Cooperative Multi-UAV Active SLAM for Traversing Partially Mapped Terrain

Mitch Bryson and Salah Sukkarieh

Abstract—This paper develops both single and multi-UAV path planning schemes for navigating over partially known terrain in GPS-denied environments. The UAVs perform Simultaneous Localisation And Mapping (SLAM) using a combination of on-board inertial sensors and an on-board terrain sensor and share map information with one another over a data fusion network. UAV paths across the terrain are planned such that the expected SLAM localisation accuracy for each vehicle is maintained above specified constraints. The path planner thus balances between traversing known and unknown sections of the terrain while attempting to reach a predefined objective point in minimum time. Simulation results are presented using a team of UAVs.

I. INTRODUCTION

Consider the scenario where a team of cooperating UAVs is tasked with navigating from their starting locations to a defined destination point in a time efficient manner. The vehicles operate in a GPS-denied area and thus are left only with terrain-aided navigation as a source of localisation (the platform’s own position and attitude) information for each platform. Some prior terrain information is provided to the platforms (for example from previous satellite imagery of the area). The quality of this terrain information however may be low; noted terrain features in the given map have low position accuracy, are sparsely situated and only cover a limited region of the map. In this scenario, the vehicles therefore perform Simultaneous Localisation And Mapping (SLAM) [3], [4], [13] to supplement the navigation process by observing and introducing new features into the map, while also improving the accuracy of the prior terrain information. The team of UAVs also shares this information over a communications/data fusion network in order to assist one another. Quality UAV position and attitude information is required on each platform in order to act as feedback for on-board platform control and to assist in other information gathering tasks such as positioning other features and targets in the environment. There exists a constraint that the localisation errors on each UAV should not grow above a given threshold; vehicles should therefore carefully plan their trajectories across the terrain in order to reach their destination point while not violating this error threshold.

The concept of vehicle trajectory planning for improving localisation and mapping estimates (commonly referred to as ‘active SLAM’) has been studied in the past; [8], [11], [17] all examine planning algorithms for indoor robots performing SLAM where the planning is performed over previously mapped features and the objective is to maximise the accuracy of localisation and mapping estimates. [2] and [15] present path planning algorithms for indoor robots where part of the objective is to explore new areas of the environment but where the objectives of exploration and revisiting mapped areas is weighted in a heuristic fashion. Similarly, [6] and [9] present heuristically weighted multi-robot path planning systems in SLAM for choosing between the tasks of exploration of new features and map re-visititation.

In [14] the authors present an active SLAM planner that ‘attracts’ the vehicle towards new areas to explore by artificially weighting the utility of slowly moving to unmapped areas. The concept of active path planning for localisation has also been investigated for aerial vehicles; [1] investigated a path planning system for an aerial vehicle that maximises localisation accuracy when flying over prior mapped terrain only. Active SLAM has also been explored for both a single UAV [5] and a team of UAVs [4] where the objectives of the path planning were simply to maximise localisation and mapping information while exploring a new area.

In all of the above active SLAM strategies the objective has been for the vehicle to explore a new area as efficiently as possible where the objectives of exploring new features in the environment versus maintaining localisation accuracy has been tackled in a heuristic manner. The scenario described at the start of this paper requires a more principled way of ensuring that localisation accuracy will be maintained while flying over unmapped terrain. The UAVs could choose a path that ‘hops’ from one known region of the map to another, thus constraining the localisation error growth associated with flying over unexplored terrain, however such a path may not exist or be inefficient given low quality initial terrain information available. The goal of the path planning should be to carefully balance between traversing unknown terrain and previously mapped terrain such that the localisation accuracy requirements are met but also so that travel time is minimised. This is the objective of our work in this paper; our path planning system computes the expected localisation error growth when exploring new areas based on the expected density of features we shall find and when these features will be observable. Our path planner thus proposes paths for a team of UAVs that minimise the time taken to reach a destination but where the expected localisation errors are constrained within specified limits. Simulation results are presented which demonstrate and compare the planned paths for both a single UAV and for a team of UAVs over different types of initial terrain information.
II. INERTIAL SLAM WITH PARTIAL PRIOR MAP INFORMATION

In our formulation of inertial SLAM, we estimate the position, velocity, attitude and inertial sensors biases of each UAV, along with the position of 3D point features in the environment using information from inertial sensors and a terrain-sensor on-board each vehicle. The inertial SLAM algorithm was first presented in [12] and was applied to a single UAV performing a navigation and mapping task. Subsequently, [16] presented two UAVs performing inertial SLAM while sharing information about the terrain map across a decentralised data fusion network. When some prior information about parts of the terrain is available, the same SLAM estimation framework can be used to both increase the accuracy of the prior terrain map and localise the vehicle from observations of the terrain features made by the vehicle.

