

# ALL-SOURCE NAVIGATION FOR ENHANCING UAV OPERATIONS IN GPS-DENIED ENVIRONMENTS

**Paul Williams, Michael Crump**  
**BAE Systems Australia**

*paul.williams6@baesystems.com; michael.crump@baesystems.com*

**Keywords:** *UAVs, Navigation, SLAM, GPS Denied*

## Abstract

*The drive towards utilizing small, cheap, autonomous aerial vehicles for military operations means that navigation systems that are robust to GPS denial must be employed. The simplest option available is to increase the accuracy of the inertial measurement unit (IMU), but this can substantially increase the price per operational unit. This paper presents an overview of the All-Source Navigation system developed by BAE Systems Australia based on inexpensive MEMS IMUs and a supporting image processing unit. The navigation system is capable of sustaining the operational flight capability of the vehicle for prolonged periods of time compared to the pure inertial solution. At its core, All-Source Navigation makes use of SLAM techniques. A variety of additional aiding sources are fused into the inertial navigation solution to give improved navigational accuracy during flight. The system is capable of performing both conventional static, as well as in-flight alignment. All-Source Navigation is demonstrated on the Kingfisher 2 UAV platform at the West Sale test facility.*

## 1 Introduction

UNMANNED air vehicles (UAVs) have gained prominence in recent years due to their wide applicability in defense. Deep and first day strike roles such as SEAD (Suppression of Enemy Air Defenses) or neutralizing enemy C3 (Command, Control and Communication) systems are considered ideal roles for unmanned vehicles. In such roles, the vehicle must be able to penetrate a remote region unobserved and be

capable of generating rapid targeting solutions with sufficient accuracy to allow target prosecution. This places the sensing platform in a hostile environment where the enemy may be capable of disrupting vehicle communications systems and denying access to GPS (Global Positioning System) signals, which is a key element of most UAV navigation systems. In order to remove the human from the loop, an autonomous platform must therefore provide a means for maintaining accurate navigation solutions without the use of GPS and be able to autonomously detect, recognize, identify and locate targets in the offending area.

Most modern inertial navigation systems for UAVs utilize strapdown inertial navigation algorithms. The key requirement of an accurate navigation solution is a high accuracy inertial measurement unit (IMU). Unfortunately, most commercially available high accuracy units are either too expensive or are export-controlled, making them infeasible or undesirable for use on Australian assets. This means that UAV operations of significant length rely almost exclusively on GPS technology for external corrections. Thus, in the presence of GPS denial, many UAVs can become unusable.

This paper presents flight test results for a navigation system based on SLAM (Simultaneous Localization and Mapping), a technique which couples targeting and navigation. SLAM provides autonomous systems with a real-time navigation, mapping and precision target location capability. SLAM does not require external information such as that provided by GPS or by *a priori* map data, though if available, can be used to generate more accurate solutions. This reduces the

requirement for GPS and presents an opportunity for using a lower fidelity sensing suite to generate targeting solutions. The use of SLAM allows UAVs to employ passive sensors, which are inherently less accurate than active sensors, to produce targeting solutions. Passive sensors preserve the requirement for low observability and ultimately results in a more survivable system.

An unaided integrated navigation solution with an IMU as the primary sensor input can become unacceptable within a few minutes so that guidance and targeting are impossible. On the other hand, in the absence of GPS, a SLAM-based navigation solution can constrain the flight vehicle position estimates to within an accuracy that is suitable for autonomous navigation. Furthermore, SLAM also enables the tracking of targets to an accuracy required to support third party prosecution.

SLAM can be thought of as a hybrid of two well-known problems: tracking and navigation through localization. In tracking, the position and attitude of the sensing platform is assumed to be known to a degree of certainty and feature/target locations in the environment are to be determined. In navigation through localization, the feature/target locations are known to some degree and the objective is to determine the sensing platform location and attitude. If both the feature/target locations in the environment and the sensing platform location and attitude are unknown, then the two problems are coupled. The process of solving the coupled problem is known as SLAM and was first postulated in the late 1980's [1].

This paper presents a solution to the real-time application of SLAM within the UAV environment. Our implementation is embedded within a new navigation architecture called All-Source Navigation (ASN), which is designed to be a plug-and-play navigation system. That is, any number of sensors can be utilized to enhance the accuracy of the onboard navigation solution. This paper will outline the development of ASN and its implementation of a SLAM solution. The system is rigorously tested in a realistic simulation and hardware-in-loop environment. Flight test results of the

system with ASN providing the navigation solution to the Kingfisher 2 UAV are presented. ASN is employed in the Intelligent Landing System described in [2].

This paper is organized as follows: First, a comprehensive review of SLAM techniques is given; next, some of the key constraints on the ASN system are highlighted; the sensor suite used by the Kingfisher 2 UAV is summarized; the design methodology for ASN is presented; the simulation architecture is presented; an overview of the real-time embedded implementation is provided; finally, flight test results are presented.

## 2 Simultaneous Localization and Mapping

In recent years, SLAM has attracted an enormous amount of interest [3][4]. A number of approaches for solving SLAM have been proposed and demonstrated in a range of environments including indoor [5], outdoor [6], underwater [7], and air [8]. In many cases, even in the absence of absolute position information, it is still possible to build a relative map of features or targets within the environment and to use that map for navigation. Results show that navigation errors can remain small over reasonably long periods of time [9][10].

The two most popular solutions to SLAM have been the Extended Kalman Filter (EKF) [5]-[17] and Rao-Blackwellized particle filter solutions (FastSLAM [18]-[20]). EKF SLAM maintains a fully-correlated feature-based map using a moments parameterization (mean and covariance). This implies that the vehicle and map states are represented by Gaussian distributions, although this assumption is often violated in practice. The EKF relies on the propagation and update of the covariance matrix using a first-order linearization of the process and measurement models. This can lead to large errors in the posterior mean and covariance for highly nonlinear systems. The EKF computational complexity grows as  $O(N^2)$  where  $N$  is the number of landmarks in the map. This limits the number of fully-correlated features that can be in the SLAM map at any

time. Hence, map management techniques need to be implemented for large scale applications. A number of map management techniques have been developed ([21]-[23]) and Dissanayake ([14]) showed that features can be removed from the map without making the SLAM process statistically inconsistent. More recently, the unscented Kalman filter (UKF) [24] has emerged as a popular and robust nonlinear filtering method that preserves the approximation of the covariance to higher order than the EKF [11]. However, the UKF relies on the propagation of a set of deterministically sampled sigma points to construct its state estimate, which generally requires significantly more computation than the construction of analytic Jacobians that are used by the EKF. Hence, for SLAM, the EKF is generally more efficient for larger maps.

FastSLAM uses a set of Rao-Blackwellized particle filters to solve the SLAM problem. Particle filtering is used to estimate the vehicle pose, whereas EKFs are used to represent each feature in the map. FastSLAM exploits the fact that the location of features can be estimated independently if the vehicle state is known. The associated computational complexity of the FastSLAM algorithm is  $O(\log N)$ . One of the key advantages of FastSLAM over EKF SLAM is that FastSLAM is significantly more robust to data association errors than EKF SLAM. Recently, however, Bailey [25] showed that FastSLAM can become inconsistent for long-term SLAM implementations.

Application of SLAM to air vehicles requires the implementation of a 6-DOF SLAM solution for platform localization, and the maintenance of a three-dimensional map. This differs from the majority of SLAM implementations that build maps in a single plane. Davison [26]-[27] demonstrated SLAM using a three-dimensional map using a handheld, wide angle camera. SLAM incorporating high speed 6-DOF vehicle motion has been investigated and implemented in the air domain using a fixed wing UAV [28]. Low speed 6-DOF SLAM has also been implemented using a

blimp [29]. In [28], an EKF solution was implemented on a UAV platform using a passive vision sensor. The range to the features used by the SLAM algorithm was estimated by placing features of known size on the ground and then inferring range from the feature size in the images. An EKF SLAM solution was also implemented in [29] on an autonomous blimp using stereovision. Bryson and Sukkarieh [30] investigated bearings-only SLAM in the air domain within an EKF framework. Their approach initializes features using range-parameterized extended Kalman filters. However, their approach was not implemented in real-time on target hardware.

