation in [12]). This decomposition allows $\Sigma_{ij}^t$ to be distributed among interacting agents $i, j$, and each stores $\sigma_{ij}^t$ and $\sigma_{ji}^t$ respectively. These decomposed cross covariance terms are iteratively updated along with the belief by each agent as shown below. Each agent $i \in \mathcal{O}$ performs the following steps.

1) Prediction step: Each agent’s motion is independent and governed by a generic motion model with white gaussian noise. The motion model may be different for different agents. Following equations govern the prediction step for agent $i$.

$$X_{ij}^{t+1} = f(X_{ij}^t, u_i^t)$$

(6)

$$\Sigma_{ij}^{t+1} = F_i^t \Sigma_{ij}^t (F_i^t)^T + V_i^t R_i^t (V_i^t)^T$$

(7)

$$\sigma_{ij}^{t+1} = F_i^t \sigma_{ij}^t$$

(8)

for all $j \neq i$ with linearizations $F_i^t = \frac{\partial f(X,u)}{\partial X}(X_i^t, u_i^t)$ and $V_i^t = \frac{\partial f(X,u)}{\partial u}(X_i^t, u_i^t)$, control input $u_i^t$, motion model $f(\cdot)$, and control noise covariance matrix $R_i^t$. As pointed by Luft et al. in [13], separately updating $\sigma_{ij}^{t+1}$, $\sigma_{ji}^{t+1}$ in agents $i, j$ respectively can be correctly combined to obtain $\Sigma_{ij}^{t+1}$ simply by equation (5).

2) Private update step: Measurements not dependent on other agents may be made by each agent. Such measurements include GPS, landmark observation etc. When observations are made by such sensors, the following equations govern this step for agent $i$.

$$X_{ij}^{t+1} = X_{ij}^t + K_i^t [z_i^t - h(X_{ij}^t)]$$

(9)

$$\Sigma_{ij}^{t+1} = (I - K_i^t H_i^t) \Sigma_{ij}^t$$

(10)

$$\sigma_{ij}^{t+1} = (I - K_i^t H_i^t) \sigma_{ij}^t$$

(11)

with identity matrix of appropriate dimension $I$, sensor observation $z_i^t$, sensor observation model $h(X)$, linearization $H_i^t = \frac{\partial h_i(X)}{\partial X}(X_i^t)$, Kalman gain $K_i^t = \Sigma_{ij}^t (H_i^t)^T S_i^{-1}$, residual covariance $S = H_i^t \Sigma_{ij}^t (H_i^t)^T + Q_i^t$, and the sensor noise $Q_i^t$.

3) Relative update step: When an agent $i$ performs a relative measurement to agent $j$, agents $i, j$ exchange $\text{bel}_i, \text{bel}_j, \sigma_{ij}, \sigma_{ji}$ and perform an update according to the following equations

$$\Sigma_{ij}^{t+1} = \sigma_{ij}^{t+1} \left( \frac{\sigma_{ij}^{t+1}}{\sigma_{ji}^{t+1}} \right)^T$$

(12)

$$X_{ij}^{t+1} = X_{ij}^t + K_i^t \left[ r_{ij}^t - g(X_{ij}^t, X_j^t) \right]$$

(13)

$$\Sigma_{ij}^{t+1} = \left( I - K_i^t G_i^t \right) \Sigma_{ij}^t$$

(14)

$$\Sigma_{ij}^{t+1} = \left( I - K_i^t G_i^t \right) \Sigma_{ij}^t - K_i^t G_i^t \Sigma_{ji}^t$$

(15)

$$\sigma_{ij}^{t+1} = \sigma_{ij}^{t+1}$$

(16)

$$\sigma_{ji}^{t+1} = I$$

(17)

$$\sigma_{ik}^{t+1} = \sigma_{ik}^{t+1} \left( \frac{\sigma_{ik}^{t+1}}{\sigma_{jk}^{t+1}} \right)^{-1}$$

