Decentralised approach to UAV navigation: without the use of GPS and preloaded maps

Jonghyuk Kim[1], Lee Ling (Sharon) Ong[1], Eric Nettleton[2] and Salah Sukkarieh[1]
[1]Australian Centre for Field Robotics (ACFR)
School of Aerospace, Mechanical and Mechatronic Engineering
University of Sydney, NSW 2006, Australia
Email: {jhkim/s.ong/salah}@acfr.usyd.edu.au
[2]Advanced Information Processing Department, Advanced Technology Centre, Sowerby Building - 20R
BAE SYSTEMS, PO Box 5, FPC 267 Filton, Bristol BS34 7QW United Kingdom
Email: eric.nettleton@baesystems.com

Abstract

In this paper we address the issue of autonomous navigation, that is, the ability for a navigation system to provide information about the states of a vehicle without the need for a priori infrastructures such as GPS, beacons, or preloaded maps of the area of interest. The algorithm applied is known as Simultaneous Localisation and Mapping (SLAM). It is a terrain aided navigation system which has the capability for online map building, while simultaneously utilising the generated map to bound the errors in the navigation solution. As no a priori terrain information nor initial knowledge of the vehicle location is required, this algorithm presents a powerful navigation augmentation system. More importantly, it can be implemented as an independent navigation system. This paper also describes a decentralised SLAM algorithm which allows multiple vehicles to acquire a joint 3D map via a decentralised information fusion network. The key idea behind this decentralised SLAM is to represent the map in information form (negative log-likelihood) for communication. Experimental results are provided using computer simulation to demonstrate the single-vehicle and multi-vehicles SLAM without the use of GPS and preloaded maps.

Keywords

Simultaneous localisation and mapping, decentralised data fusion, IMU, vision sensor, UAV.

I. Introduction

Airborne navigation systems can generally be divided into two categories: inertial (or dead-reckoning) based navigation, and reference (or absolute) based navigation.

An Inertial Navigation System (INS) makes use of an Inertial Measurement Unit (IMU) to sense the vehicle's rotation rate and acceleration. This data is then used to obtain vehicle states such as position, velocity and attitude. The IMU provides this data at high rates which is crucial for guidance and control. However this sensor error diverges in nature due to the integration process. Hence absolute sensors are required in order to constrain the drift.

Absolute sensors are categorised into two groups: beacon based or terrain based. The most common beacon based navigation system is a Global Positioning System (GPS). There has been extensive research activities in the fusion of INS and GPS systems [1]-[4]. The GPS aided inertial navigation system provides long-term stability with high accuracy in addition to worldwide coverage in any weather condition. The main drawback is its dependency on external satellite signals which can be easily blocked or jammed by intentional interferences.

As a result, research into Terrain Aided Navigation Systems (TANS) as an alternative to relieve the dependency on GPS is an active area [5]-[8]. This type of navigation system typically makes use of onboard sensors and a terrain database. The TERrain COntour Matching (TERCOM) system has been successfully applied in cruise missile systems [5]. It combines onboard radar-altimeter readings with a pre-stored Digitised Terrain Elevation (DTE) map to estimate the INS errors as well as guiding the low-flying missile at a fixed height above the ground. The TERrain PROfile Matching (TERPROM) system correlates passive sensor data with a terrain database. It can provide terrain proximity and avoidance information as well as INS aiding capability and it has been widely adapted as a navigation system within various aircrafts. [9] presents a scene or image matching correlation system which makes use of a passive camera or an infrared camera with an onboard image correlator. The observed image is matched with the pre-stored digital image database. If a correlation peak exists above a given threshold, the position of the image centre can be identified and used to estimate the INS errors. Due to its passive and non-jamming nature, it has been adapted in the terminal guidance stages of missiles.

Both forms of satellite and terrain based absolute navigation systems have their advantages and disadvantages, and in fact the more robust navigation system would have a combination of the two. However, if the mission exists within a GPS denied environment, whether within a military scenario, or for underwater systems, or whether on another planet, then one is left with the implementation of the TAN system. In TANS, the DTE is the key element. However it usually requires some sort of Space mapping infrastructure as it is typically built from high resolution satellite radar images around the mission area. Furthermore, it has a constrained degree of autonomy since the mission is bound to the knowledge of the terrain database. One would like a system which can further expand on the existing DTE, by either augmenting information in the form of new frontiers that have been seen outside of the spatial scope of the DTE, or by adding information in terms of higher quality data within the existing map. The objective however is to use this information to then bound the uncertainty in the navigation solution. Thus in order to extend the benefit of TANS the navigation system requires the ability to augment map data as it is generated, and to use the newly generated map to constrain the drift of the INS, that is, to simultaneously build a map and to localise the vehicle within it. If implemented properly, this concept can be used when there is no a priori information whatsoever about the map, about the landmarks within the map, or about the vehicle location within the map as well.

Simultaneous Localisation And Mapping (SLAM) was first addressed in the paper by Smith and Cheeseman [10] and has evolved from the indoor robotics research community to explore unknown environments, where absolute information is not available [11]-[16].

The SLAM structure can be described as shown in Fig. 1. The vehicle starts its navigation at an unknown location in an unknown environment. The vehicle navigates using its dead-reckoning sensors or vehicle model. As the onboard sensors detect features from the environment, the SLAM estimator augments the landmark locations to a map in some global reference frame and begins to estimate the vehicle and map states together with successive observations. The ability to estimate both the vehicle location and the map is due to the statistical correlations which exist within the estimator between the vehicle and landmarks, and between the landmarks themselves. As the vehicle proceeds through the environment and re-observes old landmarks, the map accuracy converges to a lower limit which is a function of the initial vehicle uncertainty when the first landmark was observed [15]. In addition, the vehicle uncertainty is also constrained simultaneously.

The SLAM architecture has four interesting characteristics:

- **Point feature:** In the context of SLAM, landmarks are the features of the environment that can be consistently and reliably observed using the vehicle's onboard sensors. Landmarks must be described in parametric form so that they can be incorporated into a state model. Point feature representation is a simple but efficient representation for this purpose, while conners, lines and polyline feature models which are useful in indoor environments have also been implemented [13].
- **Correlation:** The key element in SLAM is that an error in estimated vehicle location leads to a common error in the estimated location of landmarks as shown in Fig. 2. The vehicle starts at an unknown location and begins to estimate landmark locations from relative observations. As the vehicle traverses, the integrated data from the internal dead-reckoning sensors drift which in turn causes a common error in the landmark location as well. Indeed, it is possible to show that the correlation caused by this common error between landmarks tends to unity with sufficient observations, and thus in the limit a perfect relative map of landmarks can be constructed [13]. It is because of this correlation between the landmarks and the vehicle, that when a re-observation of a previously known stationary landmark occurs, then vehicle state estimation can proceed given this map data.
- **Map complexity:** The need to maintain these correlations is an integral part of the SLAM solution. This leads to enormous computational problems, as the location of each landmark in the environment must, in theory, be updated at each step in the estimation cycle. To retain all correlations requires $\mathcal{O}(n^3)$ computation and $\mathcal{O}(n^2)$ storage requirement, where n is the number of features, which is intractable as the size of the operation environment is increased. This leads inevitably for a need to find effective map management policies for large scale problems [13][16].
- **Revisiting Landmarks:** The most interesting aspect of SLAM is "closing-the-loop" or the revisiting process. The vehicle's error grows without bound due to the drifting nature of the dead-reckoning sensor and this affects the generated map accuracy as well. However if the vehicle has a chance to revisit a former registered landmark, the accumulated vehicle error can be estimated which in turn, improves the overall map accuracy as well. This process makes it possible to build a perfect relative map of landmarks in the limit.

There have been substantial advances over the recent years in developing the SLAM algorithm for field robotics particularly for land and underwater vehicles [12][14][16], all of which however assume a flat and 2D environment. The research conducted has illustrated the problems and remedies associated with the construction of the algorithm, the requirement for re-observing landmarks for model drift containment, and issues relating to data association.

Decentralised SLAM addresses the problem in which large numbers of vehicles cooperatively acquire a joint map, while simultaneously localising themselves in the map. Previous work has been carried out on decentralised SLAM on a flat 2D environment [17]. Here, the Extended Information Filter (EIF) is used to represent information acquired by the vehicle. The EIF is the information form of the EKF. The update of this information is additive. Hence, the incremental new information can be integrated across different vehicles with arbitrary network latencies.

In this paper, we will present the first implementation of the decentralised SLAM algorithm for a 6DoF platform navigating within a 3D environment, thus providing a revolutionary step for navigation systems for airborne applications.

The work described in this paper is part of the Autonomous Navigation and Sensing Experimental Research (ANSER) project which aims at developing a multiple flight vehicle demonstration of decentralised SLAM. The system consists of four uninhabited air vehicles, each equipped with inertial sensors, GPS and one or two payloads; consisting of either a vision system, a mm-wave radar or a vision/laser hybrid sensor.

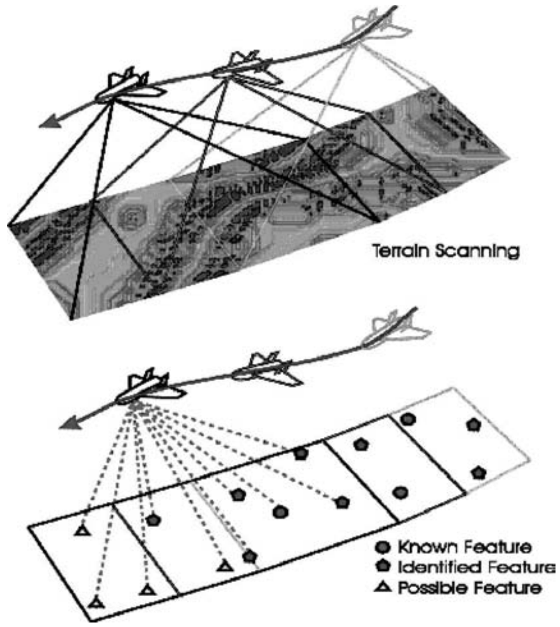


Fig. 1. SLAM system is building a relative map of feature-based landmarks using relative observations, defining a map, and using this map to localise the vehicle simultaneously.

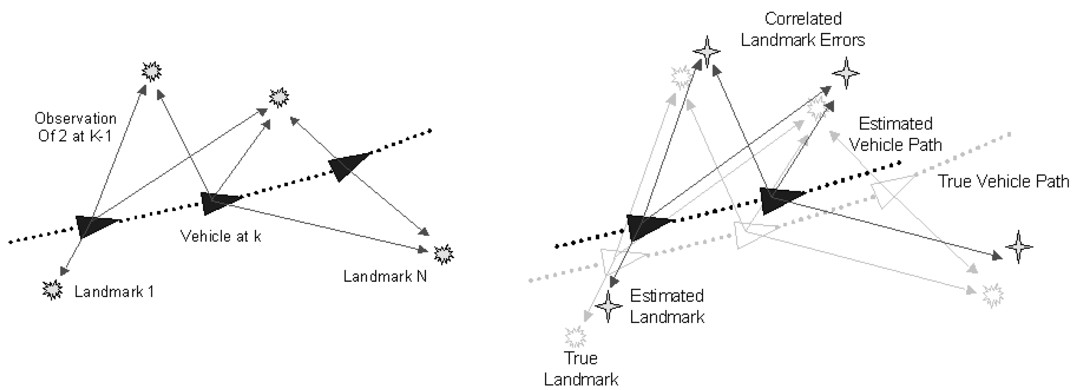


Fig. 2. The vehicle starts at an unknown location with no a priori knowledge of landmark locations and estimates the vehicle and landmark locations (left). The landmark estimates are subject to a common error from the vehicle uncertainty and eventually, all landmarks will be completely correlated (right).

