Uncertainty Based Online Planning for UAV Target Finding in Cluttered and GPS-Denied Environments

Fernando Vanegas
Queensland University of Technology (QUT)
Australian Research Centre for Aerospace Automation
22-24 Boronia Road
Brisbane Airport, QLD 4008
+61 (0) 7 3138 1772
fernando.vanegasalvarez@hdr.qut.edu.au

Felipe Gonzalez
Queensland University of Technology (QUT)
Australian Research Centre for Aerospace Automation
22-24 Boronia Road
Brisbane Airport, QLD 4008
+61 (0) 7 3138 1363
felipe.gonzalez@qut.edu.au

***Abstract*—There are some scenarios in which Unmmaned Aerial Vehicle (UAV) navigation becomes a challenge due to the occlusion of GPS systems signal, the presence of obstacles and constraints in the space in which a UAV operates. An additional challenge is presented when a target whose location is unknown must be found within a confined space. In this paper we present a UAV navigation and target finding mission, modelled as a Partially Observable Markov Decision Process (POMDP) using a state-of-the-art online solver in a real scenario using a low cost commercial multi rotor UAV and a modular system architecture running under the Robotic Operative System (ROS). Using POMDP has several advantages to conventional approaches as they take into account uncertainties in sensor information. We present a framework for testing the mission with simulation tests and real flight tests in which we model the system dynamics and motion and perception uncertainties. The system uses a quad-copter aircraft with an board downwards looking camera without the need of GPS systems while avoiding obstacles within a confined area. Results indicate that the system has 100% success rate in simulation and 80% rate during flight test for finding targets located at different locations.**

TABLE OF CONTENTS

1. INTRODUCTION

The number of applications of Unmanned Aerial Vehicles (UAV) is increasing every day in a variety of fields that include surveillance, aerial photography and filming, environmental sampling and remote sensing [1], [2], [3], crop inspection and more [4], [5]. In order to increase the level of autonomy, these type of vehicles have to rely on safe and reliable navigation and guidance systems that are able to overcome perturbations and uncertainties in the environment in which these aircraft operate.

978-1-4673-7676-1/16/$31.00 ©2016 IEEE

There are several works involving target finding and tracking using UAVs. Most of these works are based on simulated scenarios [6], [7], [8], [9], [10], [11].

Other works present real flight solutions [15]. However these type of systems rely on very precise and/or external perception systems for localisation, which may not present in real world scenarios. Flying in GPS-denied environments and with only on-board sensors as the source of perception presents a big challenge and remains an open problem. A robust method for decision making with uncertainty in GPS denied environments are Partially Observable Markov Decision Processes.

Some works propose a POMDP solution for target tracking but only show results in a simulated environment [13], [15], [14]. Other work models the uncertainty in target localisation using a belief, and plan series of actions to solve the mission but relies on accurate localisation of the UAV in its environment, [12]. A model and a system that incorporates the uncertainty in the UAV state due to imperfect sensing, besides the uncertainty in the target's location is still under exploited.

There are many algorithms for solving POMDPs including [16], [17], [18], [19]. In recent years there has been some progress in the performance of these solvers, specially in their ability to cope with scenarios that have a large number of states $|S| > 2000$ and observations $|O| > 100$ [20], [21]. These improvements and the increase on computing power resources makes feasible to implement UAV motion planning tasks as POMDPs.

In this paper we present a UAV navigation and target finding problem in which the planned trajectory has to be recalculated after every iteration in order to account for uncertainties in motion and in low cost sensors in an indoor environment with obstacles. The target is on the ground but its location within the flying area is unknown and the UAV has to search and find it. Calculations to account for a large number of possible sequences of actions and states of the vehicle in the scenario is desirable but they must be done within a limited time depending on the system dynamics.

We develop a framework to test different POMDP solvers and UAV missions and consider an online solver; *Adaptive Belief Tree* (ABT) [21] for illustration purposes.

The framework uses a low cost quad-copter platform incorporated into a modular system running the Robotic Operating System (ROS).

The main contributions of this paper are:

1. A model of a UAV Navigation and Target Finding mission as a POMDP in which the actions are a set of 13 predefined controlled motions in 4 degrees of freedom, taking advantage of the holonomic capabilities of the multirotor.
2. An online system that is able to accomplish the Target Finding mission using an state of the art POMDP solver and a modular architecture in real, cluttered and GPS-denied scenarios.

This paper is organised as follows; section 2 covers a description of POMDP and an online solver ABT. Section 3 describes the system architecture. Section 4 describes the method used for calculating the update frequency of the POMDP solver. Section 5 describes the Navigation and Target Finding problem formulation as a POMDP. Results are discussed in section 6 and section 7 provides conclusions and future areas of research.

2. BACKGROUND

MDP and POMDP

Markov Decision Processes are a means to model the sequential decision problem under uncertainties using a mathematical framework[22]. When MDP is used for robotic missions the objective is to generate a policy that allows the robot to decide what sequence of actions should be taken by a robot (agent) in order to maximise a return or cost function, taking into account the uncertainties in motion. Plain MDPs assume that the states are completely observable which is not the case for a robot that has limitations in perception.

In real situations the robot perception is limited by the type of sensors and the environment in which the robot is moving. The perception of the robot is not completely accurate or is insufficient and consequently, knowledge of its current state would have errors or deviations from the real state. This limitation in the sensory systems of robots is also known as partial observability [23], [19], [20].

Partially Observable Markov Decision Processes (POMDPs) can incorporate the uncertainties in sensing and the partial observability of the agent in the environment [17].

