Abstract—This paper presents an intelligent, on-line guidance scheme for maximising vehicle navigation and map information for an uninhabited aerial vehicle while operating over unknown terrain. The tasks of localisation and mapping are performed concurrently using the well known extended Kalman filter implementation of the Simultaneous Localisation And Mapping (SLAM) algorithm. We analyse the effect different vehicle maneuvers and actions have on two main navigation performance metrics: the entropic mutual information gain and the observability of the navigation system estimates. A guidance scheme is proposed that uses a combination of decision rules based on the performance metrics in order to maximise navigation estimate accuracy. Results are presented using a high-fidelity six-degree of freedom simulation of an aerial vehicle using inertial navigation and a range/bearing sensor.

Index Terms—Navigation, Mapping, SLAM, Autonomous Vehicles, Observability.

I. INTRODUCTION

There has been much recent interest in the autonomous systems research community for the use of airborne platforms in applications such as surveillance, mapping and even planetary exploration [1], where a higher degree of maneuverability, high-speed/fast response time and large area coverage ability of aerial vehicles provides a significant advantage over ground-based vehicles. In these types of applications, commonly used external navigational aids such as the Global Positioning System (GPS) or a priori information such as a map or terrain data may not be available to aid in localisation of the vehicle. Un-aided Inertial Navigation Systems (INS) could be used, however the high-cost, weight and power requirements of the necessary unit plus the limited operational time due to the eventual growth in system errors is prohibitive to many projects. Instead, a low-cost INS can be used along with a terrain sensor as the basis for an implementation of Simultaneous Localisation And Mapping (SLAM), where no initial terrain data is required. There are several existing implementations of SLAM using land [2], underwater [3] and aerial [4] vehicles for applications with similar requirements.

In SLAM the accuracy of the navigation estimates are inherently coupled to the actions taken by the vehicle in terms of the vehicle motion and of the observations of different features at different times. Thus when operating in large-scale environments, the uncertainty in vehicle pose estimates may become large given that the vehicle does not return to known features or perform appropriate vehicle motions. Large values of uncertainty in the estimates effects the stability of the aerial vehicle, where inaccurate feedback to the on-board control system can result in dangerous platform motions leading to the catastrophic failure of the vehicle. These problems may be less apparent at least for indoor ground based systems where the speed of the vehicle is relatively low and the roll and pitch angles of the platform remain relatively small.

In this paper we begin to focus on the issue of ensuring integrity in the navigation of a UAV by controlling the actions taken by the vehicle. By integrity we refer to both the stability of the navigation algorithm and the ensured quality of localisation estimates necessary for the control of the vehicle. We present an autonomous guidance scheme for a UAV that specifies vehicle actions while exploring over unknown terrain. The actions of the vehicle are specified in terms of groups of behaviors or when to explore an unexplored area or to re-observe a known region of the map such that certain performance measures of the quality of navigation estimates are maximised. The approach simultaneously ensures a growth in the size of the explored area, a key requirement in many of the applications stated above, and a necessary part of navigation by increasing the known regions of the terrain and thus expanding the area of operation available to the vehicle.

It is fair to say that given unlimited computational power one could simply use optimisation techniques in order to evaluate the best control actions given the expected measures of uncertainty in the filter. However in the case of SLAM running on a aerial vehicle, there are three main impracticalities to computing the optimal solution. Firstly, there is a need for high-rate control due to the fast-dynamics of the platform motion. Secondly the space of available vehicle actions to optimise is large when for example compared to ground vehicles, as the vehicle is not restricted to 3-DoF motion. Thirdly the size of the state space to optimise is related to the number of features in the map and is hence very large.

We consider two metrics of performance: firstly, the entropic Mutual Information (MI) gain and secondly the Observability of the estimated state vector. The combination of performance metrics provides both a quantitative and qualitative analysis of navigation system performance,
a property that is desirable from the view point of the designer of the navigation system. In our control architecture we use a combination of behavior-based control and sub-optimal approximations of the performance metrics in order to plan vehicle actions.

Intelligent guidance schemes for improving navigation system performance have been shown in the past. In [11], the authors present a method for generating optimal a priori trajectories over a known terrain map for an airborne vehicle using a terrain-aided INS by using the entropy of the terrain as a cost function. A similar idea was proposed for an indoor robot in [12] where a map of the navigation information contained along the walls of a room is constructed from a known map of the environment. In these applications prior information about the environment is present thus removing the need for real-time online planning.