The accuracy of SLAM can be increased in a cooperative vehicle scenario, where multiple vehicles work together to build a common map. Fenwick [31] implemented an EKF-based multi-vehicle system in an indoor environment. Nettleton et al. [32] proposed an Extended Information Filter (EIF) solution in a decentralized network and in later work [33], Nettleton showed that the low bandwidth problem resulting from an inability to send full covariance matrices can be solved by communicating submaps of the SLAM map rather than the whole map. Nieto [34] proposed a FastSLAM multi-vehicle solution where one robot acts as a central filter and maintains the map information and localization of all vehicles within the map. Walter et al. [35] proposed a multi-vehicle underwater master-slave SLAM architecture solution where one vehicle is used to maintain map estimates and vehicle pose information for all vehicles in the network. In this approach, slaves are equipped with a lower fidelity sensor suite compared to the master vehicle. The slave's only job is to send raw sensor data to the master, which then produces a map based on all received data. Ong [36] proposed a decentralized data fusion SLAM solution using an EIF architecture for the air domain. Bryson investigated a centralized multi-UAV SLAM solution [37] and in later work extended this to a decentralized network [38].

Other approaches to SLAM include purely vision based SLAM using SIFT descriptors for data association [39][40]. Methods used to reduce the computational complexity of SLAM have mainly been based around partitioning of the SLAM map into sub-maps. One popular method is the Constrained Local Submap Filter (CLSF) proposed by Williams [17], where an independent local submap of the features in the immediate vicinity of the vehicle is maintained and these local maps are then fused periodically into the global map. GraphSLAM [41] has also been developed to tackle large-scale SLAM where the vehicle carries out the mission and gathers sensor data first before the algorithm is applied. Although this approach can generate exact solutions to the SLAM problem, it cannot be applied for real-time operations. Eustice [42] showed that the SLAM information matrix is exactly sparse in a delayed-state framework meaning that a sparse EIF requires less storage space and offers the same accuracy as the full EKF solution.

Adrien [39] discussed vision based SLAM for micro UAV's in an EKF framework using SIFT descriptors for feature extraction and data association. The approach details development for 2D MAV navigation allowing construction for a metric map of visual ground landmarks. Other Airborne SLAM approaches are principally based on the EKF. The most widely known examples of UAV SLAM is on the Brumby Experimental UAV [30][8][28]. Decentralized networks consisting of a group of UAVs implementing SLAM has been addressed using information space [37][38][28][33].

Hygounenc [29] applied SLAM to an autonomous blimp. Low altitude stereo vision was used in an EKF framework. The SLAM algorithm was tested on sensor data gathered from the blimp and experimental results showed centimeter accuracy positioning in 3D space. Langelaan [43] examined airborne SLAM using a monocular camera in a UKF framework to provide a consistent estimate of the states.

The major focus in SLAM research in the last few years has been in using vision only sensors for extracting natural features for SLAM. Vision sensors are lightweight passive

sensors which provide greater detail of the environment. Vision using SIFT descriptors for feature extraction and data association has been a key enabling technology for SLAM.

The majority of research on SLAM has taken place within the indoor robotics community. Outdoor land environment results have been produced by using sensor data gathered on an outdoor vehicle. The dataset gathered at Victoria Park has featured prominently in SLAM papers. SLAM has also been implemented and tested in outdoor vehicles using natural feature sensor data as a secondary navigation solution.

SLAM in the air domain has been implemented in real-time by ACFR, but the solution was not fed back into the main navigation filter. One of the challenges in airborne SLAM is the devastating impact that a divergent solution has on the vehicle motion. Therefore, great care must be exercised in SLAM algorithm design to prevent this from occurring.

Research shows that there is a push towards more stochastic based methods such as particle filters as they provide more robust data association and they can better handle non-linearity. Sigma-point Kalman filters have also been used and some have even combined sigma-point filters with particle filters. The advantage of the UKF is that it can prevent inconsistency problems that arise from the linearization used in the EKF. Papers have shown that particle filters are scalable to a large number of landmarks but only produce consistent results over a short period of time. The robustness offered by particle filters in terms of data association and its ability to handle non-linearities warrants further investigation. In particular, its application to 6-DOF must be properly assessed. Sigma point filters do not offer any additional data association robustness.

### 3 All-Source Navigation Requirements

The ASN system is designed to supplement an existing flight control computer (FCC) developed by BAE Systems Australia. The FCC was developed using DO-178B processes and procedures. It is a triplex flight control

computer architecture, where each lane contains its own inertial measurement unit (IMU) and global positioning system (GPS) receiver. The FCC is shown in Fig. 1. Each FCC computes an independent navigation solution and flight control commands. The solutions and commands are compared across all three computers, and a single consolidated set of commands is sent out to the control actuators to fly the plane. The FCC is designed to meet stringent timing requirements and will not accept an ASN navigation solution with more than a certain amount of latency. In particular, if SLAM is used and causes a lag due to computational delay, this is deemed unacceptable and therefore unusable by the FCC. The FCC runs its control loops at 100 Hz, and requires a navigation solution in hard real-time at the same rate. The allowable latency between successive ASN packets is 20 ms.



FIG. 1. BAE SYSTEMS AUSTRALIA FLIGHT CONTROL COMPUTER.

In addition to strict timing requirements, the ASN navigation solution must be deemed to be acceptably accurate by the FCC. In its deployed configuration, ASN is intended to be the only navigation solution available to the FCC. However, for testing purposes, we do not disable GPS to the FCC but only to ASN. This is because we operate the vehicle in civilian airspace and must have situational awareness of the vehicle and its intended actions. Based on this, the FCC compares the ASN navigation

solution with its own internally computed one. The ASN solution must satisfy the accuracy requirements shown in Table 1 if it is to be in control of the flight vehicle.

TABLE 1. REQUIRED ASN SOLUTION ACCURACY.

Error Component	Accuracy
ECEF Position, X	30 m
ECEF Position, Y	30 m
ECEF Position, Z	30 m
ECEF Velocity, X	3 m/s
ECEF Velocity, Y	3 m/s
ECEF Velocity, Z	3 m/s
Euler Roll	2 deg
Euler Pitch	2 deg
Euler Yaw	5 deg

#### 4 Sensor Characteristics

In commercial aviation, accuracy of the navigation solution is required for flights that last on the order of hours. For marine vehicles, however, where the amount of time between significant landmarks can be large, much higher accuracy is needed. For inertial navigation systems, the error budget is usually specified in terms of the error incurred without an aiding source.

A fundamental constraint of the ASN system is that it must operate using low cost sensing equipment. This is based on keeping the unit price of the vehicle management system as low as possible to maximize return on investment. If money were no object, the navigation designer would always prefer higher grade inertial sensors, as navigation performance is directly coupled to the performance of the inertial sensing package.

The primary sensor used in virtually all navigation systems is an inertial measurement unit (IMU). An IMU consists of three orthogonally mounted accelerometers and three orthogonally mounted gyroscopes. In a strapdown inertial navigation system, the measured angular rates and accelerations are integrated to form estimates of position and orientation. Due to various errors, the integrated solution drifts over time. When high

grade IMUs are used, the drift rate will be relatively small. Table 2 shows a comparison of different grade IMUs with their relative drift and cost. Clearly, if one wants a very high accuracy inertial solution, then it comes at a significant cost.

In most UAV business models, the drive is to reduce the cost of the platform and avionics subsystems while maintaining a reliable flight platform. The FCC utilizes an IMU that is accurate to the tactical/industrial grade (it is closer to tactical grade). The consequence of this is that without any aiding source, the errors make the platform unusable in a matter of minutes.

TABLE 2. COMPARISON OF IMU GRADES.

Sensor grade	Cost	Drift performance
Marine grade	> \$1,000,000	< 1.8 km/day
Navigation grade	> \$100,000	< 1.5 km/hour
Tactical grade	\$5000 - \$30,000	~ minutes
Industrial grade	\$500 - \$5000	~ seconds

The most widely used aiding source in modern-day navigation systems is GPS. GPS provides position and velocity measurements at the receiver. The fusion of GPS data by means of a Kalman filter enables the correction of errors in position, velocity, and attitude. Attitude is observable by maintaining cross-correlations between errors. GPS data is typically fused at 1 Hz. Hence, when the GPS signal is lost, the frequent aiding it normally provides disappears and the aircraft is likely to be lost.

#### 4.1 Inertial Measurement Unit

The inertial measurement unit that is used onboard our unmanned systems is the SiIMU04 from Atlantic Inertial Systems. The IMU utilizes MEMS accelerometers and gyroscopes. The error characteristics are shown in Table 3.