Formally a Partially Observable Markov Decision Process is a tuple that consists of the following elements $(S, A, O, T, Z, R, \gamma)$ where S represents the set of states in the environment, A stands for set of possible action the agent can execute, O is the set of observations, T is the transition function for the state after taking an action, Z is the distribution function describing the probability of observing o from state s after taking action a, R is the set of rewards for every state and γ is the discount factor. POMDPs rely on the concept of belief or belief-state which is a probability distribution of the system over all the possible states in its state-space representation at a particular time. It is denoted by *belief* b.

In a POMDP the state of the agent can not be observed exactly, instead the agent receives an observation $o \in O$ determined by the probability distribution Z. A policy $\pi : \mathcal{B} \rightarrow \mathcal{A}$ allocates an action a to each belief-state $b \in \mathcal{B}$, which is the set of possible belief-states. Given the current $belief - state\ b$, the objective of a POMDP algorithm is to find an optimal policy that maximizes the accumulated return when following a sequence of actions suggested by the policy π. The accumulated *discounted return* is the sum of the discounted rewards after executing every action in the sequence from time t onwards $R_t = \sum_{k=t}^{\infty} \gamma^{k-t} r_k$, where r_k is the immediate reward received at particular time step t for taking action a_t. The *Value function* is the expected return from belief-state b when following policy π, $V^{\pi}(b) = \mathbb{E}\left[\sum_{k=t}^{\infty} \gamma^{k-t} r_k | b, \pi\right]$. An optimal policy for the POMDP is the one that maximizes the value function $\pi^*(b) = \arg\max_{\pi} V^{\pi}(b)$.

ABT

In order to test the framework and target detection and navigation scenario we use one of the fastest on-line POMDP algorithms to the authors knowledge, ABT.

ABT [21] is an online algorithm that uses the MCTS algorithm [24] to produce a search tree of possible subsequent belief-states and has proved to be successful in problems with large domains. The algorithm reduces the search domain by concentrating only on the states that can be reached by the agent from its initial belief-state. In order to calculate the transition to the next states the algorithm uses a black box simulator that has the transition functions for the agent. In this black box all the uncertainties in motion and in the environment are modelled. Using this simulator the dynamics of the system can be modelled using a continuous state space. The algorithm uses Monte-Carlo simulations to sample states from an initial belief-state and builds a search tree according to the observations received after performing actions.

ABT runs first a planning stage, then selects an action for the robot to execute based on the calculated policy and performs an update of the belief-state based on the observation received by the robot. The algorithm creates the search tree by doing Monte-Carlo simulations for a number of steps in the future until a planning horizon is reached. The longer the planning horizon, the more time the algorithm will take to build the search tree. The root of tree is a node containing the state particles representing the initial belief-state of the environment. The tree branches out according to the probability of selecting actions and receiving observations.

ABT updates after receiving an observation and improves the policy based on the observation received. Moreover, ABT provides a mechanism for accepting changes in the environment and adapts its policy accordingly.

3. SYSTEM ARCHITECTURE

In order to test the navigation and target finding mission in a real scenario, a modular system was created. The system is composed of four main modules, shown in Fig. 1. The four nodes run in parallel using the Robotic Operating System (ROS), they communicate with each other by publishing to standard ROS topics and requesting and providing ROS services.

The four modules are: a POMDP online solver node for motion planning, a motion controller node for performing actions in four degrees of freedom, a perception node that receives data from the UAV sensors and processes them to produce the observations for the POMDP model and an existing driver [25] for using the Parrot AR Drone multirotor in ROS. All nodes running in parallel in different threads allows the system to have different update rates for each of the nodes.

On-line POMDP Module

This module runs an online POMDP solver algorithm, it contains the formulation of the POMDP, i.e. the transition function with system dynamics and uncertainty in motion model, the observation model, the set of actions, the reward/cost function and the state representation.

The POMDP ROS module initialises the belief-state $\mathcal{B}$ according to the initial state representation in POMDP formulation (section 5). This module is a ROS node that runs at a constant rate of τ_{solver}. It first produces a policy based on the initial belief-state b_0, using a POMDP algorithm and then outputs an action a that maximises a return, to be executed by the motion control ROS node.

An observation o is obtained with data from perception module that calculates a pose estimation $\boldsymbol{P_r} = (x_r, y_r, z_r, \Psi_r)$ using the on-board sensed velocity and signs whether a target has been detected with the downward looking camera.

A prediction of the observation is conducted in order to allow for the calculation time of the online POMDP solver. This prediction is based on the current pose $\boldsymbol{P_r}$ obtained from the perception module and the action execution time τ_{action} and the action a being executed.

Once the observation is received, the POMDP node updates the belief-state b by matching the observation received with a node in the tree search and replenishes particles until a timeout $\tau_{replenish}$ is reached. Based on the current belief-state b the POMDP solver updates the policy in its tree search and outputs the subsequent action based on the updated policy.

