Progress on GPS-Denied, Multi-Vehicle, Fixed-Wing Cooperative Localization

Gary Ellingson* Tim McLain
Department of Mechanical Engineering
Brigham Young University
Provo, Utah

***Abstract*—This paper first summarizes recent results of a proposed method for multiple, small, fixed-wing aircraft cooperatively localizing in GPS-denied environments. It then provides a significant future works discussion to provide a vision for the future of cooperative navigation. The goal of this work is to show that many, small, potentially-lower-cost vehicles could collaboratively localize better than a single, more-accurate, higher-cost GPS-denied system. This work is guided by a novel methodology called relative navigation, which has been developed in prior work. Initial work focused on the development and testing of a monocular, visual-inertial odometry for fixed-wing aircraft that accounts for fixed-wing flight characteristics and sensing requirements. The front-end publishes information that enables a back-end where the odometry from multiple vehicles is combined with inter-vehicle measurements and is communicated and shared between vehicles. Each vehicle is able to create a global, back-end, graph-based map and optimize it as new information is gained and measurements between vehicles overconstrain the graph. These inter-vehicle measurements allow the optimization to remove accumulated drift for more accurate estimates.**

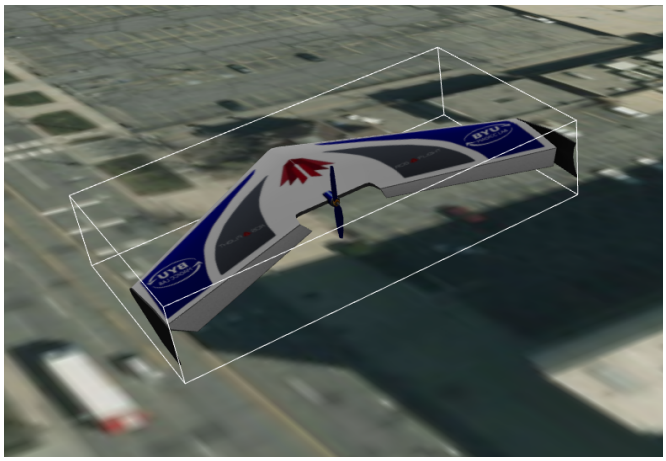

Fig. 1. This work enables GPS-denied navigation on fixed-wing aircraft. This high-fidelity, aircraft simulation was used to test the front-end, odometry estimator.

I. Introduction

More than ever before unmanned aircraft systems (UAS) need the ability to accurately navigate in GPS-denied environments. In both civil and defense applications, UAS need to have an accurate knowledge of their motion to complete their mission objectives. The creation of highly-accurate, miniaturized navigation systems that fuse inertial measurements with GPS measurements (GPS-INS) has allowed UAS to operate in a variety of new applications. These navigation capabilities remain limited because most GPS-INS solutions are brittle to GPS signal degradation and dropout. For example, civil autonomous drone delivery services will need to accurately navigate in and around obstacles where GPS signals are partially or fully obstructed.

Many military defense applications require aerial navigation in areas where GPS signals have been spoofed or jammed. Some applications require long-distance, high-speed flights under the constraint of limited communication with ground-based command centers. In contrast to low-flying delivery and inspection aircraft, these vehicles require less precision because of their distance from obstacles, but need to limit the accumulation of drift over time to achieve their mission objective.

*gary.ellingson@byu.edu

There are many advantages for UAS to be small and inexpensive. Aircraft designers often speak of size, weight, and power (SWaP) constraints that influence trade-offs in the design. Navigation capabilities have similar constraints. GPS-denied solutions that only use inertial measurement units (IMUs) have been successfully implemented, but these solutions are only possible with highly-accurate, prohibitively-expensive, military-grade IMUs that have been precisely calibrated. Small UAS often must utilize sensors that are much less precise and instead use advanced algorithms to account for sensor noise and remove drift from state estimates. Constructing small, lower-quality vehicles make it possible to economically produce more vehicles to perform the mission rather than one, highly reliable vehicle.