Section II will formulate the airborne SLAM algorithm and filter structure using an extended Kalman filter. Section III will describe the decentralised architecture. Section IV will describe the platform and sensors on which SLAM and decentralised SLAM will be applied. Section V simulation results are provided using a high fidelity simulator based on the flight vehicles. In particular the focus of the simulation results is to determine the impact on the map and localisation accuracy based on variability in landmark spatial density and the quality of both the inertial and observation sensor. The simulation results consist of a single vehicle SLAM and two aircraft decentralised SLAM. Section VI will finish with conclusion and future work.

II. Airborne SLAM Algorithm

The mathematical framework of the SLAM algorithm is based on an estimation process which, when given a kinematic/dynamic model of the vehicle and relative observations between the vehicle and landmarks, estimates the structure of the map and the vehicle's position, velocity and orientation within that map. In this work, the Extended Kalman Filter (EKF) is used as the state estimator.

A. Nonlinear Process Model

The process model includes the vehicle and map dynamic model and can be written as a first-order vector difference equation in discrete time,

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k), \mathbf{w}(k), k) \tag{1}$$

where $\mathbf{f}(\cdot,\cdot,k)$ is a non-linear state transition function at time k which forms the current vehicle and map state, $\mathbf{x}(k)$, from the previous state, $\mathbf{x}(k-1)$, and the current control input, $\mathbf{u}(k)$. $\mathbf{w}(k)$ is the process noise vector. The state can be partitioned into vehicle state and map state and the process model can be separated in each state as,

$$\begin{bmatrix} \mathbf{x}_v(k) \\ \mathbf{x}_m(k) \end{bmatrix} = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_v(k-1), \mathbf{u}(k), \mathbf{w}_v(k), k) \\ \mathbf{f}_m(\mathbf{x}_m(k-1), \mathbf{w}_m(k), k) \end{bmatrix}, \tag{2}$$

where $\mathbf{x}_v(k)$ is the vehicle state comprising of position, velocity and attitude and $\mathbf{x}_m(k)$ is the landmark position. The new landmark position, $\mathbf{x}_{mi}(k) = [\, x \quad y \quad z \,]^T$, is augmented into the state vector so its size increases during flight time, that is,

$$\mathbf{x}_m(k) = [\; \mathbf{x}_{m1}^T(k) \;\; \mathbf{x}_{m2}^T(k) \;\; \ldots \;\; \mathbf{x}_{mN}^T(k) \;]^T, \tag{3}$$

where N is the current number of landmarks in the filter.

The nonlinear vehicle model is a strapdown INS algorithm and it represents the position, velocity and attitude of the UAV. In this paper it is mechanized in the earth fixed tangent frame with Euler angle parameters:

$$\begin{bmatrix} \mathbf{P}^n(k) \\ \mathbf{V}^n(k) \\ \mathbf{\Psi}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{P}^n(k-1) + \mathbf{V}^n(k-1)\triangle t \\ \mathbf{V}^n(k-1) + [\mathbf{C}_b^n(k-1)\mathbf{f}^b(k) + \mathbf{g}^n]\triangle t \\ \mathbf{\Psi}(k-1) + \mathbf{E}_b^n(k-1)\boldsymbol{\omega}^b(k)\triangle t \end{bmatrix},$$

where $\mathbf{P}^n(k)$ and $\mathbf{V}^n(k)$ is the position and velocity respectively, and $\mathbf{\Psi}(k)$ represents the Euler angles: roll (ϕ), pitch (θ) and yaw (ψ). $\mathbf{f}^b(k)$ and $\boldsymbol{\omega}^b(k)$ are acceleration and rotation rates measured in the body frame. $\mathbf{C}_b^n$ is the direction cosine matrix and $\mathbf{E}_b^n$ is the matrix that transforms the rotation rates in the body frame to the Euler angle rates:

$$\mathbf{C}_b^n = \begin{bmatrix} C_\theta C_\psi & -C_\phi S_\psi + S_\phi S_\theta C_\psi & S_\phi S_\psi + C_\phi S_\theta C_\psi \\ C_\theta S_\psi & C_\phi C_\psi + S_\phi S_\theta C_\psi & -S_\phi C_\psi + C_\phi S_\theta S_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix}, \quad \mathbf{E}_b^n = \begin{bmatrix} 1 & S_\phi S_\theta / C_\theta & C_\phi S_\theta / C_\theta \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi / C_\theta & C_\phi / C_\theta \end{bmatrix},$$

where $S_{(\cdot)}$ and $C_{(\cdot)}$ represents $sin(\cdot)$ and $cos(\cdot)$ respectively.

The landmark dynamic model is a stationary model which has no disturbance input noise. Hence the state transition equation for the i^{th} landmark simply becomes

$$\mathbf{x}_{mi}(k) = \mathbf{x}_{mi}(k-1). \tag{4}$$

B. *Nonlinear Observation Model*

The onboard range/bearing sensor provides relative observations between vehicle and landmarks. The nonlinear observation model relates these observations to the state as

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{x}(k)) + \mathbf{v}(k), \tag{5}$$

where $\mathbf{h}(\cdot)$ is the non-linear observation model at time k, and $\mathbf{v}(k)$ is the observation noise vector.

Since the observation model predicts the range, bearing, and elevation for the i^{th} landmark, it is only a function of the i^{th} landmark and the vehicle state. Therefore equation 5 can be further simplified as

$$\mathbf{z}_i(k) = \mathbf{h}(\mathbf{x}_v(k), \mathbf{x}_{mi}(k)) + \mathbf{v}_i(k), \tag{6}$$

where $\mathbf{z}_i(k)$ and $\mathbf{v}_i(k)$ is the i^{th} observation and its associated noise in range, bearing and elevation with zero mean and variance of $\mathbf{R}(k)$.

To formulate the observation model in equation 6, the landmark position in the navigation frame should be related to the sensor observation in the sensor frame as shown in Fig. 3. The landmark position in the navigation frame is

$$\mathbf{x}_{mi}^n(k) = \mathbf{P}^n(k) + \mathbf{C}_b^n(k)\mathbf{P}_{sb}^b + \mathbf{C}_b^n(k)\mathbf{C}_s^b(k)\mathbf{P}_{ms}^s(k), \tag{7}$$

where $\mathbf{P}_{sb}^b$ is the lever arm offset from the vehicle's centre of gravity in the body frame, $\mathbf{C}_s^b(k)$ is a direction cosine matrix which transforms the vector in the sensor frame to the body frame, and $\mathbf{P}_{ms}^s(k) = [\, x \quad y \quad z \,]^T$ is the relative distance of the landmark in the sensor frame which is converted from range, bearing and elevation observations,

$$\mathbf{P}_{ms}^s(k) = \begin{bmatrix} \rho\cos(\varphi)\cos(\vartheta) \\ \rho\sin(\varphi)\cos(\vartheta) \\ \rho\sin(\vartheta) \end{bmatrix}. \tag{8}$$

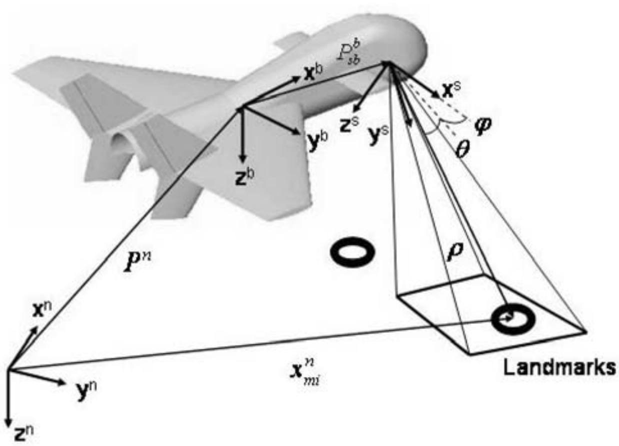


Fig. 3. The range, bearing and elevation observations from the onboard sensor can be related to the location of the landmark in the navigation frame through the flight platform's position and attitude as provided in equation 7.

ρ, φ and ϑ are the range, bearing and elevation values respectively, measured from the onboard sensor.

The predicted range, bearing and elevation between the vehicle and the i^{th} landmark in equation 6 can be obtained by rearranging equation 8,

$$\begin{aligned} \mathbf{z}_i(k) &= [\ \rho \quad \varphi \quad \vartheta\ ]^T \\ &= \begin{bmatrix} \sqrt{x^2+y^2+z^2} \\ \tan^{-1}(y/x) \\ \tan^{-1}(z/\sqrt{x^2+y^2}) \end{bmatrix}, \end{aligned} \tag{9}$$

where $[\ x \quad y \quad z\ ]^T$ is obtained from the vehicle and landmark position in the navigation frame in equation 7 as,

$$\mathbf{P}^s_{ms}(k) = \mathbf{C}^s_b(k)\mathbf{C}^b_n(k)\left[\mathbf{x}^n_{mi}(k) - \mathbf{P}^n(k) - \mathbf{C}^n_b(k)\mathbf{P}^b_{sb}\right]. \tag{10}$$

C. Estimation Process

The EKF is implemented for the estimation of both the vehicle and map states. With the state and observation models defined in the previous section, the estimation procedure can proceed. The state and its covariance are predicted using the control input which typically runs at a high-frequency to track the maneuvering UAV. Whenever a landmark is observed, a data association process is conducted to check to see if the landmark has been previously observed. If the landmark has been previously registered in the filter the observation is used to update the state and covariance, and if the landmark is a new one then a new landmark state is augmented to the filter state.

The state covariance is propagated using the Jacobians of the state transition model and process noise matrix by,

$$\mathbf{P}(k|k-1) = \nabla\mathbf{f_x}(k)\mathbf{P}(k-1|k-1)\nabla\mathbf{f_x^T} + \nabla\mathbf{f_w}(k)\mathbf{Q}(k)\nabla\mathbf{f_w^T}(k),$$

where the terms $\nabla\mathbf{f_x}(k)$ and $\nabla\mathbf{f_w}(k)$ are Jacobians of the non-linear state transition function with respect to the state and sensor noise respectively and they are defined in Appendix A.