Related methods such as exploration strategies in unknown environments where the focus is on building accurate maps have also been examined: [7] demonstrates an autonomous exploration strategy for an indoor robot based on the geometric and spatial qualities of the known sections of the map. The strategies presented in [8],[9],[10] use several utilities relating to map information, area coverage and ability of localisation as the basis for the control of indoor robots. In these papers the focus is on the task of building an accurate environment map, given no prior map information and no external localisation aids.

This paper provides two contributions: Firstly we provide insight into the relationship between the performance of the SLAM navigation system and vehicle actions for a UAV. Secondly we develop a strategy for controlling the vehicle based on navigation system performance by building on ideas in ground-based exploration strategies.

In section II we present the SLAM algorithms used on the aerial vehicle. Section III examines the different metrics used to evaluate control actions taken by the vehicle. Section IV describes the control architecture on board the vehicle and the use of the metrics in section III. A 6-DoF UAV mapping scenario and the setup of a high-fidelity simulation is described in section V with initial simulation results presented in section VI. Conclusions and future work are covered in section VII.

II. THE INERTIAL SLAM ALGORITHM

In this section we describe the SLAM algorithms used on board the UAV. The inertial SLAM algorithm can be cast in two different forms: the direct form in which the estimated states are the nominal values of vehicle pose and map feature positions and the indirect form in which the estimated state is the error in computed nominal values. The direct inertial SLAM algorithm, as shown in [4], is formulated using an Extended Kalman Filter (EKF) in which landmark-feature locations and the vehicle’s position, velocity and attitude are estimated using relative observations between the vehicle and each landmark.

1) Process Model: In the direct form, the estimated state vector \( \hat{x}(k) \) contains the three-dimensional vehicle position \( (p^n, v^n) \), velocity \( (v^n) \) and Euler angles \( (\Psi^n = [\phi, \theta, \psi]) \) and the N landmark-feature locations \( (m^i_n) \) (all in the navigation frame \( n \)) where \( i = 1, \ldots, N \). The state estimate \( \hat{x}(k) \) is predicted forward in time from \( \hat{x}(k-1) \) via the process model:

\[
\hat{x}(k) = \mathbf{F}(\hat{x}(k-1), u(k), k) + w(k)
\]

where \( \mathbf{F}(...) \) is the non-linear state transition function at time \( k \), \( u(k) \) is the system input at time \( k \) and \( w(k) \) is uncorrelated, zero-mean vehicle process noise errors of covariance \( \mathbf{Q}(k) \). The vehicle process model uses the six-degree of freedom equations of inertial navigation to predict the position, velocity and attitude of the vehicle. An inertial-frame mechanization is used where position, velocity and attitude are found via:

\[
\begin{bmatrix}
  p^n(k) \\
  v^n(k) \\
  \Psi^n(k)
\end{bmatrix} = \begin{bmatrix}
  p^n(k-1) + v^n(k)\Delta t \\
  v^n(k-1) + [C^n_b(k-1)F^n_b(k) + g^n]\Delta t \\
  \Psi^n(k-1) + E^n_b(k-1)\omega^b(k)\Delta t
\end{bmatrix}
\]

where \( F^b \) and \( \omega^b \) are the body-frame referenced vehicle accelerations and rotation rates which are provided by inertial sensors on the vehicle and \( g^n \) is the acceleration due to gravity. The direction cosine matrix \( C^n_b \) and rotation rate transformation matrix \( E^n_b \) between the body and navigation frames are given by:

\[
C^n_b = \begin{bmatrix}
  c_{\phi}c_{\theta} & c_{\phi}s_{\theta} - s_{\phi}c_{\psi} & s_{\phi}c_{\psi} + c_{\phi}s_{\theta} \\
  s_{\phi}c_{\theta} & c_{\phi}c_{\psi} + s_{\phi}s_{\theta} & c_{\phi}s_{\psi} - s_{\phi}c_{\theta} \\
  -s_{\theta} & c_{\theta} & 0
\end{bmatrix}
\]