GPS-denied navigation on fixed-wing UAS requires specific sensing and estimation considerations. The majority of previous GPS-denied research and development has mainly focused on multirotor aircraft. Fixed-wing UAS differ from multirotors because they have different aircraft dynamics, they generally fly at higher speeds, and they are unable to stop and hover in place. Multirotor UAS are often able to utilize depth sensors, such as laser scanners, to effectively measure their motion because they can fly in and around structure in the environment. On fixed-wing UAS, depth sensors are less

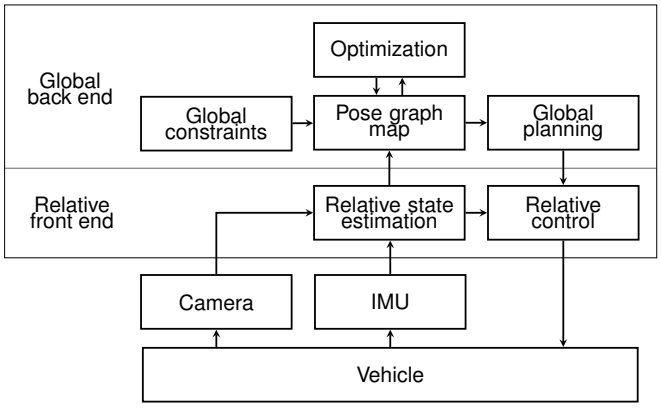

Fig. 2. The relative navigation architecture was developed for GPS-denied navigation. Estimation and control are performed in a front end where the vehicle operated relative to a local coordinate frame. The back end accounts for global information by utilizing odometry from the front end and optimizing it in a global pose-graph map.

effective at measuring the motion of the aircraft because they usually fly high above the environment.

In general small UAS can also benefit from collaboration to produce synergistic effects. Specifically for GPS-denied navigation, collaboration may provide significant advantages. Without position measurements to limit drift, global position and yaw angle are unobservable [1]–[3]. Vehicles must use other exteroceptive sensing to limit how fast estimate drift accumulates. If multiple vehicles could share measurements then the drift of all the vehicles could be further limited and provide even better navigation accuracy.

This paper provides a summary of results from testing the method that was proposed in last year's report [4] to enable multiple, small, fixed-wing UAS to collaboratively localize. A major focus of this paper is to also provide a significant future works section that will discuss the efforts necessary to have the proposed method fully functional and fielded to autonomously complete a mission. This paper summarizes work that is presented in several other papers, including papers that are already published [4], [10], currently under review [16], and yet to be submitted.

II. Previous Works

Our method draws from previous research in three areas: The overall GPS-denied architecture utilizes the relative navigation framework, the front-end, visual-inertial odometry is a modification to the multi-state-constraint Kalman filter (MSCKF), and the back-end optimization comes from the wealth of literature on pose-graph optimization. Relevant work in these areas is summarized in the following sections.

A. Relative Navigation

Researchers have previously proposed a new approach to GPS-denied navigation called relative navigation [5], [6]. It is a methodology and framework that separates the navigation into two sub-tasks. It separates a front-end estimator from a back-end optimization. The front end operates relative to the local surroundings and the back end uses regular updates from the front end to create and maintain a global map. Figure 2 shows the framework architecture.

Relative navigation is motivated by a fairly simple concept called the relative-reset step [7], which is closely related to keyframe-based methods. The concept is for the front-end estimator to regularly declare a new local origin at the location of the vehicle. This also serves to remove uncertainty from the filter because the new origin is defined to be exact. At each new origin the prior transform can be sent to the back end as an edge in a directed pose graph.

The relative navigation approach has several advantages over contemporary methods. It is locally observable by construction and it has better filter consistency compared to other state-of-the-art approaches [8]. The front end has the computational advantages of an extended Kalman filter (EKF). The pose graph used in the back end is able to better represent large, nonlinear errors in odometry estimates. The back end can also incorporate other constraints, such as opportunistic GPS measurements or place-recognition loop closures.

Several tests have been perform to demonstrate relative navigation [9]. Assumptions about vehicle dynamics, sensing, and filtering have mostly limited the tests to multirotor aircraft at relatively low speeds. The approach has also be implemented with the entire architecture on a single vehicle that has enough computational resources. Sensing requirements have ensured the paths are in and around structured environments which have allowed the paths be relatively short and include loops back on themselves. These factors have limited the impact of the relative navigation architecture as a solution to the GPS-denied navigation problem.