When an observation occurs, the state vector and its covariance are updated according to

$$\begin{aligned} \hat{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k-1) + \mathbf{W}(k)\nu(k) \\ \mathbf{P}(k|k) &= [\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h_x}(k)]\mathbf{P}(k|k-1) \\ &\quad \times[\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h_x}(k)]^T + \mathbf{W}(k)\mathbf{R}(k)\mathbf{W}^T(k), \end{aligned}$$

where the innovation vector, Kalman gain, and innovation covariance are computed as,

$$\begin{aligned} \nu(k) &= \mathbf{z}(k) - \mathbf{h}(\hat{\mathbf{x}}(k|k-1)) \\ \mathbf{W}(k) &= \mathbf{P}(k|k-1)\nabla\mathbf{h_x^T}(k)\mathbf{S}^{-1}(k) \\ \mathbf{S}(k) &= \nabla\mathbf{h_x}(k)\mathbf{P}(k|k-1)\nabla\mathbf{h_x^T}(k) + \mathbf{R}. \end{aligned}$$

$\nabla\mathbf{h_x}(k)$ is the Jacobian of the non-linear observation function $\mathbf{h}(\cdot)$ with respect to the predicted state $\mathbf{x}(k|k-1)$ and is defined in Appendix B.

D. Data Association and New landmark Augmentation

Data association is a process that finds out if there is a correspondence between observations at time k and landmarks registered within the filter state. Correct correspondence of sensed landmark observations to mapped landmarks is essential for consistent map construction. A single false match may invalidate the entire estimation process.

As a statistical validation gate, the Normalised Innovation Square (NIS) (also known as the *Mahalanobis distance*) is used to associate observations [18]. Association validation is performed in observation space. Given an innovation and its covariance with the assumption of Gaussian distribution, the NIS forms a χ^2 (chi-square) distribution. If the NIS has a value within a threshold

$$\nu(k)^T\mathbf{S}(k)^{-1}\nu(k) \leq \lambda_n, \tag{11}$$

where n is the dimension of the innovation, the observation and the landmark that were used to form the innovation are then associated. The associated innovation is used to update the state and covariance.

If the landmark is reobserved then the estimation cycle proceeds, otherwise it is a new landmark and must be augmented into both the state vector and the covariance matrix by

$$\mathbf{x}_{aug}(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{g}(\mathbf{x}(k), \mathbf{z}(k)) \end{bmatrix} \tag{12}$$

$$\mathbf{P}_{aug}(k) = \begin{bmatrix} \mathbf{I} & 0 \\ \nabla\mathbf{g}_x(k) & \nabla\mathbf{g}_z(k) \end{bmatrix} \begin{bmatrix} \mathbf{P}(k) & 0 \\ 0 & \mathbf{R}(k) \end{bmatrix} \begin{bmatrix} \mathbf{I} & 0 \\ \nabla\mathbf{g}_x(k) & \nabla\mathbf{g}_z(k) \end{bmatrix}^T \tag{13}$$

where $\nabla\mathbf{g}_x(k)$ and $\nabla\mathbf{g}_z(k)$ are Jacobians for the landmark initialisation function with respect to the current state and observation respectively and are defined in Appendix C. In equation 13 the new landmark state becomes strongly correlated to the vehicle state and the map states become fully correlated to each other as time progresses. This correlation exhibits a unique characteristic in the SLAM implementation: the revisiting process. When the vehicle revisits a former landmark, the accumulated INS drift during this period can be identified and the uncertainties in landmark states can also be reduced due to the strong cross correlation between landmarks.

This completes the EKF cycle for the single vehicle SLAM implementation.

III. Decentralised Airborne SLAM Algorithm

The decentralised architecture and decentralised data fusion (DDF) technique used in this application is based on the information (or inverse covariance) form of the Kalman filter. This builds on the work of Grime [19] on decentralised tracking.

This "information" is propagated to all vehicles or nodes in the network via point-to-point links with no loops in the network, as illustrated in Figure 4. The internal structure of each of these sensing nodes is illustrated in Fig. 5. This section gives an overview of the key elements of the architecture such as the local filter, channel filters and the channel manager as detailed in [20]. For a decentralised SLAM application, the landmark map information in each SLAM node or platform, determined using the algorithms in Section II, has to be communicated to the other node or platforms in the network. Map information received by one node from another has to be fused with its local map to get a better estimate. This section will describe the algorithm applied to achieve these requirements.

A. The Information Filter

The state, $\hat{\mathbf{x}}(i|j)$, and its covariance, $\mathbf{P}(i|j)$, of each node is communicated in its information form which is given by the information vector, $\hat{\mathbf{y}}(i|j)$, and information matrix, $\mathbf{Y}(i|j)$. The relation between the state and covariance and the information vector and matrix is given as:

$$\hat{\mathbf{y}}(i|j) = \mathbf{P}^{-1}(i|j)\hat{\mathbf{x}}(i|j) \tag{14}$$

$$\mathbf{Y}(i|j) = \mathbf{P}^{-1}(i|j) \tag{15}$$

In integration of the information, all the update operations are additive. As a result, the specific order in which updates from other vehicles are applied is irrelevant to the results, provided the features do not change over time. This observation is of central importance in multi-vehicle SLAM, as the map features are specially selected to be stationary. If map update messages from other vehicles arrive under arbitrary latency, they can still simply be added on in the information form regardless of their "age".

B. The Local Filter

The local information filter, also known as the nodal filter, generates information state estimates on the basis of observed, predicted and communicated information. Other infrastructure such as the channel filter and channel manager exist only to support the correct implementation of the local filter.

The local filter takes map information from local SLAM node (if present) and from the channel manager (if connected). The state estimates and covariance estimates of these landmarks are converted into information form to produce an information vector, $\hat{\mathbf{y}}(k|k)$, and information matrix, $\mathbf{Y}(k|k)$. This information vector and information matrix are then output to the channel manager for transmission to neighbouring nodes.

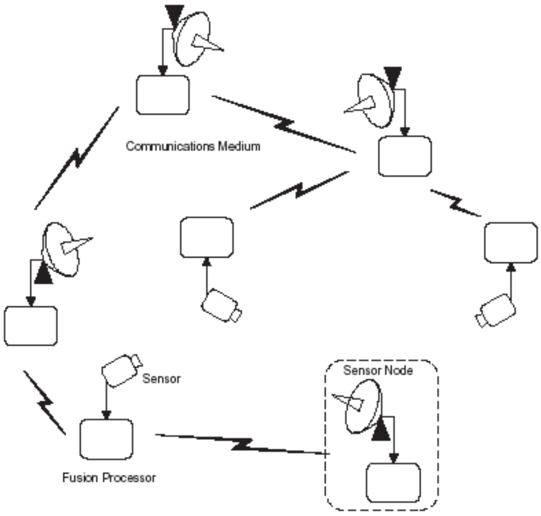


Fig. 4. A decentralised architecture

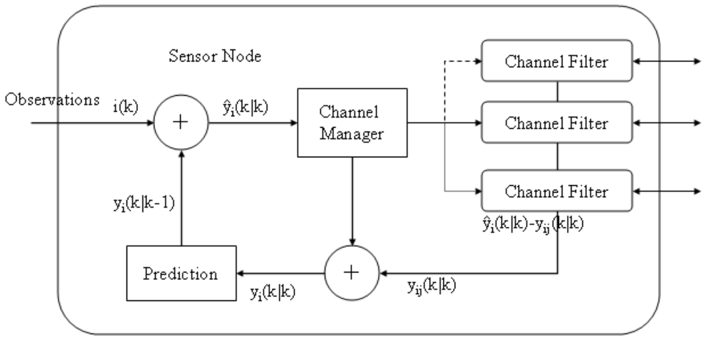


Fig. 5. A decentralised sensor node structre

C. The Channel Manager

The channel manager serves as the interface between the local filter and the channel filters (and through these, the other nodes or platforms in the network). On each platform, a channel filter would be allocated to each remote platform it is connected to.

Incoming data is collected from the channels at a time horizon and assimilated using the additive update equations. The result is communicated to the local filter which updates the SLAM estimate in a single step. Outgoing updated information states from its local filter are also received by the channel manager. This information is disseminated to the channel filters for transmission.

D. The Channel Filter

The channel filter is a conventional information filter which is used for maintaining an estimate of common data passed through a particular channel. A channel filter on node *i* that is connected to node *j* maintains a common information matrix, $\mathbf{Y}(i|j)$, and a common information vector, $\hat{\mathbf{y}}(i|j)$, between the two nodes.

Channel filters have two important characteristics. Firstly, incoming data from remote sensor nodes is assimilated by the local sensor node before being communicated on to subsequent nodes. Therefore, regardless of the number of incoming messages, there is only a single outgoing message to each platform. Secondly, a channel filter compares what has been previously communicated with the total local information at the node. Thus, if the operation of the channel is suspended, the filter simply accumulates information in an additive fashion. When the channel is re-opened, the total accumulated information in the channel is communicated in one single message.

Information previously communicated is used to compute new information gain from other vehicles in the network. The map information from remote vehicles arrives asynchronously at each channel. The channel filter calculates the new information received on any given channel and transmits this to the channel manager, before updating itself. In the event that the channel becomes blocked or disconnected, the channel filter effectively fuses the new data and cycles to the next available communication time.

E. Communication of map information

The channels take the total local information, $\hat{\mathbf{y}}_i(k|k)$, and subtract out all information that has previously been communicated down the channel, $\hat{\mathbf{y}}_{ij}(k|k)$, thus transmitting only new information obtained by node *i* since the last communication. Intuitively, communicated data from node *i* thus consists only of map information not previously transmitted to a node *j*. As common map information has already been removed from the communication, node *j* can simply assimilate incoming information measures by addition.

The vehicle information is never communicated. Hence, the channel filters will never maintain any states other than the map. As the full map of all the features is being transmitted, the channel filter update is a simple update of all the features:

$$\begin{aligned}\mathbf{Y}_{Chan}(k|k) &= \mathbf{Y}^*(k|k) &(16)\\ \hat{\mathbf{y}}_{Chan}(k|k) &= \mathbf{y}^*(k|k) &(17)\end{aligned}$$

This update is only possible as the map information $\mathbf{Y}^*(k|k)$ and $\mathbf{y}^*(k|k)$ are of the dimension of the complete map and all cross information terms are being transmitted.

When the receiving node obtains the new submap information, it updates its own channel filter using exactly the same update steps as above. Once updated, it calculates the increment of new information it has just received from node *i* that has not already been fused locally at node *j*.

$$\begin{aligned}\mathbf{I}^*_{ij}(k|k) &= \mathbf{Y}_{Chan}(k|k) - \mathbf{Y}_{Chan}(k|k-1) &(18)\\ \mathbf{i}^*_{ij}(k|k) &= \hat{\mathbf{y}}_{Chan}(k|k) - \hat{\mathbf{y}}_{Chan}(k|k-1) &(19)\end{aligned}$$

This information increment is then sent to the local filter to be fused into the SLAM estimate.