(18)

for all $k \in \mathcal{O} \setminus \{i, j\}$, with relative sensor observation $r_{ij}^t$, relative measurement model $g(X_i, X_j)$, linearizations $G_i^t = \frac{\partial g_i(X, X_j)}{\partial X_i}(X_i^t, X_j^t)$ and $G_j^t = \frac{\partial g_i(X, X_j)}{\partial X_j}(X_i^t, X_j^t)$, Kalman gain

$$K_i^t = \left( K_i^t \right) = \left( \Sigma_{ij}^t G_i^t + \Sigma_{ji}^t G_j^t \right) \left( F_i^t G_i^t + \Sigma_{ji}^t G_j^t \right)^{-1}$$

(19)

and residual covariance

$$S = [G_i^t G_j^T] \left( \Sigma_{ij}^t \Sigma_{ji}^t \right) \left( G_i^T G_j \right)^{-1}$$

Note that the measurement model in the relative update step is assumed to be generic and requires no special structure. Also, we can easily see that communication is restricted to only when a relative measurement is obtained and that too between only the agents involved in the measurement.

B. Hybrid-DCL Algorithm

The proposed algorithm is a variation of the DCL framework and described in algorithm 1. Steps 5, 7, 18, 27 are original contributions of this paper, while the remaining derives from algorithm 1 in [13]. We now describe the algorithm in detail.

Each agent’s belief is initialized at $t = 0$ as a guess, and we assume these initial beliefs to be uncorrelated i.e.
Algorithm 1  H-DCL procedure for each agent i

Require: bel_i(t), \{\sigma_{ij}^t\}_{\forall j \neq i}

1: **Initialisation:** at \(t = 0\), choose \(\text{bel}_i(0)\) as feasible, and \(\sigma_{ij}^0 = 0 \forall j \neq i\)
2: **for** each time step \(t\) do
3: \(\mathcal{V}_i^{t+1} \leftarrow \{\phi\}\)
4: \(\text{bel}_t^i, \{\sigma_{ij}^t\}_{\forall j \neq i} \leftarrow \text{predictStep}(\text{bel}_i^t, \{\sigma_{ij}^t\}_{\forall j \neq i})\)
5: \(\mathcal{V}_i^t \leftarrow \text{correctPseudorange}(\text{bel}_i^t, \text{bel}_i^t, \mathcal{V}_i^t)\)
6: **if** obtain pseudorange measurements \(r_{si}^t\) **then**
7: \(S_i^t \leftarrow \text{storeSats}(\text{all } s \in S_i^t)\)
8: \(\text{bel}_i^t \leftarrow \text{privUpdate}(\text{bel}_i^t, S_i^t, \mathcal{V}_i^t)\)
9: \(\{\sigma_{ij}^t\}_{\forall j \neq i} \leftarrow \text{privCov}(\{\sigma_{ij}^t\}_{\forall j \neq i})\)
10: **end if**
11: **if** obtain relative measurement \(r_{ij}^t\) to agent \(j\) **then**
12: send to agent \(j\): \(\text{bel}_i^t, \{\sigma_{ij}^t\}_{\forall j \neq i}, S_i^t\)
13: receive from agent \(j\): \(\text{bel}_j^t, \{\sigma_{ij}^t\}_{\forall j \neq i}, S_j^t\)
14: \(\Sigma_{ij}^t \leftarrow \Sigma_{ij}^t \cdot (\Sigma_{ij}^t)^T\)
15: \(\text{bel}_i^t, \Sigma_{ij}^t \leftarrow \text{reUpdate}^a(\text{bel}_i^t, \Sigma_{ij}^t, r_{ij}^t)\)
16: \(\{\sigma_{ik}^t\}_{\forall k \in \mathcal{O} \cup \{i\}} \leftarrow \text{reCov}^a(\{\sigma_{ik}^t\}_{\forall k \in \mathcal{O} \cup \{i\}})\)
17: \(\sigma_{ij}^t \leftarrow \Sigma_{ij}^t\)
18: \(\mathcal{V}_i^{t+1} \leftarrow \text{compareSats}((\text{bel}_i^t, S_i^t), \mathcal{V}_i^t)\)
19: **end if**
20: **if** obtain relative measurement \(r_{ij}^t\) from agent \(j\) **then**
21: receive from agent \(j\): \(\text{bel}_i^t, \{\sigma_{ij}^t\}_{\forall j \neq i}, S_i^t, r_{ji}^t\)
22: send to agent \(j\): \(\text{bel}_i^t, \{\sigma_{ij}^t\}_{\forall j \neq i}, S_i^t\)
23: \(\Sigma_{ij}^t \leftarrow \Sigma_{ij}^t \cdot (\Sigma_{ij}^t)^T\)
24: \(\text{bel}_i^t, \Sigma_{ij}^t \leftarrow \text{reUpdate}^b(\text{bel}_i^t, \Sigma_{ij}^t, r_{ij}^t)\)
25: \(\{\sigma_{ik}^t\}_{\forall k \in \mathcal{O} \cup \{i\}} \leftarrow \text{reCov}^b(\{\sigma_{ik}^t\}_{\forall k \in \mathcal{O} \cup \{i\}})\)
26: \(\sigma_{ij}^t \leftarrow \Sigma_{ij}^t\)
27: \(\mathcal{V}_i^{t+1} \leftarrow \text{compareSats}((\text{bel}_i^t, S_i^t), \mathcal{V}_i^t)\)
28: **end if**
29: \(\text{bel}_i^{t+1} \leftarrow \text{bel}_i^t\)
30: \(\{\sigma_{ij}^{t+1}\}_{\forall j \neq i} \leftarrow \{\sigma_{ij}^t\}_{\forall j \neq i}\)
31: **if** \(\mathcal{V}_i^{t+1} = \{\phi\}\ **then**
32: \(\mathcal{V}_i^{t+1} \leftarrow \mathcal{V}_i^t\)
33: **end if**
34: **end for**

