IMAV2017-10
http://www.imavs.org/pdf/imav.2017.10

Towards a MOMDP model for UAV safe path planning in urban environment

Jean-Alexis Delamer[*], Yoko Watanabe[†]
ONERA - The French Aerospace Lab, 2 Avenue Edouard Belin, 31000 Toulouse
Caroline P. Carvalho Chanel[‡]
University of Toulouse, ISAE-SUPAERO, 10 Avenue Edouard Belin, 31400 Toulouse

Abstract

This paper tackles a problem of UAV safe path planning in an urban environment in which UAV is at risks of GPS signal occlusion and obstacle collision. The key idea is to perform the UAV path planning along with its navigation and guidance mode planning, where each of these modes uses different sensors whose availability and performance are environment-dependent. A partial knowledge on the environment is supposed to be available in the form of probability maps of obstacles and sensor availabilities. This paper proposes a planner model based on Mixed Observability Markov Decision Process (MOMDP). It allows the planner to propagate such probability map information to the future path for choosing the best action. This paper provides a MOMDP model for the planner with an approximation of the belief states by Mixture of Gaussians.

1 Introduction

These last years there has been a growing need of UAVs (Unmanned Aerial Vehicles) to accomplish distant missions in urban environments. The feasibility and success of these missions are directly linked to the UAV navigation capacity which relies on onboard sensors (e.g., GPS, vision, . . .). Availability and performance of these navigation sensors may depend on the environment. For example, GPS precision can be degraded due to signal occlusion or multi-path effect in a cluttered environment. Applicability and accuracy of vision-based approaches depend on textures of an image. All of these environment-dependent sensor availabilities and precision affect the navigation performance and thus the safety of the path.

The community has proposed different frameworks and algorithms to solve vehicle safe path planning problem in a cluttered and continuous environment. A method suggested in [1] computes an efficient collision-free path for MAV (Micro Aerial Vehicles) while taking into account uncertainty issued by an onboard camera-based localization. It applies the RRBT (Rapidly-exploring Random Belief Trees, [2]) algorithm. A method proposed by [3] estimates the probability of collision under Gaussian motion and sensor uncertainty. The developed algorithm truncates a state distribution in function of the environment and propagates this truncated state distribution along a path. Other frameworks consider dynamic path being able to self-adapt in function of the events. One could cite the work of [4] that proposes a method based on POMDP (Partially Observable Markov Decision Process) for autonomous vehicles in real situations.

Most of these works do not take into account the variance of sensor availability and precision in function of the environment. Furthermore, the navigation system and its associated uncertainty are often treated in a deterministic manner in the path planner, and a choice of navigation sensors is not included in the elements to be planned.

This paper proposes a planner model which enables the planner to incorporate a priori partial information of the environment-related elements (obstacles, sensor availabilities) in the probabilistic state transition. In this work, such information is given in a form of probability grids which are overlaid on the continuous environment space (Figure 1). Depending on the availability of the sensors, different navigation and guidance modes can be selected, resulting in different localization and path execution error propagation. The planner will be designed to compute a policy which will allow, at each instant, the UAV to choose the best motion-direction and the adapted sensors (i.e. navigation and guidance modes), with regard to the mission efficiency (minimum flight time/length) and safety (minimum collision risk).

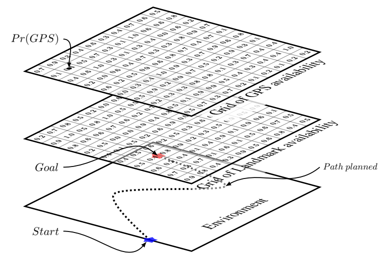


Figure 1: Continuous state space with discretized sensor availabilities maps

[*]Information Processing and Systems Departement (DTIS), email address: jean-alexis.delamer@onera.fr
[†]Information Processing and Systems Department (DTIS), email address: yoko.watanabe@onera.fr
[‡]Design and Control of Aerospace Vehicles Department (DCAS), email address:Caroline.CHANEL@isae.fr

The main contribution of this paper is providing a model which combines the state transition function of the decision process to the localization and path execution error propagation function. Furthermore, the proposed model enables the planner to propagate a priori knowledge on the availability of the navigation sensors on the future path. Considering an uncertainty model propagation, we have chosen to lean our planning model on the Markov Decision Process [5] and more precisely on its extension: Mixed Observability Markov Decision Processes (MOMDP)[6]. The paper also proposes to apply Machine Learning technique to approximate the belief states by a Gaussian mixture.