\[
E^n_b = \begin{bmatrix}
  1 & s_{\phi}l_{\theta} & c_{\phi}l_{\theta} \\
  0 & c_{\phi} & -s_{\phi} \\
  0 & s_{\phi}sec_{\theta} & c_{\phi}sec_{\theta}
\end{bmatrix}
\]

where \( s(\cdot) \), \( c(\cdot) \) and \( t(\cdot) \) represent \( \sin(\cdot) \), \( \cos(\cdot) \) and \( \tan(\cdot) \) respectively. Landmark-feature locations are assumed to be stationary and thus the process model for the position of the \( i^{th} \) landmark is given as:

\[
m^i_n(k) = m^i_n(k-1)
\]

where \( i = 1, \ldots, N \), \( N \) is the number of landmarks.

2) Observation Model: An on-board range/bearing sensor makes relative observations \( z_i(k) \) from the vehicle to an observed landmark. The observations are related to the estimated states via:

\[
z_i(k) = \mathbf{H}_i(p^n(k), \Psi^n(k), m^i_n(k), k) + v(k)
\]

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

\[
z_i(k) = \begin{bmatrix}
  R_i \\
  \varphi_i \\
  \theta_i
\end{bmatrix} = \begin{bmatrix}
  \sqrt{x_i^2 + y_i^2} \\
  tan^{-1} \frac{y_i}{x_i} \\
  tan^{-1} \left( \frac{z_i}{\sqrt{x_i^2+y_i^2}} \right)
\end{bmatrix}
\]
where \([R_i, \varphi_i, \theta_i]\) are the range, azimuth and elevation angles to the feature and \(x_i, y_i, z_i\) are the cartesian co-ordinates of \(p_{mis}^s\), the relative position of the landmark w.r.t the sensor, measured in the sensor frame. \(p_{mis}^s\) is given by:

\[
p_{mis}^s = C_b^s C_n^b m_i^n - p^n - C_n^b p_{sb}^b
\]

where \(C_b^s\) is the transformation matrix from the body frame to sensor frame and \(p_{sb}^b\) is the sensor offset from the vehicle centre of mass, measured in the body frame.

3) Estimation Process: The estimation process is recursive and is broken into three steps:

**Prediction:** The vehicle position, velocity and attitude are predicted forward in time using (1) and (2) with data supplied by the inertial sensors. The state covariance \(P\) is propagated forward via:

\[
P(k|k-1) = \nabla F_x(k)P(k-1|k-1)\nabla F_x^T(k) + \nabla F_w(k)Q\nabla F_w^T(k)
\]

where \(\nabla F_x\) and \(\nabla F_w\) are the jacobians of the state transition function w.r.t the state vector \(\hat{x}(k)\) and the noise input \(w(k)\) respectively.

**Update:** When an observation of a previously observed landmark is made, the state estimate is updated via:

\[
\hat{x}(k|k) = \hat{x}(k|k-1) + \mathbf{W}(k)\nu(k)
\]

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

\[
\nu(k) = z_i(k) - \mathbf{H}_i(\hat{x}(k|k-1))
\]

\[
\mathbf{W}(k) = P(k|k-1)\mathbf{H}_i^T(k)S^{-1}(k)
\]

\[
S(k) = \nabla H_x(k)P(k|k-1)\nabla H_x^T(k) + \mathbf{R}(k)
\]

where \(\nabla H_x(k)\) is the jacobian of the observation function w.r.t the predicted state vector \(\hat{x}(k|k-1)\). The state covariance \(P(k|k)\) after the observation is updated via:

\[
P(k|k) = P(k|k-1) - \mathbf{W}(k)S(k)\mathbf{W}^T(k)
\]

**Landmark Initialization:** When a new landmark is observed, its position is calculated using the initialization function \(G_1[\hat{x}(k), G_2(z_i(k))]\) which is given as:

\[
G_1 \rightarrow m_i^n = p^n + C_n^b p_{sb}^b + C_n^b C_b^s p_{mis}^s
\]

\[
G_2 \rightarrow p_{mis}^s = \begin{bmatrix}
R_i \cos(\varphi_i) \cos(\theta_i) \\
R_i \sin(\varphi_i) \cos(\theta_i) \\
R_i \sin(\theta_i)
\end{bmatrix}
\]

The state vector and covariance are then augmented via:

\[
\hat{x}_{aug}(k) = \begin{bmatrix}
\hat{x}(k) \\
m_i^n(k)
\end{bmatrix}
\]

\[
P_{aug}(k) = \begin{bmatrix}
I & 0 \\
\nabla G_x & \nabla G_z
\end{bmatrix} \begin{bmatrix}
P(k) & 0 \\
0 & \mathbf{R}(k)
\end{bmatrix}^T
\]

where \(\nabla G_x\) and \(\nabla G_z\) are the jacobians of the initialization w.r.t the state estimate \(\hat{x}(k)\) and the observation \(z_i(k)\) respectively.