In Algorithm 1, \(\mathcal{V}_i^t\) is the virtual satellite set for agent \(i\) at time \(t\), \(\Sigma_{ij}^t\) is the measurement noise covariance, \(\text{storeSats}\) stores a corrected pseudorange value, \(\text{privUpdate}\) implements equation (22), \(\text{compareSats}\) compares satellites available to agent \(i\) with those available to agent \(j\), \(\text{predictStep}\) implements equation (2), and \(\text{correctPseudorange}\) implements equation (3).

**Algorithm 2** compareSats function for agent \(i\)

Require: \(\text{bel}_i^t, \{\Sigma_{ij}^t\}, r_{ij}^t, S_i^t, S_j^t\)

1: \(\mathcal{V}_{i,\text{pos}} \leftarrow \text{identifyNewSats}(S_i^t, S_j^t)\)
2: \(\mathcal{V}_{i,\sigma} \leftarrow \text{computePseudorange}(\text{bel}_i^t, \text{bel}_j^t, S_i^t, S_j^t)\)
3: \(\mathcal{V}_{i,\sigma} \leftarrow \text{nonLinearity}(\text{bel}_i^t, \{\Sigma_{ij}^t\}, r_{ij}^t, S_i^t, S_j^t)\)
4: return \(\mathcal{V}_i\)

C. Calculation and propagation of virtual pseudoranges

Consider two agents \(i, j \in \mathcal{O}\). Let \(s\) be a satellite visible to only agent \(j\). Hence, we have an observed pseudorange from agent \(j\) given by \(\rho_{sj}\). Also, we have a ranging estimate between agents \(i, j\) given by \(r_{ij}\). Our aim is to obtain an estimate of a pseudorange \(\rho_{si}\) if the satellite was visible to agent \(i\). This geometry is depicted in Fig. 2, where points \(A, B, S\) correspond to the position of agents \(i, j\) and satellite \(s\).
\( \rho_{s_i} = \|p_i - p_s\| + \varepsilon_i \)  \hspace{1cm} (19)

\( \rho_{s_j} = \|p_j - p_s\| + \varepsilon_j \)  \hspace{1cm} (20)

Subtracting equation (20) from 19, we have

\[
\rho_{s_i} = \rho_{s_j} + (\|p_i - p_s\| - \|p_j - p_s\|) + (b_i - b_j) + (\varepsilon_i - \varepsilon_j)
\]

\hspace{1cm} (21)

