Trilateration-Based Cooperative Localization for Multi-Agent Systems During GNSS Degradation

Eric J. Kim
Department of Electrical and Computer Engineering
Queen's University
Kingston, Canada

***Abstract*—This project explores a trilateration-based cooperative localization strategy for multi-agent aerial systems under noisy sensor and environmental conditions. For a given agent, a cascaded Kalman filtering architecture integrates IMU mechanization with GNSS and cooperative state estimates obtained from neighbouring agents. A decentralized consensus-based proportional-derivative control law maintains formation in a constant velocity state while accounting for estimation errors and external disturbances such as wind gusts. The project showcases through simulated experiments how the cooperative localization system could be used in uncertain operating environments and provides insight into the tradeoffs between sensor fidelity, filter design, and swarm coordination.**

I. Introduction

Unmanned aerial vehicle (UAV) swarms are emerging as effective tools in applications ranging from search and rescue operations and structure monitoring to environmental data collection and defense. A key consideration in enabling multi-agent autonomy is ensuring reliable and accurate localization, particularly in GNSS-denied or GNSS-degraded environments. Traditional localization methods relying solely on global navigation satellite systems (GNSS) are vulnerable to signal obstruction, multipath effects, and intentional jamming, while strict dead-reckoning-based schemes fail to provide absolute positioning. These considerations have therefore prompted investigations into robust navigation techniques. This work investigates a GNSS-aware cooperative localization method that fuses inter-agent range measurements, inertial states, and partial GNSS data through a cascaded Kalman filtering framework. One of the more novel areas of research involves cooperative localization, where agents share relative measurements and fuse local information to improve overall state estimation. This approach takes advantage of inter-agent ranging, inertial measurements, and the idea of cooperative filtering to infer absolute or relative positions even when direct GNSS signals are unreliable. In this context, Kalman filtering remains a core technique due to its well-established theoretical and practical framework and suitability for sensor fusion. Recent works have demonstrated various sensor fusion strategies for cooperative localization. Xu et al. [10] proposed a decentralized visual-inertial ultra-wideband (UWB) framework enabling robust relative state estimation without external infrastructure, achieving centimeter-level accuracy through optimization-based fusion of camera, inertial measurement unit (IMU), and UWB data. Guo et al. [2] introduced a consensus-based approach using (UWB) range measurements and odometry for cooperative localization, also emphasizing infrastructure-free deployment while ensuring bounded estimation errors even in the presence of noise and communication disruptions. Han et al. [3] integrated relative localization with formation control using distance and velocity measurements under a leader–follower topology. Their method demonstrated global convergence under persistent excitation, highlighting the close relation between localization and formation control. Other works, such as those by Nguyen et al. [5]–[8], explored fusing UWB with vision or IMU sensors to achieve robust pose estimation in dynamic environments.

These methods focus strictly on relative positioning, and only mention GNSS as a comparative baseline or reject it outright due to its limitations. On the contrary, this work proposes a Kalman filter-based cooperative localization scheme for multi-agent systems that mimics GNSS trilateration theory by integrating relative distances to neighbouring agents. The novelty lies in mimicking GNSS single-point positioning using trilateration to neighbouring agents, while dynamically weighting their GNSS states based on their current degradation status. This allows an agent to use partially degraded or cooperative GNSS data to correct its inertial state, without full reliance on its own GNSS, therefore bridging the gap between GNSS-based and GNSS-free cooperative localization. The approach is designed for three dimensional aerial swarms and exploits the geometric structure of multi-agent systems for cooperative state estimation.

The structure of this report is as follows: Section II describes the theoretical framework and implementation methodology. Section III describes the simulation environment that was created to test the cooperative localization method. Section IV discusses simulation results under various GNSS degradation scenarios. Finally, Section V concludes the report with key insights and future research directions.

II. Methodology

A. Integrated Cooperative Solution

This work asssumes that each agent in the multi-agent system contains both an INS and a GNSS receiver. Therefore, during periods of GNSS signal degradation or complete outage, agents can use either dead-reckoning or a cooperative solution to localize themselves, while during periods of reliable GNSS data, both the relatively more biased INS and

the relatively more noisy GNSS can be fused to mitigate their respective downsides and provide an overall more accurate position estimate.

The overall localization pipeline consists of three Kalman filters:

1) a cooperative trilateration-based GNSS estimator
2) a fusion stage combining cooperative and local GNSS measurements
3) a final integration stage with INS

This hierarchy is designed to isolate and progressively refine state estimates based on the trustworthiness of each information source. The corresponding cascaded Kalman filter solution is illustrated in Figure 1.

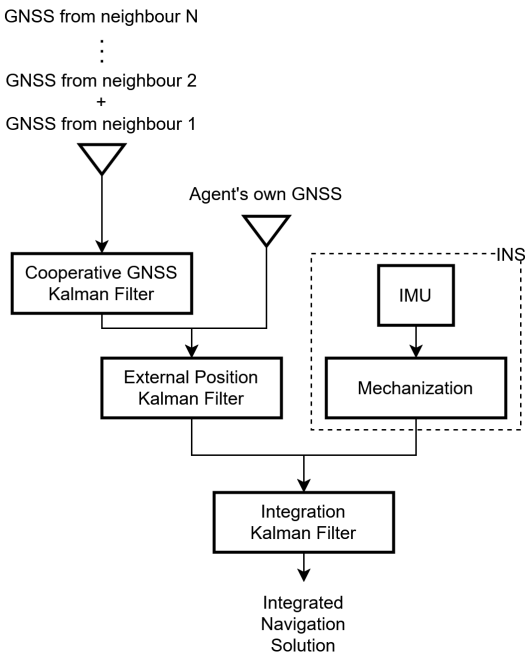

Fig. 1. Block diagram of complete positioning solution

The process begins with the cooperative GNSS Kalman Filter. This stage mimics GNSS single-point positioning (SPP), using inter-agent range measurements and the GNSS-reported positions of neighbours to trilaterate the target agent's position. The result is a cooperative state estimate, derived from external sources rather than the agent's own sensors. These estimates may themselves originate from degraded GNSS data on neighbouring agents, so downstream filters incorporate adaptive weighting to account for varying data quality.

