A Vision-aided Inertial Navigation System for Agile High-speed Flight in Unmapped Environments

Distribution Statement A: Approved for Public Release, Distribution Unlimited

Ted J. Steiner
Draper
555 Technology Square
Cambridge, MA 02139
617-258-4165
tsteiner@draper.com

Robert D. Truax
Draper
555 Technology Square
Cambridge, MA 02139
617-258-2872
rtruax@draper.com

Kristoffer Frey
Massachusetts Institute of Technology
77 Massachusetts Ave
Cambridge, MA 02139
617-258-4165
kfrey@mit.edu

***Abstract*—Small, lightweight flight vehicles, such as consumer-grade quadrotors, are becoming increasingly common. These vehicles' on-board state estimators are typically reliant upon frequent and accurate updates from external systems such as the Global Positioning System (GPS) to provide state estimates required for stable flight. However, in many cases GPS signals may be unavailable or unreliable, and loss of GPS can cause these vehicles to go unstable or crash, potentially putting operators, bystanders, and property in danger. Thus reliance on GPS severely limits the robustness and operational capabilities of lightweight flight vehicles.**

This paper introduces the Smoothing And Mapping With Inertial State Estimation (SAMWISE) navigation system. SAMWISE is a vision-aided inertial navigation system capable of providing high-rate, low-latency state estimates to enable high-dynamic flight through obstacle-laden unmapped indoor and outdoor environments. SAMWISE offers a flexible framework for inertial navigation with nonlinear measurements, such as those produced by visual feature trackers, by utilizing an incremental smoother to efficiently optimize a set of nonlinear measurement constraints, estimating the vehicle trajectory in a sliding window in real-time with a slight processing delay. To overcome this delay and consistently produce state estimates at the high rates necessary for agile flight, we propose a novel formulation in which the smoother runs in a background thread while a low-latency inertial strapdown propagator outputs position, attitude, and velocity estimates at high-rate. We additionally propose a novel measurement buffering approach to seamlessly handle delayed measurements, measurements produced at inconsistent rates, and sensor data requiring significant processing time, such as camera imagery.

We present experimental results high-speed flight with a fully autonomous quadrotor using SAMWISE for closed-loop state estimation from flight demonstrations during the DARPA Fast Lightweight Autonomy (FLA) program in April and November of 2016. SAMWISE achieved less than 1% position error and up to 5.5 m/s (12 mph) flight in a simulated indoor warehouse environment using a scanning-lidar, inertial measurement unit, and laser altimeter during the first FLA milestone event in April 2016. In November 2016, SAMWISE achieved approximately 3% error and up to 20 m/s (45 mph) flight in an open outdoor environment with large obstacles during the second FLA milestone event. The results of these flight tests demonstrate that our navigation system works robustly at high speed across multiple distinct environments.

TABLE OF CONTENTS

978-1-5090-1613-6/17/$31.00 ©2017 IEEE.

Figure 1: Draper/MIT FLA quadrotor configured for autonomous flight.

1. INTRODUCTION

Small, lightweight flight vehicles, such as consumer-grade quadrotors, have become commonplace for numerous applications, such as movie and sports filmography, remote inspection, and hobby usage. These vehicles require on-board state estimation (real-time determination of vehicle position, attitude, and velocity) to maintain stability, to provide controllability for manual pilots remotely operating the vehicle, and to enable autonomous operation, including obstacle avoidance and path planning.

The DARPA Fast Lightweight Autonomy (FLA) Program [1] aims to develop algorithms for high-speed, robust, autonomous flight in unknown environments using off-the-shelf quadrotor hardware. The program goals include several flight demonstration events in which academic and industry teams demonstrate their capability to fly a standardized vehicle platform to a goal coordinate and back through an environment containing unknown, unmapped obstacles as fast as possible, without any communication to the vehicle during the flight (including GPS signals). As part of the DARPA FLA program, Draper and MIT are currently developing novel algorithms for lightweight multi-rotor autonomy, guidance, navigation, and control.

High-speed, aggressive flight requires high-rate, low-latency

attitude control to enable rapid, stable maneuvering and quick response to disturbances, such as wind gusts. Additionally, high-rate position and velocity estimates are generally required to enable basic autonomy "primitives" such as navigating to a goal, mapping a region, and tracking of dynamic trajectories.

SAMWISE

This paper introduces a new system for robust state estimation for high-speed autonomous flight in unknown environments called Smoothing And Mapping With Inertial State Estimation (SAMWISE). SAMWISE utilizes a factor-graph representation and a sliding-window incremental smoother, which provides several advantages over more traditional extended Kalman filter (EKF) approaches such as [2]–[4]. The factor-graph, described in detail in Section 4, provides a convenient and powerful representation for adding new variables and measurements to the estimator [5]. Smoothing, as opposed to traditional filtering, optimizes a MAP estimate over a *set* of recent states, rather than simply the current one. This facilitates iterative re-linearization of nonlinear measurements, such as visual feature observations, significantly reducing the error due linearization. Additionally, this means that old measurements can be re-evaluated given the information provided by new measurements (i.e., non-causally). Smoothing also provides a natural means to handle latent measurements without additional overhead. We use a sliding window to marginalize out states that are older than a certain time threshold, effectively bounding the number of variables being estimated at any one time.

In addition to the application of an incremental smoother to fast, GPS-denied autonomous flight, SAMWISE provides three novel contributions for vision and lidar-aided inertial navigation:

1. High-rate, low-latency state estimate output using strap-down inertial propagation, resulting in 100 Hz output with sub-millisecond latency, enabling agile flight and obstacle avoidance.
2. A measurement buffering system capable of handling variable-latency sensor data from multiple input channels.
3. The use of "ghost" measurements to maintain temporal correctness of measurements requiring high-latency pre-processing or outlier detection without estimator output delays.