Hence, equation (22) is used to estimate \( \rho_{s_i} \). In equation (22), \( \rho_{s_j} \) is obtained from the pseudorange measurement, \( \|p_i - p_s\| \) is obtained from the ranging measurement and \( 1_{si}, 1_{sj}, b_i, b_j \) are obtained from states of agents \( i, j \). Note that the variables on the right in equation (22) are in fact random variables with associated covariances. By sampling points from their joint distribution and passing each point through the non-linear equation (22), we obtain a probability distribution for \( \rho_{s_i} \). This constitutes the function nonLinearity in algorithm 2. The resultant probability distribution of \( \rho_{s_i} \) is assumed Gaussian as the algorithm uses an EKF framework.

Once we obtain \( \rho_{s_i} \), it is crucial to update this estimate as the agent executes a motion. This update is dependent on the measurement and motion model used. We use a first order Taylor expansion of the measurement model in equation (1) to obtain

\[
\Delta \rho_{s_i} = L^i_{i} \Delta k
\]

where \( L^i_{i} = \left[ \begin{array}{cc} -1_{si}^T & 1 \end{array} \right] \), \( \Delta k = [\delta p_i^T, \delta b_i]^T \), with \( \delta p_i = p_{i+1}^i - p_i^i \) and \( \delta b_i = b_{i+1}^i - b_i^i \). This gives us

\[
\rho_{s_i}^{t+1} = \rho_{s_i}^{t} + \Delta \rho_{s_i}^{t}
\]

(24)

Also, the noise covariance \( (\sigma_{sn}) \) associated with each pseudorange measurement is updated as

\[
\sigma_{sn}^{t+1} = L^i_{i} M^i_{i} \left( L^i_{i} \right)^T + \sigma_{sn}^{t}
\]

where \( M^i_{i} \) is 4 x 4 matrix constituted by the covariance submatrices for \( p_i^T, b_i^T \) in the agent’s belief \( b_i^t \).

**TABLE I: Data Specifics for the Experiment.**

<table>
<thead>
<tr>
<th>numSV</th>
<th>Agents’ Coordinates</th>
<th>Ag-Ag Distance (in m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lat (°)</td>
<td>Lon (°)</td>
</tr>
<tr>
<td>Ag 1</td>
<td>4</td>
<td>-88.226885</td>
</tr>
<tr>
<td>Ag 2</td>
<td>6</td>
<td>-88.225894</td>
</tr>
<tr>
<td>Ag 3</td>
<td>6</td>
<td>-88.225974</td>
</tr>
</tbody>
</table>

In our experimental setup, we consider three agents in an urban scenario. Two agents are stationary ground stations while the third is an iBQR UAV designed and built by our research group. Each agent is equipped with a Decawave Ultra-Wideband (UWB) ranging sensor and a u-blox LEA-6T GPS receiver connected to an antenna. Computers at each agent’s location log the data from these sensors. While the proposed algorithm does not impose any requirements on the connectivity of the network, our agent network is fully connected over Wi-Fi. However, we use the communication link only when measurements are obtained from the UWB ranging sensor to allow for sparse communication. The UWB sensors on each of the agents operate asynchronously and obtain a measurement from the two neighbors at a rate of 10 Hz. GPS pseudorange measurements are obtained at a rate of about 5 Hz.

Fig. 3 shows the waveform of a typical parsed UWB sensor output. These measurements suffer from impulse noises and we threshold them to filter out this noise. Typically, the data from a Decawave UWB sensor has a mean error of about 5 cm when agents are within 30 m of each other.

For our experiment, we position the iBQR UAV at a fixed location in an alley ∼5 meters wide and with walls 3 stories high on both sides. This effectively blocks the majority of the sky from the receiver. The ground agents are placed close to...
At initialization, we allow the ground agents to have better estimates of their position than the UAV, i.e., low initial state covariance matrices, and an initial position estimate close to the true position. The idea is to demonstrate that if some agents in a network possess good localization estimates, then the agents that are “lost” should be able to utilize this information to localize themselves better.