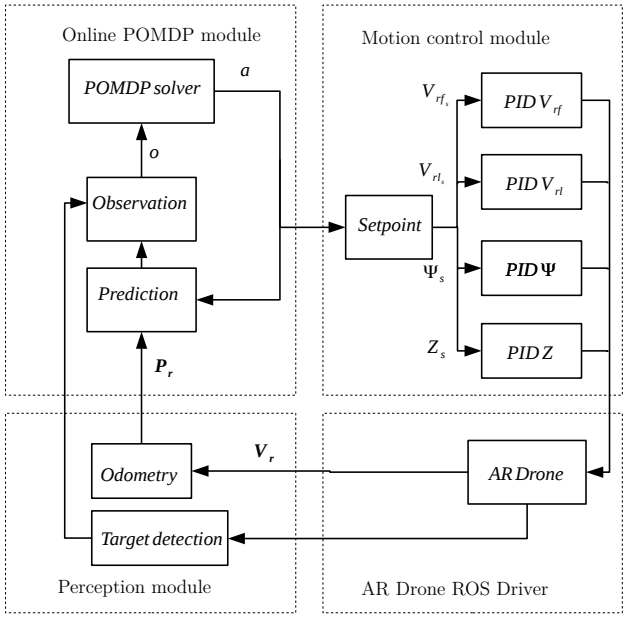


Figure 1. POMDP ROS System architecture

Motion Control Module

This is a ROS node developed to execute 4 PID controllers with an update rate of 100Hz. We use decoupled PID controllers in each degree of freedom: Forward velocity V_f, lateral velocity V_l, yaw angle Ψ_a and altitude z_a. Each PID controller is tuned to obtain a fast response, i.e. reach a steady state within the action execution time τ_{action} and the reference for each DOF are set by the commanded action from the planner(see table 1).

The input is a setpoint action given by the POMDP ROS node. The output of the PID controllers are pitch $\dot{\theta}$, roll $\dot{\phi}$ and yaw $\dot{\psi}$ rates and z rate $\dot{z}$ that are sent to the AR Drone Autonomy lab driver, that executes them. The motion control node runs in a continuous loop and updates the references for PID controller every time the POMDP node outputs a new action.

Step responses of every PID controller to setpoints commanded by all the actions in the set A were acquired in order to characterise the PID performances and to approximate the system dynamics. These characteristic responses are included in the transition function in order to approximate the real behaviour of the system after a commanded action.

Perception module

The perception module runs at a 100Hz rate. It calculates an estimation of the current pose based on the sensed forward and lateral velocities, UAV orientation and action execution time τ_{action}. It converts the forward and lateral velocities in the quadrocopter's frame to the scenario frame, and calculates x_r and y_r coordinates of the UAV position in the world frame. It also reads the altitude or z_r position from the on-board sensors.

Equation 1 is used to estimate the x_r and y_r positions, based on the onboard velocities measured at $\Delta t = 0.01s$

$$\begin{bmatrix} x_{r_{t+1}} \\ y_{r_{t+1}} \end{bmatrix} = \begin{bmatrix} x_{r_t} \\ y_{r_t} \end{bmatrix} + \begin{bmatrix} \cos(\Psi_{r_t}) & -\sin(\Psi_{r_t}) \\ \sin(\Psi_{r_t}) & \cos(\Psi_{r_t}) \end{bmatrix} \begin{bmatrix} V_{r_{f_t}} \\ V_{r_{l_t}} \end{bmatrix} \Delta t \quad (1)$$

4. NAVIGATION AND TARGET FINDING PROBLEM DESCRIPTION AND FORMULATION

Problem Description and Formulation

We illustrate the use of the framework for target finding and tracking mission with a scenario in which a multi-rotor unmanned aerial vehicle (UAV) flying in a 3D confined space without access to external GPS localisation and in the presence of obstacles must search and find a ground target which location is unknown. The aircraft is flying within a limited region of the airspace of size $(x = 6, y = 7, z = 4)m$. An example setup and this confined flying space is shown in Fig. 2. In the figure there are 5 obstacles inside the flying space that must be avoided.

The multi-rotor starts the mission from the ground. After take off, it hovers for a few seconds to initialise its orientation with readings from its on-board sensors. There is some initial drift produced in this initial hovering position. We use a gaussian probability distribution, Eq.2 ,with mean value μ around the take off position for the x_r and y_r UAV coordinates to model this initial uncertainty.

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \quad (2)$$

The problem is formulated as a POMDP that has the following elements: the state of the aircraft in the environment

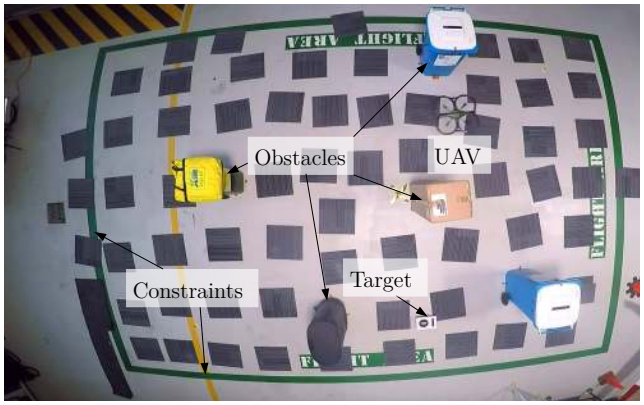


Figure 2. Example of real scenario for Navigation and Target Finding

(S), the set of actions that the multi-rotor can execute (A), the transition function describing the state transition after applying a specific action (T), the observation model that represents the sensed state of the aircraft after taking an action (O), and the reward and cost function (R).

State Variables (S)

The state variables considered in the POMDP formulation are the quad-rotor pose $\boldsymbol{P_r} = (x_r, y_r, z_r, \Psi_r)$, the target pose $\boldsymbol{P_t} = (x_t, y_t, z_t, \Psi_t)$ and the aircraft's velocity $\boldsymbol{V_r}$, all measured in the world frame. The aircraft velocity can be decomposed into two components in the aircraft's frame, forward velocity V_{r_f} and lateral velocity V_{r_l}.

Actions (A)

The quad-copter can actuate in 4 degrees of freedom. By changing its pitch angle, it moves forward or backward; changing its roll angle allows it to move to left or right; its heading is controlled by its yaw angle and it can change its altitude by varying the thrust to its motors at the same time.