Next, this estimate is combined with the agent's own degraded or partial GNSS data in the External Position Kalman Filter, which fuses both sources to produce a refined global position estimate that incorporates both independent and cooperative information. This stage provides positioning robustness for the dual situations when the agent's own GNSS data is unreliable, or the neighbouring agents from which the cooperative state is obtained experience GNSS degradation themselves.

Finally, the output of this fusion is passed into the Integration Kalman Filter, which performs traditional INS/GNSS fusion. This final stage integrates the external GNSS-derived position with inertial navigation data from the agent's IMU, which undergoes mechanization. This full integration provides a resilient and drift-limited integrated navigation solution, robust to both temporary GNSS outages or noisy GNSS data, as well as long-term drift in inertial measurements.

To facilitate examining the performance of the cooperative stateing method, a baseline Kalman filter that integrated simulated IMU mechanized states with the agent's own GNSS states was also implemented. The estimated position and velocity from this filter was used in certain parts of the process to eliminate feedback errors arising from the open-loop architecture of the system and preclude more involved examinations into stability.

In addition, for this project, the multi-agent system was modeled under constant velocity assumptions to better showcase the cooperative localization technique in the more practical and useful scenario where the agents are collectively moving instead of simply stationary.

B. Cooperative Trilateration

This section details the technique used to trilaterate the position of agent i, which is based on the way GNSS measurements are integrated to obtain a single-point-positioning estimate. This approach adapts the single-point positioning method used in GNSS, but replaces satellites with neighbouring agents acting as relative range anchors.

In GNSS positioning, the receiver's unknown position in ECEF coordinates is determined using pseudorange measurements. This model can be adopted so that the satellites represent the neighbouring agents of an agent i, that is, agents that are within a sensing range R, and the receiver represents agent i itself. The true range r_i^m from agent i's position $p \equiv (x, y, z)$ to the m^{th} satellite at $p^m \equiv (x^m, y^m, z^m)$, can be expressed as:

$$r_i^m = \sqrt{(x - x^m)^2 + (y - y^m)^2 + (z - z^m)^2} \tag{1}$$

Then, the pseudorange ρ between satellite m and agent i can be expressed as follows:

$$\rho_i^m = ||p - p^m|| \tag{2}$$

In the context of this work, this pseudorange is assumed to be obtained through UWB sensors relaying data through a two-way-ranging (TWR) method. These relative distance measurements are assumed to be subject to Gaussian noise, and arrive synchronously from all neighbours within sensing range.

For N agents, we obtain a system of nonlinear equations, which we linearize using a first-order Taylor series expansion around the current best estimate of agent i's position $p_{EST} \equiv (x_{EST}, y_{EST}, z_{EST})$ to obtain ρ_i^m, as previously described in [4]. The estimated pseudorange $\rho_{i_{EST}}^m$ is:

$$\rho_{i_{EST}}^m = ||p_{EST} - p^m|| \tag{3}$$

And represent the system for N neighbouring agents can be represented in matrix form:

$$\delta\boldsymbol{\rho}_i = \rho_i^m - \rho_{i_{EST}}^m \tag{4}$$

$$= \begin{bmatrix} \delta\rho_i^1 \\ \delta\rho_i^2 \\ \vdots \\ \delta\rho_i^N \end{bmatrix} = \begin{bmatrix} \mathbf{1}_{EST}^1 \\ \mathbf{1}_{EST}^2 \\ \vdots \\ \mathbf{1}_{EST}^M \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x} \end{bmatrix} = G\delta\boldsymbol{S}_\rho \tag{5}$$

where

$$\mathbf{1}_{EST}^m = \frac{[(x_{EST} - x^m), (y_{EST} - y^m), (z_{EST} - z^m)]}{||p_{EST} - p^m||} \tag{6}$$

To calculate the cooperative velocity from neighbouring agents, the system becomes:

$$\delta\boldsymbol{\rho}_i = \begin{bmatrix} \delta\rho_i^1 \\ \delta\rho_i^2 \\ \vdots \\ \delta\rho_i^N \\ \delta\dot{\rho}_i^1 \\ \delta\dot{\rho}_i^2 \\ \vdots \\ \delta\dot{\rho}_i^N \end{bmatrix}_{2N\times 1} = \begin{bmatrix} \mathbf{1}_{EST}^1 & 0_{3\times 1} \\ \vdots & \vdots \\ \mathbf{1}_{EST}^N & 0_{3\times 1} \\ 0_{3\times 1} & \mathbf{1}_{EST}^1 \\ \vdots & \vdots \\ 0_{3\times 1} & \mathbf{1}_{EST}^N \end{bmatrix}_{2N\times 6} \begin{bmatrix} \delta\boldsymbol{x} \\ \delta\boldsymbol{v} \end{bmatrix}_{6\times 1} \tag{7}$$

$$= H\delta\boldsymbol{S}$$

Note that the velocity estimate is constrained only along the line-of-sight (LOS) vectors between the agent and each neighbour, which limits full 3D velocity observability when LOS directions are poorly conditioned.

If $N \geq 4$, that is, if there are at least four neighbouring agents, the solution to the above equation is:

$$\delta\hat{\boldsymbol{S}} = (H^T H)^{-1} H^T \delta\boldsymbol{z} \tag{8}$$

The position and velocity estimates can be improved by applying this correction $\delta\hat{\boldsymbol{S}}$ to the current estimate.

C. Kalman Filters

The Kalman filter is a recursive algorithm for estimating the state of a dynamic system in the presence of noise. It operates in two stages: a prediction step, which projects the current state forward in time, and an update step, which corrects this prediction using incoming measurements. The following equations describe the general Kalman filter framework used in each stage of the architecture. Specific modifications for each of the three filters follow in subsequent sections.