TABLE 3. SiIMU04 ERROR CHARACTERISTICS [44].

Error Characteristic	Value
Gyro Bias repeatability	650 deg/hr
Gyro Bias instability	8 deg/hr
Gyro random walk	0.4 deg/ $\sqrt{\text{hr}}$
Gyro noise	0.72 deg/sec
Accel Bias repeatability	100/20 mg
Accel Bias instability	3/1 mg (axis 1/axes 2,3)
Accel random walk	0.6/0.25 m/s/ $\sqrt{\text{hr}}$ (axis 1/axes 2,3)
Accel noise	22 mg

#### 4.2 Air Data

All fixed-wing air vehicles utilize air data sensors for measuring the pressure altitude and true airspeed of the vehicle. In unmanned systems, such sensors are flight critical. Hence, failure of the air data system would most likely result in the loss of the aircraft. For that reason, the air data is generally robust and therefore reliable enough to use as an aiding source for the navigation system. Fig. 2 shows the location of the air data system on the Kingfisher 2 UAV.

The air data system measures dynamic pressure and absolute (static) pressure. The static pressure is calibrated to provide altitude relative to a reference, for example ISA sealevel at +15 deg C, or the local QNH pressure altitude. The dynamic pressure is used to calculate true airspeed. Unfortunately, neither of these aiding sources is absolute. They both provide information relative to a reference. True airspeed, in particular, is the speed of the aircraft relative to the air mass. For navigation purposes, the altitude required is the altitude relative to the WGS84 ellipsoid. For these reasons, the inclusion of pressure altitude and true airspeed measurements requires the estimation of additional states (one for the pressure altitude offset, and an additional 2 for the wind speed estimates in the navigation frame).



FIG. 2. AIR DATA PROBE ON KINGFISHER 2 UAV.

### 4.3 Downward-Looking Camera

The Kingfisher 2 UAV is equipped with a gimbaled camera that is used in our associated work for Intelligent Landing [2]. For navigation purposes, it is utilized in a downward-pointing configuration. Note that in general a dedicated camera would be used for this purpose. However, in our test system, the same camera is used for multiple purposes.

The downward-looking camera provides three sources of information:

- 1) Relative displacement information (frame-to-frame tracking)
- 2) Salient feature identification for map-building
- 3) Landmark identification for incorporation of known location information

#### 4.3.1 Frame-to-Frame Tracking

Our algorithm employs a modified form of optical flow to help constrain errors. In “standard” optical flow, the motion of pixels within an image are tracked to give a flow vector. In our approach, a region of interest in the center of the image is used to calculate the weighted flow of the center pixel. The center pixel is then tracked frame-to-frame to give the displacement over time.

Pixel displacements inherently contain both translational and rotational degrees of freedom. Therefore, the pixel displacement alone is not enough to be able to constrain errors. However, we have developed a novel methodology to maintain the cross-correlation of errors in the system so as to optimally constrain the propagation of errors. Fig. 3

shows a screen capture of the modified optical flow, showing the frame to frame pixel displacement of the region around the center pixel.



FIG. 3. EXAMPLE OF OPTICAL FLOW VECTORS IN REGION AROUND CENTER OF FRAME.

#### 4.3.2 Salient Feature Tracking

Simultaneous Localisation and Mapping (SLAM) is a technique for using external features for navigation. Each feature is added to a database of features, and typically comes with a unique identifier. As the vehicle moves around the map, each feature that is revisited helps to constrain the position and attitude error of the vehicle.

In previous work (FURI CTD), we demonstrated a basic SLAM algorithm using white targets placed paddocks around the West Sale aerodrome. A very simple image processing technique based on thresholding color and size was used for identifying the “features”. Fig. 4 shows two of the white targets used for this purpose.

Unfortunately, using white targets is not at all representative of an operational scenario. Instead, we utilize an algorithm based on SURF (Speeded-Up Robust Features) [45]. SURF is able to provide a fingerprint for a feature in the environment that is robust to viewing angle and scale. This enables the algorithm to detect and match features that were previously observed. SURF is similar to the SIFT [46] algorithm, but is faster. Fig. 5 shows an example of a typical

match obtained from a different flight angle of the same area. The reference feature set shown in Fig. 5 is taken during an eastward pass over the area, whereas the test feature set is produced from a westward pass over the same area. Note that the features detected and used by SURF are not necessarily easily distinguishable to the human eye.



FIG. 4. EXAMPLE OF WHITE TARGETS USED IN FURI CTD FOR SLAM FEATURES.



FIG. 5. EXAMPLE OF SURF FEATURE DETECTION AND MATCHING.

#### 4.3.3 Landmarks

Certain features in an image can be easily identified by a human operator, or are unique enough that the onboard software can identify it. The navigation solution we have developed is able to use this information provided either a priori or via an external interface. Fig. 6 shows an example of matching a house in a satellite image to a house in an image captured by the aircraft. Note that features around the house make the particular house quite unique and easily identified within the image. Feeding the true location of the house to the navigation computer, combined with the pixel coordinates, gives an absolute reference for the navigation system. This enables a large portion of the absolute error to be removed.

In addition to fusing image information about landmarks directly into the navigation filter, we also enable the option of an operator

to correct the SLAM map. In such a case, the operator matches a distinct feature in the map (using a corresponding image from the onboard system) to a map feature, and sends a reference

set of coordinates for the feature to the navigation system.

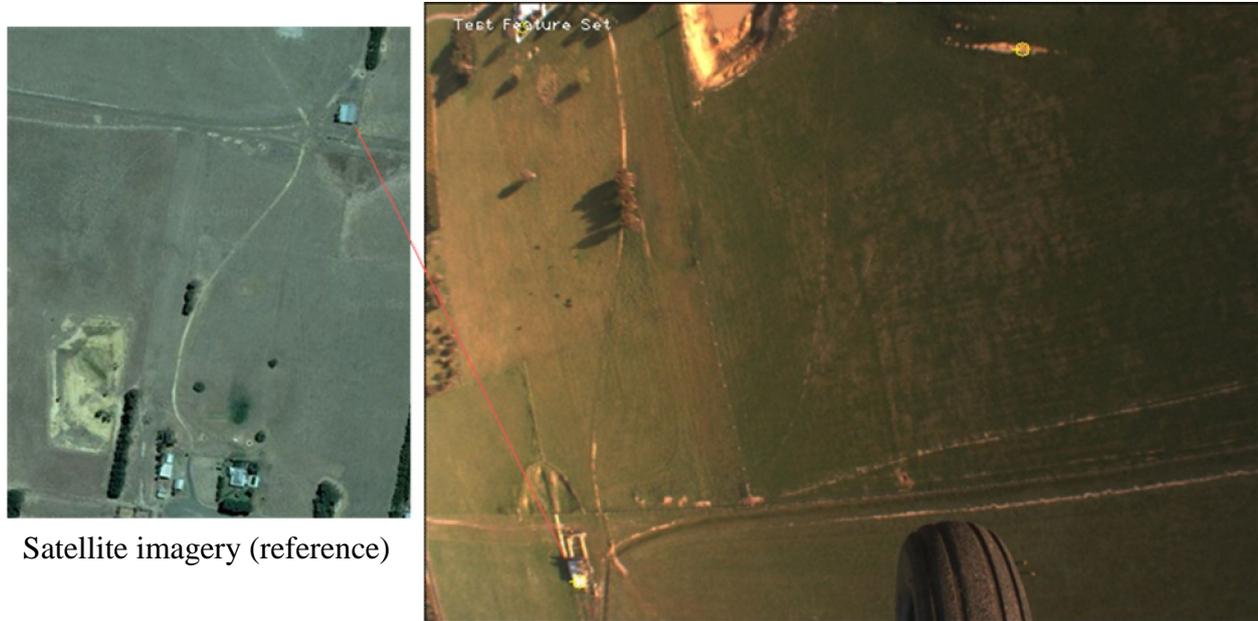


FIG. 6. EXAMPLE OF IDENTIFICATION OF A BUILDING IN IMAGE FRAME.