The set of actions was created to account for movements in all four degrees of freedom and consists of 13 actions. An action to keep the aircraft static, i.e. $\boldsymbol{V_r} = 0\ m/s$; two actions to go forward and backward, with current heading angle Ψ_r, lateral velocity $V_{r_l} = 0\ m/s$, and forward velocity, $V_{r_f} = 0.6\ m/s$ and $V_{r_f} = -0.6\ m/s$, respectively. Six actions to change the heading angle to turn the multi-rotor to left or right with increments of $15°$, at a constant forward speed of $0.6\ m/s$. Actions Up and Down, increase or decrease altitude in 0.3 m, respectively, with aircraft's velocity fixed at $0\ m/s$, and two actions to roll left and right with with current heading angle Ψ_r and $V_{r_f} = 0\ m/s$, at constant speed, of $V_{r_l} = 0.6$ m/s and $V_{r_f} = -0.6\ m/s$, respectively. The set of actions is summarised in table 1.

Transition Function (T)

The motion of the quad-rotor is based on the set of actions described in table 1. These actions are in fact, step inputs or references to a controller in each of the four DOF. This allows to incorporate step responses, that are acquired experimentally, into the kinematic model of the aircraft using a decoupled model.

The kinematic model is described in equations (3) to (4). The next aircraft position is calculated by obtaining the change in position taking into account the system step responses, the initial and requested values for state variables and the action execution step time. A transformation from the aircraft's frame to the world frame is also calculated in these equations.

Table 1. **Summary of UAV actions for Target Finding and Tracking mission**

Action a	**Forward velocity** V_{r_f} (m/s)	**Lateral velocity** V_{r_l} (m/s)	**Heading change** $\Delta\Psi_a$ $(°)$	**Altitude change** Δz_a (m)
Forward	0.6	0	0	0
Steer 15	0.6	0	15	0
Steer 30	0.6	0	30	0
Steer 45	0.6	0	45	0
Steer -15	0.6	0	-15	0
Steer -30	0.6	0	-30	0
Steer -45	0.6	0	-45	0
Up	0	0	0	0.3
Down	0	0	0	-0.3
Hover	0	0	0	0
Roll left	0	0.6	0	0
Roll right	-0.6	0	0	0

The orientation of the aircraft is determined by its heading angle Ψ_r. The uncertainty in motion is included in the system by adding a small deviation to the heading angle σ_a using a normal probability distribution with mean value equal to the desired heading and restricted to the range $-2.0° < \sigma_a < 2.0°$, which represents the uncertainty in the yaw angle when executed by the control system.

$$\begin{bmatrix}\Delta x_{r_t}\\ \Delta y_{r_t}\\ \Delta z_{r_t}\end{bmatrix} = \begin{bmatrix}\cos(\Psi_{r_t}+\sigma_{r_t}) & -\sin(\Psi_{r_t}+\sigma_{r_t}) & 0\\ \sin(\Psi_{r_t}+\sigma_{r_t}) & \cos(\Psi_{r_t}+\sigma_{r_t}) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}V_{r_{f_t}}\Delta t\\ V_{r_{l_t}}\Delta t\\ \Delta z_r\end{bmatrix} \tag{3}$$

$$\begin{bmatrix}\Delta x_{r_{t+1}}\\ \Delta y_{r_{t+1}}\\ \Delta z_{r_{t+1}}\end{bmatrix} = \begin{bmatrix}x_{r_t}\\ y_{r_t}\\ z_{r_t}\end{bmatrix} + \begin{bmatrix}\Delta x_{r_t}\\ \Delta y_{r_t}\\ \Delta z_{r_t}\end{bmatrix} \tag{4}$$

Where x_{r_t}, y_{r_t} and z_{r_t} are the x, y and z aircraft coordinates at time t, $V_{r_{f_t}}$ and $V_{r_{l_t}}$ are forward and lateral velocities in the quad-rotor's frame at time t, and Ψ_{r_t} and σ_{r_t} are heading and heading deviation at time t.

Observation Model (O)

An observation for the POMDP model is composed of a) the estimated quad-rotor position in the world frame and b) the target's pose if it is detected by the downward looking camera. The UAV odometry reading whose uncertainty is approximated by generating a position for the quadrotor to be inside a cell of size (x_B, y_B, z_B). A typical cell size is $(1, 1, 1)m$. The size of this box, i.e. the amount of uncertainty in the measurement, can be changed and it is set as a parameter in the framework.

If the target is detected by the onboard downward looking camera, the AR Drone ROS driver provides the target position within the image. This position is transformed to a position in the world frame.

Downward Camera Field of View

The field of view (FOV) of the downward looking camera, shown in Fig.3 is modelled experimentally by repeated testing. This model is included in the observation model by equations 5, 7 and 6. The area covered by the UAV and camera field of view is proportional to the altitude, the UAV is flying at.

$$h_c = z_r * \alpha_h \tag{5}$$

$$w_c = z_r * \alpha_w \tag{6}$$

$$shift = z_r * \alpha_s \tag{7}$$

Where $\alpha_h = 0.56$, $\alpha_w = 0.96$ and $\alpha_s = 0.2256$, values are obtained by real flight tests.

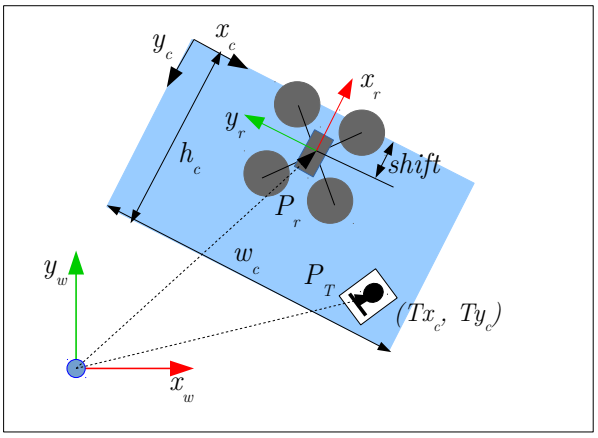


Figure 3. Field of view and frames

The downward looking camera is positioned towards the rear of the quad-copter, causing a shift in the FOV in the $-x$ direction in the UAV frame. When the target is detected the AR Drone ROS driver provides the position of target in the image frame, shown in black in Fig. 3. The target coordinates in the image are Tx_c, Ty_c.