The general system is often represented in a discrete-time state-space form:

$$\boldsymbol{x}_k = \boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{G}_{k-1}\boldsymbol{w}_{k-1} \tag{9}$$

$$\boldsymbol{z}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{\eta}_k \tag{10}$$

where:

- $\boldsymbol{x}_k$ is the state vector at time step k.
- $\boldsymbol{F}_{k-1}$ is the state transition matrix.
- $\boldsymbol{G}_{k-1}$ is the noise coupling matrix.
- $\boldsymbol{w}_{k-1}$ is the system noise, assumed to be Gaussian with covariance $\boldsymbol{Q}_k$.
- $\boldsymbol{z}_k$ is the measurement vector.
- $\boldsymbol{H}_k$ is the observation matrix.
- $\boldsymbol{\eta}_k$ is the measurement noise, assumed to be Gaussian with covariance $\boldsymbol{R}_k$.

In the prediction step, the filter calculates the next state estimate and its associated uncertainty:

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{\Phi}_{k-1}\hat{\boldsymbol{x}}_{k-1}^+ \tag{11}$$

$$\boldsymbol{P}_k^- = \boldsymbol{\Phi}_{k-1}\boldsymbol{P}_{k-1}^+\boldsymbol{\Phi}_{k-1}^T + \boldsymbol{G}_{k-1}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^T \tag{12}$$

where:

- $\hat{\boldsymbol{x}}_k^-$ is the predicted state.
- $\boldsymbol{P}_k^-$ is the predicted covariance matrix.
- $\boldsymbol{P}_{k-1}^+$ is the estimated state covariance from the previous step.

The update step then refines the prediction using the measurement $\boldsymbol{z}_k$:

$$\boldsymbol{K}_{k-1} = \boldsymbol{P}_k^-\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1} \tag{13}$$

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-) \tag{14}$$

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^- \tag{15}$$

where:

- $\boldsymbol{K}_{k-1}$ is the Kalman gain, which determines how much weight is given to the new measurement.
- $\hat{\boldsymbol{x}}_k^+$ is the updated state estimate.
- $\boldsymbol{P}_k^+$ is the updated error covariance.

This base framework was adapted for each of the three Kalman filters described in Figure 1.

D. Cooperative GNSS Kalman Filter

The first Kalman filter in the system estimates a correction to agent i's position and velocity using cooperative pseudorange and Doppler-like range-rate measurements from neighbouring agents.

We define the state vector as the position and velocity errors, which are calculated as the difference between the estimated position of agent i, and its trilaterated position from the estimated and measured pseudoranges from agent i to each of its neighbours.

$$\boldsymbol{X}_[k] = \begin{bmatrix} \delta\boldsymbol{x}[k] \\ \delta\boldsymbol{v}[k] \end{bmatrix} \in \mathbb{R}^6 \tag{16}$$

The measurement vector $\boldsymbol{z}_k$ is defined as $\delta\boldsymbol{\rho}_i$ from Equation 7, and the state evolves under a constant velocity model:

$$\boldsymbol{X}[k+1] = \boldsymbol{F}\boldsymbol{X}[k] + \boldsymbol{w}[k], \tag{17}$$

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{I}_3 & \Delta t \cdot \boldsymbol{I}_3 \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{bmatrix} \tag{18}$$

where $\boldsymbol{w}_i[k] \sim \mathcal{N}(0, \boldsymbol{Q})$ is zero-mean process noise.

The measurement model relates the error state to the pseudorange and relative velocity residuals using the matrix $H \in \mathbb{R}^{2N\times 6}$, derived in the earlier linearization:

$$\boldsymbol{z}[k] = H[k]\boldsymbol{X}_i[k] + \boldsymbol{\varepsilon}[k], \quad \boldsymbol{\varepsilon}[k] \sim \mathcal{N}(0, \boldsymbol{R}) \tag{19}$$

The measurement noise covariance matrix $\boldsymbol{R} \in \mathbb{R}^{2N \times 2N}$ accounts for uncertainty in both the pseudorange and relative velocity measurements. It is constructed adaptively at each timestep based on the status of neighbouring agents. For each neighbour j of agent i, the corresponding entries in $\boldsymbol{R}$ are determined as follows:

- If agent$j$ is currently in GNSS outage or blockage, large constant noise values are assigned to both its position and velocity components to reflect high uncertainty.
- Otherwise, the values are taken from the neighbour's known GNSS error characteristics, which is assumed to be communicated to agent i.

The final matrix is diagonal and structured as:

$$\boldsymbol{R} = \text{diag}\left(\sigma^2_{\text{pos},1}, \ldots, \sigma^2_{\text{pos},N}, \sigma^2_{\text{vel},1}, \ldots, \sigma^2_{\text{vel},N}\right) \tag{20}$$

This design ensures that unreliable neighbours contribute less to the update step, while reliable neighbours have greater influence.

The process noise covariance matrix $\boldsymbol{Q} \in \mathbb{R}^{6\times 6}$ encodes uncertainty in the motion model, which assumes constant velocity between timesteps. In this implementation, $\boldsymbol{Q}$ is kept constant, with a tuning parameter k_Q reflecting expected deviations from the constant-velocity assumption.

$$\boldsymbol{Q} = k_Q \boldsymbol{I}_6 \tag{21}$$

Despite its simplicity, this formulation is effective in a cooperative context where most of the positional information comes from external agents rather than the motion model itself.

At each time step, the Kalman filter proceeds with the standard prediction and update steps, after which the agent's position and velocity estimates are corrected using the estimated error state:

$$\boldsymbol{p}_i \leftarrow \boldsymbol{p}_i + \delta\hat{\boldsymbol{x}}_i \tag{22}$$

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \delta\hat{\boldsymbol{v}}_i \tag{23}$$

The output of this filter is a refined estimate of agent i's position and velocity, incorporating cooperative pseudorange and Doppler-like measurements from neighbours. This cooperative GNSS estimate is passed to the next stage of the architecture for fusion with the agent's own GNSS measurements.