2 Problem statement

2.1 System architecture and time differentiation

The architecture of the overall system considered in this problem combines the vehicle Guidance, Navigation and Control (GNC) transition model and the MOMDP planning model (Figure 2). The GNC transition model includes the vehicle motion model, onboard sensor models, and flight control system. The policy given by the MOMDP planner takes as inputs the probability distribution over the current state (called belief state) b_{s_c} and a vector of boolean on sensor availabilities s_v. And it will return an action a which selects the motion direction, and the navigation and guidance modes to realize it. The belief state update is performed with an observation s'_v after GNC state transition. Then this new belief state is used by the policy to define the next action. The MDP

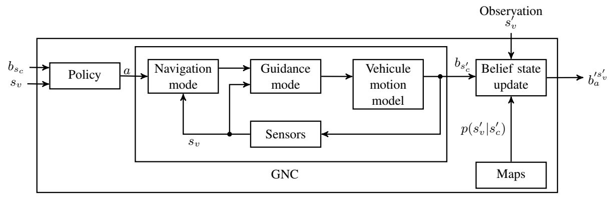


Figure 2: Architecture diagram

formalisms, in their majority and most of the variants, suppose that actions are accomplished instantly. However, in this problem, the actions are durative and the state space is continuous implies that the real state (such as sensor availability) changes during the actions. Therefore, the unit of time of the GNC transition model will be distinguished from that of the MOMDP planning model named *epoch*. The MOMDP planning model works at a lower frequency than the GNC system. Thus an epoch is equivalent to several units of time of the GNC transition model. It is for lowering the complexity of the planning algorithm by reducing the total number of actions to complete the task. The GNC transition unit of time is denoted by k and the planning epoch by t in this paper (Figure 3).

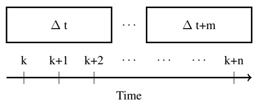


Figure 3: Difference between the units of time

2.2 GNC transition model

This section presents the vehicle GNC model which constructs a state transition model in the planning model.

2.2.1 State transition model

The UAV state $x = \begin{bmatrix}\mathcal{X}^T & \mathcal{V}^T & b_a^T\end{bmatrix}^T$ is defined respectively by its positon, velocity and the accelerometer bias. Then the state transition can be defined such as :

$$x_{k+1} = \Phi x_k + B a_k + v_{k+1} \tag{1}$$

where a_k is the acceleration, $v_{k+1} \sim N(0, Q)$ is the discretized process noise and

$$\Phi = \begin{bmatrix} I & \Delta t I & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, B = \begin{bmatrix} \frac{\Delta t^2}{2} I \\ \Delta t I \\ 0 \end{bmatrix}$$

2.2.2 State estimator

x is estimated by sensor measurements available onboard (Figure 2). The state estimator is based on an EKF (Extended Kalman Filter, [7]) which proceeds in two steps, firstly with the INS prediction then the correction by the sensor measurements, if available. The INS measurement integration enables a high-frequency state estimation, but it suffers from the drift in the estimate. In order to correct such drift, other sensors (e.g. GPS, Vision) are fused with INS through the second correction step.

INS Prediction : The accelerometer measurement a_{IMU_k} is used to propagate the estimated state.

$$a_{\text{IMU}_k} = R_{BI_k}(a_k - g) + b_{a_k} + \xi_{\text{IMU}_k} \tag{2}$$

where R_{BI_k} is a rotation matrix from the inertial to the UAV body frames provided by the INS, g is the gravity vector and $\xi_{\text{IMU}} \sim N(0, R_{\text{IMU}})$ is the IMU measurement noise. According to the process model (1), the estimated state $\hat{x}_k$ is propagated to :

$$\hat{x}_{k+1}^- = \Phi \hat{x}_k + B\left(R_{BI_k}^T\left(a_{\text{IMU}_k} - \hat{b}_{a_k}\right) + g\right) \tag{3}$$