SAMWISE can use any combination of lidar, GPS, vision, magnetometer, barometer, and laser altimeter sensors with an inertial measurement unit (IMU) to produce state estimates for autonomous flight. In this paper we focus specifically on two sensor configurations that provide robust state estimation capabilties for vehicles moving through unmapped indoor or outdoor environments at high speeds without the use of GPS:

1. 2D scanning lidar, vertical laser altimeter, and IMU, and
2. Camera, IMU, and barometer or laser altimeter.

Section 5 provides experimental results from closed-loop flight in both sensor configurations during DARPA FLA events using the fully autonomous Draper/MIT FLA quadrotor, shown in Figure 1.

2. Inertial Navigation Systems

Inertial sensors are ideal for high-rate, low-latency state estimation. An inertial measurement unit (IMU) provides high-rate acceleration and rotation rates in all 6 degrees of freedom (DOF) and is capable of capturing the highly dynamic motion of agile flight vehicles, which can achieve several G's of acceleration. Acceleration measurements from the IMU's three accelerometers can be integrated to estimate vehicle velocity and position, and rotation rate measurements from the IMU's three gyroscopes can be integrated to estimate attitude (roll, pitch, yaw angles). However, inertial sensors typically have significant, time-varying biases, and the integration error due to sensor noise accumulates rapidly, especially in position. Therefore, while an IMU provides an excellent core for a flight state estimator, additional sensors are required in order to observe the IMU biases and correct for long-term drift due to accumulated integration error. This is often referred to as an "aided" inertial navigation system (INS).

GPS-aided Inertial Navigation

One frequently used source of INS-aiding measurement data is GPS position estimates. GPS position estimates are ideal due to their ease of use, high availability in outdoor environments, and lack of long-term drift. However, in many cases GPS signals are unavailable (e.g., indoor environments), unreliable (e.g., urban canyons with limited visibility), or jammed (e.g., war zones). The temporary loss or degradation of GPS signals leads to a corresponding loss of bias observability, which can cause unmanned aerial vehicles to go unstable and crash. Even if IMU biases have been well-calibrated offline and can be treated as constant in time, the lack of external observability to correct position and velocity drift can make even basic autonomous tasks like hovering in place unreliable. Therefore, robust operation independent of GPS and the ability to exploit *opportunistic* sensor modalities such as vision are highly desirable for many real-world UAV tasks.

Lidar-aided Inertial Navigation

Scanning lidar and visual-light cameras provide two alternative sensing modalities for GPS-degraded and GPS-denied navigation. A scanning lidar actively senses the range to nearby surfaces in the environment, most typically along a single body-fixed plane to the vehicle, and can also be used for obstacle detection. Subsequent lidar scans can be matched against one another ("scan-matching") to measure changes in position and rotation within the plane of the scan (3-DOF). Scan-matching is relatively computationally efficient and performs well in environments with extensive vertical structure and orthonormal planes, but quickly degrades when such structure is not present. Furthermore, scan-matching assumes that subsequent scans are co-planar, but high-speed quadrotor flight requires large out-of-plane motions with non-zero pitch and roll angles for vehicle maneuvering. This can be somewhat mitigated by removing "ground-plane hits" and projecting the 3D points into the horizontal plane, but at a fundamental level, UAVs reliant on scan-matching must limit their pitch and roll angles, and correspondingly their acceleration, in order to preserve the quality of their state estimates.

2D scan-matching alone cannot provide observability of vehicle altitude or tilt (pitch and yaw), and therefore additional sensors must be used. Pitch and yaw are observable directly from the IMU due to the observability of the gravitational vector with known magnitude. An altimeter, such as a barometer or laser ranger can be used to estimate the vehicle's vertical position. Thus, the combination of an IMU, scanning lidar, and laser or barometric altimeter results in a viable UAV navigation system configuration with observability into all

6 DOF for sufficiently orthonormal environments, such as a warehouse.

Vision-aided Inertial Navigation

Visual feature tracking from camera imagery offers a highly informative alternative to both GPS and lidar. Cameras use a projective transformation to provide a 2D snapshot of the 3D world. Tracking opportunistic visual features detected in camera imagery provides a full 6-DOF relative motion estimate (with a scale ambiguity), enabling estimation of all six IMU biases. However, visual feature observations are highly nonlinear and are poorly modeled by standard Gaussian noise models, making them non-trivial to properly represent in traditional Extended Kalman Filter (EKF) frameworks. Numerous clever approaches have been presented to handle their uncertainty [6]–[8].

Like lidar, cameras are fully self-contained and are not reliant on external infrastructure, greatly reducing the risk of adversarial interference. Cameras are low-cost and pervasive, in many cases already serving as a valuable payload on UAVs. However, images are data rich and visual processing is expensive, requiring a data-reducing pre-processing step to generate usable measurements, many of which are erroneous. Cameras also require precise calibration, and the quality of feature measurements is highly dependent on properties of the environment, such as texture and light levels, and vehicle motion, which can induce blurring. This means that in many cases an image (or several consecutive images) may fail to produce valid measurements.