E. External Position Kalman Filter

The second Kalman filter fuses agent i's own GNSS measurements with the cooperative GNSS estimate computed from neighbouring agents. This integration stage provides a robust position and velocity estimate that reduces noise when the agent's GNSS is unreliable, or when the cooperative state is noisy. It as a reconciliation layer, resolving differences between the agent's own GNSS data and the cooperative estimate, especially when either is degraded.

The state vector in this case is the position and velocity errors between the estimated and measured states, which are used to correct the agent's own GNSS estimate. The state transition matrix is defined identically to that of the first filter through Equation 18. Correspondingly, the measurement model is constructed from the discrepancy between the internal GNSS state and the external cooperative estimate.

The measurement noise covariance $\boldsymbol{R}[k] \in \mathbb{R}^{6\times 6}$ is adaptively estimated at each time step based on the GNSS status of neighbouring agents contributing to the cooperative estimate:

$$\boldsymbol{R}[k] = \sigma^2_{\text{collab}}[k] \cdot \boldsymbol{I}_6, \tag{24}$$

where $\sigma^2_{\text{collab}}[k]$ is computed as the mean of the GNSS noise variances among neighbours.

Once again, the process noise covariance $\boldsymbol{Q} \in \mathbb{R}^{6\times 6}$ is fixed and tuned through a scalar gain k_Q, as was shown in Equation 21. Finally, each epoch proceeds with the standard Kalman prediction and update steps, after which the updated state is then used to correct the agent's state obtained through its GNSS data, all in open-loop fashion.

This formulation ensures that high-confidence cooperative updates influence the agent's GNSS-based estimate while suppressing unreliable corrections in cases of neighbour degradation or outage. Also, as with the previous filter, the update operates in open-loop. The corrected state is not fed back to upstream filters but passed forward for integration.

F. Integration Kalman Filter

The last Kalman filter combines agent i's INS-estimated position and velocity with the external position estimate, obtained by correcting the agent's GNSS data with the output of the second Kalman filter. This stage allows the system to function in the event that both the GNSS and cooperative GNSS-derived states are unreliable, thereby reducing the position and velocity estimation to a dead-reckoning system based on IMU mechanization. The output of this filter forms the final position and velocity estimate used for performance evaluation.

This filter is modelled nearly identically to the previously detailed external position filter, with differently tuned gain constants. However, the measurement covariance matrix $\boldsymbol{R}$ was taken to be the running variance of the last M measurements of the external position and velocity estimates:

$$\begin{aligned}\mathbf{R}[k] = \text{diag}\Big(&\sigma^2_x[k-M:k],\ \sigma^2_y[k-M:k],\ \sigma^2_z[k-M:k],\\ &\sigma^2_{v_x}[k-M:k],\ \sigma^2_{v_y}[k-M:k],\ \sigma^2_{v_z}[k-M:k]\Big)\end{aligned}$$

III. Simulation

To examine the performance of the proposed cooperative localization method, a multi-agent simulation environment was built with various stochastic elements.

A. Dynamics and Sensors

First, the truth states, position p and velocity v of a given agent, are computed using a simple double integrator model:

$$\dot{v} = u(t)$$

$$\dot{p} = v$$

As the scope of this project is to examine cooperative localization, the simulation of GNSS data and IMU mechanization were bypassed using this truth state. To obtain

simulated GNSS position and velocity measurements, zero mean Gaussian noise with standard deviations σ_{GNSS_p} and σ_{GNSS_v} were added to the truth states. It must be noted that GNSS measurements are assumed to be in the same global frame as the truth states. Using this model, GNSS blockages were simulated by multiplying the noise standard deviations by an outage multiplier, and outages were simulated through an internal boolean flag that forced the system to carry forward the last known GNSS position.

$$p_{GNSS} = p_{truth} + \mathcal{N}(0, \sigma^2_{GNSS_p}) \tag{25}$$
$$v_{GNSS} = v_{truth} + \mathcal{N}(0, \sigma^2_{GNSS_v}) \tag{26}$$

Similarly, the IMU and subsequent mechanization process were also not explicitly simulated. Instead, the INS was initialized with the truth states, after which it evolved through the same double integrator model described previously. To simulate drift and noise, zero mean Gaussian noise sampled at every timestep and an accumulating bias term $b(t)$ were added to the control input u (at the acceleration level) as the velocity term underwent integration.

$$a_{IMU} = b(t) + \mathcal{N}(0, \sigma^2_{IMU}) \tag{27}$$

During the trilateration process, the relative distance measurements between agents must be simulated. However, to ensure a more realistic scenario and more appropriately assess the performance of the cooperative localization method, zero-mean Gaussian noise was added to the true position and velocity states at every timestep:

$$p_{coop} = p_{truth} + \mathcal{N}(0, \sigma^2_{coop}) \tag{28}$$
$$v_{coop} = v_{truth} + \mathcal{N}(0, (0.1\sigma_{coop})^2) \tag{29}$$

B. Environmental Stochasticity

In addition to the sensor data, randomness was also added to the environment in the form of acceleration jitter and wind gusts. To simulate small-scale randomness, a zero mean Gaussian variable with a constant standard deviation was also sampled at every timestep and added to the control input. In addition, a simulated wind gust mechanism added an exponentially-decaying three-dimensional gust force $g(t)$ to the control input, which is illustrated in Figure 2. At every discrete simulation time, a uniform distribution was sampled such that there would be a probability p_g that a gust would be generated. The gust's magnitude a_0 and duration T were also sampled from uniform distributions based on maximum values.

$$g(t) = a_0 e^{-t/\tau} \tag{30}$$
$$\tau = T/ln(100) \tag{31}$$

Gust forces are added to the agent's control input, affecting the truth trajectory but not directly modifying sensor outputs. These mechanisms further disrupted the position of each agent, bringing the simulation closer to reality.