Then the state prediction error is given by :

$$\tilde{x}_{k+1}^- = x_{k+1} - \hat{x}_{k+1}^- = \left(\Phi - \Delta\Phi_k^a\right)\tilde{x}_k + v_{k+1} - BR_{BI_k}^T \xi_{\text{IMU}_k} \tag{4}$$

where $\Delta\Phi_k^a = BR_{BI_k}^T \begin{bmatrix} 0 & 0 & I \end{bmatrix}$. The associated error covariance is given by :

$$P_{k+1}^- = (\Phi - \Delta\Phi_k^a) P_k (\Phi - \Delta\Phi_k^a)^T + Q + \tilde{R}_{\mathrm{IMU}_k} \quad (5)$$

where $\tilde{R}_{\mathrm{IMU_k}} = BR_{BI_k}^T R_{\mathrm{IMU}} R_{BI_k} B^T$. For simplicity, we can consider the case of $R_{\mathrm{IMU}} = \sigma_{IMU}^2 I$ and hence $\tilde{R}_{\mathrm{IMU}} = BR_{\mathrm{IMU}} B^T$ remains constant for all k.

GPS correction : When GPS is available at t_{k+1}, the predicted state (1) can be corrected by using its position and velocity measurement $z_{\mathrm{GPS}_{k+1}}$.

$$z_{\mathrm{GPS}_{k+1}} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \end{bmatrix} x_{k+1} + \xi_{\mathrm{GPS}_{k+1}} = H_{\mathrm{GPS}} x_{k+1} + \xi_{\mathrm{GPS}_{k+1}}$$

where $\xi_{\mathrm{GPS}} \sim N(0, R_{\mathrm{GPS}})$ is the GPS measurement error. Then, the estimated state is corrected such as :

$$\hat{x}_{k+1} = \hat{x}_{k+1}^- + K_{\mathrm{GPS}_{k+1}} H_{\mathrm{GPS}} \left(z_{\mathrm{GPS}_{k+1}} - H_{\mathrm{GPS}} \hat{x}_{k+1}^- \right) \quad (6)$$

where $K_{\mathrm{GPS}_{k+1}} = P_{k+1}^- H_{\mathrm{GPS}}^T \left(H_{\mathrm{GPS}} P_{k+1}^- H_{\mathrm{GPS}}^T + R_{\mathrm{GPS}} \right)^{-1}$ is a Kalman gain. Then the error estimate and its covariance are given by :

$$\begin{aligned} \tilde{x}_{k+1} &= \left(I - K_{\mathrm{GPS}_{k+1}} H_{\mathrm{GPS}} \right) \tilde{x}_{k+1} - K_{\mathrm{GPS}_{k+1}} \xi_{\mathrm{GPS}_{k+1}} \\ P_{k+1} &= \left(I - K_{\mathrm{GPS}_{k+1}} H_{\mathrm{GPS}} \right) P_{k+1}^- \end{aligned} \quad (7)$$

Landmark by an onboard camera : By image processor, a landmark with a known position $\mathcal{X}_{\mathrm{LM}}$ is visible and detectable at t_{k+1}, it can be used to correct the predicted state (1). By assuming a pin-hole camera model, the pixel-coordinates measurement is given by :

$$z_{\mathrm{LM}_{k+1}} = C \frac{\mathcal{X}_{\mathrm{LM}_{k+1}}^C}{e_3^T \mathcal{X}_{\mathrm{LM}_{k+1}}^C} + \xi_{\mathrm{LM}_{k+1}} = h_{\mathrm{LM}}(x_{k+1}) + \xi_{\mathrm{LM}_{k+1}} \quad (8)$$

where C is a known camera matrix, $\mathcal{X}_{\mathrm{LM}_{k+1}}^C = R_{CB} R_{BI_{k+1}} (\mathcal{X}_{\mathrm{LM}} - \mathcal{X}_{k+1})$ and $\xi_{\mathrm{LM}} \sim N(0, R_{\mathrm{LM}})$ is the landmark image-detection error in pixels. R_{CB} is a camera orientation with respect to the UAV body. An EKF can be applied, and similarly to (7), the resulting estimation error and its covariance matrix are given by :