Because of these drawbacks, vision alone is insufficient for UAV navigation. However, inertial and vision modalities are highly complementary and together can facilitate robust UAV navigation. The IMU provides guaranteed continuity (aside from sensor failure) by producing full 6-DOF measurements independent of environment or trajectory at a constant rate, and especially capturing high-rate, short-period motion. Inertial data is computationally inexpensive to process, and captures the absolute scale information from the observation of acceleration due to gravity that is unobservable from opportunistic imagery alone. Vision provides multi-pose constraints which reduce drift from IMU integration and grant observability into IMU biases. However, the accuracy, quantity, and rank of these constraints are environment and trajectory dependent. A fused vision and inertial navigation system combines the advantages of both approaches and accumulates error more slowly over time than either technique on its own, producing a full position, attitude, and velocity state estimate throughout the vehicle trajectory.

Through observability of the gravity vector of known magnitude, an INS can compute drift-free estimates of *absolute* vehicle pitch and roll. Additional sensors such as a magnetometer and barometer can be used to provide observability into *absolute* altitude and yaw (heading) angle, correcting for long-term drift. The *global* x, y position is unobservable but could be corrected through *a priori* knowledge of the initial location, occasional GPS measurements, or visual observation of distinct landmarks with *a priori* knowledge of their positions, if available.

3. Related Work

Visual odometry (VO) systems, which estimate the motion of a vehicle purely via the vision data from a rigidly-mounted camera or set of cameras, can provide highly-accurate odometry estimates while requiring minimal integration with the rest of the system. Scaramuzza and Fraundorfer give an excellent overview of VO and it's major developments over the past 30 years in their tutorial article [9]. Modern VO systems are very accurate over short durations and smooth, slow motion (typically between 0.1 and 2% relative error at each frame) and are often combined with an IMU (termed visual-inertial odometry, or VIO) to help handle brief periods of rapid motion[10]–[13].

Nistér, Naroditsky, and Bergen presented the first real-time implementation of a visual odometry system with robust outlier detection in a calibrated framework [14]. This was achieved by computing frame-to-frame motion estimates using the 5-point relative pose algorithm [15], [16], eliminating outlier features using Random Sample Consensus (RANSAC) [17], and performing a bundle adjustment refinement on the most recent frames. A large number of subsequent VO developments have been presented [18]–[21], leading to reliable systems that can provide high accuracy over long durations, such as 0.1% error on 10 km of rough terrain by [22] or 0.3% error on 21 km of city streets [12].

One state-of-the-art approach to VIO is the is the Multi-State Constraint Kalman Filter (MSCKF) [12], [23]. The MSCKF is a reformulation of the traditional EKF-SLAM solution [2], [3] which, instead of explicitly representing each unknown landmark within the state vector, incorporates it only *after* it has left the field of view as a single measurement constraining the set of poses from which it was observed. This requires keeping a window of recent poses in the filter rather than a single state, but this allows many more landmarks to be tracked as they do not contribute to the size of the state vector or error covariance matrix. Additionally, landmarks are only triangulated and linearized *once* rather than at each observation, significantly reducing linearization error.

A growing, alternative class of approaches optimize a bundle adjustment or "smoothing" formulation over a *set* of poses and landmarks at each update. They exploit sparsity in the graph structure of the SLAM problem to efficiently and iteratively solve for the trajectory and landmark locations simultaneously. In contrast to the EKF or MSCKF, such approaches iteratively re-linearize the nonlinear system according to a Gauss-Newton or Levenburg-Marquardt optimization, leading to a more accurate solution. Additionally, they avoid representing the full (dense) error covariance or information matrix over all the variables, making estimation of a large number of landmarks feasible in real-time.

In order to keep computation bounded, such systems often marginalize or condition out old variables, effectively maintaining a "sliding window" of variables being actively estimated [20], [24]. Additionally, iterative approaches such as Incremental Smoothing and Mapping (iSAM) [25] use a COLAMD heuristic to represent arbitrary graph optimization problems in a tree structure, and then limit update computation only to the set of variables "locally" affected by new measurements.

Many of the above methods have demonstrated good accuracy and consistency results, at least in slow, smooth trajectories, though challenges remain in actually integrating such systems with high-rate control and planning. The robust use of VO or VIO in a closed-loop system under fast, aggressive motion has not yet been demonstrated. In this paper, we develop an incremental, sliding-window optimization framework and explicitly address some of the challenges of closed-

loop integration. Additionally, we demonstrate the versatility and robustness of this system in two different configurations and environments for the DARPA FLA program in April and November 2016.

4. SAMWISE NAVIGATION SYSTEM

The SAMWISE navigation system provides optimal multi-sensor fusion while maintaining high-rate, low-latency state estimate output by using a combination of an IMU and any of several aiding sensors. SAMWISE uses sensor data from multiple input streams to construct a Bayesian factor-graph representation similar to that of Indelman, et al. [26], which is then solved using incremental Smoothing and Mapping 2 (iSAM2) [25], [27], available in the GTSAM library [28]. SAMWISE provides a flexible inertial navigation front-end to GTSAM allowing incorporation of multiple other sensor modalities, including vision, laser scan-matcher, barometer, and laser altimeter measurements. SAMWISE handles measurement generation from raw sensor data and is robust to transmission and processing latencies, relying on low-latency inertial state estimation to maintain high-rate state output between lower-rate iSAM2 optimization calls.

In order to maintain bounded computation and solve time, we implement a modified form of the Incremental Fixed-Lag Smoother (IFLS) [24], which provides a method for efficient marginalization of old state variables within the iSAM2 framework. Our modification allows the IFLS to additionally handle variables that are added to the factor graph at lower rates than the vehicle state (pose and velocity). This enables the use of online IMU, barometer, and camera calibration variables in the IFLS framework by temporarily extending the window as necessary to ensure all variables are optimized at least once before being marginalized and that active calibration variables remain within the window.