Rewards and Cost Function (R)

The values of the reward and cost functions were selected as a result of tuning the system to resemble existing test cases for state of the art POMDP problems. This values had to be tuned by experimentation in simulated environments. The aircraft receives a high reward of 300 if it detects the target within the downwards looking camera field of view. Hitting an obstacle or going out of the scenario incur a penalty of 70 and every other movement will carry a cost of 5. A summary of the reward and cost function is shown in table 2.

5. RESULTS

Simulation

We conducted simulation results in order to explore the influence that the planning horizon length has in the quality of the solution. We ran 100 simulations for each case. Results indicate that the system spends in average slightly more than 60s to find a target within the confined space described. The results also indicate that for this particular problem the obtained accumulated discounted return, in average, increases for longer planning horizons. These results are important because they indicate that there is an optimal planning horizon for a specific POMDP formulated problem. The selected planning horizon affects the quality of the solution and this is a parameter can be tuned in order to increase the performance of the POMDP solver according to a specific problem.

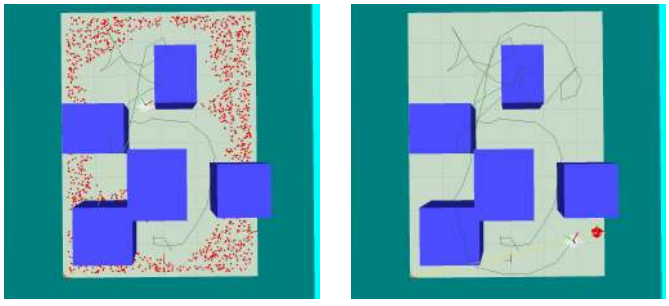
Table 2. Summary of the reward and cost function for the Target finding and tracking mission.

Reward/Cost	Value
Detecting the target	300
Hitting an obstacle	$(-)70$
Out of region	$(-)70$
Movement	$(-)5$

Table 3. Simulation results for target finding problem

UCB	Planning horizon	Flight time to target (s) (Number of steps)	Discounted return
300	25	65	-53
300	35	65	-49
300	45	64	-46
300	55	62	-36

We use 3d environment RVIZ to visualise the scenario, we model the obstacle as bounding boxes with increased size to account for the quad-copter size. We also use point clouds to visualise the particles in the POMDP model while it runs in simulation and in real flight. Fig 4(a) shows the belief-state after the UAV has explored the scenario, the knowledge that the system has of the target location (red particles) improves after exploration. Notice how the uncertainty is reduced in the areas that were already covered by the camera FOV. Fig. 4(b) shows the belief-state after the mission is accomplished by finding the target and its location is extracted from the image by transforming its position to world frame. See Appendix 1.



(a) Exploring scenario (b) Target Found

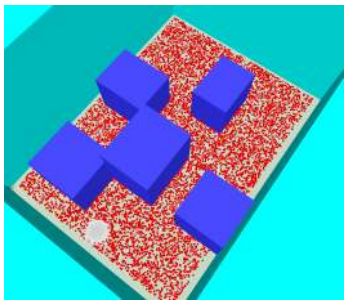
Figure 4. Example of a trajectory for UAV (black) in simulation. Red particles represent Knowledge of Target location and white particles represent UAV location.

Real Flight

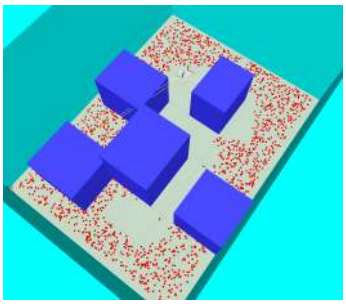
We conducted experiments to analyse the performance of the system flying in a real scenario. We tested a mission in which the quad-copter has to take off from an initial position and hover for 4 seconds to initialise its on-board sensors. The forward (pitch) and lateral (roll) velocity controllers start to actuate when taking off and the altitude controller sets an initial value of $0.7m$ above ground.

Fig. 5 shows the evolution of the belief-state in the POMDP model with the particles represented by point clouds. Fig. 5(a) shows the initial belief-state where the target location (red particles) is unknown and the particles are distributed all over the ground in the flying area. A uniform distribution for x and y coordinates and $z = 0.15m$ is used to generate the target initial location in the model. The UAV location is represented by the white particles around coordinates $(3.0, 0.5, 0.7)$ and using a gaussian distribution for x and y coordinates as described in Eq. 2. Once the UAV starts moving searching the target, the belief-state in the POMDP model updates according to the observations received. It can be seen that once the UAV flies over a region where it does not detect the target by the downward looking camera, the belief-state is updated and the uncertainty is reduced (Fig. 5(b)). Places with no red particles indicate that UAV has searched, and the target is not there, the update takes into account the field of view model of the downward looking camera, so that flying at higher altitudes increases the field of view.

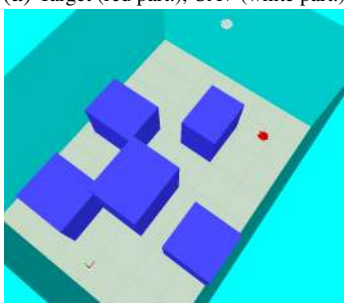
Once target detected, the belief-state is updated and the target's position is obtained based on its position within the image. A typical image of the target detected by the UAV is shown in Fig. 5(d).