This section provides an overview of the state vector, process and observation models used in the inertial SLAM algorithms for both single-UAV and multi-UAV. These models are used to help predict the localisation accuracy in the path planning system described in Section II-B below. For an overview of the inertial SLAM estimation algorithms, the reader is referred to [3], [4].

A. Single Vehicle Inertial SLAM

The inertial SLAM algorithm is formulated using an Extended Kalman Filter (EKF) and uses information from inertial sensors and from feature observations from on-board terrain sensors in order to estimate the three-dimensional vehicle position (p^m), velocity (v^m) and Euler angles (Ψ^m = [φ, θ, ψ]^T), Inertial Measuring Unit (IMU) sensor biases (δf^b and δω^b) and the N three-dimensional feature locations (m^n_i) in the environment:

$$\dot{x}_{SV,k} = [p^m_{k}, v^m_{k}, \Psi^m_{k}, \delta f^b_{k}, \delta \omega^b_{k}, \ldots, m^n_{1,k}, m^n_{2,k}, \ldots, m^n_{N,k}]^T$$

(1)

where the superscript n indicates local navigation frame referenced vectors, the superscript b indicates body-fixed frame referenced vectors and the subscript k represents the state vector at the kth time step. The SLAM algorithm maintains a state estimate ˆx_{SV,k} and state covariance estimate P_{SV,k} at the current time k within an EKF framework.

1) Process Model Equations: The state estimate ˆx_{SV} is predicted forward in time via the non-linear, discrete-time process model equation:

$$\dot{x}_{SV,k} = F[\dot{x}_{SV,k-1}, u_k, k] + G[w_k]$$

(2)

where F[.,.,.] is the non-linear state transition function at time k and G[.,.,.] is the input model transition function at time k. The process model equations can be expanded to:

$$\delta f^b_k = \delta f^b_{k-1}$$

(6)

$$\delta \omega^b_{ib,k} = \delta \omega^b_{ib,k-1}$$

(7)

$$m^n_{i,k} = m^n_{i,k-1}$$

(8)

where Δt is the time difference between the k and k − 1 discrete time segments, g^n = [0, 0, g]^T is the vector of acceleration due to gravity in the local navigation frame (g = 9.81 m/s²), E^n_k is the body to navigation frame rotation rate transformation matrix, C^n_b is the Direction Cosine Matrix (DCM) transformation from the body to local navigation frame and terrain features are treated as being stationary. The vectors ̂F_b and ̂ω_b are the accelerometer specific force vector reading and gyroscope rotation rate reading respectively (where u_k = [F_b, ω_b]^T), ̂δf_b and ̂δω_b are the accelerometer and gyro biases respectively and w_f and w_g are the accelerometer and gyro noise values where:

$$w_k = [w_f,k, w_g,k]^T$$

(9)

w(k) is the kth time sample of an uncorrelated, zero-mean vehicle process noise vector (IMU noise errors) of covariance Q.

2) Landmark/Terrain Observation Model: The observation model equations describe the relationship between the sensor observation of a map feature in the sensor co-ordinates of the terrain sensor to the estimated states in SLAM. The observation z_i(k) (of the i-th feature in the map) is related to the estimated states:

$$z_i(k) = \mathbf{H}_i(p^n_{i,k}, \Psi^n_{k}, m^n_{i,k}, k) + v_k$$

(10)

where H_i(.,.) is a function of the feature location, vehicle position and Euler angles and v_k is uncorrelated, zero-mean observation noise errors of covariance R. We consider the case where a vision camera/laser range finder system is used to measure the range and bearing to features in the terrain. The observation model is given by:

$$z_i(k) = \begin{bmatrix} \rho_i \\ \varphi_i \\ \psi_i \\ \phi_i \\ \theta_i \\ \psi_i \end{bmatrix} = \begin{bmatrix} \sqrt{(x^s)^2 + (y^s)^2 + (z^s)^2} \\ \tan^{-1}\left(\frac{y^s}{x^s}\right) \\ \tan^{-1}\left(\frac{z^s}{\sqrt{(x^s)^2 + (y^s)^2}}\right) \\ \tan^{-1}\left(\frac{y^s}{z^s}\right) \\ \tan^{-1}\left(\frac{x^s}{z^s}\right) \end{bmatrix}$$

(11)

where ρ, φ, ψ, φ, θ and ψ are the observed range, azimuth and elevation angles to the feature and x_s, y_s and z_s are the cartesian co-ordinates of p_{i,s}, the relative position of the feature w.r.t the sensor, measured in the sensor frame:

$$p_{i,s} = C^n_b C^n_{i,n}[m^n_{i,k} - p^n - C^n_b p^b_{i,s}]$$

(12)

where C^n_b is the transformation matrix from the body frame to the sensor frame, p^b_{i,s} is the sensor offset from the vehicle centre of mass, measured in the body frame, otherwise known as the ‘lever-arm’ and C^n_{i,n} = (C^n_b)^T.

When an observation of a new feature is made for the first time, it’s initial position estimate is computed from the sensor observations and estimated UAV position and attitude:

$$m^n_{i,k} = J_i(p^n_{i,k}, \Psi^n_{k}, z_i,k + v_k, k)$$

(13)
where \( \mathbf{J}_i(.,k) \) is a function of the vehicle position, Euler angles and feature observation:

\[
\mathbf{m}_i^n = \mathbf{p}^n + C_{b}^{n} \mathbf{p}_{sb}^b + C_{b}^{n} \mathbf{C}_{x}^{b} \mathbf{p}_{ms}^s
\]  

(14)

3) Loading Initial Terrain Information: Prior information about the operating terrain is provided as an initial estimate to the SLAM algorithm as a set of terrain feature locations and uncertainties. The position of each feature is initialised directly into the state vector \( \mathbf{x}_{SV} \) in Equation 1 and the state covariance \( \mathbf{P}_{SV} \) is initialised:

\[
\hat{\mathbf{x}}_{SV,\text{init}} = [\mathbf{x}^v(1), \mathbf{m}_1^n, \mathbf{m}_2^n, \ldots, \mathbf{m}_N^n]^T
\]  

(15)

\[
\mathbf{P}_{SV,\text{init}} = \begin{bmatrix}
\mathbf{P}_{vv}(1) & 0 & \ldots & 0 \\
0 & \mathbf{P}_{m1} & \ldots & 0 \\
\vdots & \vdots & \ddots & 0 \\
0 & 0 & 0 & \mathbf{P}_{mN}
\end{bmatrix}
\]  

(16)

where \( \mathbf{x}^v(1) \) and \( \mathbf{P}_{vv}(1) \) are the initial vehicle states and covariance, \( \mathbf{m}_1^n, \mathbf{m}_2^n, \ldots, \mathbf{m}_N^n \) are the positions of the prior terrain features and:

\[
\mathbf{x}_i^n = [\mathbf{p}_i^n, \mathbf{v}_i^n, \mathbf{a}_i^n, \mathbf{\Psi}_i^n, \mathbf{\delta f}_i^n, \mathbf{\delta b}_{ib,k}]
\]  

(17)

are the vehicle states, \( \mathbf{P}_{m1}, \ldots, \mathbf{P}_{mN} \) are:

\[
\mathbf{P}_m = \begin{bmatrix}
\sigma_N^2 & 0 & 0 \\
0 & \sigma_E^2 & 0 \\
0 & 0 & \sigma_D^2
\end{bmatrix}
\]  

(18)

where \( \sigma_N, \sigma_E, \sigma_D \) are the one-standard deviation uncertainties of each initial terrain feature position in the North, East and Downwards directions.