F. Fusing Information from other nodes

When the local filter receives the information $\mathbf{I}^*_{ij}(k|k)$ and $\mathbf{i}^*_{ij}(k|k)$ from the channel filter, this information is used in the SLAM estimate. In order to do this, it is necessary to firstly define a matrix $\mathbf{G}_s$ that inflates the map to the dimension of the entire SLAM state by padding vehicle elements with zeros. The update is then done by adding the new information from the other node as:

$$\begin{aligned}\hat{\mathbf{y}}(k|k) &= \hat{\mathbf{y}}(k|k-1) + \mathbf{G}_s\hat{\mathbf{y}}_{ij}(k|k) &(20)\\ \mathbf{Y}(k|k) &= \mathbf{Y}(k|k-1) + \mathbf{G}_s\mathbf{Y}_{ij}(k|k)\mathbf{G}_s^T &(21)\end{aligned}$$

It is worth noting that this update step is identical to updating with information from locally attached sensors.

G. Data Association

In decentralised SLAM, map information about the same landmark could come from different sources. Data association is necessary to correctly match this information. When an observation is made, it is necessary to determine if the landmark is same as one that has already been seen. Also, in a decentralised system, it is necessary to associate information from other nodes with that stored locally.

Figure 6 illustrates this notion where two nodes are estimating the same landmark set, but the landmarks are ordered differently on each node. When node 1 communicates information about its landmark-1, node 2 must correctly associate it with its own landmark-4.

The information gate [21] can be used for data association with the information filter. Shown in equation below, it is the information equivalent of the state space innovation gate. The primary advantage of this algorithm is that it is cast in terms of the information states.

$$\mathbf{v}^T(k)\mathbf{B}(k)^+\mathbf{v}^T(k) < \lambda_n, \tag{22}$$

where n is the dimension of the innovation, the observation and the landmark that were used to form the innovation are then associated.

$$\begin{aligned}\mathbf{v}(k) &= \mathbf{I}(k)[\mathbf{I}(k)^+\mathbf{i}(k) - \mathbf{Y}^{-1}(k \mid k-1)\hat{\mathbf{y}}(k|k-1)] &(23)\\ \mathbf{B}(k) &= \mathbf{I}(k)[\mathbf{I}(k)\mathbf{Y}^{-1}(k \mid k-1)]\mathbf{I}(k) &(24)\\ \mathbf{B}^+(k) &= \mathbf{H}^T(k)[\mathbf{H}(k)\mathbf{B}(k)\mathbf{H}^T(k)]\mathbf{H}(k) &(25)\\ \mathbf{I}^+(k) &= \mathbf{H}^T(k)[\mathbf{H}(k)\mathbf{I}(k)\mathbf{H}^T(k)]\mathbf{H}(k) &(26)\end{aligned}$$

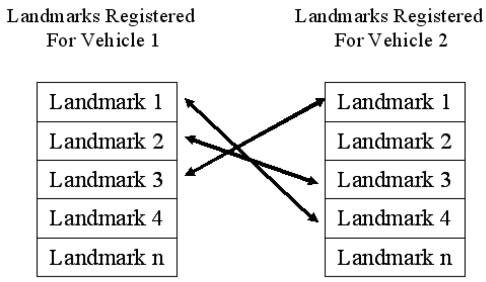


Fig. 6. Different nodes may have the same physical landmarks stored in different orders.

The nodes would also include a data association index with each information communication. The data association index is the location of the landmark at the transmitting node. When received for the first time, the landmarks will pass through the data association algorithm to determine if they match any landmarks at the receiving node. Once the receiving node knows the index of that landmark locally, it can store the relationship between the landmarks on different nodes. In this way, a look up table is generated once landmarks are identified rather than having to apply a computationally expensive association algorithm at each update iteration.

IV. The Physical System

The physical system of the UAVs, on which decentralised SLAM will be demonstrated, comprises of the flight platform, payload sensors and mission sensors.

Fig. 7. Two Brumby Mk3 aircrafts

The flight platform shown in Fig. 7 is a delta fixed wing platform with a pusher prop configuration and is capable of flying at 50m/s and up to 500m. The payload sensors are part of the flight platform. The platform can carry up to 11kg of additional mission sensors.

The payload sensors comprise of an INS/GPS/Baro navigation system. This system provides the position, velocity and attitude solutions at 100Hz for the flight control system. The IMU developed by Inertial Science is installed in the fuselage as shown in Fig. 8. The GPS unit is a 12 channel AllStar receiver. Data from these sensors have a rated accuracy of 0.5m in position, 0.1m/s in velocity and 0.5° in attitude. The data output is at 100Hz.

The mission sensor used in this paper is a vision payload sensor connected to a PC104 computer. Range is estimated from the size of the landmarks and hence poor range information is obtained. The vision sensor is a passive and low cost sensor with good bearing accuracy. The frame rate can be up to 50Hz and the images are captured in monochrome. The landmarks placed on the ground are $2{\times}2$m and are painted white for ease of detection.

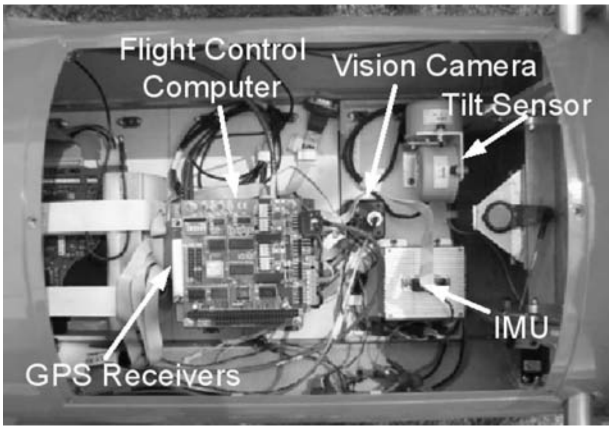


Fig. 8. The IMU and tilt sensors are installed for the PC-104 flight control computer (FCC). A low cost camera is installed next to the IMU to minimise sensor offset. Two GPS receivers are stacked on the FCC

V. Experimental Results

A. Simulation Environment

The simulation parameters and assumptions are based on a real UAV implementation and are shown in Table I. In the simulation, UAV undergoes a figure-of-eight trajectory approximately 100m above the ground with average flight speeds of 40m/s. A low-cost (or equivalently low quality) IMU is simulated with a passive vision sensor. The IMU implemented in the real system has a gyro bias of 0.01°/s and an accelerometer bias of 0.01m/s^2 which can be rated as a low-cost tactical grade inertial unit. The biases are calibrated precisely using onboard inclinometers in the real implementation, hence the biases are not explicitly modelled and studied in this simulation analysis and only white noise is modelled. The vision sensor is a passive and low-cost sensor. It has good bearing accuracy but it can only provide poor range data if the size of landmark is known otherwise it cannot provide range information at all. In this paper vision with poor range quality is incorporated.

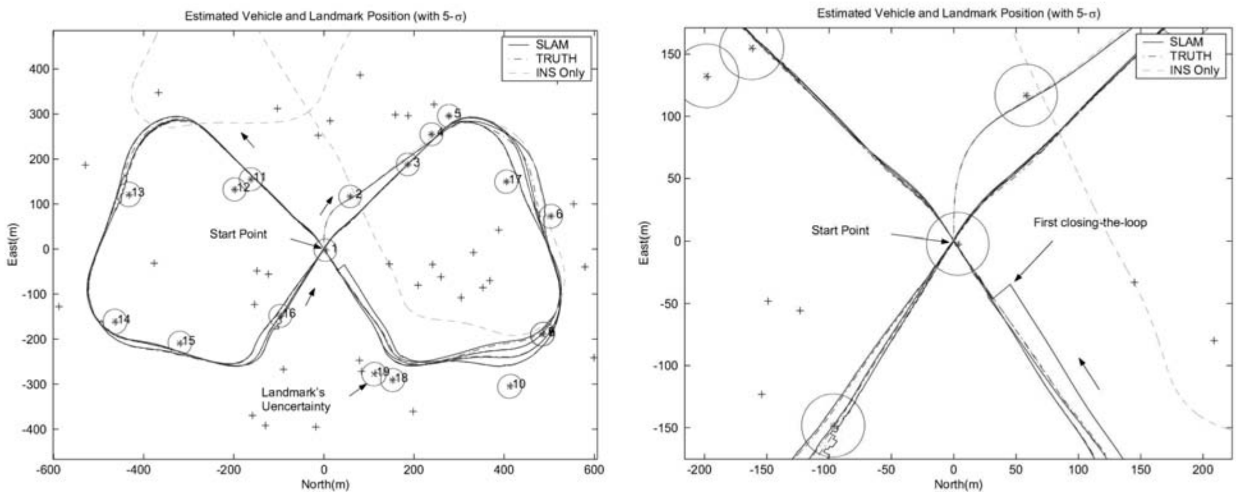
TABLE I
The parameters used in simulation.

Sensor	Type	Spec.
	Sampling rate	400 Hz
IMU	Accel noise	0.1m/s$^2/\sqrt{Hz}$
	Gyro noise	0.1°/s/$\sqrt{Hz}$
	Frame rate	25Hz
Vision	FOV angle	$\pm$15°
	Bearing noise strength	0.1604°
	Elevation noise strength	0.1206°
	Range noise strength	$\geq$20m
Alignment	Horizontal axis	0.5°
error	Vertical axis	0.0°

B. Single-vehicle SLAM results

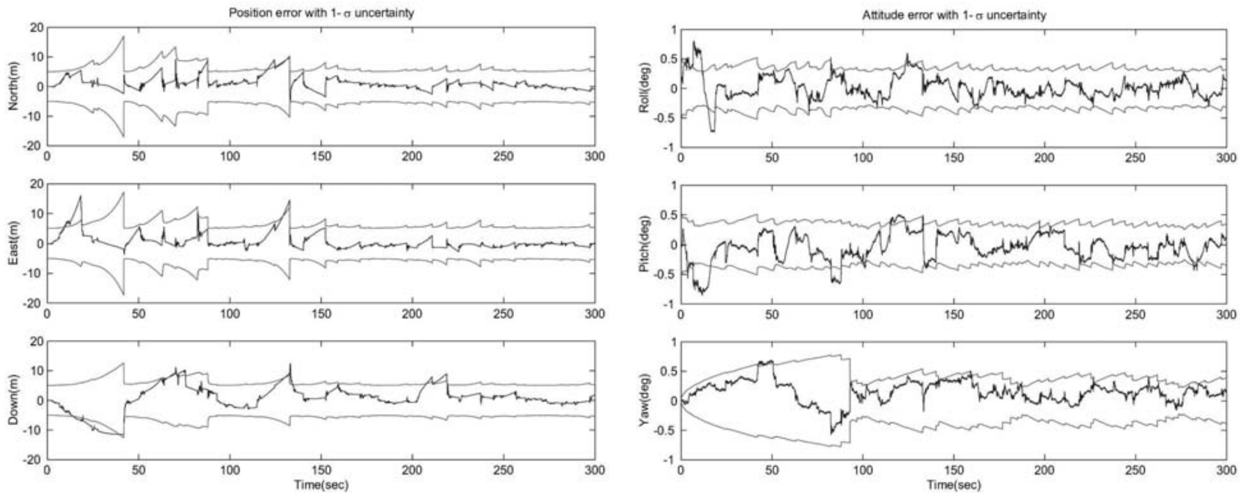
The vision sensor detects landmarks below the flight paths and registers 19 landmarks from the total 50 landmarks on the ground. The landmark extraction rate is 25Hz which is typical in most monochrome cameras.