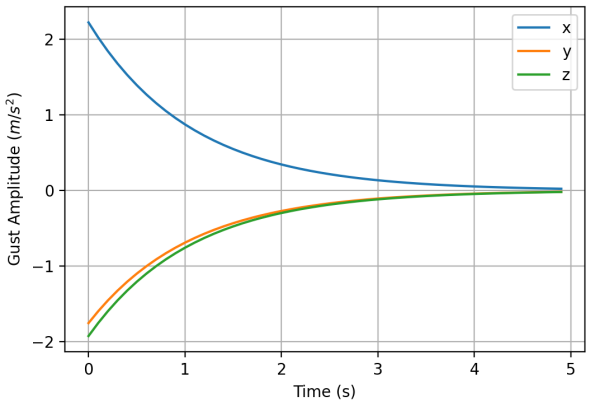

Fig. 2. Sample of Gust with Maximum Magnitude of 3.0 m/s^2 and Maximum Duration of 5.0 s

C. Control Law

To prevent agents from accelerating indefinitely due to accumulating drift and stochastic gust forces, a decentralized, consensus-based control input was applied in the form of a proportional-derivative (PD) formation controller with an additional velocity-tracking term. The control input u_i for agent i at time t is computed based on the relative position e_p and velocity e_d errors with respect to its neighbours $j \in \mathcal{N}_i$, as well as a desired global velocity error e_v:

$$u_k = -\frac{1}{N_i} \sum_{j \in \mathcal{N}_i} [K_p e_p + K_d e_d] - K_v e_v \tag{32}$$
$$e_p = (p_i - p_j) - (p_i^0 - p_j^0) \tag{33}$$
$$e_d = v_i - v_j \tag{34}$$
$$e_v = v_i - v_d \tag{35}$$

Here, p_i and v_i denote the estimated position and velocity of agent i at the current time, and p_i^0 is the initial position. The set $\mathcal{N}_i$ contains the neighbours of agent i as determined by the adjacency matrix (a function of the maximum sensing distance), and N_i is the number of neighbours. The gains K_p, K_d, and K_v are tunable parameters corresponding to position correction, velocity damping, and velocity tracking, respectively. In short, e_p aims to set the relative distance between an agent and its neighbours equal to the initial distance for formation preservation, e_d works to set relative velocities equal to zero for swarm cohesion, and e_v exists so the entire swarm is moving through space at a constant velocity to satisfy the constant-velocity model used in the Kalman filters.

In implementation, the controller uses state estimates rather than ground truth data, under the rationale that true states are not observable in reality. During the simulation, the control law uses the solution from a "benchmark" Kalman filter that integrates simulated IMU measurements with simulated GNSS data. Although using the output of the final integrated solution seems like the more applicable choice, the benchmark filter's

output was used to prevent instability caused by large initial estimation errors, as well as potential regions of instability in the overall system. Lastly, all neighbors are weighted equally in the control input, and neither proximity nor trust weighting is applied.

D. Simulation Process

During initialization of the swarm simulation, the global and relative positions of agents are randomly set so that for any given pair of agents, they exist within a maximum sensor range of each other, ensuring global initial connectivity. In the case of this project, where connections between agents are assumed to be bidirectional and therefore undirected, this initialization method represents the strongly connected case.

The complete formulation of the simulation environment is provided in pseudocode.

Algorithm 1 Simulation Routine

```
Generate initial positions for N agents with spacing constraints
Initialize each agent with IMU, GNSS, drift, outages, blockages, and Kalman filters
for t from t_0 to t_f with step Δt do
    Compute adjacency matrix A based on agent positions
    for each agent i do
        Sample random jitter
        Sample random gust
        Compute control input using neighbour estimates
        Update truth state
        Update IMU and GNSS measurements
        Update all Kalman filters:
            Benchmark: IMU and GNSS
            Cooperative: Trilaterated state estimate
            External: GNSS and cooperative state
            Integrated: IMU and external output
        Save states for plotting and analysis
    end for
end for
```

IV. Results and Discussion

A. Simulation Setup

For the following 5-agent simulation, the amount of noise and bias in the simulated IMU-mechanized position and velocity states were set to be analogous to using a low-cost Murata SCC1300-D04 IMU [1]. The accelerometer root-mean squared (RMS) noise and offset errors of $5mg$ and $70mg$ respectively were converted to standard deviations for a zero-mean Gaussian random variable of $0.05m/s^2$ and $0.70m/s^2$. The offset was interpreted as a constant bias term in acceleration, contributing directly to long-term IMU drift.

The values for GNSS position and velocity noise were obtained from the specifications of NovaTel's OEM7500 GNSS receiver module [9], which has a single point L1 band RMS value of $1.5m$ and a velocity accuracy of less than $0.03m/s$ RMS. These values have been translated to position and velocity standard deviations (for a zero-mean Gaussian random variable), of $1.5m$ and $0.1m/s$ respectively.

For the stochastic environmental noise, the acceleration jitter was set to have a standard deviation of $0.005m/s^2$, and gusts had a maximum amplitude of $3.0m/s^2$, with a per-timestep probability of 5.0% and a maximum duration of $5.0s$.

The results in this section use a target velocity v_d in x, y, z coordinates of $(2.0, 1.0, 0.1)$ m/s, where one agent $i = 0$ underwent a GNSS blockage between 10 and 30 seconds with an outage multiplier of 10, and a complete GNSS outage from 40 to 50 seconds. Furthermore, to allow filter convergence, simulation data was collected after a 60-second warm-up period.

B. Simulation Data

Figure 3 shows how the simulated IMU-mechanized position estimate diverges from the truth state due to an accumulating bias and acceleration noise.