B. Multi-Vehicle Inertial SLAM

In multi-vehicle inertial SLAM, several vehicles share estimates of the terrain feature locations they have mapped over data fusion network. The complete vector of estimated positions is now:

\[
\hat{\mathbf{x}}_{MV,k} = [\mathbf{x}_1^n, \mathbf{x}_2^n, \ldots, \mathbf{x}_k^n, \mathbf{m}_1^n, \mathbf{m}_2^n, \ldots, \mathbf{m}_N^n]^T
\]  

(19)

where the subscript in \( \mathbf{x}_k^n \) represents the vehicle number and where \( M \) is the total number of vehicles. Where data fusion between UAVs is tackled in a centralised manner, an EKF is used to estimate the state vector in Equation 19 above, along with a covariance matrix \( \mathbf{P}_{MV}(k) \) from the process model:

\[
\hat{\mathbf{x}}_{MV,k} = \mathbf{F}_{MV}[\hat{\mathbf{x}}_{MV,k-1}, \mathbf{u}_{1,k}, \mathbf{u}_{2,k}, \ldots, \mathbf{u}_{M,k}, \mathbf{k}] + \ldots + \mathbf{G}_{MV}[\mathbf{w}_{1,k}, \mathbf{w}_{2,k}, \ldots, \mathbf{w}_{M,k}]
\]  

(20)

which incorporates all of the vehicle states and inertial sensor data, and also uses the observation model shown in Equation 10. \( \mathbf{F}_{MV}[.,k] \) is composed from \( M \) copies of Equations 3 to 7, one for each vehicle, and Equation 8, one for each feature. The process model shown in Section II-A.1 is augmented to encompass localisation and inertial sensor bias values for each of the vehicles in the team. Observations from each of the vehicles are fused into the joint feature map using Equations 11 to 12. For more details on algorithms for centralised and decentralised multi-UAV SLAM the reader is referred to [4].

III. SINGLE UAV PATH PLANNING ALGORITHMS

In this section we discuss a single UAV path planning algorithm for reaching a fixed destination point while satisfying given localisation error constraints using the terrain-aided navigation algorithms discussed in the above section. The aim of the path planner is to specify a 3D trajectory for the vehicle to follow such that (1) the planned trajectory takes the vehicle from its initial starting location to a fixed, pre-specified destination point, (2) the trajectory aims to minimise the flight time of the UAV (and reaches the destination in a given time constraint) and (3) the trajectory is planned such that several constraints on the expected localisation errors (UAV position and attitude estimation) are met (i.e. localisation errors don’t rise over a specified threshold at any point in time along the trajectory).

A. Potential Planning Action Space: Trajectory Segments

The total trajectory for the UAV to follow is composed of a series of straight lines and steady turning circle flight segments that connect a list of horizontal waypoints the vehicle must fly to. The UAV maintains a constant altitude and flight speed during the trajectory. When a waypoint is reached the vehicle performs a steady turn (change in heading angle) towards the direction of the next waypoint, at the maximum rate of turn, followed by straight and steady level flight until the waypoint is reached, moving through the entire list of waypoints until the final destination is reached.

B. Minimum-time, Localisation Error Constrained Path Planner

In order to plan the series of waypoints and thus the trajectory for the UAV to follow, the path planner takes the following steps:

1) From the starting location a list of local trajectory segments \( \mathcal{A}_{local,SV} = \{a_{local,SV,1}, a_{local,SV,2}, \ldots, a_{local,SV,n}\} \), each composed of a steady turn and straight line to a single waypoint, are proposed.

2) The cost of each action is computed as the distance of the final destination to the end of the local trajectory segment (i.e. local waypoints that move the UAV closer to the final destination induce a lower cost). The local actions are then ordered from lowest cost to highest cost.

3) Beginning with the local action with lowest associated cost, we determine whether or not we expect performing the local action will violate the set localisation error constraints (see Section III-C below). If the constraints are expected to be violated, we move on to the next lowest cost local action in the list until we find an action that does not violate the constraint.

4) When an action is found that does not violate the constraints, we repeat the process from step 1 (i.e. define another list of potential local actions starting from the end of our chosen action).