The SLAM result using a vision sensor is shown in Figs 9 to 11. The first round is mainly an exploration stage since new landmarks are detected and augmented into the state. In the second round the map is improved dramatically because of the revisiting process and the vehicle error begins to be effectively constrained. Figure 9a shows the estimated flight path and landmark positions with 5σ uncertainty bounds for clarification. The revisiting effect can be seen in Fig. 9b with a correction of approximately 20m. During the second revisiting process the corrections were much smaller than that of the first visit since less information is added to the map. Figure 10 depict the estimation errors in position, and attitude with the 1σ uncertainties (the velocity is similar to the position, hence it is not plotted). The plot shows that the navigation errors are estimated and maintained within 1σ bound from the relative vision observations. Furthermore, the covariance of the heading state is always increasing in the first round, and only after the second round during the revisiting process does the heading error improve. This is primarily due to the change in observability



(a) (b)

Fig. 9. (a) Vehicle and map 2D position estimated from SLAM filter. The UAV starts from (0,0) and flies following a figure eight shape to maximise the chance of reobservation. (b) Enhanced view of the vehicle and 2D map position at revisiting. The accumulated vehicle errors are corrected approximately 20m at the first revisit.



(a) (b)

Fig. 10. (a) SLAM Position errors and (b) attitude errors with 1σ uncertainties.

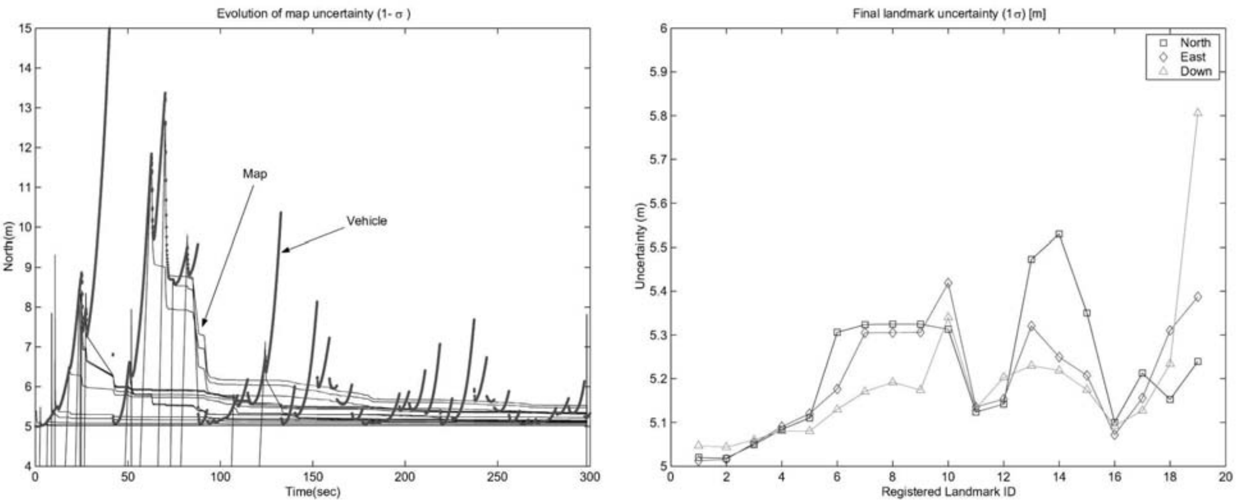
because of the addition of landmarks into the state vector. Figure 11 illustrates the evolution of the vehicle and map uncertainty along the north axis. During the first round, most of the initialised landmarks have large uncertainties because of the embedded vehicle uncertainty. After the second round, the map accuracy improves dramatically and begins to constrain the error drift of the vehicle effectively whenever the vehicle detects the landmarks. The final map accuracy with a worst case landmark uncertainty of approximately 5.8m. The initial vehicle uncertainty was 5m which is the lowest limit in map accuracy that SLAM could ideally achieve [15].

C. Decentralised SLAM results

In decentralised SLAM simulation, two UAVs fly two separate segments of the figure-of-eight trajectory, communicating map information. UAV-1 starts from the origin and flies over the right part of the figure-of-eight trajectory, while UAV-2 starts from the right lower part (around landmark-3 in Fig. 12a) and flies over the left part of trajectory. They share some common landmarks around the origin. The DDF update occurs every 2sec. Two different communication strategies were compared which were:

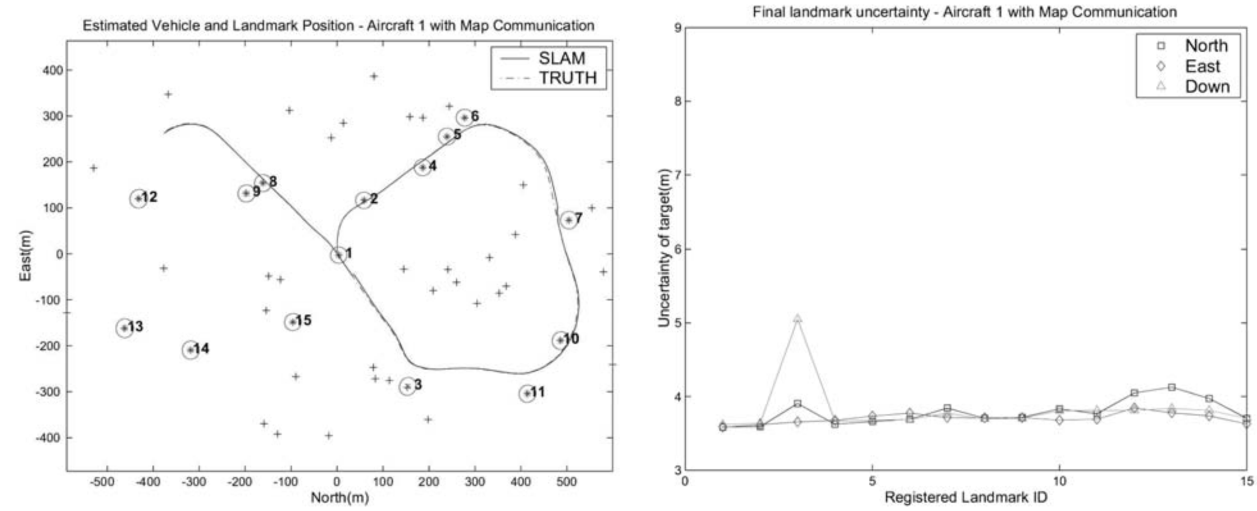
- No DDF communication between the nodes where each vehicle operates independently using only the observations it detects.
- Full DDF communication of all the observations detected by each vehicle to the other nodes.

Figures 12 and 13 show the DDF SLAM results in UAV-1 and UAV-2. Although two vehicles fly different regions, they build a common and global map from to the DDF communication. Hence the final landmark uncertainties in two vehicles are identical. Figures 14 and 15 show the standalone SLAM results without communication for comparison. As predicted, each map has a



(a) (b)

Fig. 11. (a) Evolution of uncertainties of the vehicle and landmarks on north position. During the first round, the vehicle registered new landmarks. After the second round however, the vehicle begins to re-observe the former landmarks and the map accuracy begins to improve, which in turn suppresses the vehicle errors. (b) Final map building performance achieved from SLAM navigation.

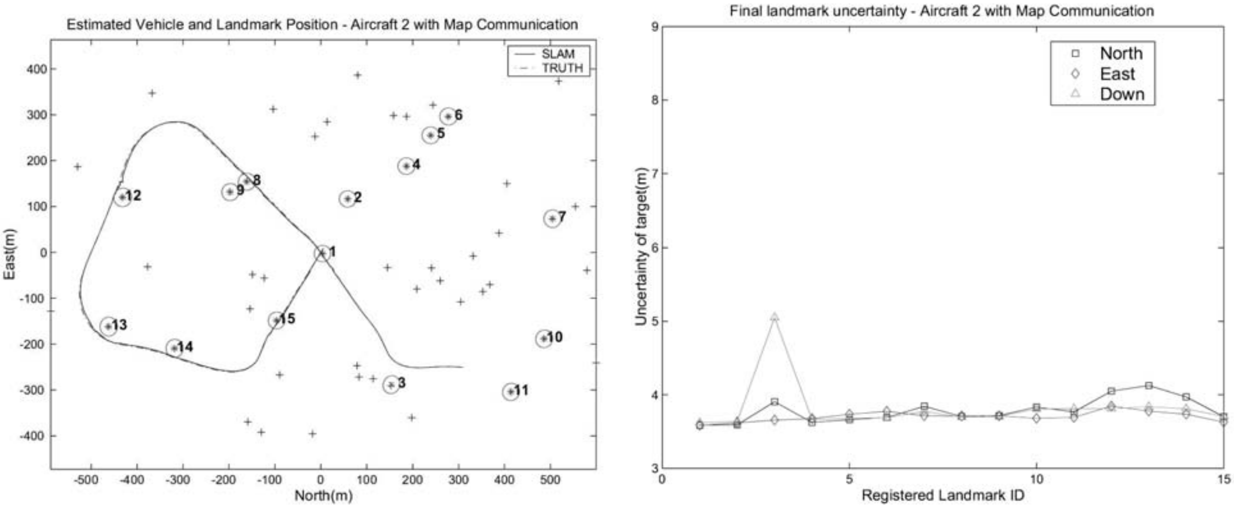


(a) (b)

Fig. 12. (a) Decentralised SLAM result in UAV-1 with the estimated vehicle and landmark position. UAV-1 flies over the right part of the figure-of-eight and UAV-2 flies over the left part of it communicating map information (5σ uncertainty ellipses of map are used). (b) Final landmark uncertainty from decentralised SLAM (the large uncertainty along the down axis in landmark-3 results from the high banking of the vehicle).