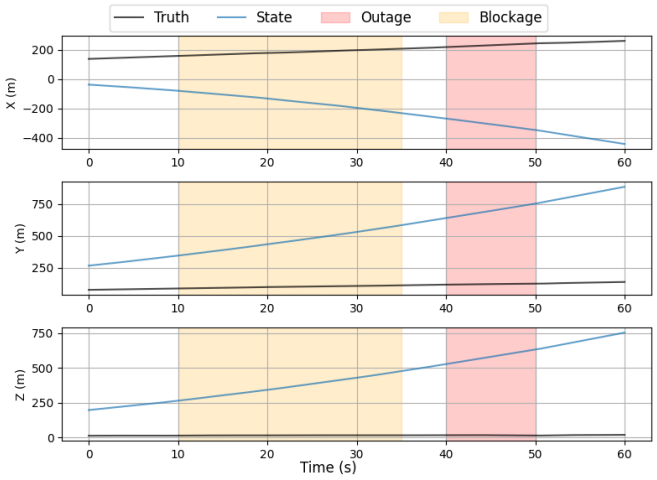

Fig. 3. Simulated IMU-Mechanized Position

Figure 4 illustrates how during blockages, the GNSS noise increases by an order of magnitude, and during outages, the GNSS output simply uses the last observed state.

Figure 5 shows how the simulated cooperative localization technique results in a position estimate that closely tracks the truth state, albeit with a noise that is proportional to the zero-mean Gaussian noise injected into the simulated relative distance measurements used for trilateration.

Figure 6 indicates how the external state Kalman filter, which integrates an agent's own GNSS with the cooperative state estimate, closely tracks the latter during periods of reduced GNSS quality. Although the figure may seem redundant in comparison with Figure 5, the external state filter exists to provide a stable external reference position and velocity during periods of both high and low-fidelity GNSS measurements. Along a similar line of reasoning, it can be expected that when agents have high quality GNSS data, the benchmark Kalman filter integration of simply the IMU and the GNSS

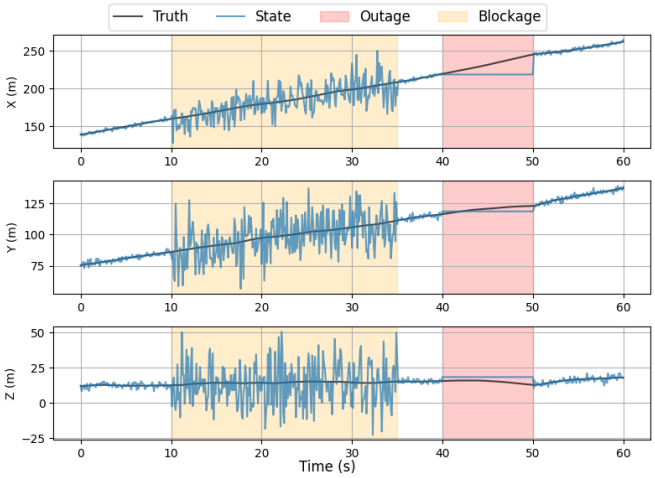

Fig. 4. Simulated GNSS Position

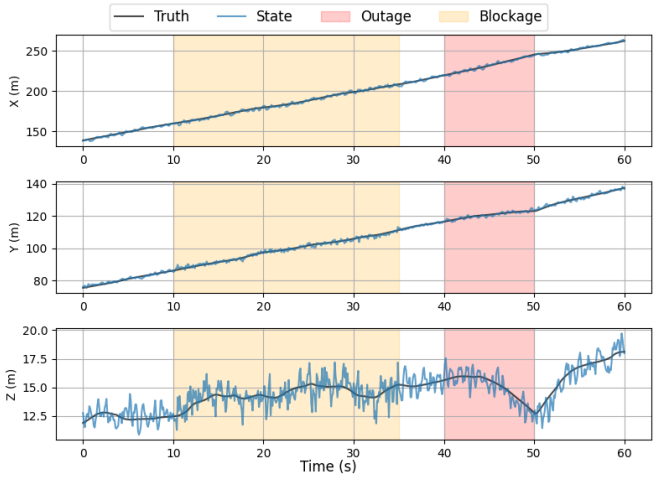

Fig. 6. External Kalman Filter Position

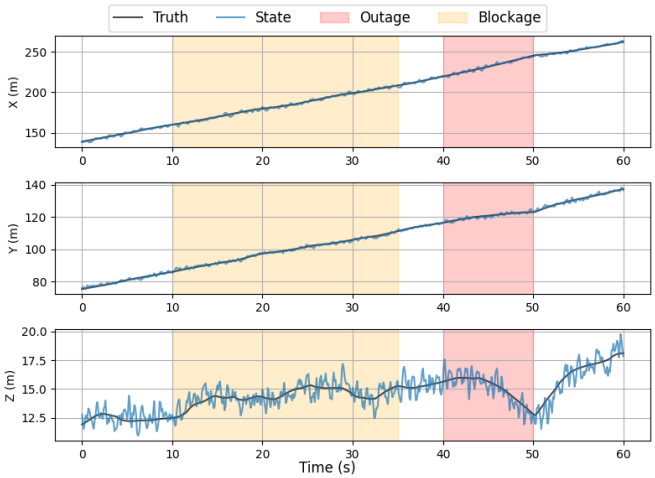

Fig. 5. Cooperative Kalman Filter Position

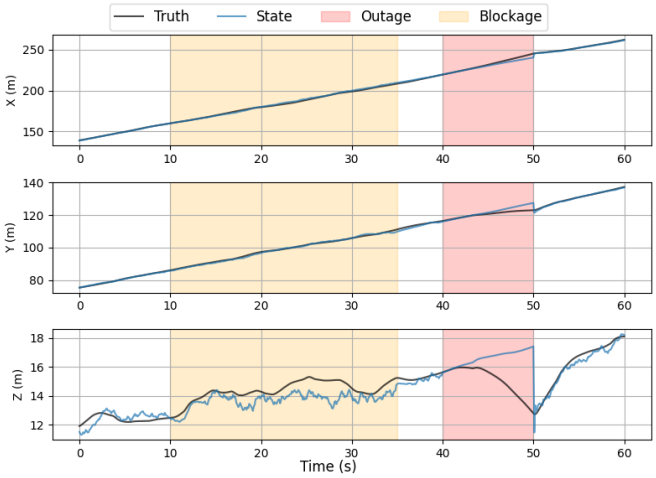

Fig. 7. Benchmark Kalman Filter Position

will perform better than the cooperative localization technique, which is subject to noise

Figure 7 represents the result of the corrections obtained from the benchmark Kalman filter, which integrates mechanized IMU with GNSS. The plots make clear how during outages, the open-loop filter has no choice but to rely solely on the mechanized IMU state, which increasingly diverges as the outage prolongs. Furthermore, there is a nonsignificant amount of noise in the position estimates, which arise from the GNSS measurements.