B. Relative MSCKF

In the majority of the relative navigation work the front-end state estimator has been called the relative multiplicative extended Kalman filter (RMEKF) [7]. The RMEKF has required a keyframe-based odometry as a measurement and the odometries have used depth sensors, such as laser scanners and RGBD cameras, to resolve scale ambiguity. Fixed-wing aircraft, where RGBD and laser depth sensors are impractical due to the increased distance to features in the environment, require a different approach. Further, the main functions of the RMEKF were to combine inertial and visual odometry measurements and to perform a relative reset at each keyframe declaration. The odometry alone would otherwise be sufficient to provide the back end with odometry edge transformations from pose to pose.

More recently, a new tightly-coupled, visual-inertial odometry has been introduced as a front-end estimator [10]. It uses only monocular imagery, without depth measurements, for exteroreceptive sensing. It combines the odometry calculations, inertial measurements, and relative-reset steps into one filter. This filter was developed specifically to enable fixed-wing UAS to use the relative navigation framework.

The new filter is based on the MSCKF. The MSCKF is more ideal for fixed-wing UAS because it makes no assumptions

about the distance to observed features, requires no depth measurements, and makes no assumptions about the vehicle dynamics. The MSCKF uses a unique measurement model that was originally presented in [11]. It avoids adding uncertainty to the filter by not initializing states that are not well known. Further, updates are performed after a image feature moves out of view and all information about that feature has been obtained.

Since its introduction, the MSCKF has seen extensive development in the literature. It has been demonstrated for use on ground vehicles [12], spacecraft [13], and even smart phones [14]. It has also been compared to several more-recent visual-inertial odometries and its accuracy and consistency properties remain comparable to the state-of-the-art with less computational burden [15]. The complete estimator development and front-end results are presented in [16].

C. Graph Optimization

The relative-navigation back end has, in the past, been used to keep track of the global map by creating a directed pose graph. During the relative reset, the position and heading angle states and covariances are zeroed and the transformation from just before the reset is sent to the global back-end as an edge in the graph. Covariance uncertainty is effectively removed from the front-end filter and sent to a global back end where the pose graph has the ability to represent non-linear uncertainties from yaw better than a Gaussian filter [8]. The back end is able to do edge optimization on the graph of pose estimates to improve global states for performing a global mission. The optimization is also able to incorporate and account for other constraints, such as opportunistic GPS measurements and place-recognition loop closures, for more accurate localization.

Graph-based optimization methods have been effectively used in robotic localization for some time [17]. Advances in computational power and sparse-matrix mathematics have, more recently, increased both the speed at which the optimizations can be performed and the number of nodes, or factors, that can be considered in the graph. Generalized graph optimization (g2o) [18], Georgia Tech smoothing and mapping (GTSAM), and incremental smoothing and mapping (ISAM) [19] are all graph optimization frameworks that have open-source implementations that are available for research. In the past, the relative-navigation back end has used the g2o graph optimization framework but recently other methods have been explored.

GTSAM is a smoothing and mapping toolbox that uses factor graphs to iteratively optimize a bipartite graph [20]. This means that it performs maximum a-posteriori inference through the relationships of states and factors that relate the states. Factors can be sensor measurements or odometry between aircraft poses. Odometry estimates are binary factors and measurements, such as opportunistic GPS or bearing to static features, are unary factors.

Because the global back end uses a pose graph that is a relatively sparse representation of the vehicle odometry, it has potential to be useful for multi-vehicle cooperative localization. Multiple vehicle cooperation has the potential to limit estimate drift over extended flights due to the increased baseline between sensors [21], [22]. Other work has shown that multiple vehicles can collaboratively estimate the vehicle state using poses as factors in factor graph smoothing frameworks [23].