(a) Target (red part.), UAV (white part.)



(b) Uncertainty reduced (red particles)



(c) Target found (Final belief-state)

(d) View of Target from UAV camera

Figure 5. Belief-state in UAV and target represented by particles for UAV (white) and target (red) location. Uncertainty in the belief-state is reduced once the UAV starts its exploration.

In order to have comprehensive test results, we placed the target at four different location for the UAV to search and find it. Figures 6, 7, 8 and 9 show the trajectories that the UAV flew in order to detect the target by its downward looking camera. The paths indicate the policy taken by the UAV, and shows that the system takes paths that are safe, avoiding hitting and obstacle by flying above them which also allows to have a bigger FOV. The system is also avoiding the space boundaries and prefers to fly closer to the center of the flying area.

Table 4. **Real flight results for Target finding problem**

Target's location (x, y, z)	**Success rate (%)**	**Flight time to target** (s) **(Number of steps)**	**Discounted return**
$(5.0, 1.0, 0.15)$	90	78	-95
$(5.0, 6.0, 0.15)$	60	124	-148
$(1.0, 6.5, 0.15)$	70	146	-157
$(1.0, 3.0, 0.15)$	100	40	60

The paths also illustrates the capabilities of the holonomic set of actions in the problem formulation. In all the paths shown in Figs. 6 to 9 the UAV ascends to have a bigger field of view, but keeps itself below the maximum altitude limit $(4m)$. It also takes advantage of the hover action in order to allow some settling time hovering in a position without moving. This might make paths look long and complex but allows the UAV to have some time at zero speed for acquiring better quality images with its downward looking camera.

Table 4 shows that when the target is located further from the initial UAV location, the number of steps to find it increases more than double, than when it is near the take off position. This is due to the fact that the UAV knowledge of the target's location has less uncertainty in regions near the initial UAV location due to the initial exploration.

Success rate was obtained for each of the target's locations by running the tests for 10 times (Table 4). The results indicate that success rate is proportional to the number of steps the UAV takes to find the target. For longer missions the uncertainty in the UAV localisation increases over time, increasing the difficulty to find the target and decreases the success rate. Overall, the system was able to find the target in average in 80% of the attempts.

6. Conclusions

The paper presents and approach and a framework for UAV navigation and target finding in GPS denied environments. The proposed POMDP model and system architecture is able to perform a full navigation and target finding mission using a low cost platform with only on-board sensors in a GPS-denied environment. The proposed system architecture incorporates a state-of-the-art on-line POMDP solver as an element of the system, separating the execution of actions and the perception computations from the solver. This means that the architecture is versatile and allows other types of planners, controllers and observers to be easily used in the system. A prediction of the observations based on the current and next states must be done in order to compensate for the time spent by the POMDP algorithm, updating the policy and calculating the next action.

The duration of each action depends on the system dynamics and overall, all the actions are modelled as step inputs to

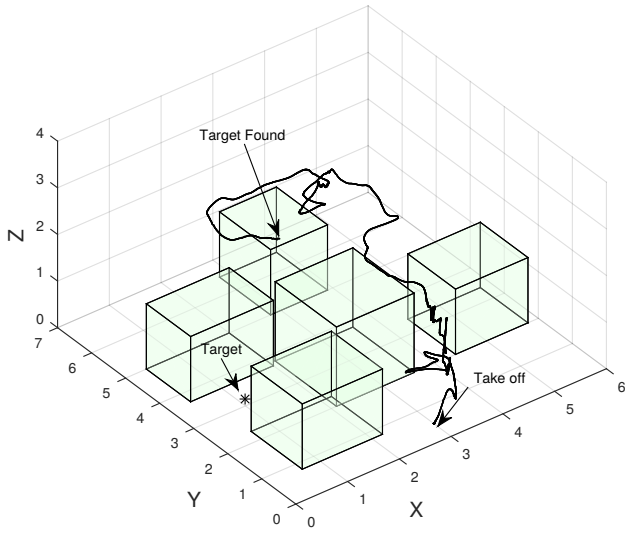


Figure 6. *Target located at (1.0, 3.0, 0.15). -UAV trajectory from take off until target is found.

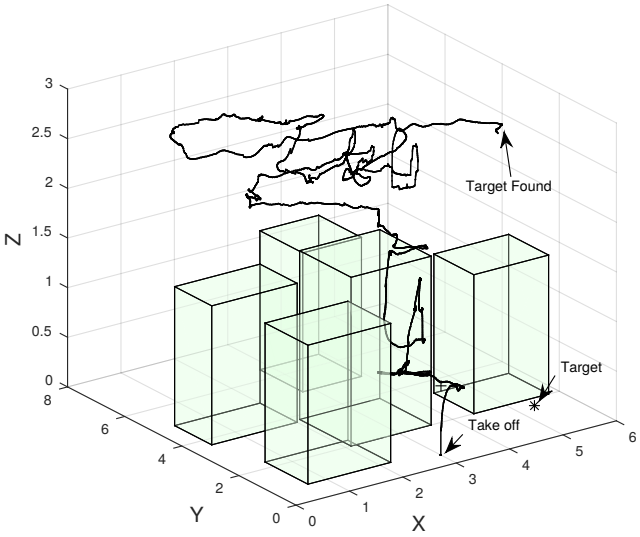


Figure 8. *Target located at (5.0, 1.0, 0.15). -UAV trajectory from take off until target is found.

