

Observability Analysis and Active Control for Airborne SLAM

MITCH BRYSON

SALAH SUKKARIEH

University of Sydney
Australia

An unmanned aerial vehicle (UAV) is tasked to explore an unknown environment and to map the features it finds, but must do so without the use of infrastructure-based localisation systems such as GPS, or any a priori terrain data. The UAV navigates using a statistical estimation technique known as simultaneous localisation and mapping (SLAM) which allows for the simultaneous estimation of the location of the UAV as well as the location of the features it sees. SLAM offers a unique approach to vehicle localisation with potential applications including planetary exploration, or when GPS is denied (for example under intentional GPS jamming, or applications where GPS signals cannot be reached), but more importantly can be used to augment already existing systems to improve robustness to navigation failure.

One key requirement for SLAM to work is that it must reobserve features, and this has two effects: firstly, the improvement of the location estimate of the feature; and secondly, the improvement of the location estimate of the platform because of the statistical correlations that link the platform to the feature. So our UAV has two options; should it explore more unknown terrain to find new features, or should it revisit known features to improve localisation quality. These options are instantiated into the online path planner for the UAV.

We present the SLAM algorithm and evaluate two important properties about the algorithm which assist in developing a path planning module for the UAV. The first of these is the use of the probabilistic measure of “entropy” as an information-based measure of the certainty in the map and vehicle locations, and is used as a utility function for planning the UAVs trajectory and determining the order in which features in the map are observed. The second is an observability analysis of SLAM which presents the unobservable states which are dependent on vehicle maneuvers. The analysis dictates the type of manoeuvres required by the UAV while observing features in order to maintain accurate statistical estimates of the map and vehicle location. This has the effect of reducing the action space that the path planner needs to search over.

Using these two properties, we demonstrate an online path planner that intelligently plans the vehicle’s trajectory while exploring unknown terrain in order to maximise the quality of both the map and vehicle location. Results of the online path planning algorithm are presented using a 6-DoF simulator of our UAV. The results show that the vehicle localisation errors are constrained and that the number of features and the size of the map steadily grows during the flight.

Manuscript received January 19, 2006; revised September 26, 2006; released for publication November 20, 2006.

IEEE Log No. T-AES/44/1/918227.

Refereeing of this contribution was handled by G. Lachapelle.

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.

Authors’ address: Australian Centre for Field Robotics, The Rose Street Building, J04, University of Sydney, NSW, 2006, Australia.

0018-9251/08/\$25.00 © 2008 IEEE

I. INTRODUCTION

Consider the scenario where an unmanned aerial vehicle (UAV) is tasked with building an internal representation of the location of features in an environment. In this scenario however, the UAV does not come equipped with its own self-localisation aid such as a Global Positioning System (GPS) receiver (or is not receiving GPS signals) and does not have any a priori terrain data information from which to localise. Such a scenario might occur during certain military application in which GPS signals may be jammed or where the UAV is used for extraterrestrial planetary exploration [1]. The problem is compounded by the requirement for the UAV to have information regarding its velocity and attitude in order to stabilise its motion. In many applications the vehicle uses an Inertial Navigation System (INS), however the eventual growth of localisation errors due to the dead-reckoning nature of the INS limits the time that the vehicle can operate. What is required is the ability to correct the INS errors by using a map that is being generated online; in many respects similar to a real-time surveying task. This is the basis for the paradigm known as simultaneous localisation and mapping (SLAM) in which the vehicle pose and a feature map are estimated simultaneously using only relative observations of the locations of features with respect to the vehicle. SLAM can also be used to augment existing navigation systems such as GPS thus improving accuracy and the robustness of the system during signal dropout. There are several existing implementations of SLAM using land [2] and underwater [3] vehicles where two-dimensional, horizontal localisation and mapping is performed. SLAM has also been demonstrated on a UAV [4, 5] where the three-dimensional position of features and the 6-DoF motion of the vehicle are generated.

SLAM is generally implemented as a statistical filter, where the prediction and estimation of the location of point features is concurrent with the prediction and estimation of the pose and velocity of the vehicle. A feature can be any object in the environment which can be represented by a point in 3D space such as houses, buildings, trees, rocks, and other landscape features. In the scope of this paper we are only interested in estimating the position of features and not any other defining characteristics such as colour, texture, or shape. The SLAM prediction stage involves the propagation of the vehicle and feature models and their corresponding uncertainties and in our implementation mainly relies on inertial navigation. The SLAM estimation step occurs when there is an observation of a feature on the ground; the observation is used to improve both the location estimate of the feature and the location estimate of the vehicle because of the correlations in the SLAM structure that link the platform to the

feature. The effect that observations have towards correcting the vehicle location estimates is dependent on the order in which features are observed and the trajectory of the vehicle. As the vehicle explores unknown terrain, initialising new features into the map, it becomes necessary to return to the well-known regions of the map in order to reduce the growth in localisation errors. This process is referred to as “closing the loop,” in which the uncertainty in both the vehicle localisation and map position estimates is reduced via the correlations made between the vehicle and map states in the SLAM filter. Additionally it is known that the maneuvers the vehicle takes during feature observations affects the accuracy in localisation estimates. Thus it becomes important to understand what effect a vehicle’s control strategy has on the accuracy of the filter estimates.

In this paper, an online path planning strategy is developed that seeks to maximise the accuracy of localisation and mapping estimates in SLAM. For SLAM to be justified as a localisation technique it must be demonstrated that the vehicle state errors can be constrained using SLAM alone, without the need for external data such as from GPS. For this reason we consider the vehicle control problem where SLAM is performed without any external information from GPS or from a known terrain map. We consider how much information is contained in the probability distributions of the SLAM estimates when differing control actions are taken by the vehicle. By information we refer to the degree to which the probability mass in a distribution is concentrated to a small volume of the distribution’s state space, a property measured by the “entropy” of the distribution (the compactness of the probability distribution). Information measures have been popular as a utility function for determining vehicle control actions that improve localisation system performance. In [6] the authors present a method using entropy as a utility function for generating optimal a priori trajectories over a known terrain map for an airborne vehicle using terrain-aided INS. In [7] the authors construct a map of the information available for localisation by an indoor robot, based on the entropy of the probability distribution of the vehicle’s position when moving to each location in the map. This information map is then used to plan the vehicle’s path that will maximise vehicle location information. More recently the problem of trajectory planning for maximising information in localisation and mapping tasks when no a priori map information is available has been considered. In [8] the author presents a purely information-theoretic approach to trajectory planning for a UAV performing SLAM where the initial location of features in the map is unknown. The strategies presented in [9], [10], [11], and [12] use several utilities relating to map information, area coverage, and ability of localisation as the basis for

the control of indoor robots where accurate a priori map information is unavailable.

There are several practical limitations to using information measures for planning in SLAM. Firstly, in the case of an airborne vehicle, the available vehicle actions to optimise over is large as the vehicle is capable of maneuvering in 6-DoF. Secondly, as the number of features N in the map grows, the computational complexity of evaluating the information measures grows in the order of $\mathbf{O}(N^2)$. This growth in computational complexity can be mitigated to some degree by computing the information utility of proposed paths using approximations such as sparse extended information filters [13]. When also considering the requirement for high-rate control of the UAV, the complexity involved in computing the information utility must be reduced before a real-time planner will be practically feasible.

In this work we do not evaluate the information gain for every feasible trajectory; instead, the path planner evaluates the information gain involved with simple straight and level flight trajectories that involve travelling to and making observations of each feature in the map. Further control of the vehicle then reduces to what trajectories should be flown when the observation of the feature is taking place. We tackle this component of the problem by undertaking an observability analysis of inertial SLAM and evaluating several behavior-based decision rules based on this analysis. The decision-rule trajectories are designed to perform vehicle motions that excite the direction of locally unobservable modes in the system, thus maximising the observability of the states over multiple time segments.

Observability analysis has had a long history in studying aided-INS; in [14] and [15] the authors present a new method of observability analysis for piece-wise constant linear systems and apply the method to the analysis of an aided-INS during in-flight alignment. More recently observability has been used to study GPS-aided INS [16, 17] during sensor platform maneuvers. In [18], the authors study the number of unobservable states in the inertial SLAM algorithm by casting the equations into the indirect or error form. It is shown that inertial SLAM is partially observable, however it is not shown which are the unobservable states and how the unobservability is related to the vehicle motions. In this paper we extend the observability analysis to study what are the unobservable modes and how specifically observability is affected by vehicle maneuvers.

The result of this paper is a practically feasible path planner that attempts to maximise SLAM estimate accuracy in both the vehicle and map states while exploring over previously unmapped terrain.

In Section II we present the inertial SLAM algorithms. Section III examines both the information metrics and observability properties of the SLAM

algorithms which are used to evaluate control actions taken by the vehicle. Section IV describes the control architecture on-board the vehicle based on the examination in Section III. A 6-DoF simulation of a UAV in a localisation and mapping scenario is described in Section V. Results of the online path planning algorithm running in the simulation are presented in Section VI. Conclusions and future work are covered in Section VII.

II. INERTIAL SLAM ALGORITHM

In this section we describe the inertial sensor-based SLAM algorithms. The inertial SLAM algorithm is formulated using an extended Kalman filter (EKF) [19] in which map feature locations and the vehicle's position, velocity, and attitude are estimated using relative observations between the vehicle and each feature. The equations for inertial sensor-based SLAM were first presented in [4] and are shown again here for clarity.

A. Process Model

The estimated state vector $\hat{\mathbf{x}}(k)$ contains the three-dimensional vehicle position (\mathbf{p}^n), velocity (\mathbf{v}^n), and Euler angles ($\Psi^n = [\phi, \theta, \psi]$) and the N three-dimensional feature locations (\mathbf{m}_i^n) in the environment:

$$\hat{\mathbf{x}}(k) = \begin{bmatrix} \mathbf{p}^n(k) \\ \mathbf{v}^n(k) \\ \Psi^n(k) \\ \mathbf{m}_1^n(k) \\ \mathbf{m}_2^n(k) \\ \vdots \\ \mathbf{m}_N^n(k) \end{bmatrix} \quad (1)$$

where $i = 1, \dots, N$ and the superscript n indicates the vector is referenced in a local-level navigation frame. The state estimate $\hat{\mathbf{x}}(k)$ is predicted forward in time from $\hat{\mathbf{x}}(k-1)$ via the process model:

$$\hat{\mathbf{x}}(k) = \mathbf{F}(\hat{\mathbf{x}}(k-1), \mathbf{u}(k), k) + \mathbf{w}(k) \quad (2)$$

where $\mathbf{F}(\cdot, \cdot, k)$ is the nonlinear state transition function at time k , $\mathbf{u}(k)$ is the system input at time k , and $\mathbf{w}(k)$ is uncorrelated, zero-mean vehicle process noise errors of covariance \mathbf{Q} . The process model is the standard 6-DoF inertial navigation equations which predict the position, velocity, and attitude of the vehicle. An inertial-frame mechanization [20] is implemented:

$$\begin{bmatrix} \mathbf{p}^n(k) \\ \mathbf{v}^n(k) \\ \Psi^n(k) \end{bmatrix} = \begin{bmatrix} \mathbf{p}^n(k-1) + \mathbf{v}^n(k)\Delta t \\ \mathbf{v}^n(k-1) + [\mathbf{C}_b^n(k-1)\mathbf{f}^b(k) + \mathbf{g}^n]\Delta t \\ \Psi^n(k-1) + \mathbf{E}_b^n(k-1)\omega^b(k)\Delta t \end{bmatrix} \quad (3)$$

where \mathbf{f}^b and ω^b are the body-frame referenced vehicle accelerations and rotation rates which are provided by the inertial sensors on the vehicle and \mathbf{g}^n is the acceleration due to gravity. The direction cosine matrix \mathbf{C}_b^n and rotation rate transformation matrix \mathbf{E}_b^n between the body and navigation frames are given as

$$\mathbf{C}_b^n = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \quad (4)$$

$$\mathbf{E}_b^n = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi \sec_\theta & c_\phi \sec_\theta \end{bmatrix} \quad (5)$$

where $s_{(.)}, c_{(.)}$ and $t_{(.)}$ represent $\sin(.), \cos(.),$ and $\tan(.),$ respectively. Feature locations are assumed to be stationary and thus the process model for the position of the i th feature is given as

$$\mathbf{m}_i^n(k) = \mathbf{m}_i^n(k-1). \quad (6)$$

B. Feature Extraction and Association

We assume that an on-board sensor makes range and bearing observations $\mathbf{z}_i(k)$ to the i th feature. Such observations can be made using either radar or by using a combination of a vision camera and laser range finder. The SLAM algorithm requires that point feature can be extracted and associated from the observation sensor data. Example feature extraction algorithms for vision include SIFT features [21] or model-based feature matching [22]. Features in this sense are points in the sensor data that are distinct and easily recognisable or else points in the sensor data that appear to correlate well with a given feature model or template that is specified offline. The sensor processing algorithms on-board the vehicle may be provided with a visual model and/or a model of the shape of a feature of interest that is likely to be in the environment, such as a tree, and the feature extraction will attempt to find areas in the sensor data that correlate with the properties of the model. Data association of extracted features from subsequent frames (i.e., associating several observations relating to a given feature) can be performed using a simple matching of the properties of the sensor data corresponding to feature (i.e., radar profile or colour/texture information from a vision sensor) or for more generic features by using innovation gating [23]. These methods are beyond the scope of this paper, as we focus more on how the vehicle control actions affect the estimation process. An example of how feature extraction and data association can be performed together for SLAM on a UAV is shown in [24].