Factor-graph Representation and Optimization

Factor-graph-based incremental smoothing provides a common framework for optimization-based sensor fusion problems. This approach builds upon recent work in the field of Simultaneous Localization and Mapping (SLAM) for mobile robotics. Factor-graphs provide a unified perspective to summary-propagation algorithms, in which "messages" are passed along the edges of a graph, which include probability-propagation algorithms such as the Kalman filter, belief propagation, and optimal smoothing [29].

A factor-graph is a bipartite graph consisting of both factor and variable nodes. Variable nodes represent unknowns in the problem, such as the vehicle's pose. Factor nodes represent probabilistic constraints on the variables, each of which is modeled as a weighted error function to be minimized. Edges in the graph connect factors to the variables they constrain. (1) defines a generic factor $\mathcal{F}_i$ that relates variables v_k^i and measurement z_i through an error function, $err(v_k^i, z_i)$, and a cost function, $d(\cdot)$. Superscript i here refers to all variable nodes constrained by the factor $\mathcal{F}_i$, and k is the current time-step.

$$f_i(v_k^i) = d(\text{err}_i(v_k^i, z_i)) \tag{1}$$

For example, (2) shows a factor for a measurement with additive Gaussian noise and a nonlinear measurement model, where z_i is the measurement, $h(v_k^i)$ is the nonlinear measurement function, and Σ is the noise covariance matrix.

$$f_i(v_k^i) = \exp\big\{-\frac{1}{2}||h(q) - z_i||_\Sigma^2\big\} \tag{2}$$

These factor functions represent the factorization structure of the joint probability distribution, allowing for optimization over the full joint without the need to ever explicitly compute it [28]. This has led to the popularization of the factor-graph representation for solving localization and SLAM problems. Section 4 shows how SAMWISE generates specific measurement factors using various sensor data. After the measurement factors have been constructed and added to the factor-graph we can proceed to perform inference on this graph to estimate the vehicle trajectory using the iSAM2 solver [25], [27].

Measurement Generation

Measurements can be modeled using one of the three basic types of measurement factors defined by [26]. The first class are unary measurement factors, which handle external measurements to an absolute frame, including prior location information. We can use unary factors to model global-frame measurements, such as from a GPS receiver, if available. The second class of factors encode relative-pose measurements that fully constrain the relationship between two poses, such as from an IMU or laser scan matcher. The third class of factors represent measurements that partially constrain (i.e., are low-rank) the relationship between either two poses or a pose and a terrain feature, such as line-of-sight observations of opportunistic visual features. Projective camera measurements are low-rank, and thus camera-based feature observations can constrain only two of an opportunistic feature's three position variables. This means that at least two observations of an opportunistic feature are required before the feature can be added to the graph. These three classes of factors can model any pose-pose and pose-landmark constraints, as well as global-frame pose measurements or priors.

Measurement Buffering and Timing

Real-time sensor fusion algorithms suffer from two significant challenges related to measurement timing. First, it is possible for measurements from different sensor streams to be received in non-chronological order. To overcome this problem, SAMWISE uses a measurement buffering system to perform time-alignment of all input data. Second, some sensors require more computation to generate measurements than others, such as feature extraction from vision or scan-matching from a 2D lidar. We describe a technique using what we call "ghost" variables that overcomes issues associated with highly-latent measurements.

Measurement Alignment—SAMWISE is capable of utilizing measurements from many different sources and data streams, such as a camera, IMU, or laser altimeter. In many cases, these sensors are read by separate pieces of hardware with independent clocks and then transmitted to a host computer, such as over USB connections. This means that sensor data can arrive at the SAMWISE interface out-of-order and with varying latencies. We account for this using two measurement techniques. First, we assign two times to all sensor measurements: The sensor's own clock tags the data with its "time-of-validity," and the host computer tags the data with a "timestamp" the moment it is received. The time-of-validity is used for precise relative timing between measurements of the same type, such as for IMU propagation, and the timestamp is used to align measurements from different channels. The offset between the timestamp and time-of-validity can

be estimated and used to correct the timestamp to improve precision, if needed.

SAMWISE maintains a series of buffers internally that are used to perform time alignment of the sensor data. Whenever a measurement is received, it is added to a priority queue designated for that input channel (e.g., all IMU measurements). In order to guarantee time-correctness, we require at at least N measurements be present in each buffer before any additional measurements can be added. We use $N = 1$ because the Draper/MIT FLA flight system guarantees that measurements *within* any given sensor input channel arrive in order (however, measurements across *different* streams often do not).

Ghost Variables—Some sensors require pre-processing steps that may induce significant latencies. For example, our feature tracking approach involves first running a KLT tracker on the image to generate feature observations [30], [31]. We then accumulate several observations of a particular visual feature across multiple images in order to reject outlier measurements, resulting in a large, variable latency between receiving the raw imagery and measurement availability. To allow the solver to proceed without delay, we "reserve" a place in the factor graph when an image is received for its associated variables *before* we have generated any measurement data. We refer to these placeholders as "ghost" variables, which perform a function similar to the "state augmentation" [23] or "clone pose" [32] techniques for EKFs. Using the timestamp of the raw input data, we generate a pose-pose IMU factor describing the measured motion from the time of the previous pose variable to the time of the new pose (reserved for the received image's measurements) and add it to the graph along with the ghost pose variable. The IMU factors connect all time-consecutive pose variables and thus guarantee full-rank connections to all poses in the graph. When the measurement becomes available the corresponding factors are then added to the graph and can improve the next iSAM2 solution. Ghost variables allow the system to continue to add and solve low-latency measurements while processing high-latency sensor data in parallel.