#### 4.4 Tightly-Coupled GPS

A typical GPS receiver will output an estimate of position and velocity of the receiver, which most navigation systems will use to aid the inertial navigation solution. However, the GPS receiver itself calculates the estimate from the time of flight of signals received from a number of different satellites, as well as the doppler velocities of the signals. Most modern receivers run a filter internally, which can introduce latencies associated with an accelerating vehicle. An alternative approach is to use the raw data produced by the GPS unit directly in the navigation filter. The raw data consists of:

- Ephemeris data (satellite position and velocity data as a function of time),
- Pseudorange data (time of flight of signal multiplied by speed of light),
- Doppler data (frequency shift converted into range rate).

Because the time of flight is computed based on the receiver's clock, an additional clock offset

must be estimated as part of the solution. This means that at least 4 satellites must be visible to obtain an initial position fix. Many receivers cease outputting position and velocity data when the number of satellites visible drops below 4. When the raw data is fused in a Kalman filter, it is possible to update the state estimate of the vehicle with less than 4 satellites. Using the raw data in this manner is referred to as a tightly-coupled GPS solution.

#### 5 Design Methodology

The FCC is rigorously tested within a simulation framework prior to being deployed onto target hardware. The simulation framework is designed and implemented in the MATLAB/Simulink environment and includes detailed models of the physical vehicle and its sensor suite. The model-based design approach enables rapid testing of prototype designs that adhere to actual system interfaces without even knowing what hardware is going to be used. The advantage of using model-based design is that models of the complete FCC are available

that allow payloads to be integrated into the system with minimal effort.

The ASN system is designed from the ground up, beginning with the strapdown algorithms. Truth data is utilized to validate the strapdown algorithm, followed by previously logged flight data. Obviously there is no substitute for real data, but the quality of the IMU used means that only short flights are useful for log data replays of the strapdown algorithm. Once the performance of the strapdown algorithm is confirmed, the EKF is designed. The last stage of design involves managing data and internal algorithm timing.

## 6 All-Source Navigation Overview

### 6.1 Coordinate Frames

Three key coordinate frames are used within the navigation system: 1) Earth-Centred Inertial (ECI) Frame, 2) Earth-Centred-Earth-Fixed (ECEF) Frame, 3) Local Level North-East-Down Frame (NED) Frame.

The ECI frame has its origin at the centre of the Earth. The  $x_i$ -axis points towards the Vernal Equinox, the  $z_i$ -axis points in the direction of the Earth's angular momentum vector (out of the North Pole), and the  $y_i$ -axis completes the right-handed triad. Fig. 7 shows the definitions of the three reference frames.

The inertial frame remains fixed relative to the stars. This is not a good reference frame for navigation because we would require knowledge of the sidereal time  $\theta_s$ , which describes the angle of Earth's prime meridian with respect to the Vernal Equinox. Even small errors in this estimate results in large position errors and hence it is not a desirable frame to use for terrestrial navigation. Since the aircraft is moving close to the Earth's surface, it is preferable to use a coordinate frame that remains fixed relative to the Earth. This is the ECEF frame, which has its  $z_e$ -axis aligned with  $z_i$ , but the  $x_e$ -axis lies in the plane of the prime meridian (zero longitude), and the  $y_e$ -axis completes the right-handed triad. The ECEF

frame is needed when considering navigating globally on the Earth. The ECEF frame remains invariant to variations in the Earth model used to represent the position of a vehicle relative to the surface.

The local level NED frame is defined such that its  $x$ -axis points North,  $y$ -axis points East, and the  $z$ -axis points downwards along the local vertical. This frame is attached to the local tangent plane at the Earth's surface. The Earth is not a perfect sphere and bulges outwards at the equator. It is modeled as an ellipsoid, and the WGS-84 ellipsoid is the current international standard. This means that, in general, the local vertical does not point towards the center of the Earth, as shown in Fig. 8. The WGS-84 ellipsoid gives a good approximation for the local direction of the gravity vector at the Earth's surface. The local direction of gravity is used for defining the orientation of the vehicle for flight control and waypoint navigation.

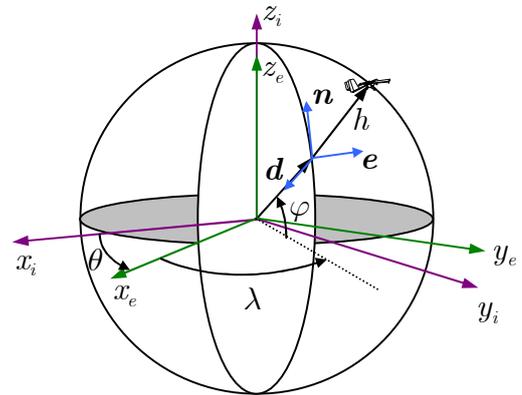


FIG. 7. REFERENCE FRAMES USED FOR TERRESTRIAL NAVIGATION.

The WGS-84 ellipsoid uses a set of geodetic variables to determine position. These are the latitude,  $\varphi$ , longitude,  $\lambda$ , and height,  $h$  (LLH). Note that the altitude above the ellipsoid is different from the altitude above ground or altitude above sea level. The ECEF coordinates for a given LLH position are given by

$$x_e = \left( \frac{R_e}{\sqrt{1 - e^2 \sin^2 \varphi}} + h \right) \cos \varphi \cos \lambda \quad (1)$$

$$y_e = \left( \frac{R_e}{\sqrt{1 - e^2 \sin^2 \varphi}} + h \right) \cos \varphi \sin \lambda \quad (2)$$

$$z_e = \left( \frac{R_e}{\sqrt{1 - e^2 \sin^2 \varphi}} [1 - e^2] + h \right) \sin \varphi \quad (3)$$

The inverse of Eqs. (1) to (3) can be computed from a variety of approximate or closed-form solutions.

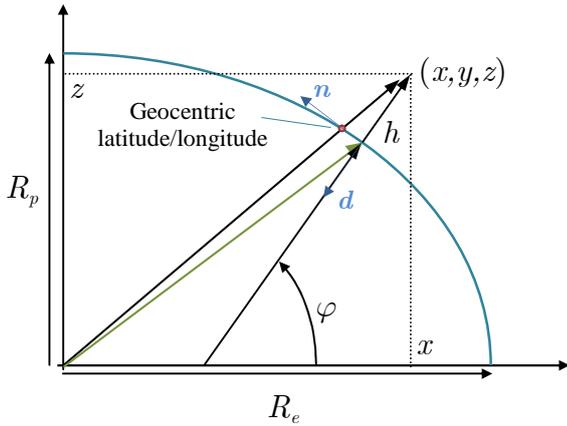


FIG. 8. COORDINATES OF THE REFERENCE ELLIPSOID.

The NED frame is related to the ECEF frame through the geodetic coordinates as follows

$$\begin{bmatrix} n \\ e \\ d \end{bmatrix} = \begin{bmatrix} -\sin \varphi \cos \lambda & -\sin \varphi \sin \lambda & \cos \varphi \\ -\sin \lambda & \cos \lambda & 0 \\ -\cos \varphi \cos \lambda & -\cos \varphi \sin \lambda & -\sin \varphi \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \quad (4)$$

The aircraft body frame is defined relative to the NED frame via the Euler angles. The IMU itself is not necessarily aligned with the aircraft body, and its orientation is defined by a 3-2-1 Euler rotation sequence relative to the aircraft body frame as shown in Fig. 9.

## 6.2 Camera Calibration

The camera used on platform is calibrated using a two-step process. In the first step, the camera is detached from the aircraft and a series of images are taken of a checkerboard pattern of known geometry from different perspectives. The camera calibration toolbox developed for MATLAB is utilized to extract the corners of

the checkerboard pattern and to obtain the intrinsic camera parameters  $f_u$ ,  $f_v$ ,  $u_0$ , and  $v_0$ . In addition to this, a fifth order distortion model is used to correct the warping of the images.

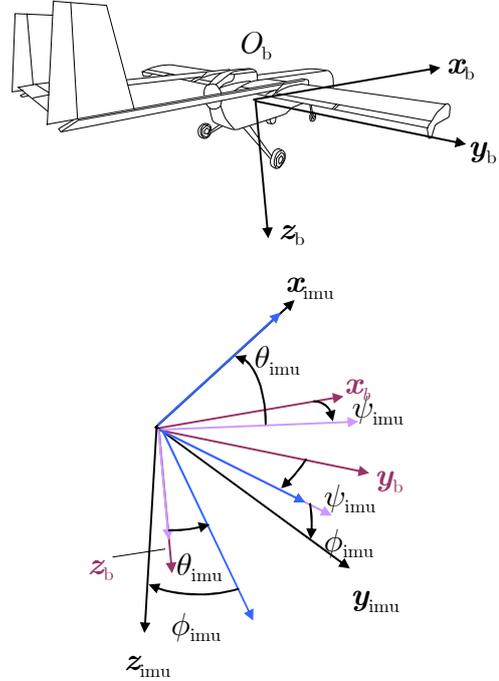


FIG. 9. DEFINITION OF THE AIRCRAFT BODY FRAME AND THE ORIENTATION OF THE IMU FRAME.