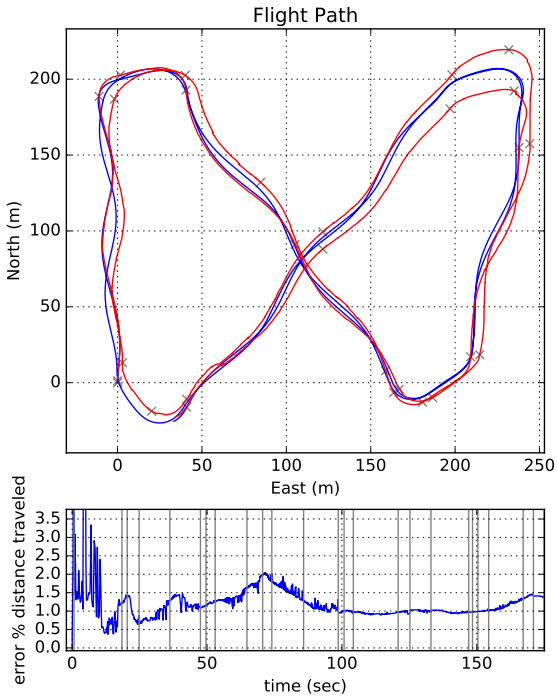

Fig. 3. The front-end estimator was tested in simulation and hardware flight tests. The simulation results show here were originally presented in [4]. Top: The path of the aircraft. The accumulated estimate (red) is compared actual path (blue). Gray $\times$ indicate relative-reset origins. Bottom: Accumulated error is less than 2% of the total distance traveled. Gray vertical lines indicated relative resets. The aircraft flew nearly 2200 m and the filter is nearly twice as accurate as previously reported results.

III. Experiments and Results

Results for this work were obtained in two distinct efforts. The front-end estimator was first developed and tested in simulation as well as on a hardware flight demonstration. The back-end map was then also tested single and multi-vehicle scenarios using simulation, recorded data front-end data, and in a flight demonstration. These results are presented below.

A. Front-end Estimator

The estimator was first tested using the ROS/Gazebo simulation tools that were developed as part of ROSplane [24]. Figure 1 shows an example of the simulation environment. The simulation testing aided the development of the filter and initial simulation results were shown in [10]. The filter was initially coded in the Python programming language and was able to achieve a total accumulated error of less than 3% of the total distance traveled. This result included the caveat that the simulation was run at half speed in order for the filter to keep up with the measurement information.

Some slight improvements to the filter were presented in last year's report [4]. The main improvements came from the reimplementation of the filter in the C++ programing language instead of Python. The greater speed from C++ allowed the filter to run in real time while using more tracked features and more images per second. These improvements allowed the filter to maintain error less than 2% of the distance traveled and were necessary for enabling the filter to run in hardware experiments. Figure 3 shows the results that were presented in that paper, where the top plot show the true and estimated trajectory and the bottom plot shows the error as a percent of the total distance traveled.

Further simulation testing is presented in [16]. This testing uses a higher fidelity simulation that includes more realistic sensor noise and error sources. The simulation testing also demonstrates a slight weakness in the algorithm. That is, when flying straight and level, the forward velocity is unobservable. This weakness is particularly problematic for a fixed-wing aircraft which often fly straight and level for long periods. The best case showed that the filter was capable of maintaining error less than 1% of the distance as long as the trajectory remained observable, but the estimates incurred a large scale bias in the forward direction otherwise.

The flight-test experiments were then performed on a small hobby-grade fixed-wing platform called the STRIX Strato-Surfer by Ready Made RC. The platform was outfitted with a Jetson TX2 flight computer, a monocular camera, and in InterialSense IMU. The IMU was also capable of producing a GPS-INS solution for truth comparison. The aircraft was manually flown over a 6 km trajectory. Several improvements had to be introduced to the filter to account for additional sources of error, including calibration, timing, and initialization errors. The largest modification included adding the camera extrinsic calibration to the state vector and estimating it throughout the trajectory. The observability of the calibration also necessitated the use of a partial update [25], a modification of the Schmidt-Kalman filter [26].