We compare the following approaches on our dataset:

- Single agent localization using Particle Filter (PF) [16], Extended Kalman Filter (EKF), and Newton Rhapson method (NR)
- Hybrid Cooperative Particle Filter (HC-PF) as proposed by Sottile et al. in [16]

and report the results on horizontal error (in ENU frame) in localization from their true positions over time by the various methods. Note that NR, PF, and EKF methods do not take into account the inter-agent ranging measurements and rely solely on GPS measurements. In these methods, we assume each agent is independent of the others.

Fig. 7 shows the plot of error in estimation from true position over time for various methods. Table II lists the RMS error in position averaged over time for the agents using the aforementioned methods. We can see from Fig. 7 that both HC-PF and H-DCL have a lower positioning error than EKF. This is expected since an additional ranging measurement should improve localization capabilities. From Table II, we can see that H-DCL has lower RMS error than HC-PF. The difference is especially larger for Agent 1. Comparing H-DCL to DCL, we see similar RMS errors for Agents 2 and 3. This is because both Agents 2 and 3 have only 1 virtual satellite in the H-DCL algorithm and their satellites-agent geometry does not

<table>
<thead>
<tr>
<th>RMS error in estimation (in m)</th>
<th>H-DCL</th>
<th>DCL</th>
<th>HC-PF</th>
<th>EKF</th>
<th>PF</th>
<th>NR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent 1</td>
<td>5.4</td>
<td>6.7</td>
<td>8.6</td>
<td>11.5</td>
<td>11.0</td>
<td>14.3</td>
</tr>
<tr>
<td>Agent 2</td>
<td>4.3</td>
<td>4.4</td>
<td>5.0</td>
<td>7.4</td>
<td>6.5</td>
<td>9.0</td>
</tr>
<tr>
<td>Agent 3</td>
<td>2.6</td>
<td>2.7</td>
<td>3.0</td>
<td>6.6</td>
<td>5.2</td>
<td>9.0</td>
</tr>
</tbody>
</table>
change much by this addition. However, for Agent 1, the RMS error is markedly lower since 3 virtual satellites are available (see Fig. 6) which provide a better satellite distribution in the sky. These results validate that sharing satellite information leads to a greater improvement in the location estimate for agents with low number of visible satellites. Overall, our method outperforms the current algorithms and provides more accurate position estimates for all agents.

While, we do not discuss the implications of agents’ geometry on the accuracy of our solution in this paper, an underlying assumption is that a non-ambiguous solution to the localization problem exists. Another consideration is the effect of approximations on the direct information use from the various sensors. As shown in [12], in an optimal filter, every private measurement such as a pseudorange measurement, affects the state estimate of all the agents in the system and not just the agent performing the measurement. However, in order to minimize information exchange, the adopted algorithm neglects this effect on other agents’ states. Similarly, approximations in the relative update step leads to information loss in the algorithm. Thus, the algorithmic framework leads to a sub-optimal use of measurements from the sensing modalities. This motivated us to introduce the virtual measurements in this approximated filter to reduce the information loss by sharing satellite information across agents. Thus, although the generated virtual measurements are not new information for the complete system (since they are not observed directly by a sensing modality) and are directly dependent on inter-agent ranging, they lead to an improvement in positioning accuracy. This can be directly seen in Fig. 8 where we compare the results of the use of only inter-agent ranging is used (shown in green).

V. Conclusion

We proposed a decentralized collaborative localization algorithm for estimating navigation solution for UAVs in urban environments. We formulated a methodology to relay satellite information between agents to obtain more constraint equations for each agent by coupling GPS measurements with ranging measurements. The algorithm improved accuracy of all agents in the system and especially aided agents suffering from signal blockage and multipath. We demonstrated this on a real world dataset collected using an iBQR UAV and ground agents.

ACKNOWLEDGMENT

The authors would like to sincerely thank Mr. Matt Peretic, Mr. Shubhendra Chauhan, Mr. Ashwin Kanhere, Mr. Craig Babiarz, Mr. Akshay Shetty, Mr. Spencer Markowitz, Mr. Arthur Lapin, Mr. Andrew Tham, and Mr. Alex Cavanaugh for their assistance. The research was carried out in part in Intelligent Robotics Laboratory, University of Illinois.

REFERENCES