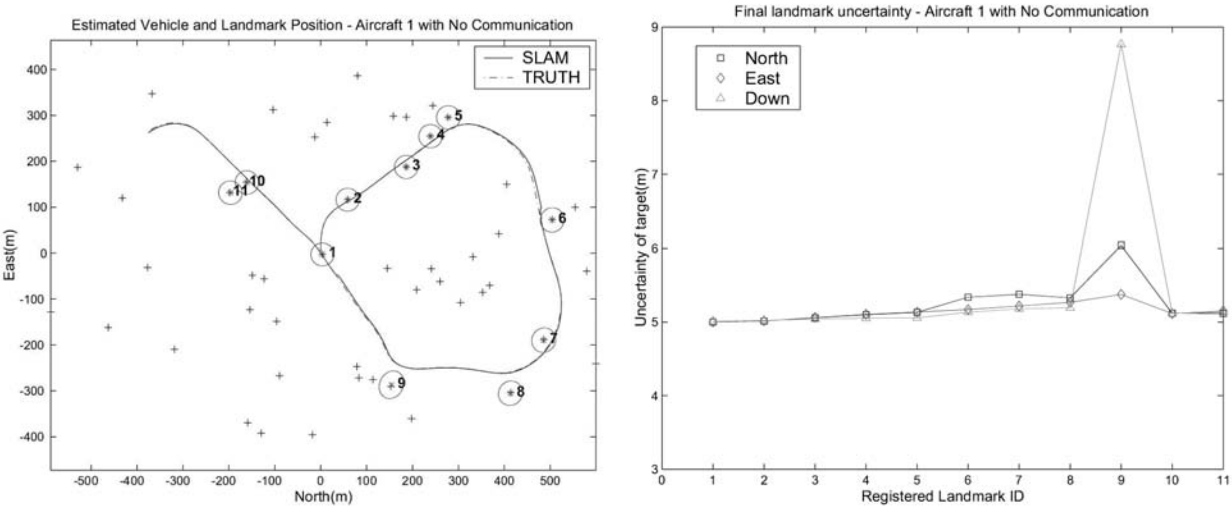
reduced number of landmarks that the vehicle detected. It should be noted that the associated landmark number is different between DDF and standalone mode map due to the separated data association process, hence they can not be compared directly from the plots. However it can be observed that the overall landmark uncertainties in DDF mode are significantly lower than the uncertainty in the standalone operation. This is because both vehicles observe some common landmarks and fuses them across information network. The large uncertainty in landmark-3 in DDF map results from the high banking of the vehicle at observation time and corresponds to landmark-9 in Fig. 14 and landmark-1 Fig. 15 respectively.

Figures 16 and 17 compare the vehicle's position uncertainty along the north axis with and without DDF communication. It can be seen clearly that DDF map communication can enhance the vehicle's position accuracy as well. This is due to the correlation structure between the vehicle and map. Once the map accuracy is improved from the DDF communication, the vehicle accuracy can also be improved via the correlation. This can also be observed in the vehicle's attitude plot (only the heading is presented in this plot) as in Fig. 17. The UAV-1 shows more rapid decrease in heading uncertainty than UAV-2, and it is due to the different trajectories and dynamics in each vehicle. The UAV-1 undergoes high banking at the start time while UAV-2 undergoes relatively straight line and this enhances the heading observability on UAV-1. To compare the landmark improvement in DDF mode, three dimensional uncertainty ellipsoids of landmark-1 are shown in Fig 18. This corresponds to landmark-1 in UAV-1 without communication and landmark-2 in UAV-2. From this plot it is clear that the DDF communication enhances the map accuracy.



(a) (b)

Fig. 13. (a) Decentralised SLAM result in UAV-2 with the estimated vehicle and landmark position. Due to DDF map communication, both UAVs build a common combined map. (b) Final landmark uncertainty which is identical to that of UAV-1.



(a) (b)

Fig. 14. (a) Standalone SLAM result in UAV-1 without DDF communication for comparison and (b) the final landmark uncertainty in UAV-1. Only a limited map is built with 11 landmarks and larger uncertainties than DDF map (The landmark-9 with large uncertainty corresponds to the landmark-3 in DDF map).

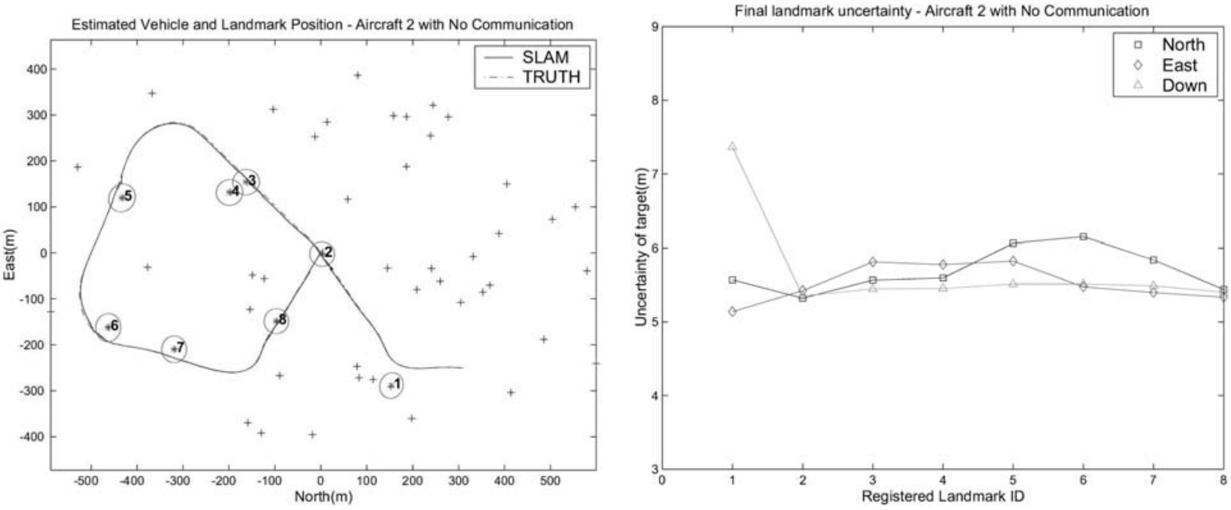
These results illustrate that navigation errors of two highly non-linear 6DoF platforms can be effectively constrained using SLAM algorithm, and its performance can be enhanced using the decentralised information fusion approach.

VI. Conclusion and Future Work

This paper has presented the SLAM and decentralised SLAM algorithms applied to a 6DoF airborne platform. The simulation analysis illustrates that the SLAM navigation system with vision sensor can perform navigation in unknown 3D terrain environments. The results also show that sharing map information reduces the navigation errors on the whole vehicles. However, should one node have higher navigation errors than the other, the errors in the first node will be reduced while there would be a slight increase in error in the second node. The uncertainty of the landmarks are reduced in the DDF communication strategy compared to the case without communication. One of the issues in multiple airborne vehicle is the limited bandwidth. Hence, only part of the map could be communicated. Strategies to maximise the amount of information to transmit will have to be investigated and applied.

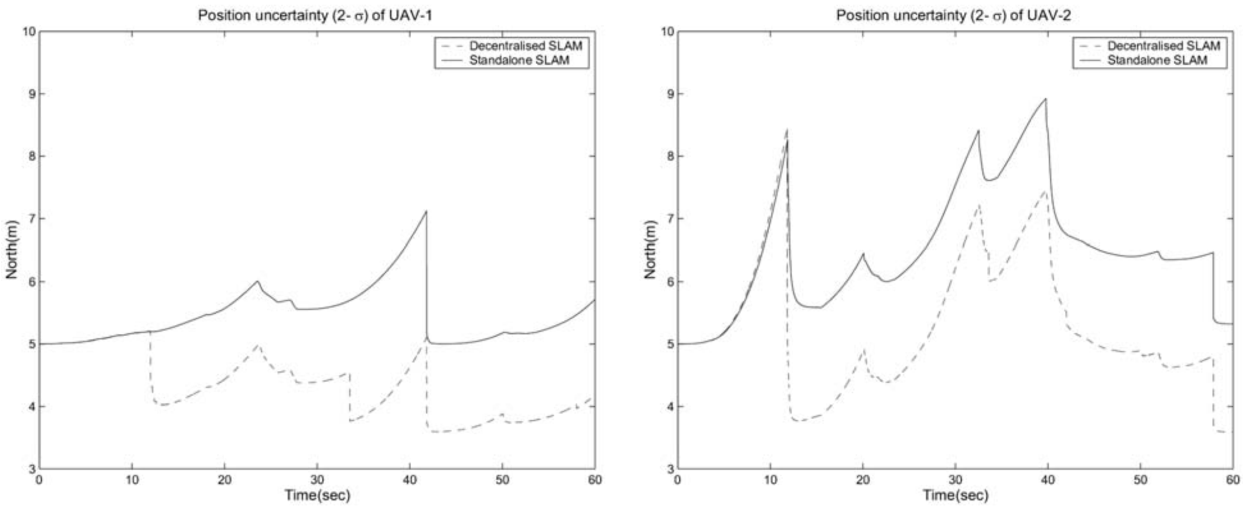
Acknowledgements

This work is supported in part by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government. The Autonomous Navigation and Sensing Experimental Research (ANSER) Project is funded by BAE SYSTEMS UK and BAE SYSTEMS Australia.



(a) (b)

Fig. 15. (a) Standalone SLAM result in UAV-2 without DDF communication for comparison and (b) the final landmark uncertainty in UAV-2. The resulting map contains only 8 landmarks with larger uncertainties (the landmark-1 with large uncertainty corresponds to the landmark-3 in DDF map).



(a) (b)

Fig. 16. Comparison of the position uncertainties along the north axis in (a) UAV-1 and (b) UAV-2 with and without DDF communications. The enhancement of map in DDF mode causes both vehicle's positions to be also improved.

REFERENCES

[1] R.L. Greenspan. GPS and Inertial Navigation. In *American Institude of Aeronautics and Astronautics*, 1996.

[2] R.E. Phillips and G.T. Schmidt. GPS/INS Integration. In *Advisory Group for Aerospace Research and Development (AGARD)*, Lecture series 207:9.1–9.10, 1996.

[3] S. Sukkarieh, E.M. Nebot, and H. Durrant-Whyte. A High Integrity IMU/GPS Navigation Loop for Autonomous Land Vehicle Applications. In *IEEE Transactions on Automatic Control*, 15:572–578, June 1999.

[4] J.H. Kim and S. Sukkarieh. Flight Test Results of a GPS/INS Navigation Loop for an Autonomous Unmanned Aerial Vehicle (UAV). In *Proceedings of the 15th International Technical Meeting of the Satellite Division of the Institute of Navigation*, pages 510–517, September, OR, USA, 2002.

[5] W.R. Baker and R.W. Clem. *Terrain contour matching (TERCOM) premier*. ASP-TR-77-61. Aeronautical Systems Division, Wright-Patterson, August 1977.

[6] I.Y. Bar-Itzhack. Optimal updating of INS using landmarks. In *Americian Institute of Aeronautics and Aestronautics*, 492–502, August 1977.

[7] J. Dezert. Improvement of strapdown inertial Navigation using PDAF. In *IEEE Transactions on Aerospace and electronic systems*, 35(3):572–578 July 1999.

[8] J.E. Pritchett and A.J Pue. Robust Guidance and Navigation for Airborne Vehicle using GPS/Terrain Aiding. In *IEEE Position Location and Navigation Symposium*, pages 457–463, 2000.