C. Observation Model

The observation $\mathbf{z}_i(k)$ is related to the estimated states using (7):

$$\mathbf{z}_i(k) = \mathbf{H}_i(\mathbf{p}^n(k), \mathbf{\Psi}^n(k), \mathbf{m}_i^n(k), k) + \mathbf{v}(k) \quad (7)$$

where $\mathbf{H}_i(\cdot, \cdot, \cdot, k)$ is a function of the feature location, vehicle position, and Euler angles and $\mathbf{v}(k)$ is uncorrelated, zero-mean observation noise errors of covariance \mathbf{R} . The observation model is given by

$$\mathbf{z}_i(k) = \begin{bmatrix} \rho_i \\ \varphi_i \\ \vartheta_i \end{bmatrix} = \begin{bmatrix} \sqrt{(x^s)^2 + (y^s)^2 + (z^s)^2} \\ \tan^{-1}\left(\frac{y^s}{x^s}\right) \\ \tan^{-1}\left(\frac{z^s}{\sqrt{(x^s)^2 + (y^s)^2}}\right) \end{bmatrix} \quad (8)$$

where ρ_i , φ_i , and ϑ_i are the observed range, azimuth, and elevation angles to the feature and x_s , y_s , and z_s are the Cartesian coordinates of \mathbf{p}_{ms}^s , the relative position of the feature w.r.t the sensor, measured in the sensor frame. \mathbf{p}_{ms}^s is given by

$$\mathbf{p}_{ms}^s = \mathbf{C}_b^s \mathbf{C}_n^b [\mathbf{m}_i^n - \mathbf{p}^n - \mathbf{C}_b^n \mathbf{p}_{sb}^b] \quad (9)$$

where \mathbf{C}_b^s is the transformation matrix from the body frame to the sensor frame and \mathbf{p}_{sb}^b is the sensor offset from the vehicle centre of mass, measured in the body frame, otherwise known as the “lever-arm.”

D. Estimation Process

The estimation process is recursive and is broken into three steps as follows.

1) *Prediction:* The vehicle position, velocity, and attitude are predicted forward in time in between feature observations using (2) and (3) with data provided by the inertial sensors. The state covariance \mathbf{P} is propagated forward:

$$\begin{aligned} \mathbf{P}(k | k-1) &= \nabla \mathbf{F}_x(k) \mathbf{P}(k-1 | k-1) \nabla \mathbf{F}_x^T(k) \\ &\quad + \nabla \mathbf{F}_w(k) \mathbf{Q} \nabla \mathbf{F}_w^T(k) \end{aligned} \quad (10)$$

where $\nabla \mathbf{F}_x$ and $\nabla \mathbf{F}_w$ are the Jacobians of the state transition function w.r.t the state vector $\hat{\mathbf{x}}(k)$ and the noise input $\mathbf{w}(k)$ respectively.

2) *Feature Initialization:* When we obtain our first range/bearing observation of a particular feature, its position is calculated using the initialization function $\mathbf{G}_1[\hat{\mathbf{x}}(k), \mathbf{G}_2(\mathbf{z}_i(k))]$ which is given as

$$\mathbf{G}_1 \rightarrow \mathbf{m}_i^n = \mathbf{p}^n + \mathbf{C}_b^n \mathbf{p}_{sb}^b + \mathbf{C}_b^n \mathbf{C}_s^b \mathbf{p}_{ms}^s \quad (11)$$

$$\mathbf{G}_2 \rightarrow \mathbf{p}_{ms}^s = \begin{bmatrix} \rho_i \cos(\varphi_i) \cos(\vartheta_i) \\ \rho_i \sin(\varphi_i) \cos(\vartheta_i) \\ \rho_i \sin(\vartheta_i) \end{bmatrix}. \quad (12)$$

The state vector and covariance are then augmented to include the new feature position:

$$\hat{\mathbf{x}}_{\text{aug}}(k) = \begin{bmatrix} \hat{\mathbf{x}}(k) \\ \mathbf{m}_i^n(k) \end{bmatrix} \quad (13)$$

$$\mathbf{P}_{\text{aug}}(k) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \nabla \mathbf{G}_x & \nabla \mathbf{G}_z \end{bmatrix} \begin{bmatrix} \mathbf{P}(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}(k) \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \nabla \mathbf{G}_x & \nabla \mathbf{G}_z \end{bmatrix}^T \quad (14)$$

where $\nabla \mathbf{G}_x$ and $\nabla \mathbf{G}_z$ are the Jacobians of the initialization w.r.t the state estimate $\hat{\mathbf{x}}(k)$ and the observation $\mathbf{z}_i(k)$, respectively. The position of this feature becomes correlated to both the pose and velocity of the vehicle and the position of other features in the map.

3) *Update:* Once a feature has been initialised into the state vector, subsequent observations of this feature are used to update the entire state vector consisting of the vehicle pose and velocity and the position of this feature and other features in the environment. The state estimate is updated using (15):

$$\hat{\mathbf{x}}(k | k) = \hat{\mathbf{x}}(k | k-1) + \mathbf{W}(k) \nu(k) \quad (15)$$

where the gain matrix $\mathbf{W}(k)$ and innovation $\nu(k)$ are calculated as

$$\nu(k) = \mathbf{z}_i(k) - \mathbf{H}_i(\hat{\mathbf{x}}(k | k-1)) \quad (16)$$

$$\mathbf{W}(k) = \mathbf{P}(k | k-1) \nabla \mathbf{H}_x^T(k) \mathbf{S}^{-1}(k) \quad (17)$$

$$\mathbf{S}(k) = \nabla \mathbf{H}_x(k) \mathbf{P}(k | k-1) \nabla \mathbf{H}_x^T(k) + \mathbf{R} \quad (18)$$

where $\nabla \mathbf{H}_x(k)$ is the Jacobian of the observation function w.r.t the predicted state vector $\hat{\mathbf{x}}(k | k-1)$. The state covariance $\mathbf{P}(k | k)$ is updated after the observation using the covariance update:

$$\mathbf{P}(k | k) = \mathbf{P}(k | k-1) - \mathbf{W}(k) \mathbf{S}(k) \mathbf{W}^T(k). \quad (19)$$

Once a feature leaves the field of view of the sensor, its position remains in the state vector and continues to be updated via its correlations to other visible features in the state vector.

III. INFORMATION MEASURES AND OBSERVABILITY ANALYSIS

In this section we analyse both the concept of information as a localisation performance metric and study the observability properties of the inertial SLAM algorithms and their relationship to vehicle maneuvers.

A. Entropy and Information Gain

The entropy $h(\mathbf{x})$ of a multivariate Gaussian probability distribution over the variable \mathbf{x} can be calculated from its covariance matrix \mathbf{P} as follows:

$$h(\mathbf{x}) = \frac{1}{2} \log[(2\pi e)^n |\mathbf{P}|]. \quad (20)$$

Entropy is a scalar-value measure of the compactness of a distribution. When we use a probability distribution to represent the knowledge we have about the variable \mathbf{x} , the smaller the entropy of the distribution, the more the probability mass is assigned to a smaller area of the state space and thus the more informative the distribution is about the state.

The evolution of the probability distributions in the EKF is a function of the state \mathbf{x} , due to the linearisation of the process and observation models. When the value of the state can be controlled to some degree, we can therefore control the evolution of the EKF probability distributions in order to minimise entropy.

Firstly we define an action \mathbf{a} as a set of controlled states and observations to be made n steps into the future:

$$\mathbf{a} \in \{\mathbf{x}(k), \mathbf{z}(k), \mathbf{x}(k+1), \mathbf{z}(k+1), \dots, \mathbf{x}(k+n), \mathbf{z}(k+n)\}. \quad (21)$$

In the case of the UAV in a localisation and mapping task, an action consists of a set of observations of different features to be made as well as the position, velocity, and attitude trajectories of the vehicle over a finite time horizon. The utility for each possible action that can be made is specified by the entropic information gain $I[\mathbf{x}, \mathbf{a}]$, which is defined as the difference between the entropies of the distributions about the estimated states before and after taking the action:

$$I[\mathbf{x}, \mathbf{a}] = h(\mathbf{x}) - h(\mathbf{x} | \mathbf{a}) \\ = -\frac{1}{2} \log \left[\frac{|\mathbf{P}(\mathbf{x} | \mathbf{a})|}{|\mathbf{P}(\mathbf{x})|} \right] \quad (22)$$

where $h(\mathbf{x})$ and $\mathbf{P}(\mathbf{x})$ are the prior entropy and covariance and $h(\mathbf{x} | \mathbf{a})$ and $\mathbf{P}(\mathbf{x} | \mathbf{a})$ are the entropy and covariance of the state \mathbf{x} subsequent to taking action \mathbf{a} (i.e., taking a particular vehicle trajectory and making observations of features along the way). The entropic information gain is a number which is negative for a loss and positive for a gain in information.

The advantage of entropy and entropic information gain as utility measures in a control problem is that they represent the whole informativeness of a multi-variate distribution in a scalar value, hence simplifying the control problem to

$$\mathbf{a}^* = \arg \max(I[\mathbf{x}, \mathbf{a}]) \quad (23)$$

where \mathbf{a}^* is the best control action. This scalar measure however can pose a disadvantage in the sense that the distribution of the information across states may be uneven. For our purposes however, the scalar measure is sufficient determining overall information gain.

B. Observability

A system is defined as observable [25] if the initial state \mathbf{x}_0 at any initial time t_0 can be determined given the state transition and observation models of the system and observations $\mathbf{z}[t_0, t]$ from time t_0 to a finite time t . When a system is fully observable, the lower bound of the error in the estimate of its state will only depend on the noise parameters of the system and will not be reliant on initial information about the states.

1) The Inertial SLAM Algorithm in its Indirect Form: In the case of inertial SLAM, the state transition and observation models are the nonlinear equations (2) and (7). In [18] the authors demonstrate how inertial SLAM can be cast into a form where the state-space of the system is composed of the errors in vehicle pose and the errors in the feature map. One main advantage with this indirect form is that the equations can be represented as a piece-wise linear system, thus simplifying the observability analysis.

In the indirect form the estimated state vector ($\delta\mathbf{x}$) is composed of the vehicle position error ($\delta\mathbf{p}^n$), vehicle velocity error ($\delta\mathbf{v}^n$), vehicle misalignment angles ($\delta\Psi^n$), and the feature position errors ($\delta\mathbf{m}_i^n$):

$$\delta\mathbf{x}(k) = \begin{bmatrix} \delta\mathbf{p}^n(k) \\ \delta\mathbf{v}^n(k) \\ \delta\Psi^n(k) \\ \delta\mathbf{m}_1^n(k) \\ \delta\mathbf{m}_2^n(k) \\ \vdots \\ \delta\mathbf{m}_N^n(k) \end{bmatrix}. \quad (24)$$

In this form the inertial navigation equations and the addition of new features to the map are run separately to the estimation process. The error states are defined as the difference between the unknown “true” system state and the estimated state. A Kalman filter estimates the error states which are then used to correct the INS and the feature map.