Experiments of the front-end were ultimately successful. The filter could maintain error less 2.5% of the distance travel, while periodically publishing transformations that were appropriate for use in the back-end map. Both simulation and hardware testing revealed the front-end estimates had unobservable forward velocity when flying straight and level and as error accumulates the estimates become biased. Further, these biases are correlated from one edge to the next because velocity estimates are not reset in the keyframe reset step. While this is an unfortunate problem associated with using a visual-inertial odometry, this problem was overcome through modifications to the back-end graph.

B. Back-end Map

To address the velocity bias observed in the front-end results, a modification to the standard graph was introduced in [16]. An addition of a scale-bias factor to the graph, which is similar to gyro bias walk factors used in accelerometer preintegration factor-graph methods.

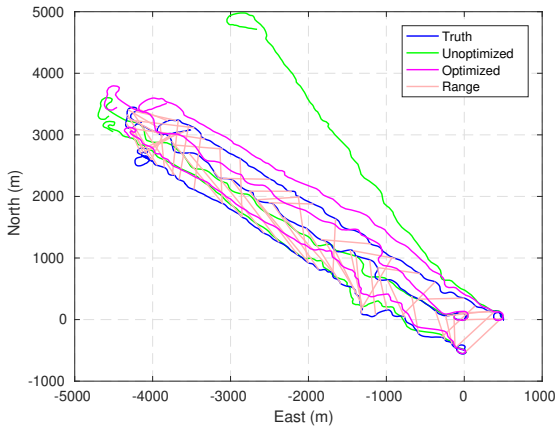

Fig. 4. This back-end graph includes edges from three vehicles, or more accurately three flight tests, with starting location of the second and third vehicles artificially offset by 500 m south and 500 m east respectively. The graph includes simulated inter-vehicle range measurements. The localization accuracy of all vehicles improve and the relative position of the swarm is maintained. Similar results were originally presented in [16].

The results showed with the scale-bias factors and opportunistic global measurements the back-end graph was able to optimize for an accurate vehicle trajectory. The back-end results in [16] used the recorded front-end transformations, as well as simulated global measurement and optimized the trajectory in post-process. The simulated global measurements utalized the true position recorded from the InterialSense GPS-INS, and consisted of global position measurements from either a GPS receiver or satellite image-based localization, as well as distance measurements to ground-based ranging systems.

To show the potential for cooperative localization, the work in [16] also combined several recorded flights into one graph. The graph also included simulated inter-vehicle range measurements. The graph was optimized using GTSAM and showed that the vehicle localization improved due to cooperation. An example of the results that provided in the paper are shown in Figure 4. The results, as well as some limited analysis, showed the potential for the multi-vehicle cooperative localization that was proposed in last year's report [4].

While the potential of cooperative localization was shown by the results demonstrated thus far, the results also have limitations. Foremost, the results have only been produced by post-processing recorded data and only optimize once at the end of the trajectory. Further, the results do not address concerns with communication, including how and when the communication between vehicles happens. Finally, one result fails to quantify the total benefit of cooperative navigation. For example, local minima in the optimization have the potential to worsen the localization accuracy in some instances.

A simulation was then developed to help address some of the shortcomings of the previous results. The simulation includes several simulated agents each with a simplified front-end estimator, where the statical accuracy matches the performance of the previously developed front end. The agents

Fig. 5. Hardware tests of the complete system were performed on a STRIX StratoSurfer, RC-grade aircraft. The cooperative localization was tested with one complete agent and other agents utilizing recorded data from previous flights. This means that the communication was real and the test was completely realistic to the aircraft's perspective.

can also periodically communicate with the each other and request the missing information for the complete graph. Each agent also has its own back-end and the graph is optimized every time a inter-vehicle measurement is taken between the vehicles. The simulation also enables Monti-Carlo testing to quantify the benefits of the approach.

The final result is to produce the complete system using the hardware setup described previously. In addition, the vehicle uses a small 915 MHz telemetry radio for communication. The protocol used in the simulation was then implemented using the MAVLink protocol. Flying multiple vehicles simultaneously was beyond the scope of the project so the single vehicle instead communicated with a ground vehicle that was playing the front-end data recorded from previous flights. The setup is shown in Figure 5. Each vehicle, both recorded and actual, had a back-end that included the front-end edges and range measurements between vehicles. The aircraft was flown high above the environment while relatively straight-and-level over the 6 km trajectory. The graph was again optimized at every range measurement and produced real-time results that were, at the end of the flight, similar to those shown in Figure 4.