### A. Indirect Inertial SLAM

As shown in [5], we can express the SLAM algorithms in indirect form where the estimated state becomes the vehicle position error \((\delta p^n)\), vehicle velocity error \((\delta v^n)\), vehicle misalignment angles \((\delta \Psi^n)\) and the feature position errors \((\delta m_i^n)\). In this form the inertial navigation equations and the addition of new features to the map can be run separately to the EKF process, where the EKF provides periodic updates to correct the errors that are present in the INS and the feature map. The error states are defined as the difference between the true system state and the estimated state (as in the INS and map). The estimated states are predicted forward in the Kalman filter prediction stage via:

\[
\delta p^n(k) = \delta p^n(k-1) + \delta v^n(k)\Delta t
\]

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

\[
\delta \Psi^n(k) = \delta \Psi^n(k-1) + C_n^b\delta \omega^b
\]

\[
\delta m_i^n(k) = \delta m_i^n(k-1) + \delta f^b
\]

where \(\delta f^b\) and \(\delta \omega^b\) are the errors in the accelerometer and gyro readings respectively (modelled as Gaussian white noise processes) and \([\times f^n]\) is the skew-symmetric matrix of the specific force vector in the navigation frame. In the update stage, the observation is \(\delta p_{mis}^s\), the difference between the observation of range and bearing to the feature, transformed into cartesian co-ordinates and the estimated value \((\hat{p}_{mis}^s)\) as computed from the existing map.

\[
\delta p_{mis}^s = G_2(z_i) - \hat{p}_{mis}^s
\]

The observation is related to the vehicle error states via:

\[
\delta p_{mis}^s = C_b^s C_n^b [\delta m_i^n - \delta p^n + [\times f^n] \delta \Psi^n + ...]
\]

\[
[\times C_b^s C_n^b m_i^n - \hat{p}^n - C_n^b p_{sb}^b] \delta \Psi^n
\]

where it is assumed that the body to sensor transformation \(C_b^s\) and the sensor offset \(p_{sb}^b\) are known with negligible error.

The indirect and direct forms of the algorithm are numerically equivalent to one another. There are however two advantages to the indirect form. Firstly the prediction cycle computational load is reduced in the filter, as the filter operation can be performed separately from the inertial navigation system. Secondly, the process and observation models in the indirect filter can be represented as piecewise linear systems thus simplifying their information and observability analysis (as will be seen in the next section).

### III. NAVIGATION PERFORMANCE METRICS

In this section we analyse several navigation performance metrics of the inertial SLAM algorithms that can be used to compare the utility of different vehicle actions.

**A. Entropy and Mutual Information Gain**

The entropy \(H(x)\) of a multivariate gaussian probability distribution can be calculated from its covariance matrix \(P\) as follows:

\[
H(x) = \frac{1}{2} \text{log}(2\pi e)^n ||P||
\]
Entropy is a measure of the compactness of a distribution and thus the informativeness. To measure the utility of taking an action \( a \) which will be to move to a certain observation, making observations along the way, we use the mutual information gain \( I[x,a] \) which is defined as the difference between the entropies of the distributions about the estimated states before and after taking the action:

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

(26)

where \( \mathbf{P}(x) \) is the prior covariance and \( \mathbf{P}(x|a) \) is the covariance subsequent to travelling to the destination. The mutual information gain is a number which is negative for a loss and positive for a gain in information.

Entropy is useful as a metric for deciding between actions as it is a single scalar value encompassing the information over the entire state space. The disadvantage with using entropy is in evaluating the expected covariance over the whole state vector for each possible action, a computationally expensive exercise.