$$\begin{aligned} \tilde{x}_{k+1} &= \left(I - K_{\mathrm{LM}_{k+1}} H_{\mathrm{LM}_{k+1}} \right) \tilde{x}_{k+1} - K_{\mathrm{LM}_{k+1}} \xi_{\mathrm{LM}_{k+1}} \\ P_{k+1} &= \left(I - K_{\mathrm{LM}_{k+1}} H_{\mathrm{LM}_{k+1}} \right) P_{k+1}^- \end{aligned} \quad (9)$$

where $H_{\mathrm{LM}_{k+1}}$ is a Jacobian matrix of the nonlinear measurement function $h_{\mathrm{LM}}(x_{k+1})$ evaluated at $x_{k+1} = \hat{x}_{k+1}^-$. It should be noted that $H_{\mathrm{LM}_{k+1}}$ thus depends on the predicted state $\hat{x}_{k+1}^-$, while H_{GPS} does not.

INS-only solution : If no correction is made with neither sensors, then state estimate at t_{k+1} is given by:

$$\begin{aligned} \tilde{x}_{k+1} &= \tilde{x}_{k+1}^- \\ P_{k+1} &= P_{k+1}^- \end{aligned} \quad (10)$$

2.2.3 Guidance laws

Given a desired velocity $\mathcal{V}_{\mathrm{ref}}$, the following linear guidance law can be applied to realize it :

$$a_k = \hat{K}_p \mathcal{V}_{\mathrm{ref}} - K_d(\hat{\mathcal{V}}_k - \mathcal{V}_{\mathrm{ref}}) = K_p \mathcal{V}_{\mathrm{ref}} - K_d \hat{\mathcal{V}}_k \quad (11)$$

where K_p,$K_d > O$ are some control gains and $\hat{\mathcal{V}}_k$ is the estimated UAV velocity at instant t_k.

This paper considers two different guidance modes for $\hat{\mathcal{V}}_k$ in (11). The first mode uses the state estimation result from the navigation module (Section 2.2.2), while the second mode uses some sensor measurements directly. The former mode corresponds to a conventional absolute guidance approach such as waypoint tracking, and the latter to a sensor-based relative guidance method such as visual servoing.

Absolute guidance : $\hat{\mathcal{V}}_k = \begin{bmatrix} 0 & I & 0 \end{bmatrix} \hat{x}_k$, where $\hat{x}_k$ is the estimated state from Section 2.2.2. Then, x_{k+1} can be obtained by substituing this guidance law (11) into the discretized process model (1) :

$$x_{k+1} = (\Phi - \Delta\Phi^{\mathcal{V}}) x_k + BK_p \mathcal{V}_{\mathrm{ref}} + \Delta\Phi^{\mathcal{V}} \tilde{x}_k + v_{k+1} \quad (12)$$

where $\Delta\Phi^{\mathcal{V}} = BK_d \begin{bmatrix} 0 & I & 0 \end{bmatrix}$. Hence, the state x_{k+1} follows the normal distribution as below.

$$\begin{aligned} x_{k+1} &\sim N((\Phi - \Delta\Phi^{\mathcal{V}}) x_k + BK_p \mathcal{V}_{\mathrm{ref}}, \Delta\Phi^{\mathcal{V}} P_k \Delta\Phi^{\mathcal{V}^T} + Q) \\ &= N(\bar{x}_{k+1|k}, \tilde{Q}_{k+1}^a) \end{aligned} \quad (13)$$

where the covariance $\tilde{Q}_{k+1}^a$ becomes a function of the estimation error covariance P_k.

Sensor-based relative guidance : In the sensor-based relative guidance mode, $\hat{\mathcal{V}}_k$ in (11) is directly given from some onboard sensors such as optical flow. Let us assume the measurement error $\tilde{\mathcal{V}}_k = (\mathcal{V}_k - \hat{\mathcal{V}}_k) \sim N(0, R_{\mathcal{V}_k})$. Then, similarly to (12),

$$\begin{aligned} x_{k+1} &= (\Phi - \Delta\Phi^{\mathcal{V}}) x_k + BK_p \mathcal{V}_{\mathrm{ref}} + BK_d \tilde{\mathcal{V}}_k + v_{k+1} \\ &\sim N(\bar{x}_{k+1|k}, BK_d R_{\mathcal{V}_k} K_d^T B^T + Q) = N(\bar{x}_{k+1|k}, \tilde{Q}_{k+1}^s) \end{aligned} \quad (14)$$

where the covariance $\tilde{Q}_{k+1}^s$ is now independent from P_k.