The error states are predicted forward in the Kalman filter prediction stage using the piece-wise linear process model:

$$\delta\mathbf{x}(k) = \mathbf{F}\delta\mathbf{x}(k-1) + \mathbf{w}(k) \quad (25)$$

where

$$\delta\mathbf{p}^n(k) = \delta\mathbf{p}^n(k-1) + \delta\mathbf{v}^n(k)\Delta t \quad (26)$$

$$\delta\mathbf{v}^n(k) = \delta\mathbf{v}^n(k-1) + [\times\mathbf{f}^n]\delta\Psi^n(k)\Delta t + \mathbf{C}_b^n\delta\mathbf{f}^b \quad (27)$$

$$\delta\Psi^n(k) = \delta\Psi^n(k-1) + \mathbf{C}_b^n\delta\omega^b \quad (28)$$

$$\delta\mathbf{m}_i^n(k) = \delta\mathbf{m}_i^n(k-1) \quad (29)$$

where $\delta\mathbf{f}^b$ and $\delta\omega^b$ are the errors in the accelerometer and gyro readings, respectively (modelled as

Gaussian white noise processes) and $[\times \mathbf{f}^n]$ is the skew-symmetric matrix of the specific force vector in the navigation frame.

Equation (25) can also be represented in continuous form as shown in (30):

$$\delta \dot{\mathbf{x}}(t) = \mathbf{F}\delta \mathbf{x}(t) + \mathbf{w}(t) \quad (30)$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & [\times \hat{\mathbf{f}}^n] & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}. \quad (31)$$

In the update stage, the observation for a single feature is $\delta \mathbf{p}_{mis}^s$, the difference between the observation of range and bearing to the feature, transformed into Cartesian coordinates and the estimated value ($\hat{\mathbf{p}}_{mis}^s$) as computed from the existing map:

$$\delta \mathbf{p}_{mis}^s = \mathbf{G}_2(\mathbf{z}_i) - \hat{\mathbf{p}}_{mis}^s. \quad (32)$$

In the following equations the time index k is implied and has been removed to make the analysis easier to read. The observation is related to the vehicle error states using the linear equation:

$$\delta \mathbf{p}_{mis}^s = \mathbf{H}_i \delta \mathbf{x} + \mathbf{v} \quad (33)$$

$$\begin{aligned} \delta \mathbf{p}_{mis}^s = \mathbf{C}_b^s \hat{\mathbf{C}}_n^b [\delta \mathbf{m}_i^n - \delta \mathbf{p}^n + [\times \hat{\mathbf{C}}_b^s \mathbf{p}_{sb}^b] \delta \Psi^n] \\ + [\times \mathbf{C}_b^s \hat{\mathbf{C}}_n^b [\hat{\mathbf{m}}_i^n - \hat{\mathbf{p}}^n - \hat{\mathbf{C}}_b^s \mathbf{p}_{sb}^b]] \delta \Psi^n. \end{aligned} \quad (34)$$

The body to sensor transformation \mathbf{C}_b^s and the sensor offset \mathbf{p}_{sb}^b are usually known to a high degree of accuracy on most UAV platforms through sensor alignment and calibration procedures performed before take-off. In the case where alignment and calibration cannot be performed accurately or when the sensor pose w.r.t the body is uncertain (for example due to significant aeroelasticity in the airframe), the variables $\delta \mathbf{p}_{sb}^b$ and $\delta \mathbf{C}_b^s$ (i.e., the error in sensor pose) may need to be added to the estimated state vector of the system. This is however beyond the scope of this paper and thus we assume that these values are known with negligible error.

We can further simplify the observation model in (34) without effecting the observability analysis of the equations. We assume that the sensor frame s lies along the body axis b (i.e., $\mathbf{C}_s^b = \mathbf{I}_{3 \times 3}$), and that the sensor offset from the body axis (\mathbf{p}_{sb}^b) is zero. The SLAM observation model in (34) can now be simplified to

$$\delta \mathbf{p}_{mis}^n = \delta \mathbf{m}_i^n - \delta \mathbf{p}^n + [\times \hat{\mathbf{r}}_{mv}^n] \delta \Psi^n \quad (35)$$

thus

$$\mathbf{H} = [-\mathbf{I}_{3 \times 3} \ \mathbf{0} \ [\times \hat{\mathbf{r}}_{mv}^n] \cdots \mathbf{I}_{3 \times 3} \cdots] \quad (36)$$

where $[\hat{\mathbf{r}}_{mv}^n \times]$ is the skew-symmetric matrix of the estimate of the relative position between the feature and vehicle positions, in the navigation frame. The inertial SLAM equations have now been defined as a piece-wise linear system, thus simplifying the observability analysis of the system.

2) *Linear System Observability*: For a system specified by the time-invariant linear equations:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{F}\mathbf{x}(t) \\ \mathbf{z}(t) &= \mathbf{H}\mathbf{x}(t) \end{aligned} \quad (37)$$

where \mathbf{F} and \mathbf{H} are time-invariant matrices, a necessary property for the system to be completely observable is that the local observability matrix [25] (LOM):

$$\mathcal{O} = [\mathbf{H}^T, (\mathbf{HF})^T, (\mathbf{HF}^2)^T, \dots, (\mathbf{HF}^{n-1})^T]^T \quad (38)$$

have rank n where n is the dimension of the state vector \mathbf{x} . If the system in (37) is time variant:

$$\begin{aligned} \mathbf{F}(t) &= \mathbf{F}_1, & \mathbf{H}(t) &= \mathbf{H}_1, & t_0 < t < t_1 \\ \mathbf{F}(t) &= \mathbf{F}_2, & \mathbf{H}(t) &= \mathbf{H}_2, & t_1 < t < t_2 \\ &\vdots &&& \\ \mathbf{F}(t) &= \mathbf{F}_k, & \mathbf{H}(t) &= \mathbf{H}_k, & t_{k-1} < t < t_k \end{aligned} \quad (39)$$

then we can define two types of system observability. Firstly we define the system in (39) to be instantaneously observable at time k if the LOM at time k :

$$\mathcal{O}_k = [\mathbf{H}_k^T, (\mathbf{H}_k \mathbf{F}_k)^T, (\mathbf{H}_k \mathbf{F}_k^2)^T, \dots, (\mathbf{H}_k \mathbf{F}_k^{n-1})^T]^T \quad (40)$$

has rank n . \mathbf{F}_k and \mathbf{H}_k are the k th time segment state transition matrix and observation models of the system. Instantaneous observability means that the system state \mathbf{x} over the time interval $[t_{k-1}, t_k]$ can be estimated using the information in the observations $\mathbf{z}[t_{k-1}, t_k]$ without the need for extra observations or prior state information.

Secondly we define the system in (39) to be completely observable over the time interval $[t_0, t_k]$ if the total observability matrix (TOM) [14]:

$$\mathcal{O}_{\text{TOM}}(k) = \begin{bmatrix} \mathcal{O}_1 \\ \mathcal{O}_2 e^{\mathbf{F}_1 \Delta t_1} \\ \vdots \\ \mathcal{O}_k e^{\mathbf{F}_{k-1} \Delta t_{k-1} \dots e^{\mathbf{F}_1 \Delta t_1}} \end{bmatrix} \quad (41)$$

has rank n . Δt_k is the difference of t_k and t_{k-1} . Complete observability means that the system state \mathbf{x} can be estimated over the time interval $[t_0, t_k]$ using the observations $\mathbf{z}[t_0, t_k]$. A system that is locally observable over every time segment $[t_{k-1}, t_k]$ in the interval $[t_0, t_k]$ will also be completely observable over the interval $[t_0, t_k]$, however a system may be

completely observable over $[t_0, t_k]$ but not locally observable over each time segment $[t_{k-1}, t_k]$.

In [18] the authors perform a rank analysis of the observability matrices of the indirect inertial SLAM equations, determining the number of observable and unobservable modes in inertial SLAM. It was shown that for a total of 9 vehicle states plus 3N map states, there were 4 instantaneously unobservable states (that is, given a single time segment), regardless of the number of map features considered. If the observability was considered over two time segments in which some parameters of the linear system changed between segments, then one more state could become observable (considering the complete observability over the two time segments) resulting in a total of 3 unobservable states. It was shown that any additional time segments do not further increase the number of observable modes.

In this paper we wish to extend the observability analysis performed in [18]: we are interested in discovering what are the unobservable states in the system and how these states are effected by the manouevres/control actions taken by the vehicle.

3) *Observability Rank Analysis*: The LOM for any single time segment of the system described in (31) and (36), where the map consists of a single feature is

$$\mathcal{O}_{\text{LOM}} = \begin{bmatrix} -\mathbf{I}_{3 \times 3} & \mathbf{0} & [\times \hat{\mathbf{r}}_{m1v}^n] & \mathbf{I}_{3 \times 3} \\ \mathbf{0} & -\mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -[\times \mathbf{f}^n] & \mathbf{0} \end{bmatrix}. \quad (42)$$

In all of the following observability matrices we only present the non-zero rows, and have removed those rows that contain only zeros as they do not contribute to the observability analysis. Given that $\hat{\mathbf{r}}_{m1v}^n \times \mathbf{f}^n \neq \mathbf{0}$, the LOM in (42) has a rank of eight where the total number of estimated states is twelve. When the map consists of two features, the LOM becomes

$$\mathcal{O}_{\text{LOM}} = \begin{bmatrix} -\mathbf{I}_{3 \times 3} & \mathbf{0} & [\times \hat{\mathbf{r}}_{m1v}^n] & \mathbf{I}_{3 \times 3} & \mathbf{0} \\ -\mathbf{I}_{3 \times 3} & \mathbf{0} & [\times \hat{\mathbf{r}}_{m2v}^n] & \mathbf{0} & \mathbf{I}_{3 \times 3} \\ \mathbf{0} & -\mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -[\times \mathbf{f}^n] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -[\times \mathbf{f}^n] & \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (43)$$

The LOM now has a rank of eleven where the total number of estimated states is fifteen. For the addition of each extra feature into the map, the rank of the LOM increases by three and thus the total number of unobservable states for a single time segment is always four.

To study the observability of the system over multiple time segments, the TOM in (41) is used. By using only the first-order term in the matrix exponential, we can approximate $e^{\mathbf{F}_1 \Delta t_1} \approx \mathbf{I}_{3 \times 3} + \mathbf{F}_1 \Delta t_1$

and thus the TOM over two time segments, with two map features is

$$\mathcal{O}_{\text{TOM}} = \begin{bmatrix} -\mathbf{I}_{3 \times 3} & \mathbf{0} & [\times \hat{\mathbf{r}}_{m1v1}^n] & \mathbf{I}_{3 \times 3} & \mathbf{0} \\ -\mathbf{I}_{3 \times 3} & \mathbf{0} & [\times \hat{\mathbf{r}}_{m2v1}^n] & \mathbf{0} & \mathbf{I}_{3 \times 3} \\ \mathbf{0} & -\mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -[\times \mathbf{f}_1^n] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -[\times \mathbf{f}_1^n] & \mathbf{0} & \mathbf{0} \\ -\mathbf{I}_{3 \times 3} & -\Delta \mathbf{I}_{3 \times 3} & [\times \hat{\mathbf{r}}_{m1v2}^n] & \mathbf{I}_{3 \times 3} & \mathbf{0} \\ -\mathbf{I}_{3 \times 3} & -\Delta \mathbf{I}_{3 \times 3} & [\times \hat{\mathbf{r}}_{m2v2}^n] & \mathbf{0} & \mathbf{I}_{3 \times 3} \\ \mathbf{0} & -\mathbf{I}_{3 \times 3} & -\Delta t [\times \mathbf{f}_2^n] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{3 \times 3} & -\Delta t [\times \mathbf{f}_2^n] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -[\times \mathbf{f}_2^n] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -[\times \mathbf{f}_2^n] & \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (44)$$

Provided that $\mathbf{f}_1^n \times \mathbf{f}_2^n \neq \mathbf{0}$ (i.e., the cross product of these vectors is not zero and thus they are not parallel in direction) and/or $\hat{\mathbf{r}}_{m1v1}^n \times \hat{\mathbf{r}}_{m2v2}^n \neq \mathbf{0}$, the extra time segment adds one linearly independent row to the matrix. This increases the rank of the system from eleven to twelve, bringing the number of unobservable states to three. Regardless of the number of features added to the map or any extra time segments, the number of unobservable states will never be less than three.