The camera is then installed on the aircraft in a turret whose location has been measured relative to the IMU. The second stage of the calibration involves estimating the transformation between the camera frame and the aircraft body frame,  $C_c^b$ . To obtain these estimates, the aircraft is flown over a series of targets that have been placed on the ground and surveyed to centimeter accuracy. The observations of the targets are then matched to the known targets and a least squares optimization is carried out using the installation angles  $(\phi_s, \theta_s, \psi_s)$  of the sensor as parameters. This procedure is discussed in more detail in the companion paper [2].

### 6.3 Navigation Alignment

Navigation alignment must be performed before the navigation system can commence its core task of predicting and correcting the navigation state. The alignment procedure is intended to provide the initial estimate of the state vector. In high grade systems, alignment can take several minutes to complete due to the use of gyro-compassing to obtain attitude. Unfortunately, MEMs IMUs cannot be used for this purpose due to the high level of noise present on the gyro readings.

#### 6.3.1 Static Alignment

During static alignment, the initial attitude is obtained from a set of inclinometers that have significantly lower bandwidth than the accelerometers. Initial heading can be provided either via an operator command, or by using dual GPS antennae. IMU bias estimates are obtained by using the measured attitude and the computed gravity vector, as well as the computed Earth rate projected into the IMU axes. Initial position is obtained from the GPS unit. For static alignment, the initial velocity is necessarily zero.

#### 6.3.2 In-Motion Alignment

In-motion alignment is much more challenging because assumptions cannot be made about the platform motion. Our in-motion alignment algorithm utilizes point samples of position and velocity from the GPS unit, and an estimate of heading from the velocity vector. The initial attitude is computed using a multiple model-like approach. When a model converges, the state vector is declared usable and the main navigation process commences. We have found that our approach works successfully in flight during maneuvers, i.e., banked turns, despite very large initial attitude errors.

## 7 Simulation Architecture

All components of the system are implemented in the Simulink environment (running MATLAB R2010b SP1). Simulink provides a powerful model-based design

environment that can be linked to a set of functional requirements described in a set of IBM Rational DOORS modules. In fact, all system interfaces are described in DOORS modules, from which a complete set of interface code for C++ and MATLAB is autogenerated.

The complete flight vehicle is modeled using Simulink blocks (engine, undercarriage, aerodynamics, actuators), together with a representative model of the environment (wind, gust, turbulence, gravity, atmosphere, ground). Flight physics are modeled in the Earth-Centered-Earth-Fixed (ECEF) coordinate frame. In addition to the physics model of the vehicle, a complete sensor simulation with interfaces and noise characteristics matching the real sensors is performed (Inertial Measurement Unit, GPS, air data, etc). The sensor data is fed into a Simulink model of the flight control computer to enable simulation of the closed-loop control system. Note that because the flight control computer code is generated from the Simulink models via Real-Time Workshop, the simulation is virtually an exact match to the implementation on the target hardware on the real vehicle. The high fidelity simulation environment allows the ASN algorithms to be tested completely without ever having an aircraft in the sky.

## 8 Embedded Real-Time Implementation

The algorithms that are designed, implemented and tested in Simulink are converted into C code by the Real-Time Workshop Embedded Coder. As noted above, a set of software interfaces to the code is autogenerated from DOORS, minimizing the amount of hand-code that is required to run the software as an executable on the target hardware. The C code is wrapped in C++ with appropriate data and message handling code. It is scheduled using a real-time scheduler which runs in the main thread at 100 Hz. The linux operating system running a real-time kernel is used for the ASN system. The Green Hills Integrity operating system is used for the actual flight control computer.

To ensure maximum efficiency, we have developed a set of highly optimized S-functions for performing the expensive matrix operations required for the Kalman filter implementation. These functions exploit features from the Intel Math Kernel Library [47]. In fact, the autocode generated from Simulink is configured to automatically use the Intel library for all matrix operations performed throughout the ASN system. This significantly speeds up the real-time implementation.

In addition to the core algorithms, an additional process is executed to log the input and output data of the algorithmic code. All the inputs and outputs are logged, which enables offline data replays to be conducted using a set of support tools that we have developed. This enables bugs to be found and removed in a rapid fashion.

## 9 Hardware Overview

The flight vehicle platform that we have used to conduct flight operations is the Kingfisher 2 vehicle, shown in Fig. 10. Kingfisher 2 was designed and built by BAE Systems Australia to allow rapid prototyping of payloads and is not a production system. The key characteristics of the platform are given in Table 4.



FIG. 10. KINGFISHER 2 VEHICLE LANDING AT WEST SALE.

The flight control computer forms the core product of the vehicle management system, which consists of an actuation unit, two GPS

units, an IMU, an air data system, weight on wheel sensors, an accurate height sensor, and a C2 communications system. The FCC uses a Radstone IMP2A as the hardware board. The FCC runs a variety of core processes in separate address spaces to maintain a high integrity system. The FCC mounting in the airframe, together with the payload computer and turret controller, described in [2], is shown in Figs. 11 and 12.

A ground station (Fig. 13) that communicates over a C2 link is used to control and monitor the vehicle. It uses the same hardware as the FCC for reliability, but additionally utilizes a Windows-based graphical user interface.

TABLE 4. KINGFISHER PLATFORM ATTRIBUTES.

Attribute	Value
<b>Mass (including payload)</b>	125 kg
<b>Wing span</b>	4.13 m
<b>Wing area</b>	2.67 m <sup>2</sup>
<b>Max. airspeed</b>	100 kts
<b>Max. cross-wind</b>	15 kts
<b>Max. tail-wind</b>	10 kts

A manual handset is used by an external pilot to take over control in the case of failure of a flight critical component (such as IMU or air data). The pilot's commands are sent to the vehicle via the ground station and bypass the core autonomous processes (the inputs are scaled and limited to prevent the pilot from overstressing the airframe). In the flight tests reported in this work, the pilot was never required to take control of the vehicle.

The ASN algorithms are housed on a Kontron CP308 board, which features a Core-2 Duo. One core is dedicated to running the All-Source Navigation system, and the second core is dedicated to running the Intelligent Landing System described in [2]. In addition, inputs and outputs from all processes are logged on a solid state hard drive. The turret control subsystem runs on a Kontron CP307, and is responsible for managing the turret control and logging all raw imagery. A 50 minute flight generates roughly 40 GB of image data.



FIG. 11. ALL-SOURCE NAVIGATION TEST HARDWARE.

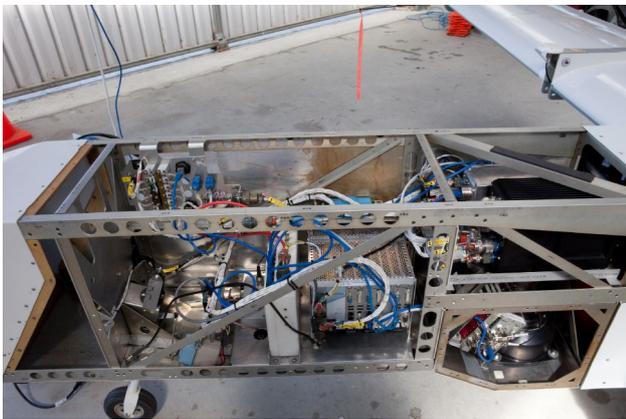


FIG. 12. FCC AND ASN HARDWARE INSTALLED IN KINGFISHER 2.