Strapdown IMU Output

iSAM2 uses a nonlinear Gauss-Newton solver to optimize the values of the variables in the factor-graph, which iterates until some termination criteria has been reached. After convergence, the optimal values of the most recent pose and velocity variables can be output in the form of a state estimate. The convergence time is a function of the number of constraints and their unobservable measurement errors, among other factors. Therefore, the exact convergence time is difficult to predict during real-time operation but can be assumed random within some bounds. Real-time vehicle control requires low-latency state estimates at rates of 100 Hz or more, which in terms means state outputs must be produced with less than 10 ms of computation time. However, in practice, we have observed that a full optimization update often takes over 250 ms on our hardware. Not only does this mean that optimization at these real-time rates cannot be guaranteed or even achieved on average, but also that the computed estimates already have high latency by the time they finish computing, which can lead to controller instability.

To address this, we run computationally efficient inertial strapdown propagation in parallel to iSAM2 in order to provide the high-rate, low-latency state estimates required for real-time vehicle control. In one thread, we run the iSAM2 solver and save the best estimate of the most recent vehicle pose and velocity variables after every solve call finishes. A new solve is started whenever new measurements are available and the previous solve has completed. In practice, this results in an updated pose and velocity value (state) at about 2-10 Hz.

In the second thread, we achieve high-rate output, by propagating every IMU measurement received upon the most recent iSAM2-computed state estimate. This results in a full position, rotation, and velocity estimate for every IMU measurement with sub-millisecond latency. We store the IMU measurements in a buffer and whenever a new state is available from the solver we re-propagate the previous measurements upon the updated state. Using this method, we can produce high-rate, low-latency state estimates at IMU-rate. We have tested this to up to 1000 Hz on our FLA flight system but typically run at 100 Hz to match our attitude controller. This parallel propagate-and-update approach makes iterative solvers viable for high-rate control applications.

For comparison, a traditional EKF is non-iterative (i.e., previous measurement errors cannot be recomputed). This results in deterministic update time, which is ideal for real-time control applications. However, in some cases the EKF update is still too time consuming to maintain full IMU-rate output. Therefore, an inertial-EKF may require the use of a similar parallel propagate-and-update technique in order to achieve high-rate output, while still suffering from lower accuracy than an iterative solver.

Software Architecture

The SAMWISE navigation system is architected to have three core layers. In the terminology of the SLAM community, the iSAM2 algorithm and GTSAM library provide the back-end solver, which optimizes the state variables given a set of measurement constraints. The SAMWISE library presented here serves as the front-end, processing raw sensor data into measurement constraints and constructing the factor-graph passed to iSAM2. SAMWISE also provides a set of custom factors that extend those already available in GTSAM, and handles internal strapdown propagation along with the corresponding thread management.

A series of lightweight wrapper libraries provides interfaces between the SAMWISE navigation system and various communication interfaces. These wrappers provide the ability to configure parameters, translate input data into the standardized SAMWISE format, and publish state estimates. "Samros" offers a ROS interface to SAMWISE and additionally provides detailed system health monitoring. Similar to Samros, "SamTask" offers an interface to Draper's own communication framework [33], and SAMWISE can be trivially wrapped to interface with additional sensor data communication frameworks through its straightforward API.

5. EXPERIMENTAL RESULTS

This section presents experimental results from autonomous quadrotor flights using the SAMWISE navigation system as the primary vehicle state estimator. We focus on the two primary flight configurations used by the Draper/MIT FLA team [1]. Laser-aided inertial navigation was used during the first DARPA FLA milestone event in April 2016, which involved flight through a simulated warehouse environment. Vision-aided inertial navigation configuration is used for GPS-denied outdoor environments, including the second DARPA FLA milestone event in November 2016. The SAMWISE nav-

Figure 2: Draper/MIT FLA quadrotor performing an autonomous gps-denied, vision-aided flight.

igation system has performed over 100 closed-loop flights across the two configurations, and is shown performing an autonomous vision-aided flight in Figure 2.

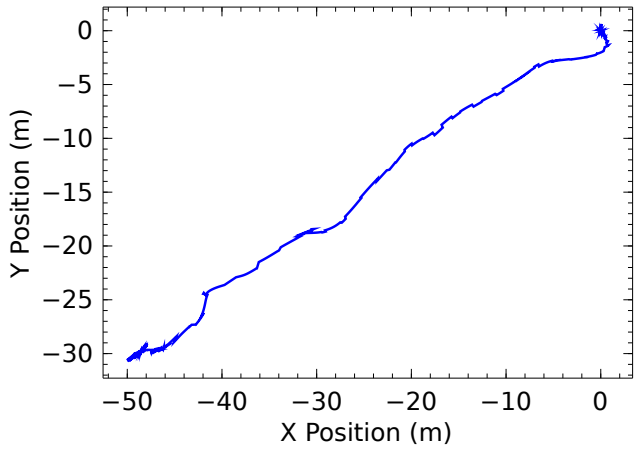
Hardware Configuration

All results presented in this section were computed in real-time during closed-loop autonomous flights with the Draper/MIT FLA quadrotor, shown in Figure 1. The DARPA FLA program requires the use of a DJI f450 quadrotor frame, the DJI E600 propulsion system, 12 inch diameter rotors, and a 6 cell lithium ion battery. Our computing platform comprises an Intel NUC5i7RYH 3.1 GHz dual core processor with 16 GB of RAM and 250 GB of SSD storage. We use a PX4 Pixhawk 1 Autopilot flight controller to perform minimum-latency attitude estimation and control, and this attitude estimate is frequently updated by SAMWISE to keep the two systems synchronized. Fully integrated, the Draper/MIT quadrotor weighs 3.3 kg (7.3 lbs).