4) *Direction of the Unobservable Modes in Inertial SLAM*: In order to determine the direction in the state space of the unobservable modes we can evaluate the observability Grammian:

$$\mathcal{N} = \mathcal{O}^T \mathcal{O} \quad (45)$$

where \mathcal{O} can be either the LOM or the TOM of the system, depending on the number of time segments we are considering. The eigenvectors corresponding to the zero eigenvalues of the Grammian \mathcal{N} are the unobservable modes of the system [14].

Considering the Grammian for the TOM in (44), the eigenvectors of the three zero eigenvalues of \mathcal{N} are

$$\begin{aligned} \mathbf{x}^{\text{unobs}} &= [\mathbf{I}_{3 \times 3}, \mathbf{0}, \mathbf{0}, \mathbf{I}_{3 \times 3}, \mathbf{I}_{3 \times 3}, \dots] \\ &= \delta \mathbf{p}^n + \delta \mathbf{m}_1^n + \delta \mathbf{m}_2^n + \dots \end{aligned} \quad (46)$$

Thus for the system with at least two time segments where $\mathbf{f}_1^n \times \mathbf{f}_2^n \neq \mathbf{0}$ and/or $\hat{\mathbf{r}}_{m1v1}^n \times \hat{\mathbf{r}}_{m2v2}^n \neq \mathbf{0}$, the three unobservable modes are the components of the vector sum of the vehicle and feature position errors. The observable and unobservable vectors form an orthogonal basis in the state-space in (24). Thus any vectors in the state-space that are orthogonal to the unobservable vectors will be observable. Some important results can be derived from this analysis as follows.

1) The unobservable modes do not intersect with the vehicle velocity and attitude error states (i.e., the unobservable modes contain no components of the velocity and attitude states). These states are therefore observable as they lie completely in the observable subspace of the system.

2) The vehicle and feature position errors are only partially observable as they form part of the unobservable subspace.

3) Some directions in the state space that are orthogonal to the unobservable modes and thus completely observable are: a) the error in the relative position of each feature to the vehicle ($\delta\mathbf{m}_1^n - \delta\mathbf{p}^n, \delta\mathbf{m}_2^n - \delta\mathbf{p}^n, \dots$), and b) the error in the relative position between each feature in the map ($\delta\mathbf{m}_1^n - \delta\mathbf{m}_2^n, \delta\mathbf{m}_2^n - \delta\mathbf{m}_3^n, \dots$).

The observability analysis of SLAM demonstrates that we are only able to build a relative map of point features and localise the position of the vehicle w.r.t the features themselves; neither the global position of the vehicle or the global position of the features can be estimated. The observability analysis formalises the intuitive concept that only a relative mapping and localisation is possible without global information about the position of the vehicle or features such as GPS or prior terrain information.

5) *Locally Unobservable Mode:* We have shown the observability of the system over two time segments where the values of the matrices \mathbf{F} and \mathbf{H} change between segments. We are also interested in determining the locally unobservable modes in the system, that is the unobservable states in the LOM, where only one time segment is considered. The locally unobservable mode indicates the direction in the state space to which no information is being added over the local time set of observations.

In order to simplify the analysis, we begin by separating out the modes of the inertial SLAM state vector which are always unobservable. Consider the modification to the state vector such that instead of estimating the global vehicle ($\delta\mathbf{p}^n$) and map feature ($\delta\mathbf{m}_i^n$) position error states, we estimate the position error of each feature minus the vehicle position error (i.e., the relative feature to vehicle position error ($\delta\mathbf{m}_i^n - \delta\mathbf{p}^n$)). We maintain the errors in the vehicle velocity ($\delta\mathbf{v}^n$) and attitude ($\delta\Psi^n$) in the state vector as before. The time dynamics and observation model of the new state is

$$(\delta\mathbf{m}_i^n - \delta\mathbf{p}^n) = -\delta\mathbf{v}^n \quad (47)$$

$$\delta\mathbf{p}_{mib}^n = (\delta\mathbf{m}_i^n - \delta\mathbf{p}^n) + [\times\hat{\mathbf{r}}_{mv}^n]\delta\Psi^n. \quad (48)$$

The state vector now becomes

$$\mathbf{x} = [(\delta\mathbf{m}_1^n - \delta\mathbf{p}^n), (\delta\mathbf{m}_2^n - \delta\mathbf{p}^n), \dots, \delta\mathbf{v}^n, \delta\Psi^n]^T.$$

The TOM of the new system over two time segments with two map features is

$$\mathcal{O}_{\text{TOM}} = \begin{bmatrix} -\mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} & [\times\hat{\mathbf{r}}_{m1v1}^n] \\ \mathbf{0} & -\mathbf{I}_{3 \times 3} & \mathbf{0} & [\times\hat{\mathbf{r}}_{m2v1}^n] \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_{3 \times 3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_{3 \times 3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & [\times\mathbf{f}_1^n] \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & [\times\mathbf{f}_1^n] \\ -\mathbf{I}_{3 \times 3} & \mathbf{0} & -\Delta t \mathbf{I}_{3 \times 3} & [\times\hat{\mathbf{r}}_{m1v2}^n] \\ \mathbf{0} & -\mathbf{I}_{3 \times 3} & -\Delta t \mathbf{I}_{3 \times 3} & [\times\hat{\mathbf{r}}_{m2v2}^n] \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_{3 \times 3} & -\Delta t [\times\mathbf{f}_2^n] \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_{3 \times 3} & -\Delta t [\times\mathbf{f}_2^n] \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & [\times\mathbf{f}_2^n] \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & [\times\mathbf{f}_2^n] \end{bmatrix}. \quad (50)$$

The first 6 rows are linearly independent, rows 7 to 9 provide three extra linearly independent rows and provided that $\mathbf{f}_1^n \times \mathbf{f}_2^n \neq 0$ and/or $\hat{\mathbf{r}}_{mv1} \times \hat{\mathbf{r}}_{mv2} \neq 0$, rows 13 to 18 and rows 31 to 36 provide an extra three linearly independent rows. Thus the TOM has rank twelve (the dimension of the state vector with two features) and thus is completely observable. Adding extra features (thus adding the relative feature to vehicle position error ($\delta\mathbf{m}_i^n - \delta\mathbf{p}^n$) to the state vector) increases the rank of the TOM by three thus resulting in full state observability for any number of landmarks greater than or equal to one.

Consider the LOM using the new state vector over a single time segment with two map features:

$$\mathcal{O}_{\text{LOM}} = \begin{bmatrix} -\mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} & [\times\hat{\mathbf{r}}_{m1v}^n] \\ \mathbf{0} & -\mathbf{I}_{3 \times 3} & \mathbf{0} & [\times\hat{\mathbf{r}}_{m2v}^n] \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_{3 \times 3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_{3 \times 3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & [\times\mathbf{f}_1^n] \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & [\times\mathbf{f}_1^n] \end{bmatrix}. \quad (51)$$

There is one unobservable mode:

$$\mathbf{x}^{\text{unobs}} = [\mathbf{f}^n \times \hat{\mathbf{r}}_{m1v}^n, \mathbf{f}^n \times \hat{\mathbf{r}}_{m2v}^n, 0, \mathbf{f}^n]^T. \quad (52)$$

This locally unobservable mode is also common to the original state vector from the system in (31) and (36) (i.e., it is the fourth locally unobservable mode in the original state vector, where the other three modes are those in (46). The three modes in (46) cannot be made observable through any amount of vehicle maneuvering).

Practically, consider an example in which a UAV is in straight and steady level (SSL) flight observing features on the ground below while the vehicle passes by. In this scenario the relative position vectors to

TABLE I
Classes of UAV Maneuvers and Expected Effect on Navigation Estimate Accuracy

Maneuver	$\Delta \mathbf{f}^n$	$\hat{\mathbf{r}}_{mv}^n$	Expected Estimate Accuracy
Motionless/Hover	none	none	continuing loss in heading accuracy
Climb or Descent	none	steady change in vector direction	continuing loss in heading accuracy
Steady Level Flight	none	steady change in vector direction	continuing loss in heading accuracy
Steady Turn/Orbit	vector traces a cone shape	range to feature constant for features at center of turn, direction changing	high accuracy on all attitude states
S-Shape Maneuver	vector oscillates back and forward	vector to features traces S-shape	high accuracy on all attitude states

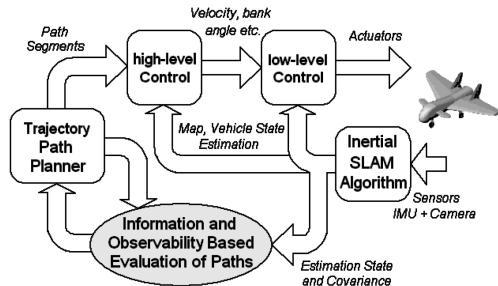


Fig. 1. The structure of the vehicle control scheme on the UAV. Control hierarchy composed of low-level and high-level control.

Trajectory planner designates trajectory segments based on evaluation of information and observability along each path.

each feature ($\hat{\mathbf{r}}_{mv}^n$) will rotate in the navigation frame as the vehicle moves past the feature, however the specific force vector direction will remain constant. In this case, the components of the locally unobservable mode in the attitude states points along the z-axis direction and thus there is a continuing loss of information about the attitude state corresponding to this axis (i.e., the platform yaw or heading angle).

6) *Observability and Vehicle Maneuvres/Control Actions:* In this section we analyse how changes in \mathbf{f}^n and $\hat{\mathbf{r}}_{mv}^n$ through aircraft maneuvers affect inertial SLAM observability. Table I describes some typical aircraft maneuvers, the resulting local observability of the localisation estimates, and the speculated result in localisation estimate information. We can see from the observability analysis that when the vehicle is in motionless hover, that the direction of the locally unobservable mode is constant. For SSL flight or a steady climb/descent in which the specific force vector (\mathbf{f}^n) does not change direction between time segments, the components of the unobservable mode in the vehicle attitude error states remain constant, thus restricting full motion of the locally unobservable mode. These maneuvers result in low heading error information (i.e., the direction in the state-space of local unobservability given the only specific force acting on the vehicle is from lift). When performing maneuvers such as steady turns or S-shape maneuvers, the acceleration vector is excited in the lateral direction thus rotating the unobservable mode between time segments and distributing the information across all of the vehicle attitude error state estimates.

IV. AERIAL VEHICLE PATH PLANNING ARCHITECTURE

In this section we describe the control architecture for improving localisation and mapping estimate accuracy when using SLAM based on a combination of information measures and insight gained from the observability analysis.

A. Vehicle Control Hierarchy

This section describes the control hierarchy of a small sized (40 kg, 2 m wing span) UAV, the Brumby MkIII, a research platform at the Australian Centre for Field Robotics. The control hierarchy, illustrated in Fig. 1, can be broken into three sections: low-level control, high-level control and the trajectory path planner. The low-level control system maintains the stability of the aircraft, using the aircraft throttle and control surfaces to control the altitude, velocity, bank angle, and sideslip of the aircraft. The high-level control system provides altitude, velocity and bank angle commands to the low-level control system in order to control the vehicle along allocated trajectory segments composed of straight lines and arcs in space. The low-level and high-level control systems are currently implemented on the flight vehicle (see [26]) and provide a basic path following capability. Combinations of straight lines and arcs that form a complete trajectory in 3D space are provided by the trajectory path planner. The vehicle position, velocity and attitude are provided by the SLAM solution as feedback to all levels of the control system.

B. Information-Based Path Planning and Trajectory Allocation

In this section we describe the method for generating potential trajectories from the trajectory path planner and the methods used to evaluate the information gain of performing each trajectory. The information-based path planner is a single-step lookahead control based on maximising the entropic information gain of the joint probability distribution, including vehicle and map states, of the SLAM estimated state. The action space for the path planner is made up of a discrete grid on waypoints, all at

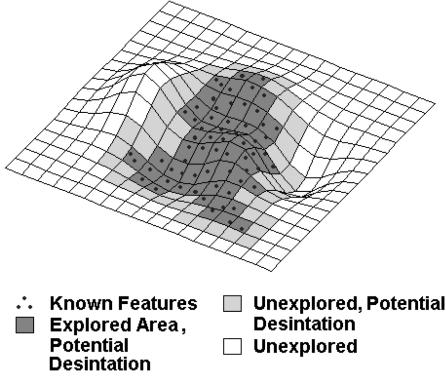


Fig. 2. Evaluating potential destinations by guidance system: destinations composed of explored region and surrounding unexplored regions broken down into grid of size 100×100 m.

constant altitude above the terrain, that the vehicle could move to. At the time of decision-making (i.e., the last allocated trajectory has been completed) the path planner takes the following steps.