These final results, both simulation and hardware tests, were recently obtained and will be included in a future publication. The publication will also describe necessary changes to the architecture in Figure 2 to allow for cooperative localization. The results show that cooperation between vehicles is valuable for improving the accuracy of the localization and that the method is robust to limited communication, including delay and momentary dropout.

IV. Future Work

While the results presented are significant for improving localization and navigation capabilities of fixed-wing UAS, localization and navigation are only part of what is necessary to fly a mission and achieve an objective. This section will outline several improvement, extensions, and alternate methods that could be used to both improve the proposed method and to extend the capabilities of the system.

A. Planning and Control

First, in this work the system architecture shown in Figure 2 is incomplete, though it has been demonstrated in it's entirety elsewhere [6], [9]. Specifically this work has focused on navigation and localization but experiments were flown manually and did not include the planner and controller necessary to fly autonomously. Thus far the front-end and back-end division of a controller has not been explored for a fixed-wing UAS but would be necessary for completing a fully autonomous mission.

The controller for cooperative navigation has also not been explored. The vehicles need a way to cooperatively complete a mission as well as de-conflict trajectories that would cause a crash. A centralized planner would be straitforward but would go against the decentralized and opportunistic nature of the proposed method. More elegant would be decentralized planner that uses a consensus algorithm to help all the vehicles arrive at the same plan. This would require a major modification to the proposed communication scheme and would require significant testing to make sure the solution remains robust to communications dropout and delay.

The cooperative controller could also be constructed to improve accuracy in the back-end graph. The graphs ability to remove accumulated drift is correlated with the quality of the constraints that it creates between aircraft. For example, when the UAS are all flying in the same direction and at the same speed, inter-vehicle range measurement only help to remove yaw error. If instead a single UAS flew perpendicular to the others, the position accuracy would also increase because the constraints on the graph would effectively provide more information. A controller could be developed to exploit this phenomenon. It could, for example, periodically have one agent do some sharp turns or even vary the commanded airspeed between agents to gain the benefit.

B. Front-End Improvements

The next group of potential improvements could focus on improving the front-end odometry method. The choice of the MSCKF measurement model has many advantages, but more modern visual-inertial odometries can be more accurate [15]. The MSCKF particularly struggles and estimates can diverge in two cases; in straight-and-level flight and also when the UAS is stopped, such as sitting on the ground or hovering in place. Other odometries, such as VINS [27], can maintain accurate estimates for longer but ultimately the states are unobservable and estimate will eventually become inaccurate.

Forward velocity unobservability can be remedied by controlling the UAS such that it remains observable, as we have demonstrated [16], or perhaps more elegantly, adding more sensing modalities. Feasible options include airspeed sensors or single-point depth sensors. An airspeed sensor would, however, measure the forward velocity with respect to the airmass and therefore would require modeling the wind, which may be challenging. Relying on a depth sensor, such as a sonar or single laser altimeter, could improve the scale of the estimates by adding constraints on the distance to observed features. A model for combining a depth sensor in combination with a camera is presented in [28]. This type of approach could be applied to the MSCKF measurements. The MSCKF performs a least-squares optimization to solve for the feature location. A simple distance constraint on the optimization could improve estimates of the scale and thus the velocity of the aircraft.

C. Map Generation

The map could also be improved and better utilized. The current capability primarily uses the map for measure the localization error. The visual map could be created to facilitate completion of various mission objectives, including surveillance and target identification. A simple but effective way to accomplish this would be by tiling the keyframe images to create a visual representation of the terrain. Tiling the keyframe images could also provide constraints to the back-end optimization to improve global estimates.