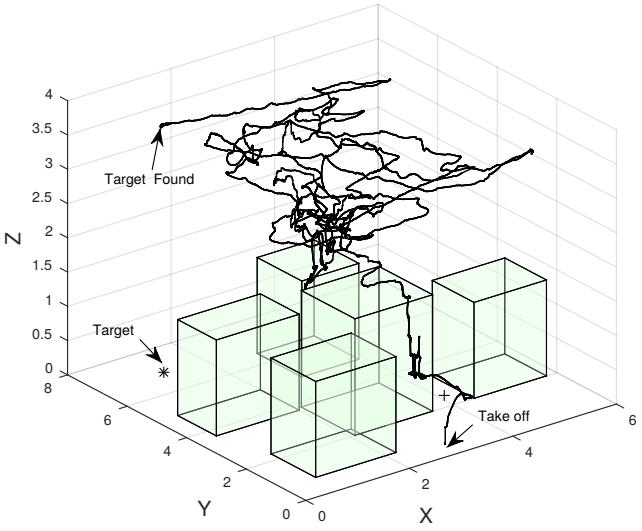


Figure 7. *Target located at (1.0, 6.5, 0.15). -UAV trajectory from take off until target is found. *Target position.

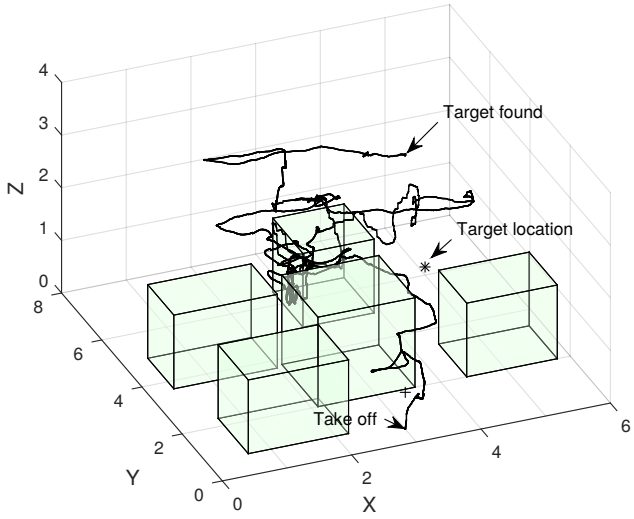


Figure 9. *Target located at (5.0, 6.0, 0.15). -UAV trajectory from take off until target is found.

PID controllers acting separately on each of the 4 degrees of freedom of the quad-copter used. This fact is taking into account to coordinate the timing for the calculation of the policy of the online POMDP solver, the execution of actions and the update of the belief-state in the POMDP solver in the system. Allowing the POMDP solver to calculate for longer times improves the quality of the policy, on the other hand, it increases the uncertainty in motion as the prediction time is longer. On the other hand, having shorter action execution time lowers the uncertainty in motion but increases the horizon search, i.e. the system will take more steps to accomplish the mission, thus making the solution more difficult to compute. Results in simulation and flight test show the robustness of the framework. Our system balances between allowing a sufficient time for the online POMDP solver to generate a good quality of the solution and not having a long horizon problem by allowing actions that execute for 1 second.

Current ongoing work focuses on introducing a source for localisation based on image processing in order to reduce the uncertainty and on extending the capabilities of the system to be able to track a moving target. Ongoing work also focuses in outdoor testing and exploration of the response of the system in an environment in windy conditions.

ACKNOWLEDGMENTS

The authors thank the Australian Research Centre for Aerospace Automation for its resources and indoor flying facilities. We thank Dr. Hanna Kurniawati (UQ), for support given on POMDP and the open source software ABT.

APPENDIX

1. FRAME TRANSFORMATIONS

In order to model the Field of View of the downward looking camera and for checking whether the Target is within it, some frame transformation must be done, as in Eqs. 8 and 9. $x_{T/R}$ and $y_{T/R}$ are x and y target coordinates seen from the quad-copter position.

$$x_{T/R} = \{x_T * cos(\Psi_r) + y_T * sin(\Psi_r) - x_r * cos(\Psi_r) - y_r * sin(\Psi_r)\} \quad (8)$$

$$y_{T/R} = \{-x_T * sin(\Psi_r) + y_T * cos(\Psi_r) - y_r * cos(\Psi_r) + x_r * sin(\Psi_r)\} \quad (9)$$

The target is within the FOV of the camera if it is inside the limits calculated in eqs. 5, 6.

When the target is detected we use equations for calculating the target position from its position in the image. First the target position in the quad-copter frame is calculated as in Eqs. 10 and 11. Then theses coordinates are transformed into the world frame by Eqs. 12 and 13.

$$x_{T/R} = shift - h_c * y_c/1000; \quad (10)$$

$$y_{T/R} = w_c/2 - w_c * x_c/1000; \quad (11)$$

$$x_T = x_{T/R} * cos(\Psi_r) - y_{T/R} * sin(\Psi_r) + x_r \quad (12)$$

$$y_T = x_{T/R} * sin(\Psi_r) + y_{T/R} * cos(\Psi_r) + y_r \quad (13)$$

REFERENCES

[1] L. F. Gonzalez, D. Lee, R. A. Walker, and J. Periaux, Optimal mission path planning (MPP) for an air sampling unmanned aerial system, Proceedings of the Austral-Asian Conference on Robotics and Automation, 2009.

[2] A. Malaver, F. Gonzalez, A. Depari, P. Corke, and N. Motta, Towards the development of a gas sensor system for monitoring pollutant gases in the low troposphere using small unmanned aerial vehicles, Proceedings of Workshop on Robotics for Environmental Monitoring, Sydney, Australia, vol. 19, 2012.