1) The path planner generates a discrete rectangular grid of potential destination waypoints that the UAV can fly to from its current location. Each waypoint is placed at an altitude of 100 m with a grid size of 100×100 m. Each grid is labelled. Explored grids are those that have known SLAM map features lying within their bounds, whereas unexplored grids are the remaining grids in which no features have been observed. Potential waypoints are placed in both the explored grids and the unexplored grids which touch upon the explored grids (see Fig. 2). Any proposed destination waypoints that are inside the turning radius of the vehicle are removed from the list.

2) For each potential waypoint, a trajectory from the current location to the waypoint is formed by a steady turn until the heading angle of the vehicle is aligned with the destination point followed by straight and level flight until the vehicle reaches the destination waypoint. In all segments of the trajectory the vehicle remains at a fixed altitude in flight.

3) The utility based on information gain for the trajectories to each waypoint is evaluated and the trajectory that results in the highest entropic information gain is chosen with the corresponding trajectory segments being sent to the high-level and low-level control systems for the vehicle to perform. Once the trajectory segment has been completed the vehicle replans to the next waypoint.

The evaluation of the information gain based utility is explained in the following subsections.

1) *Evaluating the Information Gain for a Potential Trajectory:* For a given potential trajectory, we predict the position and attitude of the vehicle at 1 s increments along the trajectory (assuming the vehicle bank angle is a maximum of 60 deg during turns and zero when flying straight). At each point in the

trajectory we also evaluate the observations the vehicle would be expected to make based on our current estimate of the feature locations and our knowledge of the vehicle pose and the orientation and constraints of the observations sensors on the vehicle. At the points along the trajectory where observations are expected to be present, the Jacobian of the observation function in (7) is evaluated. A copy of the SLAM covariance matrix is made and this covariance matrix is propagated forward in time using (10)–(19) with the evaluated Jacobians at each trajectory sample until the destination is reached. What results is a simulated approximation of the SLAM covariance matrix if we were to take the potential trajectory considered, i.e., $\mathbf{P}(\mathbf{x} | \mathbf{a})$. The expected covariance and the current actual SLAM covariance $\mathbf{P}(\mathbf{x})$ (i.e., before any trajectories are decided upon) are substituted into (22) to calculate the information gain and therefore utility for the current proposed waypoint.

2) Evaluating Utility over Unexplored Grid Points:

Since we are unaware if any observations of features will be made when we move into an unexplored area, evaluating the information gain for visiting these waypoints is more complicated. We will assume that within each unexplored area there is a number of randomly distributed unseen features based on a feature density ρ_f . As the vehicle integrates features into the SLAM map, the number of features observed and the ground coverage area that the UAV has observed are used to compute ρ_f . The expected initial positions of the features are scattered randomly within the unexplored grid square. When flying towards or over an unexplored area we create expected observations of unseen features which are added to the list of observations as explained in the previous subsection. The copy of the covariance matrix is augmented to include the initial covariance \mathbf{U}_m of these features:

$$\mathbf{P}(\mathbf{x})_{\text{aug}} = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vm} & 0 \\ \mathbf{P}_{mv} & \mathbf{P}_{mm} & 0 \\ 0 & 0 & \mathbf{U}_m \end{bmatrix}. \quad (53)$$

The covariance of the expected feature positions before any observations \mathbf{U}_m is given an appropriately large initial value (diagonal matrix with values of 10^9 m² along the diagonal). We now have an approximation of the information to be gained by visiting an unexplored area and integrating new features into the map.

C. Behavior-Based Decision Rules from Observability Analysis

In this section we describe decision rules that are applied to the path planning process based upon the mathematical insight gained from the observability analysis. The following decision rules analyse the

uncertainty in heading angle of the vehicle, briefly interrupting the information-gain path planning to perform different vehicle maneuvers when the decision rules are activated.

1) *S-Shape Maneuver Behavior*: This decision rule analyses the growth in heading uncertainty along the straight and level segments of the vehicle's trajectory. The difference between the current heading uncertainty along the straight and the heading uncertainty at the beginning of the current straight flight segment is examined. When the heading uncertainty growth reaches above a given threshold, the guidance system allocates a single s-shape maneuver along the direction the vehicle is heading such that the sideways-looking cameras are used to make observations of the features below while the vehicle is banking. At the end of the maneuver the vehicle continues along the straight flight segment previously allocated. The aim of this decision rule is to limit the growth in heading angle errors accumulated during long straight and level sections of the flight. The value of the heading uncertainty growth threshold is arbitrary. Setting the threshold to a low value will result in better heading accuracy throughout the flight but at the cost of a longer time taken to perform the mapping. The results shown here use a threshold of 1° (1σ).

2) *Orbit Maneuver Behavior*: This decision rule analyses the absolute heading angle uncertainty when the vehicle is approaching the destination set in the information-gain path planning. When the vehicle is at a distance of its minimum orbiting radius from the destination, the value of heading angle uncertainty is examined. If the absolute heading angle uncertainty is above a given threshold, then the guidance system replans a single steady 360° orbit around the destination point specified in the information-based path planner. If the absolute heading uncertainty is below the threshold then the trajectory continues as planned, flying straight and level until directly over the specified destination. When either the single orbit or straight and level flight over the destination is achieved, the information-based path planner then replans the vehicle trajectory as explained in Section IVB. The value of the absolute heading uncertainty threshold is arbitrary. Setting the threshold to a low value will result in better heading accuracy throughout the flight but at the cost of a longer time taken to perform the mapping. The results shown in this paper use a threshold of 2° (1σ).

V. UAV SIMULATION SCENARIO

A 6-DoF simulation of a UAV in a localisation and mapping task while flying over unknown terrain, without GPS, is used to examine both the effect different aircraft maneuvers have and the effectiveness of the path planning algorithms proposed in the

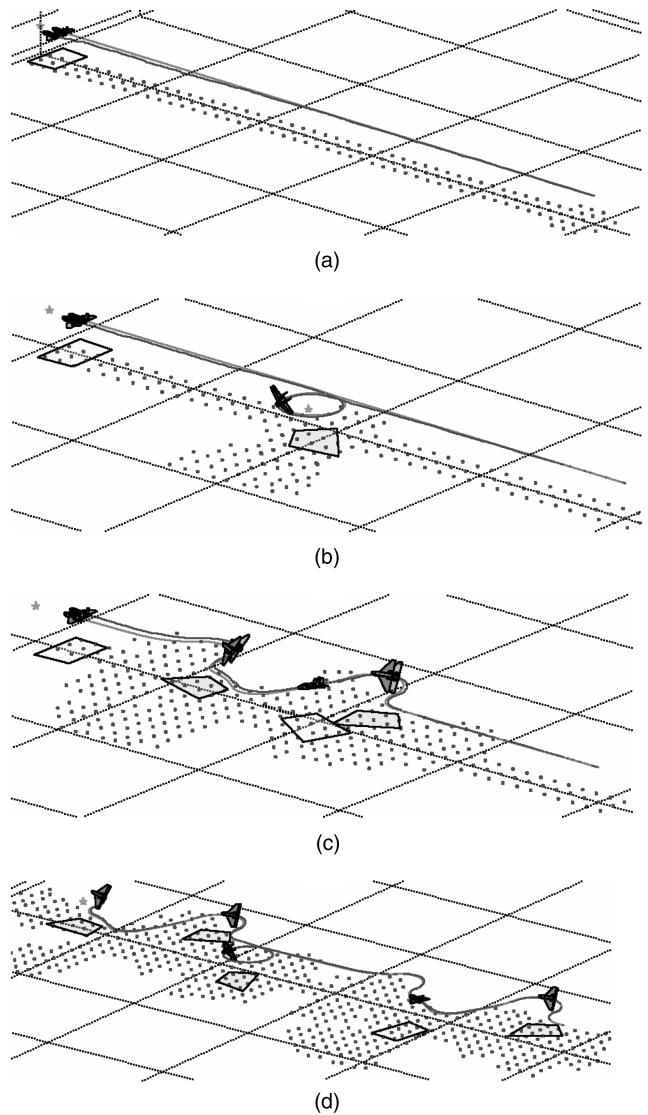


Fig. 3. Different flight paths for simulation analysis. (a) Straight and steady level. (b) Single orbit. (c) Single S-shape maneuver. (d) Combination of orbits and S-shape maneuvers. Dots in figure represent positions of features on the ground. Quadrilaterals represent downwards and sideways sensor footprints on ground.

previous sections. From the simulation model of the UAV dynamics, readings from an on-board low-cost, automotive-grade strapdown inertial measuring unit (IMU) and three simulated vision/laser sensors are made. One of the vision sensors is mounted in a downwards direction with the other two facing in each sideways direction on the aircraft body. These sensors are used to measure the range, azimuth, and elevation angles to each feature on the ground. The IMU readings are sampled at 100 Hz with average noise values of 0.05 m/s^2 for the accelerometers and $0.5^\circ/\text{s}$ for the gyros. We assume that the major component of the biases in the IMU readings remain at a constant value throughout the flight (i.e., switch on biases) and can be calibrated for on the ground before the flight. The IMU is temperature controlled and thus any other

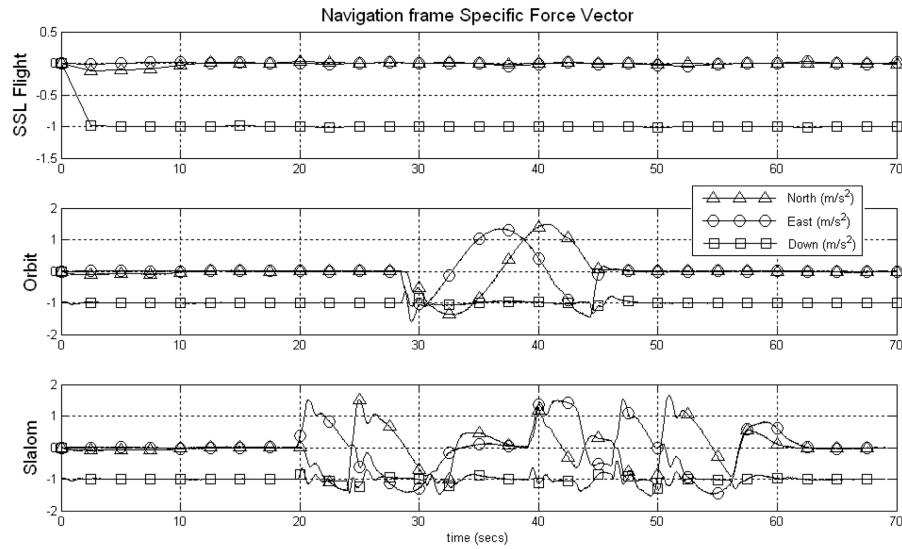


Fig. 4. Profile of components of specific force vector (f^n) acting on vehicle for different maneuver cases.