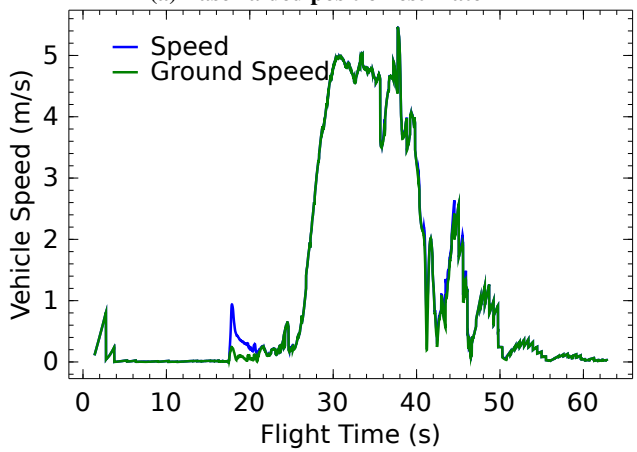
During outdoor flight, SAMWISE can optionally use the barometer and magnetometer from the Pixhawk flight controller which are transmitted to the flight computer over USB. A Lidar-LITE single-beam ranger is vertically mounted to the vehicle to provide precise altimetry when flying over flat terrain. We additionally use an Analog Devices ADIS16448 IMU, sampled at 816 Hz and transmitted to the flight computer over USB using a custom interface board. For laser-aided flight we use a Hokuyo UTM-30LX scanning range finder mounted above the vehicle with a 30 meter range and the Canonical Scan Matcher (CSM) to provide the scan matching [34]. For vision-aided flight we use a Point Grey Flea3 FL3-U3-13Y3M-C monochromatic camera with a global shutter and a 3.5 mm focal length machine vision lens running at 20 frames per second. All flight tests used the Samros ROS interface to SAMWISE, and all sensor data and state estimates were transmitted to and from SAMWISE using the ROS communication framework.

Laser-aided Inertial Navigation Results

The laser-aided configuration has been extensively flight-tested in indoor environments, including at the April 2016 DARPA FLA milestone event. The event included several flights through a 60 meter long obstacle course constructed of scaffolding, boxes, and tarps, which was designed to simulate an indoor warehouse environment complete with dead-ends and occlusions. The Draper/MIT team completed 39 flights as part of the indoor milestone event, 29 of which



(a) Laser-aided position estimate



(b) Estimated speed

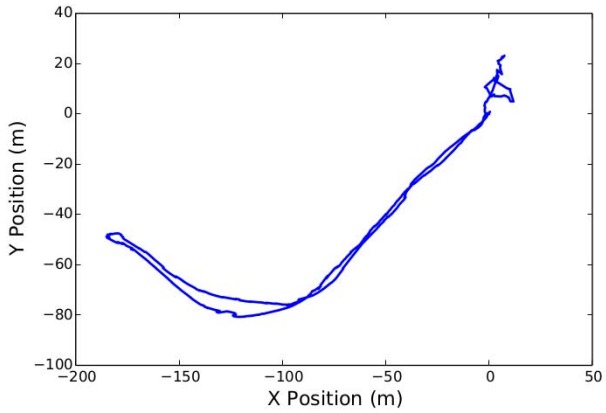
Figure 3: Real-time position and speed estimates for a closed-loop flight using a 2D laser scanner, IMU, and laser altimeter. This flight was part of the first DARPA FLA milestone event and required the vehicle to slalom around large obstacles while flying along a 60 meter aisle of a warehouse environment. The final position error was not measured, but the vehicle achieved the program goal of landing within 6 meters of the target object.

successfully reached the goal location. Post-flight analysis showed that the SAMWISE state estimator performed well during 34 of the 39 closed-loop flights, with failures mainly caused by limitations of using a 2D scan-matcher in a 3D environment, and that the maximum vehicle speed during the event was about 5.5 m/s (12 mph).

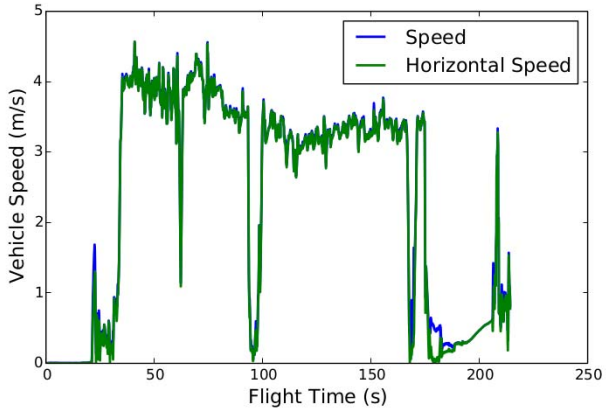
Figure 3 shows the estimated position and speed from one of the successful closed-loop flights. This flight involved a high-speed slalom down a 60 meter warehouse aisle with large obstacles obstructing portions of the vehicle path that required evasive maneuvers. The vehicle achieved a maximum speed of about 5 m/s before landing within the designated 6 meter circle surrounding the goal location.

Vision-aided Inertial Navigation Results

The vision-aided state estimation configuration was successfully demonstrated at the second DARPA FLA milestone event in November 2016, which involved GPS-denied outdoor flight. The SAMWISE estimator enabled 47 closed-



(a) Vision-aided position estimate



(b) Estimated speed

Figure 4: Real-time position and speed estimates for a closed-loop flight using visual feature tracking and an IMU, barometer, and laser altimeter. The vehicle flew approximately 600 meters while avoiding obstacles and returned to the start location with approximately 3% of position error with respect to distance traveled and a top measured speed of 4.5 m/s (10 mph).

loop, GPS-denied, autonomous flights during the week-long event, including the flight shown in Figure 2. This SAM-WISE configuration used a camera, IMU, and barometer or laser altimeter as sensors and included online calibration refinement of the camera intrinsic and extrinsic parameters, IMU bias estimation, and barometer calibration parameters. A 2D scanning laser range finder and stereo camera were also present on the vehicle during the flight tests to enable autonomous obstacle detection but were not used for state estimation.