5) If all of the potential local actions are exhausted (i.e. none of them can achieve the localisation error...
with the onboard inertial sensors to estimate the localisation below terrain are made using onboard sensors and fused trajectory composed of a steady turn and straight and level flight between two waypoints, feature observations of the trajectory is shown in Figure 1.

Fig. 1. Path Planning Process: Local paths are proposed and assessed to see if they meet the localisation error constraints. The best local path (one that takes the vehicle closest to the final destination) that also meets the constraint is stored and more local paths considered at the end of this path. The process is repeated until the goal destination is reached. If none of the local paths meet the constraints, then the path planned returns to the beginning of the last path taken and replans from the remaining choices.

6) At each waypoint, the planner also determines if the final destination point is achievable within the time constraint specified by the planner (by computing the remaining time left and the minimum time is would take to fly directly to the final destination). If it does not meet the time constraint then the planner calls the option from the list and moves onto the next potential local action.

7) The planner ends in two possible ways: either a series of waypoint is planned that eventually reaches the destination at which point the planning has finished or the planner exhausts all possible lists of waypoints and finds none that can meet both the localisation error constraints and the time constraint.

The planning process is based on an A* search algorithm [7] which is most often used for path planning of ground-based robotic vehicles in the presence of obstacles. In our case, trajectories are constrained by localisation requirements rather than any physical obstacles. A flowchart of the planning algorithm is shown in Figure 1.

C. Assessing Localisation Error Constraints - Covariance Prediction

As the vehicle flies along a given segment of the trajectory composed of a steady turn and straight and level flight between two waypoints, feature observations of the below terrain are made using onboard sensors and fused with the onboard inertial sensors to estimate the localisation and map states. While the vehicle banks to turn, feature observations are made using sideways mounted cameras and while in straight and level flight, observations are made using a downwards facing camera. Thus for a given trajectory segment and an initial estimate of localisation and map feature states (at time segment \( k \)), we wish to compute the expected covariance of estimate errors over the course of the trajectory segment.

To do this we firstly break up the trajectory into \( n \) discrete steps and approximate the set of vehicle states and expected observations that the vehicles will make along the trajectory:

\[
a_{SV} = \{x_k, z_k, x_{k+1}, z_{k+1}, \ldots, x_{k+n}, z_{k+n}\} \quad (21)
\]

Expected observations are computed by calculating when known features in the map are expected to fall within the field of view of the different onboard sensors. We may also make observations of new features along the trajectory which will assist in the localisation process. Let us assume for now (further details will be explained in the sections below) that an ‘oracle’ can tell us the locations of new features in the terrain that we currently don’t know about yet (Equation 21). We can now use a discrete, recursive Ricatti equation [10] to predict the covariance of the SLAM estimate errors along the trajectory:

\[
P_{SV,k+1} = \nabla F_{k+1} [P_{SV,k} - M_k] \nabla F_{k+1}^T + \nabla G_{k+1} Q \nabla G_{k+1}^T \quad (22)
\]

\[
M_k = P_{SV,k} \nabla H_{k} S^{-1} \nabla H_{k} P_{SV,k} \quad (23)
\]

\[
S = \nabla H_{k} P_{SV,k} \nabla H_{k}^T + R \quad (24)
\]

where \( Q \) and \( R \) are the process and terrain sensor noise covariance matrices respectively and \( \nabla F \) and \( \nabla G \) are the jacobians of the process model (Equation 2) w.r.t the state \( x_{SV} \) and w.r.t the process noise \( w_k \). \( \nabla H \) is the jacobian of the observation model (Equation 10) w.r.t the state \( x_{SV} \). \( \nabla F \), \( \nabla G \) and \( \nabla H \) are computed at each time using \( x_{SV,k}, \ldots, x_{SV,k+n} \), the sequence of expected states and \( z_k, \ldots, z_{k+n} \), the sequence of expected observations from each vehicle for a given action \( a_{SV} \).