Figure 8 shows the estimated position after being corrected with the output of the final-stage integrated Kalman filter. The position more closely tracks the truth state when compared to Figure 7, which is expected, since the estimate here is based on the cooperative localization output in addition to the IMU mechanization and GNSS measurements. Quantitatively, the 3D RMS position error for both the benchmark and final trajectories were $1.27m$ and $0.920m$ respectively. As a result, there is less noise during GNSS blockages, and the divergence in the case of outages is absent.

To better illustrate the difference in positioning performance, Figures 9 and 10 provide the magnitudes of the position errors for both the benchmark and final cases, where the state estimates are subtracted from the truth states.

The discrepancy in the filters' performances is attributed to the fact that the Doppler-based velocity estimation in the cooperative localization filter only measures the component of the agent's velocity along each neighbour's line-of-sight. This results in persistent 3D velocity errors in directions that are poorly aligned with the LOS vectors of the neighbours, reducing the effectiveness of the cooperation technique relative to simply using the agent's own velocity, even if noisy.

C. Network Size

To observe the effect of increasing the network size N, a new range of simulations were run with the same parameters as

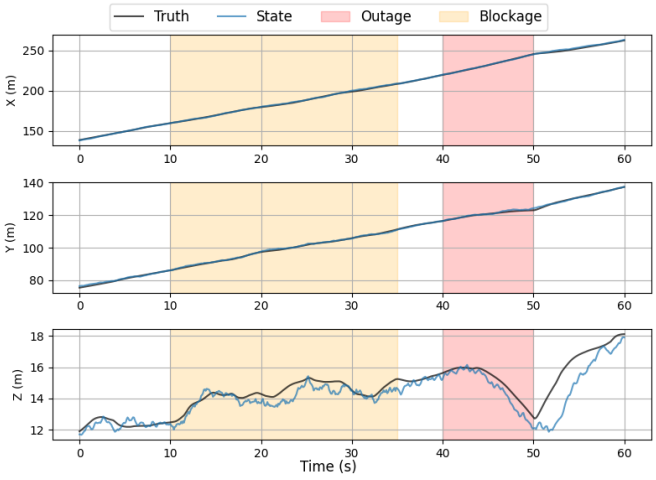

Fig. 8. Integrated Kalman Filter Position

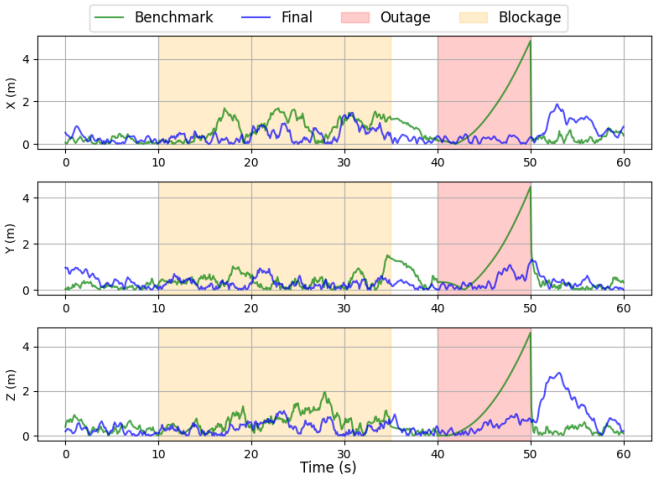

Fig. 9. Error Comparison of Benchmark and Final Kalman Filter Positions

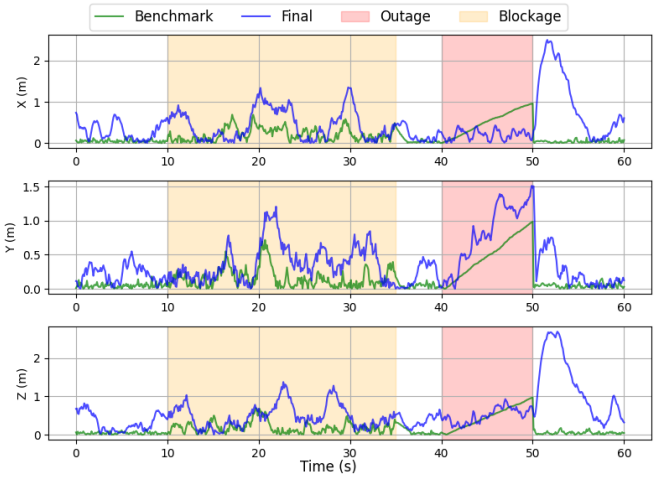

Fig. 10. Error Comparison of Benchmark and Final Kalman Filter Velocities

Metric	Benchmark	Final
$N = 5$		
Position	1.3	8.1
Velocity	0.36	1.2
$N = 6$		
Position	1.6	0.82
Velocity	0.35	0.91
$N = 7$		
Position	1.5	0.82
Velocity	0.37	0.86
$N = 8$		
Position	1.9	0.78
Velocity	0.38	0.92

TABLE I
3D RMS ERROR $(m, m/s)$ FOR VARYING NETWORK SIZE

detailed in Section IV-A, but with an additional GNSS outage for agent 1 between 30 and 40 seconds. Table I summarizes the results relative to agent 0's state estimates.