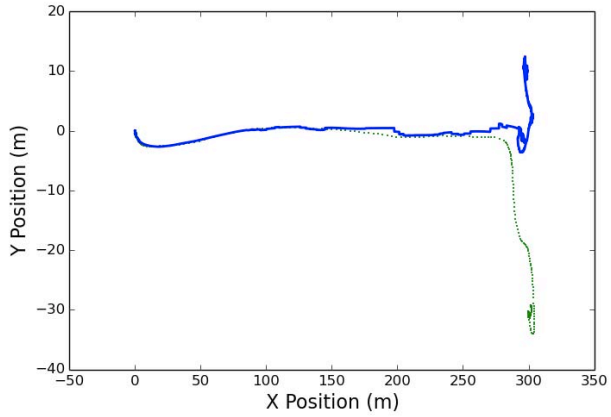
Figure 4 shows results from a closed-loop flight test involving traversing to a goal location 189 meters from the launch point followed by a return to the launch location, requiring the avoidance of large obstacles including densely wooded areas and a building as shown by the sample images in Figures 5 and 6. Due to obstacle avoidance, the actual trajectory was approximately 600 meters round-trip. The vehicle successfully flew to the goal location and returned to the initial location with less than 20 meters of accumulated position error (approx. 3% of distance traveled) and achieving a maximum speed of about 4.5 m/s (10 mph).

Figure 5: Sample image from the on-board camera during the outbound segment of the vision-aided flight shown in Figure 4, which required robustness to difficult lighting conditions and lens flare while flying into the morning sun.

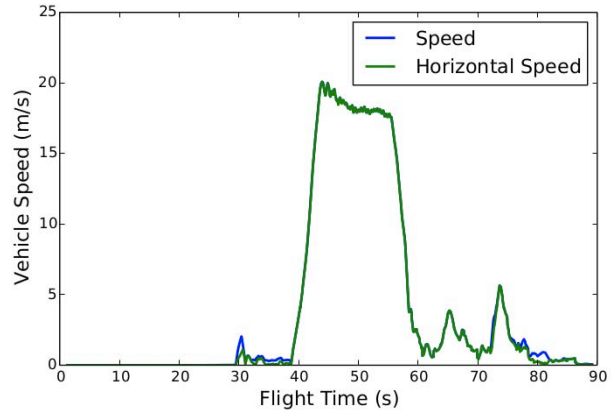
Figure 6: Sample image from the on-board camera during the inbound segment of the vision-aided flight shown in Figure 4, which demonstrated robustness to glare while flying away from the sun.

Figure 7 shows results from a high-speed closed-loop flight test during the second DARPA FLA milestone event. As shown in Figure 8, the flight environment was an empty runway with very limited visual texture for feature tracking. The vehicle ascended to an altitude of 4 meters and then flew 300 meters straight downrange, achieving a maximum velocity of 20 m/s (45 mph). For this test, a GPS receiver was added to the vehicle for logging purposes only, allowing us to more closely evaluate the accuracy of the vision-aided estimator. As seen in Figure 7a, position error accumulation caused the vehicle to stop about 10 meters short of the 300 meter goal.

The state estimator failed to regain visual tracking after the aggressive deceleration maneuver, which resulted in the approximately 30 meter drift during the landing sequence. The



(a) Vision-aided position estimate (blue) with GPS reference (green)



(b) Estimated speed

Figure 7: Real-time position and speed estimates for a closed-loop flight using vision, an IMU, and a laser altimeter. The flight involved ascending to a 4 meter altitude and traversing 300 meters forward at near-maximum throttle without obstacles. GPS position data was logged during this flight (shown as green dots above), but this data was not used by the state estimator. The flight had a measured top speed of 20 m/s (45 mph).

aggressive deceleration maneuver combined with the low-texture environment resulted in a mis-estimation of velocity when the vehicle failed to regain visual tracking and noise in the IMU began to accumulate. Our safety pilot then took control of the vehicle, halted the horizontal velocity, and safely landed the vehicle. SAMWISE regained tracking during this manual maneuver and was able to correctly estimate the vehicle velocity during the manual landing sequence. Future development work will focus on improving the robustness of this sensor configuration during and after such aggressive flight maneuvers with the goal of eliminating the need for human-aided recovery.

6. Conclusions and Future Work

This paper presented the SAMWISE navigation system, an incremental smoothing approach to GPS, laser, and vision-aided inertial navigation for high-speed autonomous flight with small, agile quadrotors. We show that the high-rate, low-latency state estimates required for agile flight are achievable through use of a parallel update and propagate framework, in which the iSAM2 solver is run in the background while computationally-efficient strapdown integration is used to propagate the vehicle state forward in time at up to the IMU sampling rate. SAMWISE is capable of accounting for sensor data transmission and processing latencies using measurement alignment buffers and the novel incorporation of "ghost" variables as temporary placeholders in the factor-graph. SAMWISE additionally uses a sliding window to bound the computation required for iSAM2 solve steps in order to maintain real-time performance.

Figure 8: Sample image from the on-board camera during the high speed flight shown in Figure 7.