### B. Observability

A system state is defined as observable if given the state transition and observation models of the system and observations relating to the system state through the observation model from some initial time \( t_0 \), that we can evaluate the state of the system at the initial time \( t_0 \). When a system is fully observable, the lower bound of the error in our 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. In [5] an observability analysis of the indirect form of the inertial SLAM equations was presented with use of the Stripped Observability Matrix [6]:

\[
\mathbf{Q}_{SOM}(k) = \begin{bmatrix}
\mathbf{Q}_1 \\
\mathbf{Q}_2 \\
\vdots \\
\mathbf{Q}_k
\end{bmatrix}
\]

(27)

\[
\mathbf{Q}_k = [\mathbf{H}_k^T(\mathbf{H}_k \mathbf{F}_k)^T (\mathbf{H}_k \mathbf{F}_k)^T \cdots (\mathbf{H}_k \mathbf{F}_k^{n-1})^T]^T
\]

(28)

where \( \mathbf{F}_k \) and \( \mathbf{H}_k \) are the \( k \)th time segment state transition matrix and observation models given in the Inertial SLAM case by rearranging equations (19-22) and (23).

It was shown that for a total of 9 vehicle states plus 3N map states (where N is the number of features), there were 4 unobservable states given a single time segment, regardless of the number of map features considered. If the observability was considered over two time segments in which the specific force vector of the vehicle in the navigation frame \( f^n \) changed direction between segments, then one more state became observable resulting in a total of 3 unobservable states. The addition of extra time segments does not increase the number of observable modes.

We are interested in the direction of the unobservable modes and the effect the maneuvers and motion of the vehicle have on the observability.

1) **Simplification of Inertial SLAM Observation Model:** Consider a simplification to the indirect SLAM observation model that does not change the observability of the system. We will assume that the sensor frame \( s \) lies along the body axis \( b \) (i.e. \( C_s^b = \mathbf{I}_{3x3} \)), and that the sensor offset from the body axis \( (\mathbf{p}_{sb}^b) \) is zero. The SLAM observation model in equation (24) can now be simplified to:

\[
\delta \mathbf{p}^n_{mb} = \delta m^n - \delta \mathbf{p}^n + [\times \mathbf{p}_{mc}^b] \delta \Psi^n
\]

(29)

where \( [\times \mathbf{p}_{mc}^b] \) is the skew symmetric matrix of the estimate of the relative position between the feature and vehicle positions, in the navigation frame. The change to the observation model assists in the analysis of the unobservable modes.

2) **Direction of the Unobservable Modes:** In order to determine the direction in the state space of the unobservable modes we can evaluate the matrix:

\[
\mathbf{S} = \mathbf{Q}^T \mathbf{Q}
\]

(30)

\( \mathbf{S} \) is a dilution of precision measure of the observability matrix and thus the eigenvectors corresponding to the zero eigenvalues of \( \mathbf{S} \) are the unobservable modes of the system.

Over two time segments where the acceleration vector changes direction and where two map features are considered, the SOM of the system becomes:

\[
\mathbf{Q}_{SOM} = \begin{bmatrix}
-I_{3x3} & 0 & \times \mathbf{f}_{m11}^n \\
-I_{3x3} & 0 & \times \mathbf{f}_{m21}^n \\
0 & -I_{3x3} & 0 \\
0 & -I_{3x3} & 0 \\
0 & 0 & -[\times \mathbf{f}_b^n] \\
0 & 0 & -[\times \mathbf{f}_b^n] \\
-I_{3x3} & 0 & \times \mathbf{f}_{m12}^n \\
-I_{3x3} & 0 & \times \mathbf{f}_{m22}^n \\
0 & -I_{3x3} & 0 \\
0 & -I_{3x3} & 0 \\
0 & 0 & -[\times \mathbf{f}_b^n] \\
0 & 0 & -[\times \mathbf{f}_b^n]
\end{bmatrix}
\]

(31)

the eigenvectors of the three zero eigenvalues of \( \mathbf{S} \) are:

\[
x_{\text{unobs}} = [I_{3x3}, 0, 0, I_{3x3}, I_{3x3}, \ldots]
\]

\[
= \delta \mathbf{p}^n + \delta m_1^n + \delta m_2^n + \ldots
\]

(32)

Thus the unobservable modes are the three components of the vector sum of the errors in the position of the vehicle and the errors in the position of each feature. An important consequence of this analysis is that the unobservable modes do not lie across the errors in velocity and attitude of the vehicle and thus these states are completely observable.