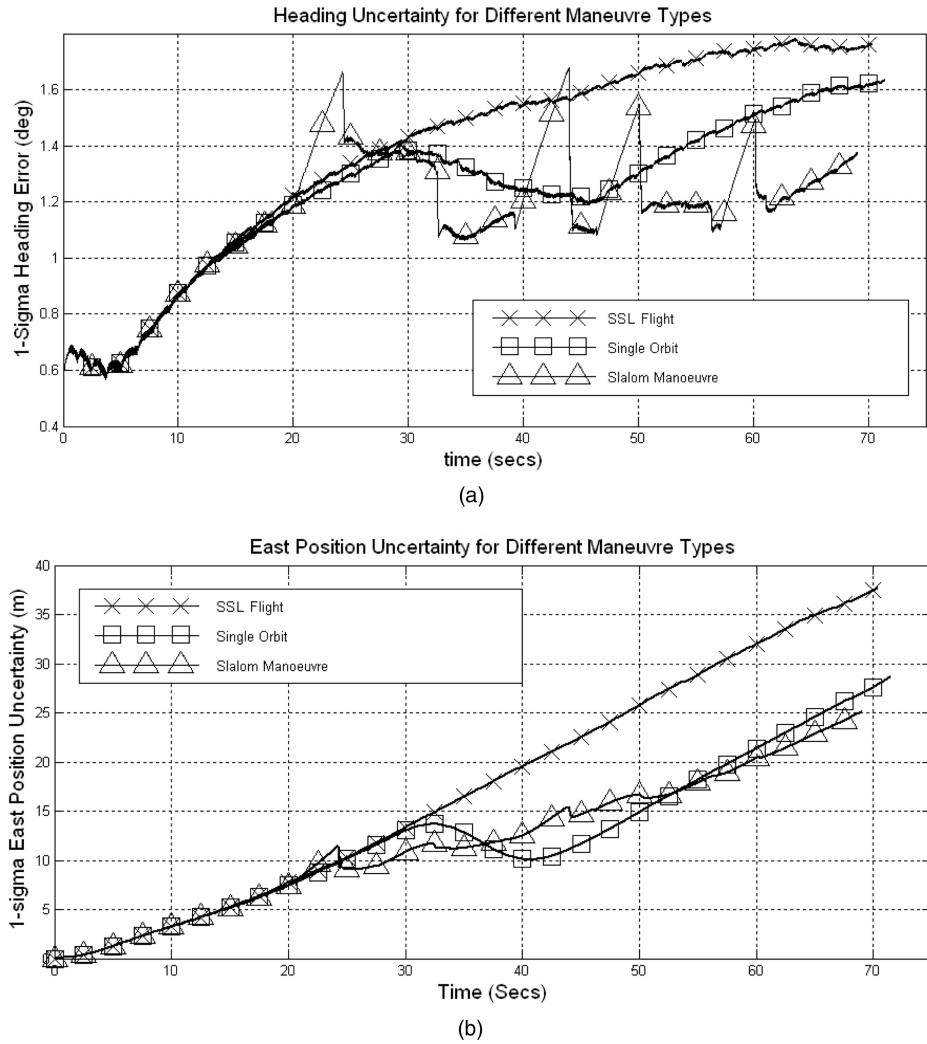


Fig. 5. 1σ SLAM estimate uncertainty for (a) heading angle (ψ) and (b) east (y-axis) vehicle position (p^n) for SSL flight, orbit and s-shape maneuvers. Heading uncertainty is reduced when orbit and s-shape maneuvers are performed due to rotation of locally unobservable mode in system.

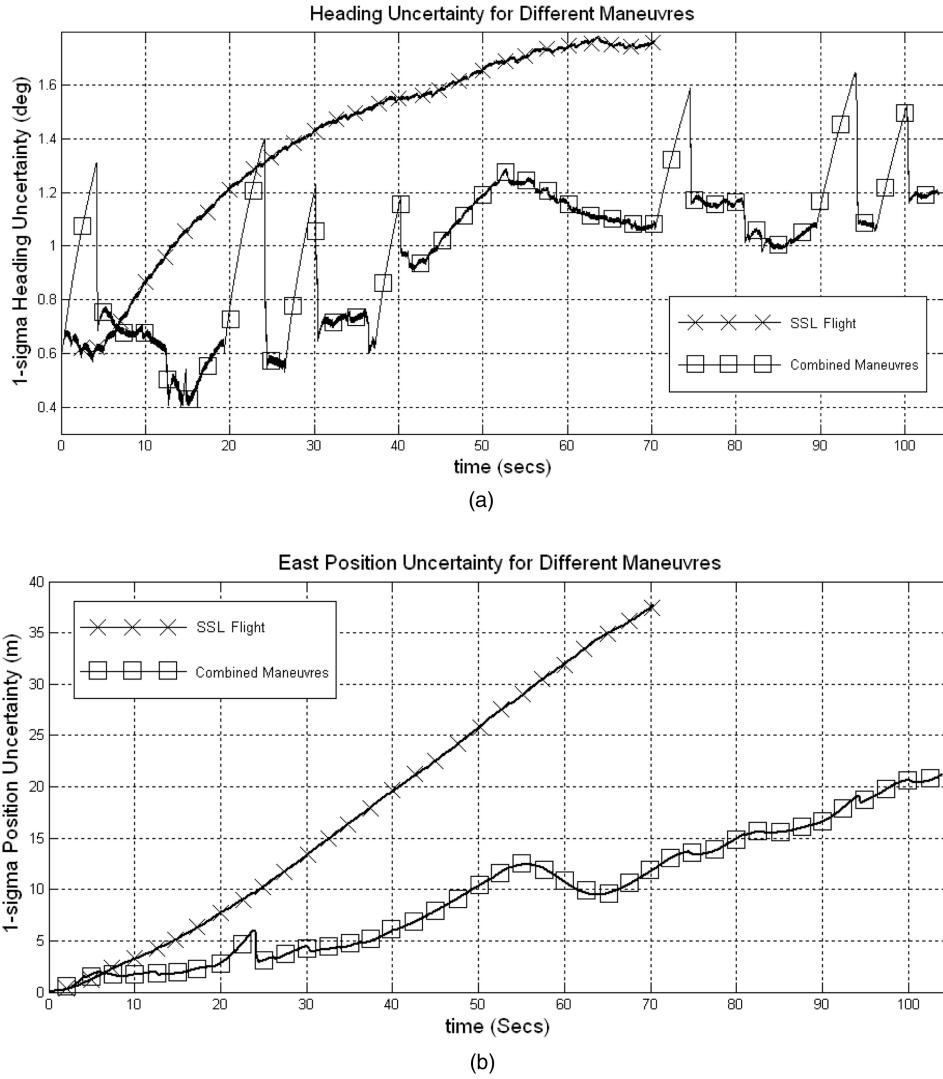


Fig. 6. 1σ SLAM estimate uncertainty for (a) heading angle (ψ) and (b) east (y-axis) vehicle position (p^y) for SSL flight and combined maneuvers. Lateral position uncertainty is reduced during the orbit and s-shape maneuvers due to reduction in heading angle uncertainty.

time-variant biases are negligibly small. If our IMU contained large time-variant, in-run biases, then online calibration would be required. This is however beyond the scope of this paper and is the topic of future work in the inertial SLAM algorithm. The only simulated errors from the IMU are therefore from noise. The vision/laser sensors run at 10 frames/sec and have a field of view of 40° across the horizontal axis of the image and a field of view of 30° across the vertical axis of the image for the downwards-facing camera and a horizontal field of view of 30° and a vertical field of view of 20° for the sideways-facing cameras. The bearing to features is measured with an average error of 0.5° and the range with an average error of 2 m. In the simulation we assume known data association of the map features and that there are no errors in the feature extraction process in order to isolate the effects in estimation accuracy for different planned paths and vehicle maneuvers.

Two different simulations are performed. In the first simulation, four different runs are performed of the vehicle travelling over a 1 km segment at an altitude of 100 m. In each different run the vehicle performs different classes of maneuvers on the way to the destination. Four different flight trajectories are analysed. The first is a simple straight and level flight segment over the 1 km distance (Fig. 3(a)). The second trajectory is a straight and level flight segment until the midpoint of the trajectory is reached followed by a complete 360° orbit of radius 50m, followed by straight and level flight to the destination (Fig. 3(b)). During the second trajectory the vehicle banks at 60° deg thus pointing one of the sideways-looking cameras towards the ground. The third trajectory contains an S-shape maneuver (Fig. 3(c)). In the S-shape maneuver, the vehicle banks right then left then right following a single S-shape. The vehicle banks at 60° deg during the turns,

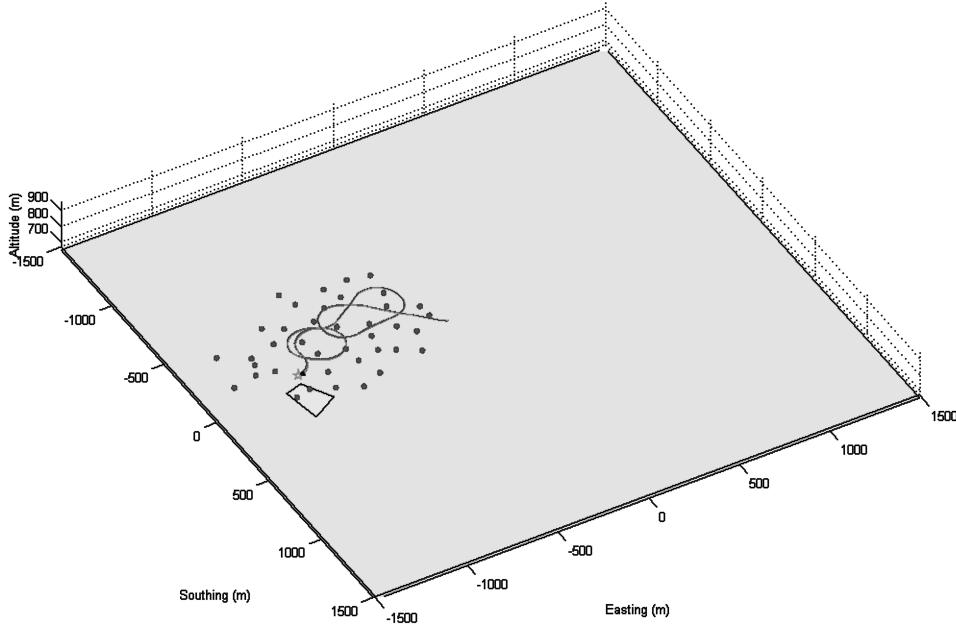


Fig. 7. Path planning results: Vehicle trajectory sequence after 60 s of flight.

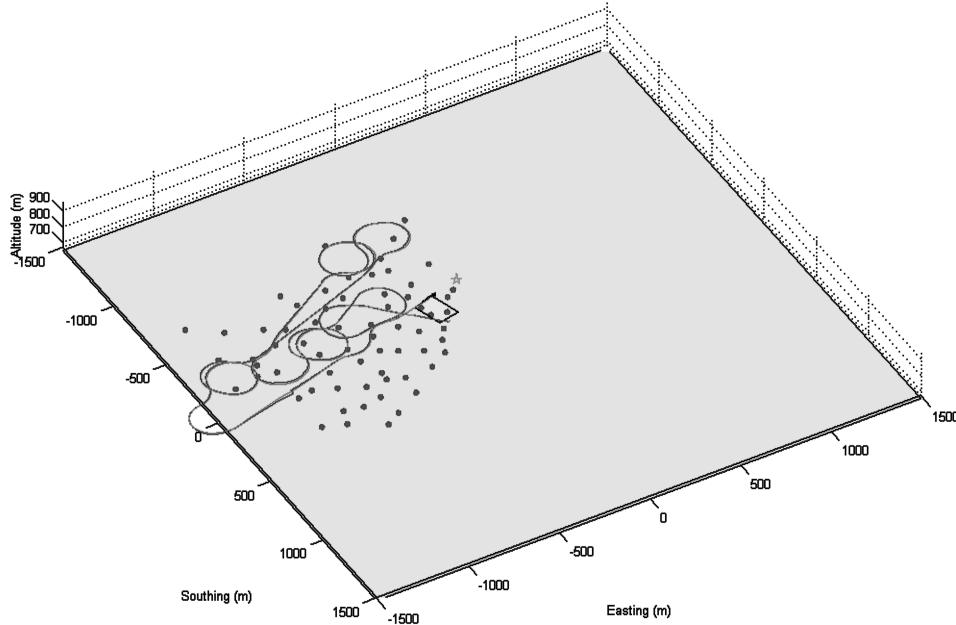


Fig. 8. Path planning results: Vehicle trajectory sequence after 340 s of flight.

where the sideways-looking camera points towards the ground. The fourth trajectory is a combination of the orbit and S-shape maneuvers (Fig. 3(d)). The vehicle performs one S-shape trajectory followed by an orbit and then a second S-shape trajectory. For all trajectories the features are dispersed across the terrain with an average density of $\rho_f = 1$ features/100 m².

In the second simulation, the UAV is given the task of building up a feature map of an unexplored region on the ground of given size. The vehicle uses the online path planning algorithms developed in Section IV in order to plan the mapping sequence while maintaining an acceptable level of localisation

estimate accuracy. In this scenario, the features are dispersed across the terrain with an average density of $\rho_f = 0.0025$ features/km².