Tiling the images for map creation requires the knowledge of distance to the images as well as estimates from one image to the next. Simple approaches use the altitude of the aircraft to obtain the distance and then assume the terrain is flat, using the so-called flat-earth model. In our approach the front end provides the inter keyframe transforms. The MSCKF measurement model also solves for the location of the observed features in the keyframe coordinates frame. This means the image provides dense appearance information and the features become a sparse point cloud with corresponding location information. If the terrain is fairly smooth, meaning not sharp cliffs or jagged corners, an image could be draped over the point cloud to provide a rough shape of the terrain for a given keyframe image. The separate keyframe images could then be combined using a combination of iterative closest point and image transformation methods. Map creation, however, greatly increases the complexity of sharing information between cooperative agents and would require significant communications changes.

D. Back-End Measurements

Exploring various other types of back-end measurements, as well as alternate use cases, may highlight the practical advantages of the proposed method. A potential scenario was partially demonstrated in a previous paper [16], where only one vehicle has access to global measurements but the reset of the vehicles gained the localization advantages of those measurements. A small alteration to the scenario would be to have one vehicle use a global measurement from a satellite image-based localization. A similar method is used in [13] to initialize an entry-decent-and-landing filter of a spacecraft approaching a planetary body. There approach uses prerecorded satellite images to find the current location of the aircraft but are complex and computationally expensive.

Another relevant scenario would be applying the relative navigation architecture and cooperative localizing when multiple vehicles are trying to achieve a goal location, are performing target tracking, or are prosecuting a enemy vehicle. In this scenario not all vehicles are able to measure the location of the target. The blind vehicles must rely on the targeting capabilities of one or more other UAS to know their relative position. Since the inter-vehicle relative position is maintained by the proposed method then this may be an effective way to increase the capability of the less capable agents.

V. Conclusion

This research contributes to the maturation of small unmanned aircraft. Before introduction into the national airspace or use in military applications, small unmanned aircraft will need greater reliability and to be robust to GPS signal degradation and dropout. This research utilizes state-of-the-art methods to expand the capabilities of these aircraft.

This work has demonstrated a feasible method for collaboratively localizing fixed-wing UAS in GPS-denied environments. The work is significant because it directly acknowledges and addresses challenges of GPS-denied, fixed-wing UAS. GPS-denied solutions for multi-rotor aircraft have been fairly common, but are less common for fixed-wing aircraft. Often, solutions that do exist make significant simplifying assumptions, such as operating over flat-earth or in Manhattan world environments, or having complex or unreasonable sensing requirements, such as downward facing camera or depth measurements. This work uses minimal sensing (only camera, IMU, and inter-vehicle range) and makes no such simplifying assumptions. It further enables GPS-denied navigation within a collaborative framework capable of incorporating inter-vehicle measurements from multiple aircraft. These measurements over constrain the graph and allow the graph smoothing and optimization to remove accumulated error from the graph.

The work also extends the impact of the relative navigation framework. It allows the value of the framework to be shown for a alternate type of vehicle with a different mission profile. Since the back-end constrains the graph with inter-vehicle measurements and not with loop closures, the aircraft will be able to fly in relatively long, straight paths at high speeds. These mission profiles may be more representative of real-world UAS scenarios.

A number of results were summarized. The approach was first demonstrated using a high fidelity aircraft simulation of a small fixed-wing aircraft with simulated sensors. The simulated aircraft dynamics and sensor-noise characteristics were representative of those from an actual small, unmanned fixed-wing aircraft. Both provided here and in previous works the accuracy and consistency of the relative odometry were

presented, as well as hardware results that use a GPS-INS for ground-truth comparison. The multi-aircraft cooperative flight demonstration, enabled by a modified relative-navigation back end and the tightly-coupled, visual-inertial front end, demonstrate the value of the complete system.

Several proposed future works were also proposed. The proposals focus on both completing the architecture and extending the capabilities of the system to additional compelling scenarios.

ACKNOWLEDGMENT

Gary Ellingson receives generous funding from the Utah NASA Space Grant Consortium Fellowship. This paper is written in partial fulfillment of a requirement for the fellowship.

This work has been funded by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF award No. IIP-1161036 along with significant contributions from C-UAS industry members.

REFERENCES

[1] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 957–964, IEEE, 2012.

[2] E. Jones, A. Vedaldi, and S. Soatto, "Inertial structure from motion with autocalibration," in *Workshop on Dynamical Vision*, vol. 25, 2007.