3) **Augmenting the State Vector for Complete Observability:** Consider the change to the state vector in which we consider the error in the relative position of each feature w.r.t the vehicle position (i.e. \((\delta m^n - \delta \mathbf{p}^n)\)) rather than the separate global vehicle \( (\delta \mathbf{p}^n) \) and map feature \( (\delta m^n) \) error states. We maintain the errors in the vehicle velocity \( (\delta \mathbf{v}) \)
and attitude ($\delta \Psi^n$) in the state vector as before. The time dynamics and observation model of the new state is:

$$
(\delta m^n_i - \delta p^n) = -\delta v^n
\quad (33)
$$

$$
\delta p_{mb}^n = (\delta m^n_i - \delta p^n) + [\times \hat{r}_{mb}^n] \delta \Psi^n
\quad (34)
$$

The state vector now becomes:

$$
\mathbf{x} = [(\delta m^n_1 - \delta p^n), (\delta m^n_2 - \delta p^n), \ldots, \delta v^n, \delta \Psi^n]^T
\quad (35)
$$

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

$$
Q_{SOM} = \begin{bmatrix}
-I_{3x3} & 0 & 0 & [\times \hat{r}_{m1v}^n] \\
0 & -I_{3x3} & 0 & [\times \hat{r}_{m2v}^n] \\
0 & 0 & -I_{3x3} & 0 \\
0 & 0 & 0 & -I_{3x3} \\
0 & 0 & 0 & 0 & [\times f^n_1] \\
-3 & 0 & 0 & 0 & [\times f^n_2] \\
0 & 0 & 0 & 0 & 0 & [\times f^n_1] \\
0 & 0 & 0 & 0 & 0 & [\times f^n_2] \\
0 & 0 & 0 & 0 & 0 & 0  \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\quad (36)
$$

The first 6 rows are linearly independent, rows 7 to 9 provide three extra linearly independent rows and provided that $f^n_1 \times f^n_2 \neq 0$ and/or $\hat{r}_{m1v} \times \hat{r}_{m2v} \neq 0$ and rows 13 to 18 and 31 to 36 provide an extra three linearly independent rows. Thus the SOM 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 ($\delta m^n_i - \delta p^n$) to the state vector) increases the rank of the SOM by three thus resulting in full state observability for any number of landmarks greater than or equal to one.

4) Instantaneously Unobservable Mode: We have shown the observability of the system over two time segments where the values of the matricies $F$ and $H$ change between segments. We are also interested in determining the instantaneously unobservable modes in the system, that is the unobservable directions when only one time segment is considered. The instantaneously unobservable mode indicates the direction in the state space to which no information is being added over the local time set of observations.

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

$$
Q_{SOM} = \begin{bmatrix}
-I_{3x3} & 0 & 0 & [\times \hat{r}_{m1v}^n] \\
0 & -I_{3x3} & 0 & [\times \hat{r}_{m2v}^n] \\
0 & 0 & -I_{3x3} & 0 \\
0 & 0 & 0 & -I_{3x3} \\
0 & 0 & 0 & 0 & [\times f^n_1] \\
0 & 0 & 0 & 0 & [\times f^n_2] \\
0 & 0 & 0 & 0 & 0 & [\times f^n_1] \\
0 & 0 & 0 & 0 & 0 & [\times f^n_2] \\
0 & 0 & 0 & 0 & 0 & 0  \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\quad (37)
$$

There is one unobservable mode:

$$
x_{unobs} = [f^n_1 \times \hat{r}_{m1v}^n, f^n_1 \times \hat{r}_{m2v}^n, 0, f^n_1]^T
\quad (38)
$$

Thus the instantaneously unobservable state is related to the component of the misalignment angle $\delta \Psi$ about the specific force vector $f^n$ in the navigation frame and the component of each of the relative map-vehicle position error states that is aligned with the cross product the specific force vector and the estimated relative range vector $\hat{r}_{mb}^n$ for each feature $i$.

Our vehicle maneuver strategy should be to perform vehicle actions that do not allow this mode to remain pointing in a single direction thus resulting in long periods of time in which information is not available to one direction in the state space. Thus we should consider vehicle maneuvers that maximise the direction changes by considering the specific force vector of the vehicle and the relative range vector to each feature.