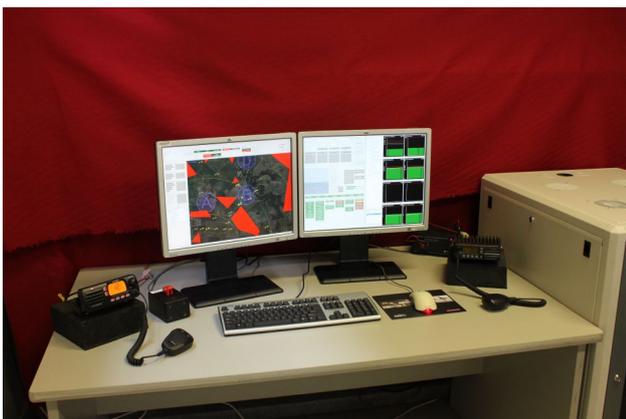


FIG. 13. GROUND CONTROL STATION USED FOR CONTROLLING KINGFISHER 2 UAV.

## 10 West Sale Aerodrome Operations

The Kingfisher 2 vehicle is operated and maintained at a facility situated near the West Sale Aerodrome, see Fig. 14. The ground station is housed in the hangar, and communication antennae as well as a differential GPS antenna are mounted on the hangar roof.

The vehicle is first prepared for flight at the hangar, and navigation alignment and pre-flight tests are conducted outside of the hangar. The aircraft is connected to an external power source to prevent the internal batteries from running low. Once standard operation of the system is confirmed, it is towed on the back of a trailer to one of the ends of the runway. The takeoff direction is specified either via air traffic control (when inside ATC hours), or by the wind direction (when outside ATC hours). After the appropriate set of radio calls are made to ATC, the vehicle is deployed to the runway and becomes airborne under autonomous control.



FIG. 14. VIEW OF BAE SYSTEMS AUSTRALIA'S HANGAR FACILITY.

During UAV operations, other aircraft frequently takeoff and land at the Aerodrome, and the operations crew must ensure separation is maintained (this is achieved via the autonomous system using features such as loiter, heading hold, altitude hold, etc.). Because the ASN system is experimental and runs SLAM closed-loop, it can be enabled or disabled manually by the vehicle operator. By default it is disabled. The FCC also performs

integrity checking on the ASN system and associated navigation data. If the FCC determines that ASN is faulty, it locks it out for a pre-determined period of time and only allows it to be re-enabled if it is healthy. This may happen, for example, during GPS denied testing if the system drifts by a large amount.

## 11 Flight Test Results

The ASN system was flight tested during November 2011 at the West Sale facility. One of the goals of the flight testing was to demonstrate the Intelligent Landing System, described in [2]. Results from those trials are presented in [2]. A total of 22 flights were undertaken. ASN was running for 21 flights, and was in active control of the vehicle for 20 flights. This means that the ASN navigation solution was flying the plane closed-loop. GPS was enabled for the majority of flights while confidence in image processing was gained, and algorithms tuned using real data. We conducted 8 flights with closed-loop GPS-denied navigation. This section only presents the results obtained from one of those flights.

### 11.1 In-Flight Alignment Results

In-flight navigation alignment was performed on a number of occasions. This was during testing of the algorithm, as well as after manually restarting the system in-air. In all cases the navigation system successfully aligned and was able to fly and land the plane.

An example of in-flight alignment is shown in Fig. 15. In-flight alignment was triggered during the turn immediately after takeoff. This represents a non-level situation where the effect of angular misalignment is more pronounced. The accuracy of the alignment can be seen by examining the filter corrections that are applied following the alignment. Note that the results shown include the actual state during the alignment. The only discernible correction occurs in the fusion following the commencement of the alignment.

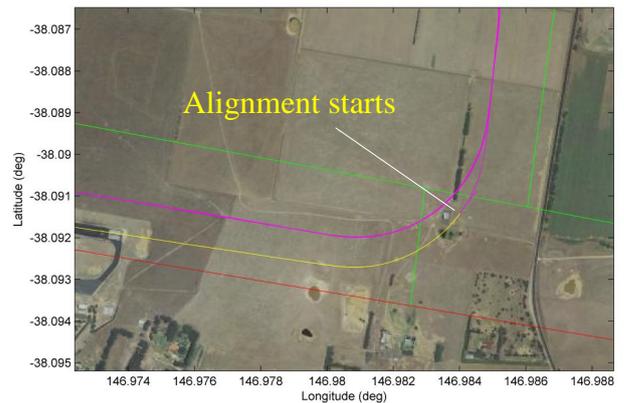


FIG. 15. ASN GROUND TRACK PRIOR TO, AND FOLLOWING IN-FLIGHT ALIGNMENT DURING A TURN.

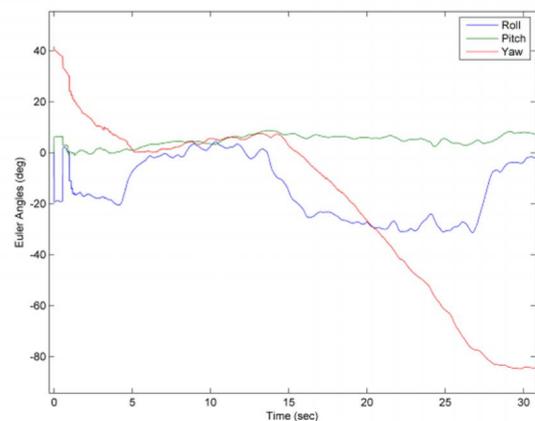


FIG. 16. ASN IN-FLIGHT ALIGNMENT FILTER RESULTS FOR EULER ANGLES.

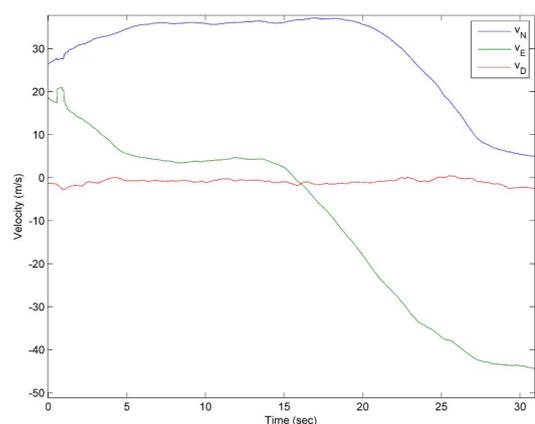


FIG. 17. ASN IN-FLIGHT ALIGNMENT VELOCITY RESULTS.

Fig. 16 shows the Euler angles following the commencement of alignment. This illustrates that the filter corrections tend to converge to small values within the first two

seconds. The initial roll angle of  $\sim 20$  deg is handled easily by the alignment algorithm. Fig. 17 shows the corresponding velocities in the period following the commencement of alignment. It can be seen that the velocity converges extremely well after approximately 2 seconds. We conclude that the in-flight alignment algorithm is suitable for non-level flight conditions and demonstrates excellent convergence.

## 11.2 GPS-Denied Results

In the GPS-denied flight test shown in this section, GPS was turned off during a portion of the flight for approximately 9 minutes. Although not shown here, ASN has been successful in navigating for 30 minutes in-flight without GPS, and for over 24 hours in hardware-in-the-loop simulations.

Fig. 18 shows an aerial plot of the ASN navigation solution. Fig. 18 shows the SLAM map formed by SLAM measurements extracted from log data. The locations where GPS is disabled and re-enabled are shown. The aircraft is put into a loiter during the mission, during which time the SLAM solution is active. The results show the varying nature of the mapped features in terms of their uncertainties. The benefit of SLAM is clearly evident when compared to the free-inertial solution. The free-

inertial solution begins to drift following the first turn after GPS is denied. The solution then rapidly diverges. In comparison, the ASN solution remains stable and accurate when compared with the GPS-enabled solution. The GPS-enabled solution is computed from the log data via a replay.

Fig. 19 shows the NED position difference between the SLAM solution and the GPS-enabled navigation solution. It can be seen that the position error remains constrained in the absence of GPS due to loop closure of the SLAM solution. The position error peaks at 11 m in the East direction, but is generally less than 10 m. The error in height tends to be more stable than horizontal position due to the use of pressure altitude. Fig. 20 shows the Euler angle difference between the SLAM solution and the GPS-enabled navigation solution. The maximum error remains less than 1.5 deg. Fig. 21 shows the execution time of the ASN computer during a SLAM flight. The peak execution time is 2.75 ms, which is substantially lower than the 10 ms available. The mean execution time is 0.39 ms. The execution time alternates between troughs and peaks. The peaks occur on fusion frames.