Although the swarm's formation was randomized between the simulations of different swarm sizes, the network remained strongly connected, meaning that agent 0, which suffered GNSS blockages and outages as described previously, always remained connected to agent 1, which lost its GNSS for 10 seconds. Therefore, since agent 1's degraded GNSS was always included in the trilateration calculations, the results are comparable.

With only five agents, the final Kalman filter's corrective output diverges after agent 1's outage, with an RMS position error of $8.1m$ compared to $0.82m$ once the network size increases to six, as there are fewer than four neighbours available to agent 0. This prevents a full trilateration solution from being computed through the Kalman filter's update step. As soon as the number of agents increases to six, the position estimate of the final filter drastically improves, with the RMS error falling below that of the benchmark (IMU and GNSS) estimation. This improvement occurs because the trilateration filter requires at least four independent neighbours for full 3D localization. With only five agents, agent 0 cannot maintain four neighbours during agent 1's outage. In the end, the pattern of improved states holds as the number of agents continues to increase. Interestingly, velocity error remains relatively constant across increasing network sizes, further supporting the conclusion that cooperative Doppler-based velocity estimation is limited by geometry, and not necessarily agent count. Regardless, these results confirm that larger networks would be more resilient to cross-agent GNSS outages.

V. CONCLUSION

This project explored a trilateration-based cooperative localization framework for multi-agent UAV systems operating in GNSS-degraded environments. The approach integrates inter-agent pseudorange and relative velocity measurements with each agent's own inertial and GNSS data using a cascaded

Kalman filter architecture. Simulation results demonstrate that this fusion scheme can improve position estimation accuracy during periods of GNSS outages or blockages, given that the agent remains connected to at least four neighbours. Moreover, the system shows resilience to sensor noise and environmental disturbances, enabled by a decentralized consensus-based PD controller that maintains formation under constant-velocity motion.

While cooperative localization improves position estimation, velocity estimates remain limited due to the geometric nature of range-rate measurements, which only constrain motion along the line-of-sight to neighbouring agents. Consequently, velocity errors persist in directions that are poorly aligned with the swarm's sensing topology.

In future work, the control law could be extended using a weighted consensus approach, where each agent changes the influence of its neighbours based on estimated localization uncertainty or GNSS quality. This would enhance swarm robustness in cross agent degradation scenarios and allow the system to selectively prioritize reliable neighbours during outage periods. Integrating trust weighting into the control law would also reduce the influence of compromised agents on the overall formation.

In addition, the stability of the cascaded Kalman filter structure used in this project should be analyzed formally. While each filter stage is designed to be individually stable, the open-loop architecture (where downstream filters depend on upstream outputs) introduces the potential for the amplification of errors.

One major assumption made in this project is the synchronicity of the relative position and GNSS data from neighbouring agents. While GNSS single-point positioning relies on accurate satellite clocks and bias correction terms, such mitigations would typically not be available on small, mobile platforms such as unmanned aerial systems. However, effective synchronicity could be assumed if the agents are moving slowly enough relative to the broadcast rate. In any case, an area of further exploration would be to loosen the synchronicity condition and consider messaging time delays between agents, as well as the communication overhead.

Overall, this work demonstrates a resilient GNSS-cooperative localization system that leverages trilateration and filtering to maintain performance in degraded environments, while identifying key limitations and future extensions needed for deployment in larger, more realistic swarm settings.

REFERENCES

[1] SCC1300-D04—Gyro Sensors—Sensors—Murata Manufacturing Co., Ltd. — murata.com. https://www.murata.com/products/productdetail?partno=SCC1300-D04.

[2] Kexin Guo, Xiuxian Li, and Lihua Xie. Ultra-Wideband and Odometry-Based Cooperative Relative Localization With Application to Multi-UAV Formation Control. *IEEE Transactions on Cybernetics*, 50(6):2590–2603, June 2020.

[3] Zhimin Han, Kexin Guo, Lihua Xie, and Zhiyun Lin. Integrated Relative Localization and Leader–Follower Formation Control. *IEEE Transactions on Automatic Control*, 64(1):20–34, January 2019.

[4] Eric J. Kim. Cisc839 project 2 gps single point positioning, 2025.

[5] Thien Hoang Nguyen, Thien-Minh Nguyen, and Lihua Xie. Range-Focused Fusion of Camera-IMU-UWB for Accurate and Drift-Reduced Localization. *IEEE Robotics and Automation Letters*, 6(2):1678–1685, April 2021.

[6] Thien-Minh Nguyen, Abdul Hanif Zaini, Chen Wang, Kexin Guo, and Lihua Xie. Robust Target-Relative Localization with Ultra-Wideband Ranging and Communication. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2312–2319, May 2018. ISSN: 2577-087X.

[7] Thien-Minh Nguyen, Thien Hoang Nguyen, Muqing Cao, Zhirong Qiu, and Lihua Xie. Integrated UWB-Vision Approach for Autonomous Docking of UAVs in GPS-denied Environments. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9603–9609, May 2019. ISSN: 2577-087X.

[8] Thien-Minh Nguyen, Zhirong Qiu, Thien Hoang Nguyen, Muqing Cao, and Lihua Xie. Distance-Based Cooperative Relative Localization for Leader-Following Control of MAVs. *IEEE Robotics and Automation Letters*, 4(4):3641–3648, October 2019.

[9] NovAtel Inc. Oem7500 series gnss receiver modules. https://www.novatel.com/products/gnss-receivers/oem-receiver-boards/oem7500/, 2021.

[10] Hao Xu, Luqi Wang, Yichen Zhang, Kejie Qiu, and Shaojie Shen. Decentralized Visual-Inertial-UWB Fusion for Relative State Estimation of Aerial Swarm. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8776–8782, May 2020. ISSN: 2577-087X.