SAMWISE has been demonstrated on over 100 successful closed-loop autonomous flights. Section 5 gave results for the two most common sensor configurations tested: laser-aided and vision-aided inertial navigation. These results show that SAMWISE is capable of autonomous, high-speed, agile flight with the FLA hardware platform, achieving flight speeds of up to 20 m/s (45 mph).

Future work will focus on additional development and robustness testing of the vision-aided configuration. This will include expanding the online camera calibration to include lens distortion, the incorporation of dense optical flow measurements to more robustly estimate vehicle velocity at low speeds, and measurement pruning to further reduce the computational requirements of vision-aided, GPS-denied flight.

Acknowledgments

We would like to acknowledge the dedicated work of the entire Draper and MIT FLA Team for their outstanding contributions to the implementation of the FLA flight system and flight test support. We would additionally like to thank the DARPA FLA team for their program and test support and Scott Rasmussen of Draper for use of his photos.

Disclaimer

The views, opinions, and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the

U.S. Government.

REFERENCES

[1] S. Paschall II and J. Rose, "Fast, lightweight autonomy through an unknown cluttered environment", in *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, 2017.

[2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: real-time single camera slam", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–67, 2007.

[3] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I", *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[4] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II", *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–17, 2006.

[5] F. Dellaert and M. Kaess, "Square root SAM", in *Robotics: Science and Systems*, 2005, pp. 177–84.

[6] T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix, "Vision-based SLAM: stereo and monocular approaches", *International Journal of Computer Vision*, vol. 74, no. 3, pp. 343–64, 2007.

[7] J. Civera, A. J. Davison, and J Montiel, "Inverse depth parametrization for monocular SLAM", *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–45, 2008.

[8] E. Eade and T. Drummond, "Scalable monocular slam", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 469–76.

[9] D. Scaramuzza and F. Fraundorfer, "Visual odometry tutorial part I: the first 30 years and fundamentals", *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[10] T. Oskiper, Z. Zhu, S. Samarasekera, and R. Kumar, "Visual odometry system using multiple stereo cameras and inertial measurement unit", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[11] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: a scalable real-time causal approach", *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–30, 2011.

[12] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry", *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[13] T. J. Steiner, S. A. Rasmussen, P. A. DeBitetto, B. E. Cohanim, and J. A. Hoffman, "Unifying inertial and relative solutions for planetary surface navigation", in *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, 2012.

[14] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2004.

[15] D. Nistér, "An efficient solution to the five-point relative pose problem", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–70, 2004.

[16] H. Stewénius, C. Engels, and D. Nistér, "Recent developments on direct relative orientation", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 4, pp. 284–94, 2006.

[17] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the ACM*, vol. 24, no. 6, pp. 381–95, 1981.

[18] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers", *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–86, 2007.

[19] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera", in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2531–8.

[20] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing", *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608, 2010.

[21] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Towards consistent vision-aided inertial navigation", in *Algorithmic Foundations of Robotics X*, 2013, pp. 559–74.

[22] K. Konolige, M. Agrawal, and J. Sola, "Large-scale visual odometry for rough terrain", in *Robotics Research*, 2011, pp. 201–12.

[23] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation", in *IEEE international Conference on Robotics and Automation*, 2007, pp. 3565–72.

[24] H.-P. Chiu, S. Williams, F. Dellaert, S. Samarasekera, and R. Kumar, "Robust vision-aided navigation using sliding-window factor graphs", in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013, pp. 46–53.

[25] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: incremental smoothing and mapping", *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–78, 2008.

[26] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Information fusion in navigation systems via factor graph based incremental smoothing", *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 721–38, 2013.

[27] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: incremental smoothing and mapping with fluid relinearization and incremental variable reordering", in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3281–8.

[28] F. Dellaert, "Factor graphs and GTSAM: a hands-on introduction", Tech. Rep., 2012.

[29] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm", *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[30] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision.", in *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, BC, 1981, pp. 674–9.

[31] J. Shi and C. Tomasi, "Good features to track", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 1994, pp. 593–600.

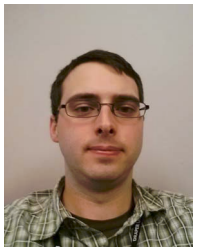
[32] P. Lommel, R. Madison, and P. DeBitetto, "Solder affixed, vision aided navigation technology (SAVANT): part 2", in *ION Joint Navigation Conference*, 2012.

[33] T. J. Steiner, "A unified vision and inertial navigation system for planetary hoppers", Master's thesis, Massachusetts Institute of Technology, 2012, p. 146.

[34] A. Censi, "An ICP variant using a point-to-line metric", in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, 2008.

BIOGRAPHY

Dr. Ted Steiner *is a Senior Member of the Technical Staff in the Perception and Localization Systems group at Draper Laboratory. His research interests include planetary exploration and vision navigation systems. He holds a B.S. in Mechanical Engineering from the University of Wisconsin-Madison and S.M. and Ph.D. in Aeronautics and Astronautics from the Massachusetts Institute of Technology.*



Robert Truax *is a Senior Member of the Technical Staff in the Perception and Localization Systems group at Draper Laboratory. He received a B.S. and M.S. in Mechanical Engineering at the Massachusetts Institute of Technology. He has participated in many robotic projects, including the DARPA Urban Challenge in 2007, an autonomous forklift with the Agile Robotics Lab at MIT, and DARPA's Fast Light Autonomy program. His main research interests are autonomous localization, mapping, and visual-inertial navigation.*

Kristoffer Frey *is a Draper Fellow and S.M. Candidate at the Massachusetts Institute of Technology in the Department of Aerospace and Aeronautical Engineering working under the advisement of Professor Jonathan How. He previously completed an S.B. in Computer Science & Electrical Engineering, also from MIT.*