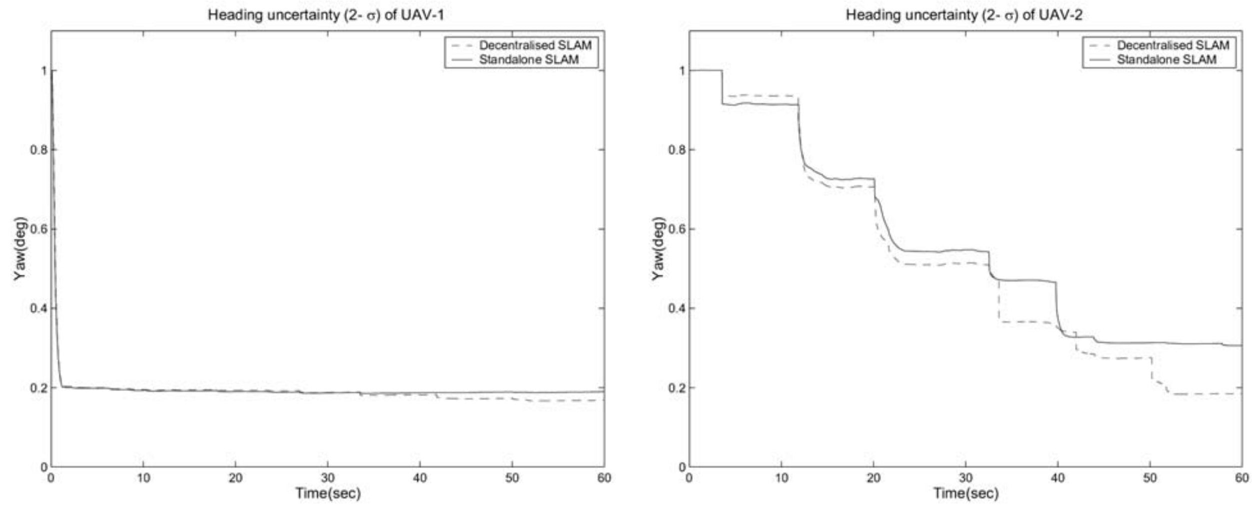
[9] R.O. Duda and P.E. Hart. *Pattern Classication and Scene Analysis*. John Wiley and Sons, 1973.

[10] R. Smith and P. Cheeseman. On the Representation of Spatial Uncertainty. In *International Journal of Robotics Research*, 5(4):56–68, 1987.

[11] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous Map Building and Localisation for an Autonomous Mobile Robot. In *IEEE International Conference on Intelligent Robot Systems*, 1991.

[12] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 53:31–29, 1998.

[13] J.J. Leonard and H.J. Feder. Experimental analysis of adaptive concurrent mapping and localisation using sonar. In *International Symposium on Experimental Robotics*, 1999.



(a) (b)

Fig. 17. Comparison of the attitude uncertainties along the heading axis in (a) UAV-1 and (b) UAV-2 with and without DDF communications. It can be seen that in DDF mode, both vehicle's attitudes are also improved (the rapid decrease of heading uncertainty in UAV-1 is due to the high maneuvers at start time which enhance the heading observability on UAV-1, while UAV-2 undergoes relatively low maneuvers at start time).

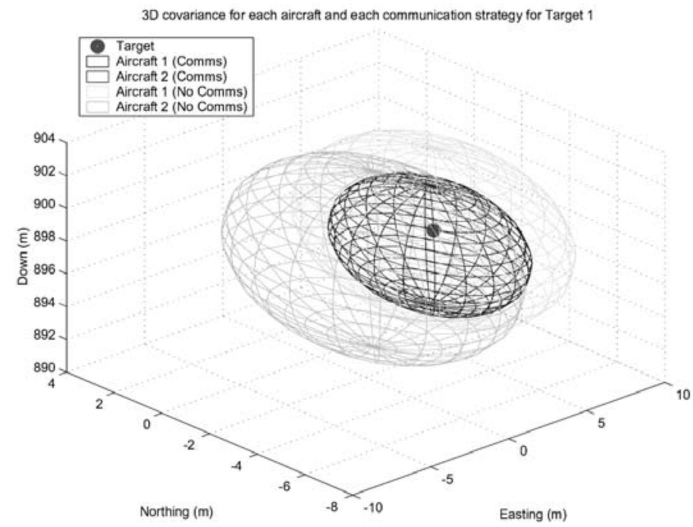


Fig. 18. 3D uncertainty ellipsoids for a landmark-1 with and without DDF communication. The DDF map (ellipsoid in center) clearly shows enhanced performance than standalone maps.

[14] S.B. Williams, G. Dissanayake, and H. Durrant-Whyte. Towards terrain-aided navigation for underwater robotics. In *Advanced Robotics*, 15(5):533–550, 2001.

[15] M.W.M.G. Dissanayake, P. Newman, H. Durrant-Whyte, S. Clark, and M. Csorba. A solution to the simultaneous localization and map building problem. In *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001.

[16] J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real-time implementation. In *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001.

[17] E. Nettleton, S. Thrun, H. Durrant-Whyte and S. Sukkarieh. Decentralised SLAM with Low-Bandwidth Communication for Teams of Vehicles *4th International Conference on Field and Service Robotics (FSR '03)*, Japan, July 2003

[18] T. Kirubarajan, Y. Bar-Shalom, and X.R.Li. *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, 2001.

[19] S. Grime and H. Durrant-Whyte. Data fusion in decentralized sensor networks. *Control Engineering Practice*,2, No 5:849863,1994

[20] E. Nettleton, H. Durrant-Whyte and S. Sukkarieh. A robust architecture for decentralised data fusion. In *Proceedings of IEEE International Conference on Advanced Robotics*, 2003

[21] M. Fernandez. Failure Detection and Isolation in Decentralised Multisensor Systems. Ph.D Thesis, The University of Oxford, 1993.

APPENDIX A

The Jacobian $\nabla \mathbf{f}_x(k)$ of the nonlinear vehicle model with respect to the vehicle state in equation 4 can be computed as,

$$\nabla \mathbf{f}_x(k) = \begin{bmatrix} \frac{\partial \mathbf{P}(k)}{\partial(\mathbf{P}(k-1),\mathbf{V}(k-1),\mathbf{\Psi}(k-1))} \\ \frac{\partial \mathbf{V}(k)}{\partial(\mathbf{P}(k-1),\mathbf{V}(k-1),\mathbf{\Psi}(k-1))} \\ \frac{\partial \mathbf{\Phi}(k)}{\partial(\mathbf{P}(k-1),\mathbf{V}(k-1),\mathbf{\Psi}(k-1))} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \triangle t\mathbf{I} & 0 \\ 0 & \mathbf{I} & \frac{\partial(\mathbf{C}_b^n(k)\mathbf{f}^b(k))}{\partial \mathbf{\Psi}(k-1)}\triangle t \\ 0 & 0 & \mathbf{I} + \frac{\partial(\mathbf{E}_b^n(k)\boldsymbol{\omega}^b(k))}{\partial \mathbf{\Psi}(k-1)}\triangle t \end{bmatrix}, \tag{27}$$

where the sub-matrix can also be computed by using Jacobians of $\mathbf{C}_b^n$ and $\mathbf{E}_b^n$ with respect to the Euler angles (ρ, ψ, θ). They are three element matrices and are computed as,

$$\frac{\partial \mathbf{C}_b^n}{\partial(\rho,\psi,\theta)} \triangleq \left[\frac{\partial \mathbf{C}_b^n}{\partial \phi}, \frac{\partial \mathbf{C}_b^n}{\partial \theta}, \frac{\partial \mathbf{C}_b^n}{\partial \psi}\right], \qquad \frac{\partial \mathbf{E}_b^n}{\partial(\rho,\psi,\theta)} \triangleq \left[\frac{\partial \mathbf{E}_b^n}{\partial \phi}, \frac{\partial \mathbf{E}_b^n}{\partial \theta}, \frac{\partial \mathbf{E}_b^n}{\partial \psi}\right] \tag{28}$$

where,

$$\begin{aligned}
\frac{\partial \mathbf{C}_b^n}{\partial \phi} &= \begin{bmatrix} 0 & S_\phi S_\psi + C_\phi S_\theta C_\psi & C_\phi S_\psi - S_\phi S_\theta C_\psi \\ 0 & -S_\phi C_\psi + C_\phi S_\theta C_\psi & -C_\phi C_\psi - S_\phi S_\theta S_\psi \\ 0 & C_\phi C_\theta & -S_\phi C_\theta \end{bmatrix} \\
\frac{\partial \mathbf{C}_b^n}{\partial \theta} &= \begin{bmatrix} -S_\theta C_\psi & S_\phi C_\theta C_\psi & C_\phi C_\theta C_\psi \\ -S_\theta S_\psi & S_\phi C_\theta C_\psi & C_\phi C_\theta S_\psi \\ -C_\theta & -S_\phi S_\theta & -C_\phi S_\theta \end{bmatrix} \\
\frac{\partial \mathbf{C}_b^n}{\partial \psi} &= \begin{bmatrix} -C_\theta S_\psi & -C_\phi C_\psi - S_\phi S_\theta S_\psi & S_\phi C_\psi - C_\phi S_\theta S_\psi \\ C_\theta C_\psi & -C_\phi S_\psi - S_\phi S_\theta S_\psi & S_\phi S_\psi + C_\phi S_\theta C_\psi \\ 0 & 0 & 0 \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathbf{E}_b^n}{\partial \phi} &= \begin{bmatrix} 0 & C_\phi S_\theta / C_\theta & -S_\phi S_\theta / C_\theta \\ 0 & -S_\phi & -C_\phi \\ 0 & C_\phi / C_\theta & -S_\phi / C_\theta \end{bmatrix} \\
\frac{\partial \mathbf{E}_b^n}{\partial \theta} &= \begin{bmatrix} 0 & S_\phi / C_\theta^2 & C_\phi / C_\theta^2 \\ 0 & 0 & 0 \\ 0 & S_\phi S_\theta / C_\theta^2 & -C_\phi S_\theta / C_\theta^2 \end{bmatrix} \\
\frac{\partial \mathbf{E}_b^n}{\partial \psi} &= \mathbf{0}
\end{aligned}$$

The Jacobian $\nabla \mathbf{f}_w(k)$ of the nonlinear vehicle model with respect to the input noise (or inertial sensor noise) in equation 4 is obtained by,

$$\nabla \mathbf{f}_w(k) = \begin{bmatrix} \frac{\partial \mathbf{P}(k)}{\partial(\mathbf{f}^b(k), \boldsymbol{\omega}^b(k))} \\ \frac{\partial \mathbf{V}(k)}{\partial(\mathbf{f}^b(k), \boldsymbol{\omega}^b(k))} \\ \frac{\partial \boldsymbol{\Psi}(k)}{\partial(\mathbf{f}^b(k), \boldsymbol{\omega}^b(k))} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \mathbf{C}_b^n(k-1) & 0 \\ 0 & \mathbf{E}_b^n(k-1) \end{bmatrix}. \tag{29}$$