2.2.4 State probability density function

Given an initial state $x(t_0) = x_0$ and initial estimation error covariance P_0 such as $\tilde{x}_0 \sim N(0, P_0)$. As defined later in

Section 3.3.2, an action a in the planner model corresponds to a combination of the direction of desired motion $\mathcal{V}_{\text{ref}}$, the navigation mode and the guidance mode. For a given action a, it is possible to obtain the distribution of the next state $x_1 = x(t_1) = x(t_0 + \Delta t)$ knowing x_0.

$$f_X(x_1|x_0) \sim N\left(\bar{x}_{1|0}, \tilde{Q}_1\right) \tag{15}$$

where $\tilde{Q}_1$ is either $\tilde{Q}_1^a$ in (13) or $\tilde{Q}_1^s$ in (14) depending on the selected guidance mode. At the same time, the state estimation error covariance is updated to P_1 by using the selected navigation mode (7, 9 or 10).

Now recall from Section 2.1 that the system's transition model and the planning model do not have the same time unit. It means that a single state transition $s = s_0$ to $s' = s_1$ with an action a in the planner corresponds to several state transitions x_0 to x_n in the system's transition model. Thus, the state transition must continue further up to $n > 1$ with the same action a. The conditional state distribution at t_k knowing x_0 can be obtained sequentially as follows.

$$f_X(x_k|x_0) \sim \int f_X(x_k|x_{k-1}) f_X(x_{k-1}|x_0) dx_{k-1}$$

where $f_X(x_k|x_{k-1}) \sim N(\bar{x}_{k|k-1}, \tilde{Q}_k)$. In parallel, the Kalman filtering process is repeated k times to obtain P_k. When $\tilde{Q}_k$ and P_k do not depend on the state x_{k-1}, the integration above will result in a normal distribution:

$$f_X(x_k|x_0) = N(\bar{x}_{k|0}, \tilde{\Sigma}_k), \quad k > 1 \tag{16}$$

The derivation of the state distribution (16) becomes more complex in a case of having a dependency of $\tilde{Q}_k$ and P_k on the state x_{k-1}. To avoid this complication, $\tilde{Q}_k$ and P_k can be approximated by those evaluated at the expected state $\bar{x}_{k-1|0}$. Then, the state transition function in the planning model can be given by (16) when $k = n$.

Besides, it is hard to know a priori if a selected sensor will be available for the entire action. To simplify we will suppose that if the sensor is available at the end of the action, the sensor was available during the entire action.

In result, the state transition function from the state $s = s_0$ to $s' = s_1$ can be re-wtitten with a notation from the planning model as below.

$$f_S(s' = s_1|s = s_0) = f_X(x_n|x_0) \sim N(\bar{x}_{n|0}, \tilde{\Sigma}_n) = N(\bar{s'}, \Sigma') \tag{17}$$

It is worth emphasizing here that this equation will be the only link to the GNC's transition model with the MOMDP planning model (Section 3.3).

2.3 Environment Maps

The planner will use a priori partial knowledge on the environment. This 3D environment map is discretized into cells. Let us denote the c_i, the i-th cell of the map, then a probability that the cell c_i is occupied, i.e., obstacle, is given as $p(Collision|c_i)$. Similarly, sensor availability maps are provided as a set of probabilities that a sensor is available at each cell c_i. They can be generated by considering the corresponding sensor characteristics in relation with the environment (given by the occupancy map). For example, GPS performance suffers from its signal occlusion or multi-path effect due to surrounding obstacles. It is common to measure the performance of GPS by metrics called DOPs (Dilutions of Precision) which corresponds to a standard deviation of the measured position error $\tilde{\mathcal{X}}_{\text{GPS}}$. Figure 4 shows an example of 3D Position DOP (PDOP) map at a given altitude created by a GNSS simulator on an obstacle map of San Diego city. In this paper, this PDOP map is transformed to GPS availability map by setting a maximum allowable position error threshold ϵ. A probability of GPS availability at each cell c_i is calculated by $p(GPS|c_i) = p(\epsilon_{\mathcal{X}_{\text{GPS}}}(c_i) < \epsilon)$.