For expected observations of new features in the list \( a_{SV} \), the expected covariance matrix \( P_{SV} \) is augmented during the Ricatti equation process to include the uncertainty of the new feature:

\[
P_{SV,aug} = \begin{bmatrix} I & 0 \\ \nabla J_x & \nabla J_z \end{bmatrix} \begin{bmatrix} P_{SV} & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} I & 0 \\ \nabla J_x & \nabla J_z \end{bmatrix}^T \quad (25)
\]

where \( \nabla J_x \) and \( \nabla J_z \) are the jacobians of the initialization function \( J \) (Equation 13) w.r.t the state \( x_{SV} \) and the expected observation \( z_i \) respectively. Further expected observations of
the unknown feature are updated in the expected covariance matrix using the Equation 22.

The resulting process allows us to predict forward in time for a given trajectory segment how we expect the SLAM navigation algorithm to perform (from a statistical standpoint). We have an explicit way of predicting forward the effect observations of prior known terrain features will have on our localisation performance. Given our expected observations of new features that are not part of our prior terrain map (as provided by our `oracle`) we can also predict forward our localisation performance when moving across unknown areas of the terrain. Section III-D below explains in more detail how we predict when and where we will make observations of new features.

D. Computing Expected Observations over Unknown Areas of Terrain

Although we cannot specify exactly when and where we will make observations of new features (since we don’t have any prior information of their whereabouts), we can make an approximation. We begin by making the assumption that there exist features in the unmapped areas of the terrain and that the density of these features is equal to the density of features we can see in the known regions of the map. Based on this density, we thus randomly distribute the expected locations of features into the unmapped regions of the terrain. From this distribution of features we then compute for a given trajectory when and where we expect to make an observation, and this information is inserted into the action data in Equation 21 and used to predict forward it’s effect on localisation performance (see Section III-C). The process is illustrated in Figure 2. Although our expected locations of features will not be the same as the actual locations of new feature we will find when we actually go out and explore the unmapped area, the methods gives us a systematic way to approximate the localisation performance over unmapped areas, which we would otherwise not be able to ascertain.

E. Assessing Localisation Error Constraints - Constraint Calculation

Once we have computed the expected estimate error covariance matrix along the trajectory, we can use the matrix to evaluate whether or not we have exceeded the localisation error on particular states. In our path planning system we apply four separate constraints of the localisation performance errors; yaw angle, roll/pitch angle, horizontal positioning and vertical positioning errors where these are taken from the diagonal values of the covariance matrix. If at any point along the proposed trajectory any of the four localisation error constraints are violated, then the trajectory is removed from the planning list as discussed in Section III-B.

IV. MULTI-UAV PATH PLANNING ALGORITHMS

In the multi vehicle version of the planning algorithms, the goal of the planner is to specify a trajectory for each of the vehicles in the team from their starting location to the destination point while not violating the localisation constraints on any of the platforms. The planner uses the same procedure as shown in the above section except that now at each stage, the planner computes trajectory segments for each vehicle and evaluates the localisation performance on each vehicle but now where map information is shared between vehicles in a centralised SLAM data fusion EKF.

V. RESULTS

In this section we present results of our path planning algorithm for both the single and multi-vehicle cases over two different sets of initial terrain. The first set of terrain data is designed such that there is a continuous path of features that lead from the starting location to the final destination point. In the second set of terrain data there are three regions of prior map information with large gaps of unexplored territory that must be traversed. In each prior terrain set, the initial feature position uncertainty is 2m (1σ) and the UAVs are required to traverse a distance of approximately 6km. The constraints values for the localisation system uncertainty are 3° of 3σ roll and pitch uncertainty, 10m of 3σ horizontal positioning uncertainty and 5m of 3σ vertical positioning uncertainty.

Figure 3 (a), (b) and (c) illustrate the path taken by a single UAV for the first set of terrain data. The path taken follows along side the known features, where observations of these features are made using the terrain sensor facing out the righthand side of the UAV. As the UAV moves past the halfway point the trajectory cuts across into unknown terrain where the vehicle loops back on it’s trajectory a few times around the 160 to 250 second mark. During this time the
Fig. 3. Planning on Prior Terrain Information Set 1: Shown are the prior terrain features and initial planned trajectory for the UAVs to follow for both the single and multi vehicle planning scenarios. For the single vehicle planning (a) after 86 seconds, (b) after 173 seconds and (c) after 295 seconds where the destination is reached. For the multi-vehicle planning (d) after 67 seconds, (e) after 144 seconds and (f) after 198 seconds where the destination is reached by each UAV.