APPENDIX B

The observation Jacobian $\nabla \mathbf{h}_x(k)$ of equation 5 with respect to the vehicle and map state can be derived from the nonlinear observation equations 9 and 10 by applying a chain rule,

$$\begin{aligned}
\nabla \mathbf{h}_x(k) &= \begin{bmatrix} \frac{\partial \rho(k)}{\partial(\mathbf{P}(k-1), \mathbf{V}(k-1), \boldsymbol{\Psi}(k-1), \mathbf{x}_{mi}(k-1))} \\ \frac{\partial \varphi(k)}{\partial(\mathbf{P}(k-1), \mathbf{V}(k-1), \boldsymbol{\Psi}(k-1), \mathbf{x}_{mi}(k-1))} \\ \frac{\partial \vartheta(k)}{\partial(\mathbf{P}(k-1), \mathbf{V}(k-1), \boldsymbol{\Psi}(k-1), \mathbf{x}_{mi}(k-1))} \end{bmatrix} &\quad (30) \\
&= \nabla \mathbf{h}_1(k) \nabla \mathbf{h}_2(k) &\quad (31)
\end{aligned}$$

where $\mathbf{h}_1(k)$ is defined as equation 9 and $\mathbf{h}_2(k)$ is defined as equation 10. The Jacobians of these functions are obtained as,

$$\begin{aligned}
\nabla \mathbf{h}_1(k) &= \begin{bmatrix} \frac{x}{\rho} & \frac{y}{\rho} & \frac{z}{\rho} \\ \frac{-y}{\rho(x^2+y^2)} & \frac{x}{\rho(x^2+y^2)} & 0 \\ \frac{-xz}{\rho^2\sqrt{x^2+y^2}} & \frac{-yz}{\rho^2\sqrt{x^2+y^2}} & \frac{(x^2+y^2)}{\rho^2} \end{bmatrix} \\
\nabla \mathbf{h}_2(k) &= \begin{bmatrix} -\mathbf{C}_b^s \mathbf{C}_n^b & \mathbf{0} & \mathbf{C}_b^s \frac{\partial \mathbf{C}_n^b(\mathbf{x}_{mi} - \mathbf{P})}{\partial \boldsymbol{\Psi}(k-1)} & \vdots & \mathbf{C}_b^s \mathbf{C}_n^b \end{bmatrix}
\end{aligned}$$

where ρ, φ and ϑ are range, bearing and elevation respectively and x, y and z are computed from $\mathbf{h}_2(k)$.

APPENDIX C

The Jacobian $\nabla \mathbf{g}_x(k)$ and $\nabla \mathbf{g}_z(k)$ used in equation 12 can be obtained from the new landmark initialisation function in equation 7,

$$\begin{aligned} \nabla \mathbf{g}_x(k) &= \begin{bmatrix} \mathbf{I} & \mathbf{0} & \frac{\partial \mathbf{C}_b^n(\mathbf{P}_{sb}^b + \mathbf{C}_s^b \mathbf{P}_{ms}^s)}{\partial \mathbf{\Psi}} \end{bmatrix} \\ \nabla \mathbf{g}_z(k) &= \mathbf{C}_b^n \mathbf{C}_s^b \frac{\partial \mathbf{P}_{ms}^s}{\partial \mathbf{z}_i} \end{aligned} \tag{32}$$

APPENDIX D

The information observation quantities $\mathbf{i}(k)$ and $\mathbf{I}(k)$ are of dimension the state space, whereas the innovation and innovation variance are of dimension the observation space. In the innovation gate, the inverse innovation covariance is used to normalise the gate. In the information gate, the inverse of the corresponding information matrix is required. This, however will generally be singular as it is of dimension the state but has rank of only observation dimension. A generalised inverse $\mathbf{I}_+(k)$ is therefore defined in the following manner.

$$\mathbf{I}(k)\mathbf{I}^+(k) = \mathbf{E} \tag{33}$$

where $\mathbf{E}$ is an idempotent matrix which acts as the identity for both $\mathbf{I}(k)$ and $\mathbf{I}^+(k)$

$$\mathbf{I}(k)\mathbf{E} = \mathbf{I}(k) \tag{34}$$

$$\mathbf{I}^+(k)\mathbf{E} = \mathbf{I}^+(k) \tag{35}$$

and

$$\mathbf{I}(k)\mathbf{I}^+(k)\mathbf{I}(k) = \mathbf{I}(k) \tag{36}$$

$$\mathbf{I}^+(k)\mathbf{I}(k)\mathbf{I}^+(k) = \mathbf{I}^+(k) \tag{37}$$

One generalised inverse which satisfies these requirements is calculated by exploiting the observation model $\mathbf{H}(k)$ as a projection operator. This matrix projects a state space into an observation space and conversely its transpose projects observations back to state space.

The generalised inverse is then

$$\mathbf{I}^+(k) = \mathbf{H}_k^T[\mathbf{H}_k\mathbf{I}(k)\mathbf{H}_k^T]^{-1}\mathbf{H}_k \tag{38}$$

The appropriateness of this selection of projection matrix is apparent as

$$\mathbf{H}_k\mathbf{I}(k)\mathbf{I}^+(k) = \mathbf{H}_k\mathbf{I}(k)\mathbf{H}_k^T[\mathbf{H}_k\mathbf{I}(k)\mathbf{H}_k^T]^{-1}\mathbf{H}_k \tag{39}$$

The innovation which is used for data association is defined as the difference between the observed and predicted observation and given in Equation 40 below:

$$\nu(k) = \mathbf{z}(k)\mathbf{H}_k\hat{\mathbf{x}}(k|k-1) \tag{40}$$

The information residual vector is:

$$\mathbf{v}(k) \triangleq \mathbf{H}_k^T\mathbf{R}_k^{-1}\nu(k) \tag{41}$$

Substituting Equation 40 into Equation 41 gives

$$\begin{aligned} \mathbf{v}(k) &= \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{z}(k) - \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\hat{\mathbf{x}}(k|k-1) \\ &= \mathbf{i}(k) - \mathbf{I}(k)\mathbf{Y}^{-1}(k \mid k-1)\hat{\mathbf{y}}(k|k-1) \end{aligned} \tag{42}$$

The covariance of this information residual is now calculated from $\mathrm{E}\{\mathbf{v}(k), \mathbf{v}^T(k)|\mathbf{Z}_{k-1}(k)\}$

$$\begin{aligned} \mathbf{B}(k) &= \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathrm{E}\{\mathbf{v}(k), \mathbf{v}^T(k)|\mathbf{Z}_k(k)\}\ \mathbf{R}_k^{-1}\mathbf{H}_k^T \\ &= \mathbf{H}_k^T\mathbf{R}_k^{-1}[\mathbf{H}_k^T\mathbf{P}(k \mid k-1)\mathbf{H}_k + \mathbf{R}_k]\mathbf{R}_k^{-1}\mathbf{H}_k^T \\ &= \mathbf{I}(k) + \mathbf{I}(k)\mathbf{Y}^{-1}(k \mid k-1)\mathbf{I}(k) \\ &= \mathbf{I}(k)[\mathbf{I}^+(k) + \mathbf{Y}^{-1}(k \mid k-1)]^{-1}\mathbf{I}(k) \end{aligned} \tag{43}$$

The normalised information residual is now given by

$$\mathbf{\Gamma}(k) = \mathbf{v}^T(k)\mathbf{B}^+(k)\mathbf{v}^T(k) \tag{44}$$

where $\mathbf{B}^+(k)$ is the generalised inverse of $\mathbf{B}(k)$ and is again calculated using the projection operation $\mathbf{H}_k$ as

$$\mathbf{B}^+(k) = \mathbf{H}_k^T[\mathbf{H}_k\mathbf{B}(k)\mathbf{H}_k^T]^{-1}\mathbf{H}_k \tag{45}$$

The relationship between the information gate to the state space innovation gate can be illustrated by noting that this equation above can be written as:

$$\begin{aligned}\mathbf{B}^+(k) &= \mathbf{H}_k^T[\mathbf{H}_k\mathbf{B}(k)\mathbf{H}_k^T]^{-1}\mathbf{H}_k \\ &= \mathbf{H}_k^T[\mathbf{H}_k\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{S}_k\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{H}_k^T]^{-1}\mathbf{H}_k \\ &= \mathbf{H}_k^T[\mathbf{H}_k\mathbf{H}_k^T]^{-1}\mathbf{R}_k\mathbf{S}_k^{-1}\mathbf{R}_k[\mathbf{H}_k\mathbf{H}_k^T]^{-1}\mathbf{H}_k\end{aligned} \tag{46}$$

$$\begin{aligned}\mathbf{v}^T(k)\mathbf{B}^+(k)\mathbf{v}^T(k) &= \nu^T(k)\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{H}_k^T[\mathbf{H}_k\mathbf{H}_k^T]^{-1}\mathbf{R}_k\mathbf{S}_k^{-1}\mathbf{R}_k[\mathbf{H}_k\mathbf{H}_k^T]^{-1}\mathbf{H}_k\mathbf{H}_k^T\mathbf{R}_k^{-1}\nu(k) \\ &= \nu^T(k)\mathbf{S}_k^{-1}\nu(k)\end{aligned} \tag{47}$$

The normalised information residual is thus an information form of the conventional innovation residual.

$\mathbf{f}(\cdot)$	non-linear state transition function
$\mathbf{h}(\cdot)$	non-linear observation function
$\mathbf{g}(\cdot)$	landmark initialisation function
$\mathbf{x}, \mathbf{u}, \mathbf{w}$	state vector, control input and system noise
$\mathbf{P}^n, \mathbf{V}^n$	aircraft position and velocity in NED axes
$\mathbf{\Psi}$	aircraft attitude (roll, pitch and yaw)
$\mathbf{f}^b, \boldsymbol{\omega}^b$	acceleration and rotation rate measurement in the body frame
$\mathbf{C}_b^n$	direction cosine matrix between body and NED axes
$\mathbf{E}_b^n$	angular rate transform matrix
$\mathbf{z}, \mathbf{v}$	observation vector and noise
ρ, φ, ϑ	range, bearing and elevation
$\mathbf{P}_{sb}^b$	sensor's lever arm offset
$\mathbf{C}_s^b$	sensor's direction cosine matrix
$\mathbf{P}_{ms}^s$	relative position of the landmark from the sensor
$\nabla\mathbf{f}_x, \nabla\mathbf{f}_w$	Jacobians of the non-linear state function with respect to the state and noise
$\nabla\mathbf{h}_x, \nabla\mathbf{h}_v$	Jacobians of the observation function with respect to state and noise
$\nabla\mathbf{g}_x, \nabla\mathbf{g}_z$	Jacobians of the landmark initialisation function
$\mathbf{P}(k\|k)$	system covariance matrix
$\mathbf{Q}(k), \mathbf{R}(k)$	system and observation noise matrix
$\nu(k), \mathbf{S}(k)$	innovation vector and innovation covariance
$\mathbf{W}(k)$	filter gain (or weight) matrix
λ_n	threshold value in chi-square distribution with n-dimension
$\mathbf{y}(k\|k), \mathbf{Y}(k\|k)$	information vector and information matrix
$\mathbf{y}_{ij}(k\|k), \mathbf{Y}_{ij}kk$	cross information vector and its matrix
$\mathbf{i}_{ij}^*(k\|k), \mathbf{I}_{ij}^*(k\|k)$	information vector and matrix in the full dimension of the map
$\mathbf{v}(k), \mathbf{B}(k)$	innovation vector and matrix in information space