VI. RESULTS

A. Observability/Maneuuvre Comparison Results

In this section we present the results of the SLAM estimate performance for the different maneuvers shown in Fig. 3. Fig. 4 illustrates the profiles of the components of the specific force vector (\mathbf{f}'') acting on the vehicle for each of the three classes of maneuvers

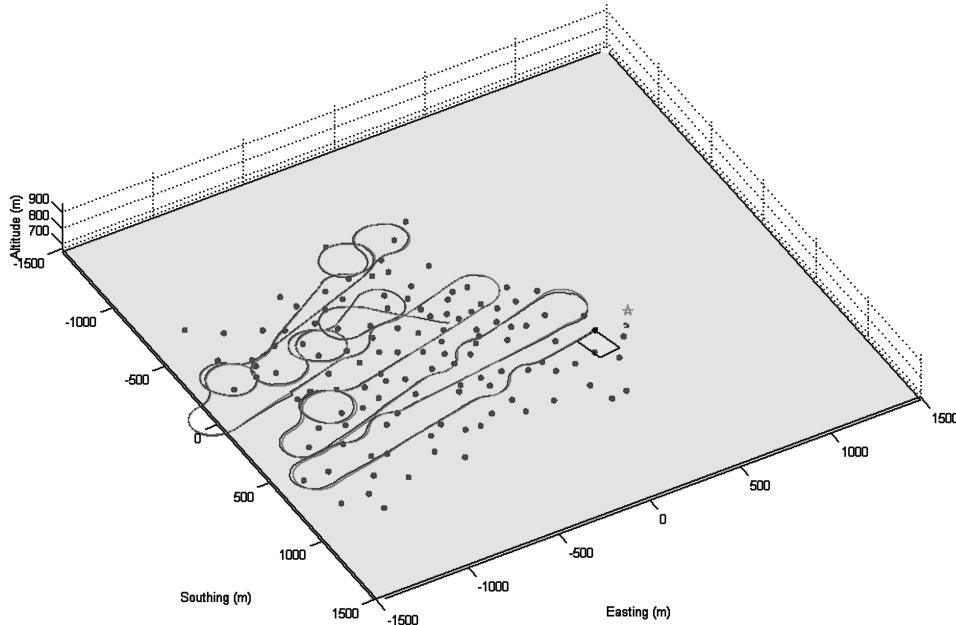


Fig. 9. Path planning results: Vehicle trajectory sequence after 630 s of flight.

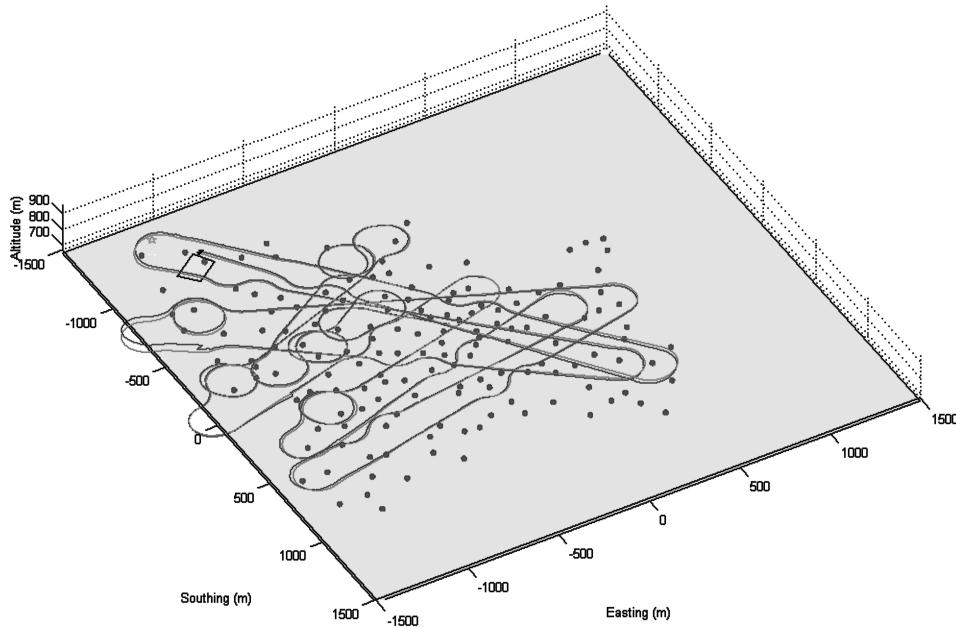


Fig. 10. Path planning results: Vehicle trajectory sequence after 1090 s of flight.

considered. We can see for the SSL flight case that the vector direction remains almost constant, pointing in the vertical direction. For both the orbit and s-shape maneuver cases the components of the vector rotate in the navigation frame directions during the maneuver.

Fig. 5 shows the uncertainty (1σ) of the SLAM estimates of the vehicle's east position and heading (yaw) angle for the three single maneuver cases demonstrated. For the SSL flight case the uncertainty in the heading angle continues to rise throughout the flight due to the constant direction of the instantaneously unobservable mode in the space of the vehicle attitude error states. It can be seen

that around the 30 s mark (the time at which each of the slalom and orbit maneuvers begin) that the vehicle's heading uncertainty for the orbit and s-shape maneuvers begins to drop corresponding to the change in direction of the specific force vector. For the orbit and s-shape cases it can be seen that the growth in the uncertainty in position estimates is reduced when compared with the SSL flight case due to the increase in heading accuracy and thus the reduction in transformation errors of the specific force in the inertial navigation equations (see (3) and (4)). For the s-shape maneuver case, the small spikes corresponding to sharp growth then drop in uncertainty relate to

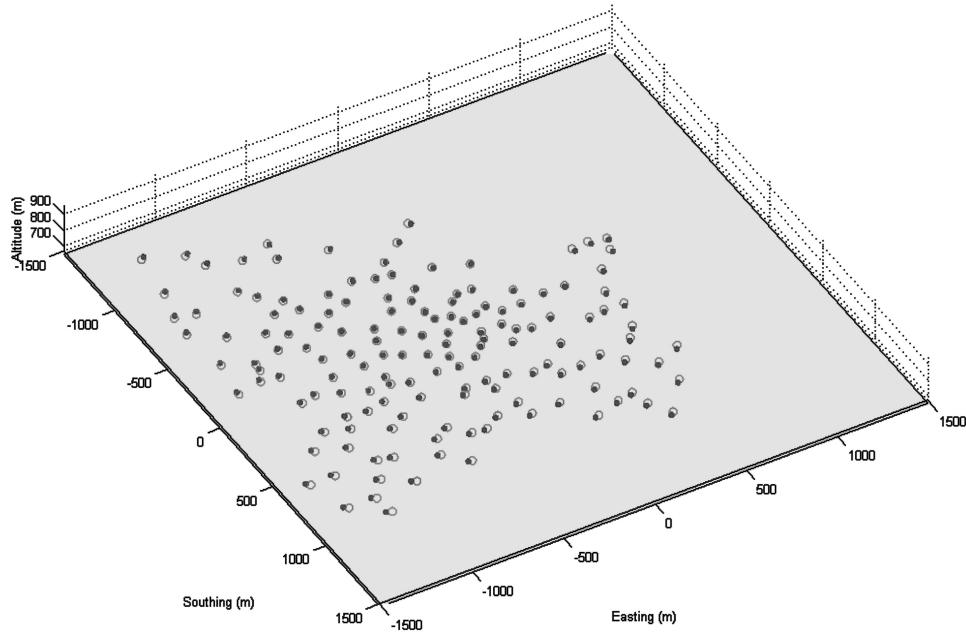


Fig. 11. Path planning results: Final feature map generated at end of simulation. Solid points represent estimated location of features in SLAM system and circles indicate simulated “true” location of features.

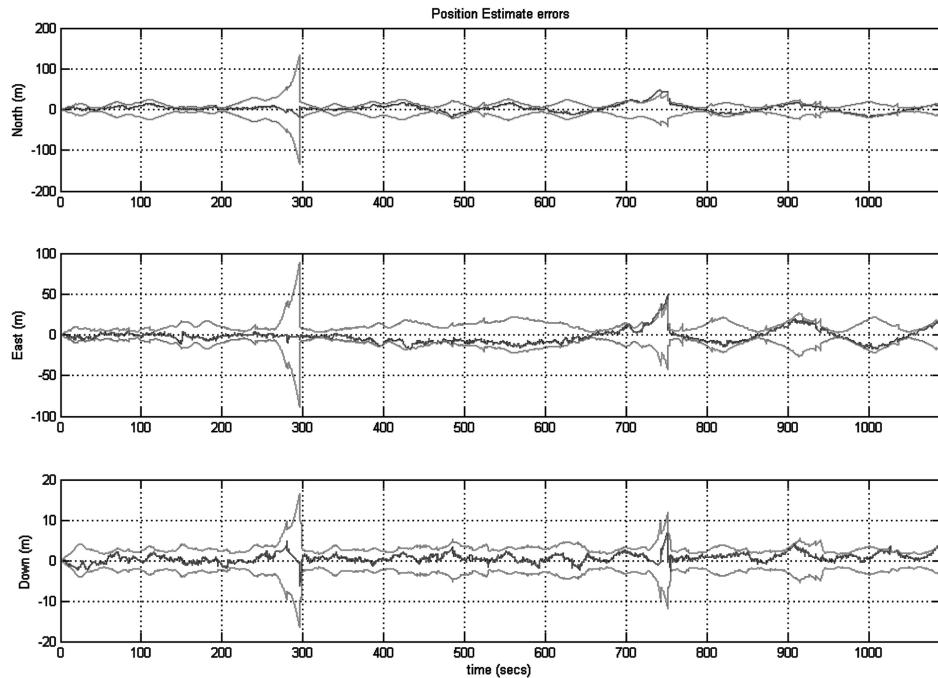


Fig. 12. Path planning results: Inertial SLAM position estimate errors and 1σ uncertainty bounds. Through maneuvers and features observations made, path planner helps to ensure that error growth in system is constrained during flight.

the on-board cameras not seeing any features when the UAV banks sharply for small periods of time. When the vehicle reobserves the features however, the uncertainty drops back close to its level before observations were lost.

Fig. 6 compares the 1σ uncertainty of the SLAM estimates for the SSL flight case and for a situation in which a combination of the orbit and s-shape maneuver is applied. It can be seen that growth in both the heading and position uncertainties are

reduced by the maneuvers. It should be noted however that by performing the maneuvers the vehicle has taken a longer period of time to reach the final destination, a necessary trade-off for the improvement in localisation estimate accuracy.

B. Online Path Planning Algorithm Results

In this section we present results of the online path planning algorithm during a localisation and mapping

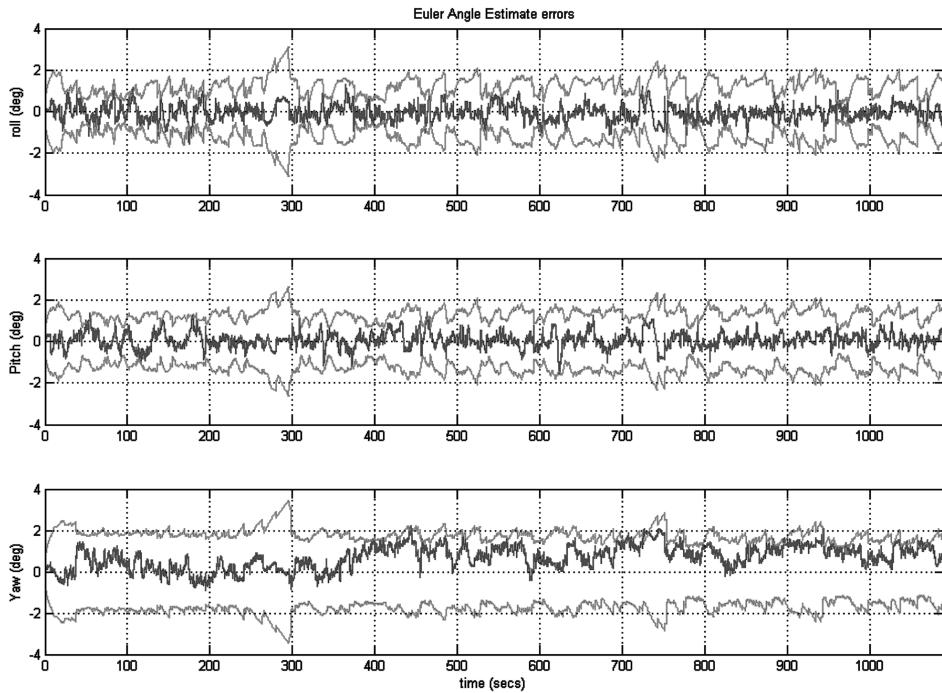


Fig. 13. Path planning results: Inertial SLAM Euler angle estimate errors and 1σ uncertainty bounds. Maneuver decision rules in path planner help to ensure that heading estimate uncertainty is maintained at low level and does not grow during flight.