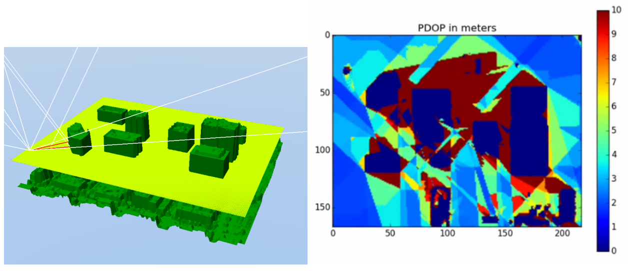


Figure 4: Example of GPS PDOP map

Availability of the navigation mode using landmark detection $p(LM|c_i)$ is obtained in function of a camera's field-of-view and detection performance of an image processor. Likewise, availability of the sensor-based relative guidance (e.g., wall following) is conditioned by a sensing range concerning an object-of-interest (wall). Its availability map as a set of probabilities $p(Wall|c_i)$ is supposed to be given.

3 Planning model

The objective of this paper is to propose a decision framework for computing the safest and shortest path. To achieve this goal, the transition model must be accurate. That is why the GNC transition model is integrated in the planning model.

3.1 Why MOMDP ?

This work is about computing a policy which gives a UAV at each epoch the best action to execute. Therefore, planning can be associated with decision-making which is the cognitive process of choosing an action to perform confronted to a situation. In real life problem, the result of an action is often synonymous of uncertainty that must be considered in the problem. POMDPs and variants provide several frameworks to model sequential decision problem under uncertainty and partial observability. The idea behind the POMDPs is that the state is not known, but several observations are possible for each state with a distinct probability. Then these observations are used to update its belief state. The Mixed Observability Markov Decision Process (MOMDP) proposed by [8] and [6] is a variant of the POMDPs. The state is not partially observable, but a part of the state is known at each epoch. In this problem, the UAV always knows the current sensor availabilities which can be considered as a part of the state. Consequently, MOMDP can be used in this problem.

3.2 *Recall on MOMDPs*

The MOMDP explores a particular structure where certain state variables are fully observable. This factored model leads to a significant time gain in policy computation, improving the efficiency of a point-based algorithms. A MOMDP is a tuple $(\mathcal{S}_v, \mathcal{S}_c, A, \mathcal{O}, T, R, b_0, \gamma)$, where:

- $\mathcal{S}_v$ and $\mathcal{S}_c$ are respectively the bounded set of fully observable state variables and the bounded set of partially observable state variables;
- $\mathcal{A}$ is a finite set of actions;
- Ω is a finite set of observation;
- $T(s_v, s_c, a, s'_v, s'_c) \to [0;1]$ is a transition function;
- $\mathcal{O}(s', a, o) \to [0;1]$ is an observation function such as $p(o_v, o_c|a, s'_v, s'_c)$
- $R : \mathcal{S}_v \times \mathcal{S}_c \times A \to \mathbb{R}$ is a reward function associated with a state-action pair
- $b_0 = (s_v^0, b_{c,0})$ an initial belief state, where $b_{c,0}$ is the initial probability distribution over the partially observable states, conditionned by s_v^0, the fully observable initial states.
- $\gamma \in [0, 1[$ is the discount factor.

The belief state b is noted by (s_v, b_c), and $\mathcal{B}_c$ is the belief state space of s_c conditioned by s_v : $\mathcal{B}_c(s_v) = \{(s_v, b_c), b_c \in \mathcal{B}_c\}$. $\mathcal{B}_c(s_v)$ is a sub-space of $\mathcal{B}$, such that $\mathcal{B} = \bigcup_{s_v \in \mathcal{S}_v} \mathcal{B}_c(s_v)$.

Solving MOMDPs consists in finding *a set of policies* $\pi_{s_v} : \mathcal{B}_c \to A$, which maximize the criterion :

$$\pi^*_{s_v} \leftarrow \underset{\pi_{s_v} \in \Pi}{\arg\max}\, E_{\pi_{s_v}} \left[\sum_{t=0}^{\infty} \gamma^t r((s_v^t, b_c^t), \pi((s_v^t, b_c^t))) \middle| b_0 = (s_v^0, b_{c,0}) \right] \tag{18}$$

For more details about MOMDP, please see [8].