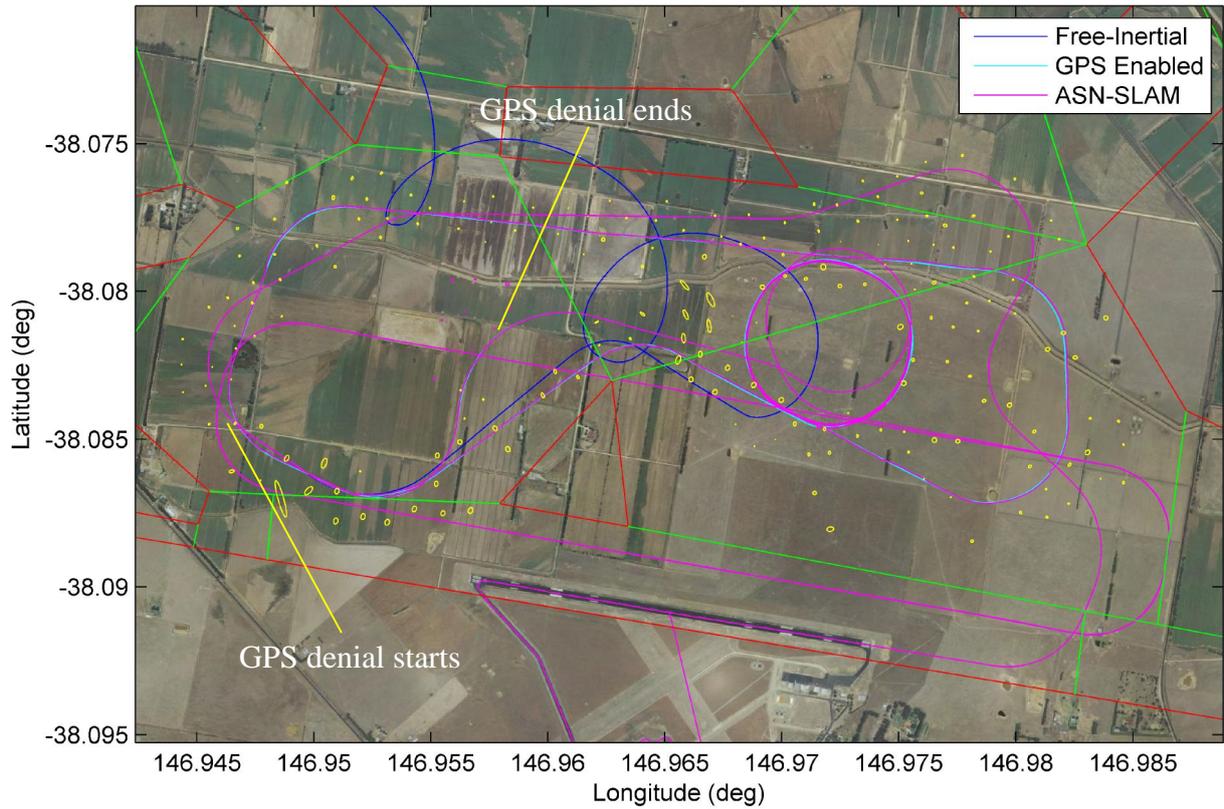


FIG. 18. ASN FLIGHT RESULTS WITH SLAM AND GPS DENIAL ACTIVE, SHOWING FREE-INERTIAL SOLUTION (NO SLAM DURING GPS DENIAL), GPS ENABLED SOLUTION, AND SLAM-ENABLED SOLUTION. YELLOW ELLIPSES ARE MAP STORE FEATURES/COVARIANCES. MAGENTA ELLIPSES ARE ACTIVE MAP FEATURES/COVARIANCES.

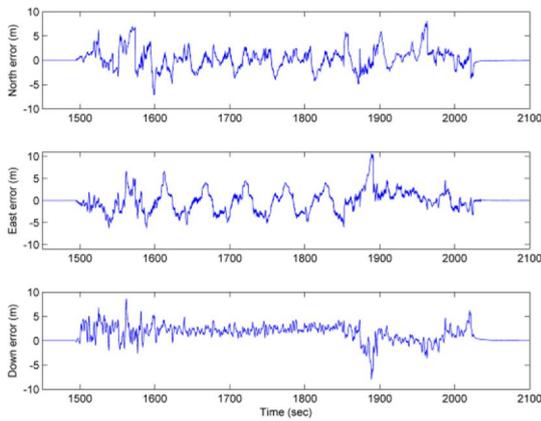


FIG. 19. NORTH, EAST, AND DOWN COMPONENTS OF POSITION ERROR OF SLAM SOLUTION COMPARED WITH GPS SOLUTION.

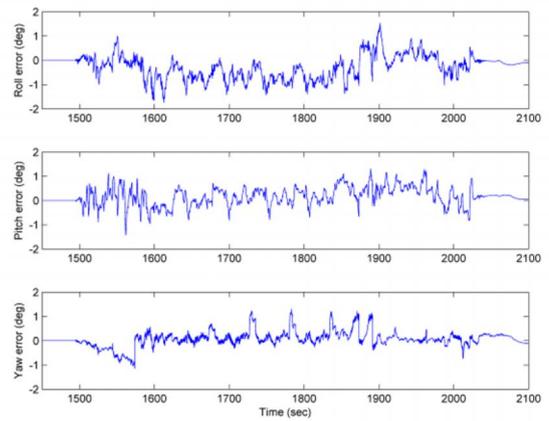


FIG. 20. EULER ANGLE ERRORS OF SLAM SOLUTION COMPARED WITH GPS SOLUTION.

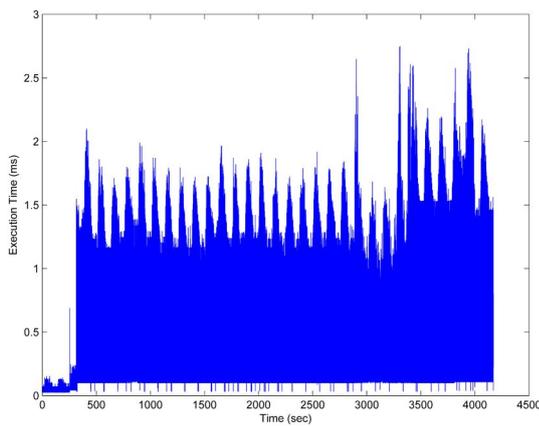


FIG. 21. ASN FRAME EXECUTION TIME FOR FLIGHT INCLUDING SLAM.

## 12 Conclusions

All-source navigation is an architecture that allows virtually any sensor to be integrated and used by the navigation system. It is designed to be robust to faulty measurements, and out of order sensor data. It handles sensors with variable and different latencies. The ASN system is able to operate in GPS-denied environments by using a combination of data gathered from a downward looking camera. These include salient features, and frame-to-frame tracking. The ASN system has been extensively tested in a fully representative simulation environment and test flown on 20 flights of the Kingfisher 2 UAV. It has been successfully used to perform in-flight alignment and GPS-denied navigation.

## Acknowledgements

The authors would like to thank Ramon Wilkinson, Kynan Graves, Andrew Fairmaid, Travis Pereira, Nelson Evans, Jason Bult, and Paul Harbison for their efforts in supporting the development of ASN, and their dedication to the flight trials

## References

[1] Smith, R.C., On the representation of spatial uncertainty. *Int. J. Robotics Research*, 5(4), 1987, pp.56-68.