[3] A. Malaver, N. Motta, P. Corke, and F. Gonzalez, Development and Integration of a Solar Powered Unmanned Aerial Vehicle and a Wireless Sensor Network to Monitor Greenhouse Gases, Sensors, vol. 15, no. 2, pp. 40724096, 2015.

[4] F. Gonzalez, M. P. G. Castro, P. Narayan, R. Walker, and L. Zeller, Development of an autonomous unmanned aerial system to collect time-stamped samples from the atmosphere and localize potential pathogen sources, Journal of Field Robotics, vol. 28, no. 6, pp. 961976, Nov. 2011.

[5] L. F. Gonzalez, Robust evolutionary methods for multi-objective and multdisciplinary design optimisation in aeronautics, 2005.

[6] C. Geyer, Active target search from UAVs in urban environments, in Robotics and Automation, ICRA 2008. IEEE International Conference on, 2008, pp. 2366-2371.

[7] S. Waharte and N. Trigoni, Supporting Search and Rescue Operations with UAVs,, Emerging Security Technologies (EST), 2010 International Conference on, 2010, pp. 142-147.

[8] H. Al-Helal and J. Sprinkle, UAV Search: Maximizing Target Acquisition, in Engineering of Computer Based Systems (ECBS), 2010 17th IEEE International Conference and Workshops on, 2010, pp. 9-18.

[9] J.-P. Ramirez-Paredes, E. A. Doucette, J. W. Curtis, and N. R. Gans, Urban target search and tracking using a UAV and unattended ground sensors, in American Control Conference (ACC), 2015, 2015, pp. 2401-2407.

[10] R. R. Pitre, X. R. Li, and R. Delbalzo, UAV route planning for joint search and track missionsAn information-value approach, Aerospace and Electronic Systems, IEEE Transactions on, vol. 48, no. 3, 2012 pp. 2551-2565.

[11] S. Zhu, D. Wang, and C. B. Low, Ground Target Tracking Using UAV with Input Constraints, Journal of Intelligent & Robotic Systems, vol. 69, no. 1-4, pp. 417-429, Jan. 2013.

[12] R. He, A. Bachrach, and N. Roy, Efficient planning under uncertainty for a target-tracking micro-aerial vehicle, in Robotics and Automation (ICRA), 2010 IEEE International Conference on, 2010, pp. 18.

[13] D. Hsu, W. S. Lee, and N. Rong, A point-based POMDP planner for target tracking, ICRA 2008, Robotics and Automation, IEEE International Conference on, 2008, pp. 2644-2650.

[14] S. Ragi and E. Chong, UAV path planning in a dynamic environment via partially observable Markov decision process, IEEE Trans. Aerosp. Electron. Syst, vol. 49, pp. 23972412, 2013.

[15] C. Ponzoni Carvalho Chanel, F. Teichteil-Knigsbuch, and C. Lesire, POMDP-based online target detection and recognition for autonomous UAVs, in ECAI 2012, 20th European Conference on Artificial Intelligence, 2012, pp. 955-960.

[16] Smith T.; Simmons R. "Heuristic search value iteration for POMDPs," in Proceedings of the 20th conference on Uncertainty in artificial intelligence, 2004, 520-527.

[17] J. Pineau, G. Gordon, and S. Thrun, Anytime point-based approximations for large POMDPs *Journal of Artificial Intelligence Research*, **2006**, *27*, 335-380.

[18] Hauser K. Randomized belief-space replanning in partially-observable continuous spaces *Algorithmic Foundations of Robotics IX* **2011**, 193-209

[19] Silver D.; Veness J.; Monte-Carlo Planning in large POMDPs *Advances in Neural Information Processing Systems*, **2010**, *46*

[20] Kurniawati H.; Du Y.; Hsu D.; Lee W. S. Motion planning under uncertainty for robotic tasks with long time horizons *The International Journal of Robotics Research*, **2011**, *30* 308-323

[21] Kurniawati H.; Yadav V. An Online POMDP Solver for Uncertainty Planning in Dynamic Environment *International Symposium on Robotics Research*, **2014**

[22] Thrun, S.; Burgard W.; Fox D. *Probabilistic Robotics* Vol. 1. MIT press Cambridge, MA, 2005.

[23] Papadimitriou C.H.; Tsitsiklis J.N. The Complexity of Markov Decision Processes. *Mathematics of Operations Research*, **1987** *12* 441-50.

[24] Kocsis, Levente, and Szepesvari C. Bandit Based Monte-Carlo Planning. *Machine Learning*, Springer, 2006; pp. 282-93.

[25] AR Drone Autonomy Lab. A Ros Driver for the Parrot AR Drone UAV. http://ardrone-autonomy.readthedocs.org/en/latest/, 2014.

BIOGRAPHY

Fernando Vanegas *received his B.S. in Mechatronics Engineering from UMNG in 2004 and M.S. in Electrical Engineering from Halmstad University in 2008. He is currently a Ph.D. candidate in Robotics and Autonomous Systems at The Australian Research Centre for Aerospace Automation and Queensland University of Technology. His current research activities include path planning for UAVs in cluttered and uncertain environments using POMDPs.*

Felipe Gonzalez *is a Senior Lecturer (UAV) in the Science and Engineering Faculty, QUT and Theme leader for UAV Remote Sensing at ARCAA. He holds a BEng (Mech) and a PhD from the University of Sydney. His research explores bioinspired optimization, uncertainty based UAV path planning and UAVs for environmental monitoring. He leads the CRC Plant Biosecurity project Evaluating Unmanned Aerial Systems for Deployment in Plant Biosecurity and the Green Falcon Solar Powered UAV. Felipe is a Chartered Professional Engineer and member of professional organizations including the RAeS,IEEE and AIAA.*