IV. VEHICLE CONTROL ARCHITECTURE

In this section we describe the control architecture for maximising navigation system performance for a UAV. Fig. 1 illustrates the control and guidance scheme on the aircraft. The SLAM navigation estimates are used as feedback for the low-level and high-level control systems. A combination of decision rules and path planning based on maximising the expected mutual information gain of the navigation estimates is used to generate the trajectories that are provided to the control system to follow.

A. Low-level and High-level Control

The low-level control system maintains the stability of the aircraft, using the aircrafts throttle and control surfaces to control the altitude ($h$), velocity ($v$), bank angle ($\phi$) and sideslip ($\beta$) of the aircraft. The high-level control system uses inputs into the low-level control system in order to control the vehicle along allocated trajectory segments composed of straight lines and arcs in space. The guidance scheme dynamically allocates the trajectory segments for the vehicle to follow based on the methods in the following subsection.
Fig. 2. Evaluating potential destinations by the guidance system: destinations are composed of the explored region and surrounding unexplored regions broken down into a grid

B. Guidance System Planning and Trajectory Allocation

The guidance system assigns one trajectory at a time consisting of a set of lines and arcs in space to the high-level control system on the UAV. Upon achieving the previous trajectory segment, the guidance system consults a discrete list of available destinations, evaluates the total utility of travelling to and making observations along the way to each destination and composes a set of paths to form a trajectory to command to the on-board control system to perform. The list of possible destinations is made up of all of the explored area and unexplored area encompassing the explored region, broken up into map grid points of size 100x100 meters (see Fig. 2).

C. Paths for Maximising Mutual Information Gain

The total utility for travelling to a destination is given as the expected mutual information gain $I[x,a]$. To calculate the expected information gain, we simply simulate forward in time the expected covariance of the estimate errors from our current knowledge using all of the expected observations along the path to the end destination and substitute into equation (26).

1) Evaluating Utilities for Travelling to Explored Regions: In order to evaluate the information gain of travelling to a known region, using (26), the expected covariance matrix $P(t_{dest}|t_{dest})$ at the time of reaching the destination $t_{dest}$, is found by propagating the current value of covariance from the filter, $P(t_0|t_0)$, through from the current time, $t_0$, to $t_{dest}$. The propagation is broken up into two sections:

a) Expected Prediction Propagation: The motion to the destination is approximated by a steady bank until the aircraft heading is aligned with the destination followed by steady straight and level flight. The covariance is propagated between expected observation using approximations to equation (9).

b) Expected Observation Propagation: Using approximations to the expected observations that the vehicle will make along the path, the covariance is updated at each expected observation using Equations (12-14) with approximated observation models.

2) Evaluating Utilities for Exploring an Unknown Region: It is assumed that there exists a certain available feature density, $\rho_f$, in terms of average number of features per map grid area across the map which we will use to propose a utility for exploring an unexplored region. Firstly we assume that within each unexplored area there is a single feature with position covariance of $U_{sf}$. This covariance is chosen as an appropriately large initial value, as if for an unknown location of the feature. We thus assume that the SLAM filter covariance matrix $P(t_0|t_0)$ at the time $t_0$ is augmented to account for the unknown feature:

$$ P(t_0|t_0)_{aug} = \begin{bmatrix} P_{uv} & P_{vm} & 0 \\ P_{mv} & P_{mm} & 0 \\ 0 & 0 & U_{mf} \end{bmatrix} $$

(39)

and the approximation of the covariance propagation is performed in the same manner as described in the sections above with the added observations of the presumed features in unexplored areas.

D. Decision Rules from Observability Analysis

Table I describes some typical aircraft maneuvers, the resulting instantaneous observability of the navigation estimates and the speculated result in navigation estimate information.