3.3 *MOMDP planning model*

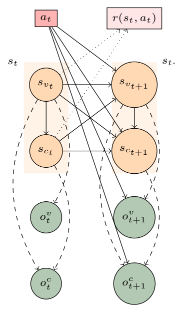
Considering the differences between the problem presented in this paper and a standard MOMDP problem from the litterature, some modifications are made to the MOMDP formalism. Our MOMDP is a tuple $(\mathcal{S}_v, \mathcal{S}_c, \mathcal{A}, \Omega, \mathcal{T}, \mathcal{O}, \mathcal{C}, b_0)$:

- $\mathcal{S}_v$ and $\mathcal{S}_c$ are respectively bounded set of fully observable states and *non* observable states.
- $\mathcal{T}$: The state transition function composed of two functions:
  - $T_{\mathcal{S}_c} : \mathcal{S}_c \times \mathcal{S}_v \times A \times \mathcal{S}_c \to [0;1]$ a transition function such as : $T_{\mathcal{S}_c}(s_c, s_v, a, s'_c) = p(s'_c|s_c, s_v, a)$.
  - $T_{\mathcal{S}_v} : \mathcal{S}_v \times \mathcal{S}_c \to [0;1]$ a transition function such as : $T_{\mathcal{S}_v}(s'_c, s'_v) = p(s'_v|s'_c)$;
- $\mathcal{O} : \Omega \times \mathcal{S}_c \to [0;1]$: observation function such as :

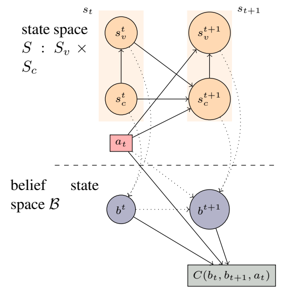
$$O(o, a, s'_c, s'_v) = p(o|s'_c, s'_v, a) = \begin{cases} 1 & o = s'_v \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

- $\mathcal{C} : \mathcal{B} \times \mathcal{B} \times A \to \mathbb{R}$: the cost function, where $\mathcal{B}$ is the belief state space defined on $\mathcal{S} = \mathcal{S}_v \times \mathcal{S}_c$.

The set of observations Ω is equal to $\mathcal{S}_v$ and consequently the observation function $\mathcal{O}$ is deterministic since $o = s'_v$ in (19). Moreover, the Bayesian dependancy in our model changes from a MOMDP proposed in [8], s'_v depends on s'_c and s'_c depends on s_v and s_c, therefore it depends on the position of the vehicle in the environment (Figure: 5).



(a) Transition model for a MOMDP



(b) MOMDP model modified with the complete belief state space $\mathcal{B}$.

Figure 5: Difference between the transition model

3.3.1 State space of the decisional problem

A state s is composed of two sets : $s_v \in \mathcal{S}_v$ and $s_c \in \mathcal{S}_c$. The state space $|\mathcal{S}| = |\mathcal{S}_v| \times |\mathcal{S}_c|$ represents all the possible states. It must be noted that for our model it is the entire state space $\mathcal{S}$ that is partially observable. More specifically, we define $s_c = x$ (as defined in section 2.2.1), which is defined on a *non observable continuous bounded space*.

s_v is a vector containing the fully observable booleans of sensor availability $[0;1]$, a boolean on the collision, and P the localization error covariance matrix. In our example, s_v becomes as follows.

$$s_v = \begin{bmatrix} FlagGPS \\ FlagLandmark \\ FlagWallFollowing \\ FlagColision \\ P \end{bmatrix}$$

P is needed by the state probability density function (15). Then, it is necessary to keep P in the state.

3.3.2 Action space of the decisional problem

In contrary to the vehicle state space, the action space is discrete and the action a is defined as a tuple (d, m_n, m_g) composed by three action variables : d, m_n and m_g. $d \in D$ is the desired directions of motion which specifies $\mathcal{V}_{\text{ref}}$ in (11), $m_n \in \mathcal{M}_n$ designates the navigation mode available on the vehicle and $m_g \in \mathcal{M}_g$ designates the guidance mode. In our problem, we define the navigation modes $\mathcal{M}_n = \{INS - only, GPS, Landmark\}$ and the guidance modes $\mathcal{M}_g = \{Waypoint\ tracking, Wall\ following\}$ which corresponds to (10, 7, 9) and (13, 14) respectively.