UA V closes the loop on features it has seen in this area, strengthening the correlations between these features and increasing their positioned accuracy such that the localisation errors remain within the constraints. Eventually the UAV cuts back towards the known parts of the terrain in the upper left-hand side of the terrain where the left-facing terrain sensor is used to observe features.

Figure 3 (d), (e) and (f) illustrate the path taken by the team of UAVs over the first set of terrain data. We can see that UAV2 and UAV3 start moving directly to the goal but that UAV1 is required to firstly sweep across the known parts of the map to the east before starting to head northwards to the goal location. after about 20 seconds, UAV2 moves across near UAV1’s path in order to help observe common features and for each vehicle to close the loop on each other’s observed features. Eventually UAV3 moves across to the west around the 30-40 second mark. Up until this point UAV3 has stuck mainly to known terrain areas and has quite good localisation accuracy. As UAV3 moves westward it starts to observe new features that UAV1 and UAV2 have added to the map and thus develops correlations between these new features and the areas of the known map UAV3 has just observed. This serves to close the loop on the explored features (increase their accuracy) thus lowering the localisation uncertainties of UAV1 and UAV2, and thus these UAVs are not forced to loop back as much on their trajectories as in the single vehicle case.

Figure 4 (a), (b) and (c) illustrate the path taken by a single
UAV for the second set of terrain data. In this case the vehicle is required to traverse two large areas of unknown terrain. As the vehicle moves into the first area of unknown terrain, it performs a trajectory that continuously loops back on areas it has begun to explore, continuously closing small loops in the exploration process. These ‘loops’ serve to strengthen the information in features the vehicle has added to the map through the exploration process, which in turn are used for localisation. The vehicle continues this looping pattern of terrain traversal until it reaches the area of prior known terrain in the middle of the map. At this point the UAV closes the loop on the explored area by observing the features with prior known information and continues straight on into the second unknown region of the map. The same type of looping exploration behavior is exhibited until the UAV reaches the known group of features near it’s destination. Similarly to the first terrain set, the vertical positioning accuracy seems to be the critical constraint. It should be noted that the UAV has taken longer to traverse the terrain than in the case of the first terrain set. This is due to the fact that the vehicle must spend more time looping back upon explored features while traversing unknown terrain.

Figure 4 (d), (e) and (f) illustrate the path taken by the team of UAVs over the second set of terrain data. A similar type of looping exploration over unknown terrain and straight and direct flight over known terrain is shown as in the single vehicle case over this terrain set. The main difference now is that the use of extra UAVs sharing terrain information has meant that the total traversal time has been reduced from 400 seconds to 221 seconds.

VI. CONCLUSIONS AND FUTURE WORK

This paper has demonstrated both single and multi-UAV path planning schemes for navigating over partially known terrain in GPS-denied environments. The system accounts for accuracy constraints in the resulting localisation state estimation process and chooses a path that balances between traversing known and unknown terrain while attempting to reach a predefined objective point in minimum time. The results of the path planning show that the method is able to plan feasible paths that meet both the localisation error and time constraints and balance between traversing known and unknown areas of the terrain in a measured, quantitative way that is dependant on the magnitude of the constraints and the expected density of features. The planner computes the expected localisation performance when moving into unknown areas of the terrain based on randomly distributing expected features into the unknown areas based on the density of features in the known areas of the map.

One issue with this approach is that the density of the features may change in different areas and thus require online corrections to the planned path that are made when the vehicle begins to explore and reevaluates the feature density. Another issue is that as the density of features becomes sparse, there may be small sections of time where features are not observed at all. This can have a somewhat unpredictable effect on the localisation accuracy and thus there is an inherent uncertainty or risk associated with exploration. In our future work, we intend to tackle this concept of risk through quantifying the risk or uncertainty associated with the localisation performance vs. a particular trajectory and incorporate this risk into the path planning process. Future work will also focus on developing a path planning system that provides real-time corrections to the initially planned path.

VII. ACKNOWLEDGMENTS

This work is supported in part by USAF/AOARD contract FA4869-08-1-4060. 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