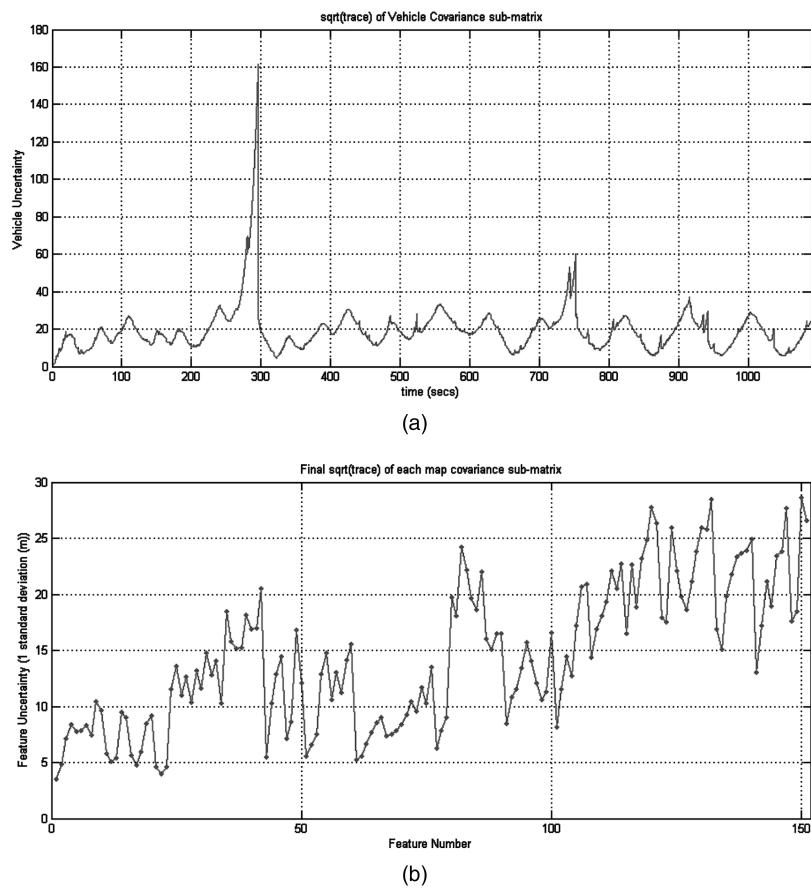


Fig. 14. Path planning results: Square root of trace of covariance matrix (a) vehicle states versus time (b) of each map state submatrix at final time. Final uncertainty in position of each map feature is below 30 m with average of about 15 m.

task. Figs. 7–10 illustrate the path taken by the vehicle over a 1100 s flight and Fig. 11 shows the final feature map generated by the SLAM algorithm. The vehicle begins by exploring the area in the direction of flight before returning to known features in order to close the loop, thus reducing the uncertainty in the vehicle pose estimates and the explored map features. The continuing pattern of the trajectory planning is to explore the nearest unexplored frontier until vehicle errors become large enough that the information gain for returning to previously seen features is greater than the information gain associated with further exploration. The choice of map location in which to fly depends mainly on reducing the uncertainty in the position of features in the map, while maintaining the uncertainty in the vehicle states to an acceptable level. At several times during the flight the decision rules in Section IVC are applied and the vehicle either orbits around the destination feature or else performs s-shape maneuvers along long periods of straight and level flight in order to maintain the heading angle uncertainty within acceptable limits.

Figs. 12–13 show the errors in the estimated position and attitude of the vehicle (compared with simulation truth) and the corresponding 1σ confidence bounds within the navigation filter. It can be seen that the actual error in the estimates of the vehicle attitude are maintained within a 1σ bound of approximately 3–4 deg, whereas the position estimates are bounded within approximately 50 m in the horizontal and about 7 m vertically, the largest errors corresponding to two small sections at about 270 and 730 s into the flight in which there were no available features in the vehicle's vicinity while exploring.

The uncertainty in the vehicle states during the flight (measured by the square root of the trace of vehicle state sub-block of the SLAM covariance matrix) and the final uncertainty in the position of each feature within the map is shown in Fig. 14. Besides the two small time segments in which the vehicle had no features to localise with, the vehicle uncertainty is maintained at a steady level and does not continue to increase over the span of flight. The final uncertainty in the position of each map feature is below 30 m with an average of approximately 15 m.

Due to the single-step lookahead in the path planner, the planned path is suboptimal and will often follow local minima. The aim however of the planner is not optimality but a feasible solution; the vehicle state uncertainty has been constrained. Multi-step lookahead could be used to plan a path such that a more accurate map could be built in less time, however the computational complexity involved in the planning and utility evaluation stages becomes infeasible due to the exponential increase in the action space.

VII. CONCLUSIONS AND FUTURE WORK

This paper has demonstrated a framework for developing intelligent guidance schemes for a UAV for improving localisation and mapping performance when operating over unknown terrain and without the use of GPS. The proposed scheme plans vehicle paths based on a combination of computed approximations to the expected mutual information gain for certain trajectories, as well as decision rules based on a qualitative knowledge of the effect maneuvers have on the observability of the system. The approach does not necessarily find the optimal path (in terms of localisation performance) however offers a practical alternative with mathematical insight to the unreasonable computational load involved in performing the full optimisation. The observability analysis of the inertial SLAM algorithm presented in this paper reveals the importance of dynamic vehicle motion in the SLAM estimation process and points towards vehicle motion behaviors that maximise localisation accuracy.

In future work, we will look at additional methods for using the insight gained through the observability analysis in inertial SLAM for online path planning. Different metrics for navigation performance such as consistency of the filter and the ability for data association will also be considered. These metrics will assist in composing additional decision rules for the vehicle actions. We will also consider how system configuration on the UAV such as the placement and pointing directions of sensors can be optimised for a particular task given the maneuver envelope available to the vehicle and the knowledge from the observability analysis presented in this paper.

REFERENCES

- [1] Braun, R. D., Wright, H. S., Croom, M. A., Levine, J. S., and Spencer, D. A.
The Mars airplane: A credible science platform.
Presented at the IEEE Aerospace Conference, Big Sky, MT, 2004.
- [2] Dissanayake, M. W. M. 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, **17**, 3 (June 2001), 229–241.
- [3] Williams, S. B., Dissanayake, G., and Durrant-Whyte, H.
Towards terrain-aided navigation for underwater robotics.
Advanced Robotics, **15**, 5 (2001), 533–550.
- [4] Kim, J. H., and Sukkarieh, S.
Airborne simultaneous localization and map building.
Presented at the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003.
- [5] Kim, J. H., and Sukkarieh, S.
Autonomous airborne navigation in unknown terrain environments.
IEEE Transactions on Aerospace and Electronic Systems, **40**, 3 (July 2004), 1031–1045.

- [6] Bar-Gill, A., Ben-Ezra, P., and Bar-Itzhack, I. Y. Improvement of terrain-aided navigation via trajectory optimisation. *IEEE Transactions on Control Systems Technology*, **2**, 4 (Dec. 1994), 336–342.
- [7] Roy, N., Burgard, W., Fox, D., and Thrun, S. Coastal navigation—Mobile robot navigation with uncertainty in dynamic environments. Presented at the IEEE/RSJ International Conference on Robotics and Automation, Detroit, MI, 1999.
- [8] Bryson, M., and Sukkarieh, S. An information-theoretic approach to autonomous navigation and guidance of an uninhabited aerial vehicle in unknown environments. IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada, 2005.
- [9] Makarenko, A., Williams, S. B., Bourgault, F., and Durrant-Whyte, H. An experiment in integrated exploration. Presented at the IEEE International Conference on Intelligent Robots and Systems, Switzerland, 2002.
- [10] Bourgault, F., Makarenko, A., Williams S. B., Grocholsky B., and Durrant-Whyte, H. Information based adaptive robotic exploration. Presented at the IEEE International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 2002.
- [11] Sim, R., Dudek, G., and Roy, N. Online control policy optimization for minimizing map uncertainty during exploration. Presented at the IEEE International Conference on Robotics and Automation, New Orleans, LA, 2004.
- [12] Huang, S., Kwok, N. M., Dissanayake, G., and Ha, Q. P. Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity. Presented at the IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005.
- [13] Lui, Y., and Thrun, S. Results for outdoor-SLAM using sparse extended information filters. Presented at the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003.
- [14] Goshen-Meskin, D., and Bar-Itzhack, I. Y. Observability analysis of piece-wise constant systems—Part 1: Theory,” *IEEE Transactions on Aerospace and Electronic Systems*, **28**, 4 (Oct. 1992), 1056–1067.
- [15] Goshen-Meskin, D., and Bar-Itzhack, I. Y. Observability analysis of piece-wise constant systems—Part 2: Application to inertial navigation in-flight alignment. *IEEE Transactions on Aerospace and Electronic Systems*, **28**, 4 (Oct. 1992), 1068–1075.
- [16] Rhee, I., Abdel-Hafez, M. F., and Speyer, J. L. Observability of an integrated GPS/INS during maneuvers. *IEEE Transactions on Aerospace and Electronic Systems*, **40**, 2 (Apr. 2004), 526–534.
- [17] Hong, S., Lee, M. H., Chun, H., Kwon, S., and Speyer, J. Observability of error states in GPS/INS integration. *IEEE Transactions on Vehicular Technology*, **54**, 2 (Mar. 2005), 731–743.
- [18] Kim, J. H., and Sukkarieh, S. Improving the real-time efficiency of inertial SLAM and understanding its observability. Presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, 2004.
- [19] Gelb, A. *Applied Optimal Estimation*. Cambridge: MIT Press, 1974.
- [20] Titterton, D., and Weston, J. *Strapdown Inertial Navigation Technology*. London: Peter Peregrinus Ltd., 1997.
- [21] Lowe, D. G. Distinctive Image features from scale-invariant keypoints. *International Journal of Computer Vision*, **60**, 2 (2004), 91–110.
- [22] Nixon, M. S., and Aguado, A. S. *Feature Extraction and Image Processing*. Oxford: Butterworth Heinemann/Newnes, 2001.
- [23] Bar-Shalom, Y., and Fortmann, T. E. *Tracking and Data Association*. New York: Academic Press, 1988.
- [24] Bryson, M., and Sukkarieh, S. Bearing-only SLAM for an airborne vehicle. Presented at the Australasian Conference on Robotics and Automation, Sydney, Australia, 2005. Available at <http://www.araa.asn.au/acra/acra2005/papers/bryson.pdf>.
- [25] Sarachik, P. E. *Principles of Linear Systems*. Cambridge: Cambridge University Press, 1997.
- [26] Cole, D. T., Sukkarieh, S., Goktogan, A. H., Stone, H., and Hardwick-Jones, R. The development of a real-time modular architecture for the control of UAV teams. Presented at the 5th International Conference on Field and Service Robotics, Port Stevens, Australia, July, 2005.



Mitch Bryson received his B.Eng. degree in aeronautical and aerospace engineering from the University of Sydney in 2003. He is currently a Ph.D. candidate at the Australian Centre for Field Robotics at the University of Sydney.

His current research areas include simultaneous localisation and mapping (SLAM), unmanned aerial vehicle (UAV) navigation and autonomous exploration strategies for UAVs.



Salah Sukkarieh obtained his Ph.D. in 2000 at the University of Sydney, Australia.

Dr. Sukkarieh is a senior lecturer at the University of Sydney in the School of Aerospace, Mechanical and Mechatronic Engineering. He is also the program manager for aerospace at the Australian Centre for Field Robotics in the Centre for Autonomous Systems. His current research areas include inertial SLAM, cooperative control, and systems of systems design.