[3] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.

[4] G. Ellingson, "Toward GPS-denied, multi-vehicle, fixed-wing cooperative localization," tech. rep., NASA Utah Space Grant Consortium, 2018. Available at https://digitalcommons.usu.edu/spacegrant/2018/Session$_two/25/$.

[5] R. C. Leishman, T. W. McLain, and R. W. Beard, "Relative navigation approach for vision-based aerial GPS-denied navigation," *Journal of Intelligent and Robotic Systems*, vol. 74, no. 1–2, pp. 97–111, 2014.

[6] D. O. Wheeler, P. W. Nyholm, D. P. Koch, G. J. Ellingson, T. W. McLain, and R. W. Beard, "Relative navigation in GPS-degraded environments," *Encyclopedia of Aerospace Engineering*, 2016.

[7] D. P. Koch, D. O. Wheeler, R. Beard, T. McLain, and K. M. Brink, "Relative multiplicative extended Kalman filter for observable GPS-denied navigation," 2017.

[8] D. Wheeler, D. Koch, J. Jackson, T. McLain, and R. Beard, "Relative navigation: An observable approach to GPS-degraded navigation," *IEEE Control Systems, Accepted*, 2017.

[9] D. O. Wheeler, D. P. Koch, J. S. Jackson, G. J. Ellingson, P. W. Nyholm, T. W. McLain, and R. W. Beard, "Relative navigation of autonomous GPS-degraded micro air vehicles," 2017.

[10] G. Ellingson, K. Brink, and T. McLain, "Relative visual-inertial odometry for fixed-wing aircraft in GPS-denied environments," in *Position, Location and Navigation Symposium (PLANS), 2018 IEEE/ION*, IEEE, 2018.

[11] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Robotics and automation, 2007 IEEE international conference on*, pp. 3565–3572, IEEE, 2007.

[12] L. E. Clement, V. Peretroukhin, J. Lambert, and J. Kelly, "The battle for filter supremacy: a comparative study of the multi-state constraint Kalman filter and the sliding window filter," in *Computer and Robot Vision (CRV), 2015 12th Conference on*, pp. 23–30, IEEE, 2015.

[13] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 264–280, 2009.

[14] M. Shelley, *Monocular visual inertial odometry on a mobile device*. PhD thesis, Masters thesis, Technischen Universitat Munchen, 2014.

[15] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," *Memory*, vol. 10, p. 20, 2018.

[16] G. Ellingson, K. Brink, and T. McLain, "Relative navigation of fixed-wing aircraft in GPS-denied environments," *under-review*, 2019. Available at https://scholarsarchive.byu.edu/.

[17] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[18] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," pp. 3607–3613, May 2011.

[19] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, "Multiple relative pose graphs for robust cooperative mapping," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 3185–3192, IEEE, 2010.

[20] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," tech. rep., Georgia Institute of Technology, 2012.

[21] R. Sharma, S. Quebe, R. W. Beard, and C. N. Taylor, "Bearing-only cooperative localization," *Journal of Intelligent & Robotic Systems*, vol. 72, no. 3-4, p. 429, 2013.

[22] R. Sharma and C. Taylor, "Cooperative navigation of MAVs in GPS denied areas," in *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pp. 481–486, IEEE, 2008.

[23] L. R. Sahawneh and K. M. Brink, "Factor graphs-based multi-robot cooperative localization: A study of shared information influence on optimization accuracy and consistency," 2017.

[24] G. Ellingson and T. McLain, "ROSplane: Fixed-wing autopilot for education and research," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, 2017.

[25] K. M. Brink, "Partial-update Schmidt–Kalman filter," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 9, pp. 2214–2228, 2017.

[26] S. F. Schmidt, "Application of state-space methods to navigation problems," in *Advances in control systems*, vol. 3, pp. 293–340, Elsevier, 1966.

[27] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[28] R. Munguía, S. Urzua, Y. Bolea, and A. Grau, "Vision-based SLAM system for unmanned aerial vehicles," *Sensors*, vol. 16, no. 3, p. 372, 2016.