- [2] Williams, P., and Crump, M., Intelligent landing system for landing UAVs at unsurveyed airfields. *Proceedings of the 28<sup>th</sup> International Congress of the Aeronautical Sciences*, Brisbane, September 2012.
- [3] Durrant-Whyte, H., and Bailey, T., Simultaneous Localization and Mapping: Part I. *IEEE Robotics & Automation Magazine*, June 2006, pp. 99-108
- [4] Bailey, T., and Durrant-Whyte, H., Simultaneous Localization and Mapping: Part II. *IEEE Robotics & Automation Magazine*, September 2006, pp. 108-117
- [5] Castellanos, J.A., Tardos, J.D., and Schmidt, G., Building A Global Map of the Environment of a Mobile Robot: The Importance of Correlations. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, 1997, pp.1053-1059.
- [6] Bailey, T., and Nebot, E., Localisation In Large-Scale Environments. *Robotics and Autonomous Systems*, Vol. 37, No. 4, pp. 261-281, 2001.
- [7] Majumder, S., Scheding, S., and Durrant-Whyte, H., Sensor Fusion and Map Building for Underwater Navigation. *Proceedings of Australian Conference on Robotics and Automation*, Melbourne, Australia, Aug 30-Sep 1, 2000.
- [8] Kim, J.-H., and Sukkarieh, S., Airborne Simultaneous Localisation and Map Building. *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, September 2003, pp.406-411.
- [9] Newman, P., Leonard, J., Tardos, J.D., and Neira, J., Explore and Return: Experimental Validation of Real-Time Concurrent Mapping and Localization. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002, pp.1802-1809.
- [10] Guivant, J., and Nebot, E., Optimization of the Simultaneous Localization and Map Building Algorithm for Real Time Implementation. *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 3, pp.242-257, 2001.
- [11] Dissanayake, G, Newman, P., Clark, S., Durrant-Whyte, H.F., and Csorba, M., A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 3, pp.229-241, 2001.
- [12] Durrant-Whyte, H., Localization, Mapping and the Simultaneous Localization and Mapping (SLAM) Problem. *SLAM Summer School*, 2002.
- [13] Fitzgibbons, T., and Nebot, E., Application of Vision in Simultaneous Localization and Mapping. *Proceedings of the 2001 Australian Conference on Robotics and Automation*, Sydney, November 2001, pp.121-127.
- [14] Dissanayake, G., Williams, S.B., Durrant-Whyte, H., and Bailey, T., Map Management for Efficient Simultaneous Localization and Mapping (SLAM). *Autonomous Robots*, Vol. 12, pp.267-286 2002.
- [15] Guivant, J., Nebot, E., and Baiker, S., Autonomous Navigation and Map building Using Laser Range Sensors in Outdoor Applications. *Journal of Robotic Systems*, Vol. 17, No. 10, pp.565-583, 2000.

- [16] Tardos, J.D., Neira, J., Newman, P.M., and Leonard, J.J., Robust Mapping and Localization in Indoor Environments using Sonar Data. *The International Journal of Robotics Research*, Vol. 21, No. 4, pp.311-330, 2002.
- [17] Williams, S.B., Dissanayake, G., and Durrant-Whyte, H., An Efficient Approach to the Simultaneous Localisation and Mapping Problem. *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, DC, May 2002, pp.406-411.
- [18] Thrun, S., Burgard, W., and Fox, D., A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots. *Autonomous Robots*, Vol. 31, No. 5, pp.1-25, 1998.
- [19] Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B., FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002.
- [20] Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B., FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. *Proceedings of the 18th international joint conference on Artificial intelligence*, 2003, pp.1151-1156.
- [21] Guivant, J., and Nebot, E., Compressed Filter for Real Time Implementation of Simultaneous Localization and Map Building. *Proceedings of the International Conference on Field and Service Robotics*, 2001, pp.309-314.
- [22] Leonard, J.J., and Feder, H.J.S., A Computationally Efficient Method for Large Scale Concurrent Mapping and Localization. *Proceedings of the Ninth International Symposium on Robotics Research*, pp.169-176, 1999.
- [23] Nieto, J.I., Guivant, J.E., and Nebot, E.M., The HYbrid Metric Maps (HYMMs): A Novel Map Representation for DenseSLAM. *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2004, pp.391-396.
- [24] Andrade-Cetto, J., Vidal-Calleja, T., and Sanfeliu, A., Unscented Transformation of Vehicle States in SLAM. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp.323-328, 2005.
- [25] Bailey, T., Nieto, J., and Nebot, E., Consistency of the FastSLAM Algorithm, *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.
- [26] Davison, A.J., Cid, Y.G., and Kita, N., Real-Time 3D SLAM With Wide-Angle Vision. *Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisboa, July 2004.
- [27] Davison, A.J., and Kita, N., 3D Simultaneous Localization and Map-Building Using Active Vision for a Robot Moving on Undulating Terrain. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, April 2001, pp. 384-391.
- [28] Kim, J., and Sukkarieh, S., Real-Time Implementation for Airborne inertial-SLAM. *Robotics and Autonomous Systems*, Vol. 55, pp.62-71, 2007.
- [29] Hygounenc, E., Jung, I.-K., Soueres, P., and Lacroix, S., The Autonomous Blimp Project of LAAS-CNRS: Achievements in Flight Control and Terrain Mapping. *The International Journal of Robotics Research*, Vol. 23, No. 4-5, pp.473-511, 2004.
- [30] Bryson, M., and Sukkarieh, S., Bearing-Only SLAM for an Airborne Vehicle. *Proceedings of the Australian Conference on Robotics and Automation*, Sydney, Australia, 2005.
- [31] Fenwick, J.W., Newman, P.M., and Leonard, J.J., Cooperative Concurrent Mapping and Localization. *Proceedings of the IEEE International Conference on Robotics and Automation*, August 2002, pp.1810-1817.
- [32] Nettleton, E.W., Gibbens, P.W., Durrant-Whyte, H.F., Closed Form Solutions to the Multiple Platform Simultaneous Localization and Map Building (SLAM) Problem. *Sensor Fusion: Architectures, Algorithms, and Applications IV*, pp.428-437, 2000
- [33] Nettleton, E., Thrun, S., Durrant-Whyte, H., and Sukkarieh, S., Decentralised SLAM with Low-Bandwidth Communication for Teams of Vehicles. *Proceedings of the 4th International Conference on Field and Service Robotics (FSR '03)*, Japan, July 2003, pp.179-188.
- [34] Nieto, J., Guivant, J., Nebot, E., and Thrun, S., Real Time Data Association for FastSLAM. *Proceedings of the IEEE International Conference on Robotics and Automation*, September 2003, pp.412-418.
- [35] Walter, M., and Leonard, J., An Experimental Investigation of Cooperative SLAM. *Proceedings of the Fifth IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, July 2004.
- [36] Ong, L.L., Ridley, M., Kim, J.-H., Nettleton, E., and Sukkarieh, "Six DoF decentralized SLAM," *Proceedings of the Australasian Conference on Robotics and Automation*, Brisbane, Australia, December 2003, pp. 10-16.
- [37] Bryson, M., and Sukkarieh, S., Co-Operative Localisation and Mapping for Multiple UAVs in Unknown Environments. *Proceedings of the IEEE/AIAA Aerospace Conference*, Big Sky, 2007.
- [38] Bryson, M., and Sukkarieh, S., Decentralised Trajectory Control for Multi- UAV SLAM. *Proceedings of the 4<sup>th</sup> International Symposium on Mechatronics and its Applications*, Shajah, 2007.
- [39] Adrien, A., Filliat, F., Doncieux, S., Meyer, J.-A., 2D Simultaneous Localization and Mapping for Micro Air Vehicles. *Proceedings of the European Micro Aerial Vehicles Conference*, Vol. 1, pp.1-10, 2006
- [40] Ellekilde, L.-P., Huang, S., Miro, J.V., and Dissanayake, G., Dense 3D Map Construction for Indoor Search and Rescue. *Journal of Field Robotics*, Vol. 24, Nos.1-2, pp.71-89, 2007.
- [41] Thrun, S., and Montemerlo, M., The GraphSLAM Algorithm with Applications to Large-Scale Mapping

- of Urban Structures. *The International Journal of Robotics Research*, Vol. 25, No. 5–6, pp.403-429, 2006.
- [42] Eustice, R.M., Singh, H., and Leonard, J., Exactly Sparse Delayed-State Filters. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp.2417-2424.
- [43] Langelaan, J., and Rock, S., Passive GPS-Free Navigation for Small UAVs, Proceedings of the 2005 IEEE Aerospace Conference, Big Sky, March 2005, pp.1-9.
- [44] Atlantic Inertial Systems, Product Specification for a Silicon IMU, SiIMU04, IMU04-01-0100-120, Revision 6, July 2007.
- [45] Bay, H., Tuytelaars, T., and Gool, L.V., SURF: Speeded Up Robust Features. *Computer Vision – ECCV 2006*, Vol. 3951, pp.404-417, 2006.
- [46] Lowe, D.G., Object Recognition from Local Scale-Invariant Features. *Proceedings of the International Conference on Computer Vision*, 1999, pp.1150-1157.
- [47] Anon., *Intel Math Kernel Library for Windows OS User's Guide, Version 10.3*, Intel Corporation, 2011.

## Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS2012 proceedings or as individual off-prints from the proceedings.