<table>
<thead>
<tr>
<th>Maneuver</th>
<th>$\Delta f^m$</th>
<th>$\Delta f^m_{rv}$</th>
<th>Observability and Estimate Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motionless /Hover</td>
<td>none</td>
<td>none</td>
<td>low accuracy on attitude states</td>
</tr>
<tr>
<td>Climb or Descent</td>
<td>none</td>
<td>steady change in vector direction</td>
<td>low heading accuracy</td>
</tr>
<tr>
<td>Steady Level Flight</td>
<td>none</td>
<td>steady change in vector direction</td>
<td>low heading accuracy</td>
</tr>
<tr>
<td>Steady Turn /Orbit</td>
<td>vector traces a cone shape</td>
<td>range to feature constant for features at center of turn, direction changing</td>
<td>high accuracy on all states</td>
</tr>
<tr>
<td>S-Shape Maneuver</td>
<td>vector oscillates back and forward</td>
<td>vector to features traces S-shape</td>
<td>high accuracy on all states</td>
</tr>
</tbody>
</table>

**TABLE I**

**CLASSES OF UAV MANEUVERS AND EXPECTED EFFECT ON NAVIGATION ESTIMATE ACCURACY FROM OBSERVABILITY ANALYSIS**

We have seen from the observability analysis that when the vehicle is in a motionless hover, straight and steady level flight or a steady climb/descent and the specific force vector does not change direction between time segments, the unobservable mode remains relatively stationary. These maneuvers thus mainly result in low heading error information (i.e. the direction in the state space of instantaneous...
Fig. 3. (a) The Brumby MkIII UAV, weighing 40kg with a wing span of 2.8 meters, capable of carrying a payload of 13.5 kg and flying at 100kts (b) Visualisation of the simulation over the Center for Autonomous Systems flight test center at Marulan, NSW. The blue and red lines illustrate the trajectory of the vehicle and the green square indicates the current footprint of the sideward mounted vision camera.

unobservability given the only specific force acting on the vehicle is from the lift or thrust generation applied to equal to downwards acceleration due to gravity). 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 more evenly across all of the vehicle attitude error state estimates.

One decision rule that we apply to the system is therefore to perform a complete 360° steady turn around the nearest set of features in the event that the heading uncertainty in the filter grows beyond a certain threshold.

V. SIMULATION DESCRIPTION

In this section we present some initial results from a high-fidelity 6-DoF simulation (see Fig. 3) of a UAV performing the guidance strategy which is run in order to test the concept before the algorithms can operate on the real system. The same simulation has also been used to verify the existing guidance and control systems that operate on the real vehicle and provides a good indication of the real issues involved in ensuring the stability of the vehicle.

In the simulation, a single UAV is given the task of building up a feature map of an unexplored region on the ground of given size. On-board the vehicle is a low-cost strapdown Inertial Measuring Unit (IMU) and two vision sensors; a downwards mounted camera and sideways mount camera. 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 0.1 m/s² for the accelerometers and 0.1 deg/s for the gyros. The vision sensors run at 10 frames/sec and have a horizontal field of view of 40° and a vertical field of view of 30° 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 camera. The features are dispersed across the terrain within a 9 square km area with an average density of \( \rho_f = 0.01 \) features/100m². In the simulation we assume known data association of the map features.

VI. SIMULATION RESULTS

In this section we present the results of the simulation. Figure 4 illustrates the path taken by the vehicle over a 200 second flight. The vehicle initially performs a circuit where the information gain is highest by closing the loop on the initially spotted features. The expected information gain then becomes higher for travelling to and re-observing a briefly observed set of features to the south of the map before returning to an area close to the start of the map again. The behavior generated by running the simulation over a long period of time will be to localise the features on the fringe of the map and occasionally returning to close the loop on well known features.

Figures 5-7 show the errors in the estimated position, velocity and attitude of the vehicle and the corresponding 1-sigma confidence bounds within the navigation filter.

VII. CONCLUSIONS AND FUTURE WORK

This paper has demonstrated a framework for developing intelligent guidance schemes for a UAV for improving navigation integrity when operating over unknown terrain. The proposed scheme plans vehicle paths based on
a combination computed approximations to the expected mutual information gain for certain trajectories for the vehicle to follow as well as decision rules based on a qualitative knowledge of the effect maneuvers have on the observability of the system. The approach is sub-optimal in the navigation performance sense however offers a practical alternative to unreasonable computationally involved in performing the full optimisation of vehicle actions to maximise navigation estimate information.

In future work, we will look at different metrics for navigation performance such as consistency of the filter and the ability for data association. These metrics will assist in composing additional decision rules for the vehicle actions. Future work will also look at expanding on the insight gained in the analysis of the existing metrics by considering alternative formulations of the inertial SLAM equations from the direct and indirect forms.

ACKNOWLEDGMENTS

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.

REFERENCES