3.3.3 Observation space and observation function of the decision problem

As explained, a set of observation Ω is considered equal to $\mathcal{S}_v$ and consequently the observation function is deterministic since $p(o|s'_c, s'_v, a) = 1$. In contrary to the standard case of POMDP, the agent does not receive any observation in the classical meaning in our problem. Considering the navigation error propagation, the UAV state s_c is estimated by noisy (and biased for INS) sensor measurements. These measurements are used to estimate the non observable state variables. Unfortunately, we cannot consider these measurements as observations, because it is hard to approach a probabilistic function allowing to anticipate the future measurements. In this sense, s_c is considered as a non observable state variable in the decision problem.

3.3.4 Transition function

The transition function from one state to other is composed of two functions :

- $T_{\mathcal{S}_c}$ a transition function such as :

  $$T_{\mathcal{S}_c}(s_c, s_v, a, s'_c) = f_{s'_c}(s'_c|s_c, s_v, a) \sim N(\bar{s'_c}, \tilde{\Sigma}'(s_v))$$

  As previously developed in (17), the probability distribution of a predicted state s'_c follows a normal distribution $N(\bar{s'_c}, \tilde{\Sigma}'(s_v))$ which is a function of the previous state s_c, the action a. Figure 6a illustrates this function.

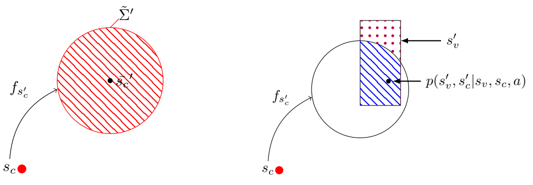
- $T_{\mathcal{S}_v}$ is a transition function such as $T_{\mathcal{S}_v}(s'_c, s'_v) = p(s'_v|s'_c)$. It represents the transition to s'_v and depends on the sensor availability map and therefore depends only on the next state s'_c. Concretely

$$p(s'_v|s'_c) = \prod_{i=1}^{|\mathcal{S}_v \setminus P|} p(s'_v(i)|s'_c) \tag{20}$$

where $|\mathcal{S}_v \setminus P|$ is the number of sensors onboard and $s'_v(i)$ is the i-th sensor. Figure 6b illustrates this function.

Then, we define the transition function :

$$\begin{aligned} T(s_v, s'_v, a, s'_c, s_c) &= T_{\mathcal{S}_c}(s_c, s_v, a, s'_c) \times T_{\mathcal{S}_v}(s'_c, s'_v) \\ &= p(s'_v|s'_c) f_{s'_c}(s'_c|s_c, s_v, a) \end{aligned} \tag{21}$$



(a) Propagation of s'_c from s_c with an action a : $T_{\mathcal{S}_c}$

(b) Reduction of the next reachable states in function of s'_v : $T_{\mathcal{S}_v}$

Figure 6: Illustration of the two transition functions : $T_{\mathcal{S}_c}$ and $T_{\mathcal{S}_v}$

3.3.5 Belief state

The belief state condenses all the accumulated information along the path of length N, which is the complete information history h defined by : $h = \{a_0, o_1, a_1, o_1, ..., a_{N-1}, o_N\}$ A belief state b is a probability density function over the possible states at each time step t,

$$b(s^t) = f_s(s = s^t), \forall s^t \in \mathcal{S} \tag{22}$$

computed from the state transition model with selected navigation and guidance modes. This belief state is updated supposing the Bayes rule after each action a done and at each observation o perceived. By factoring the state space according to our model we obtain $b = (s_v, b_{s_c})$, with $b_{s_c} = f_{s_c}(s_c|s_v)$.

3.3.6 Belief state update

As explained, the belief state must be updated after each action associated with the transition function in Section 3.3.4. The update is done by three sub-functions. The first function corresponds to the system's dynamic model which propagates the current belief state b_{s_c} to the futur belief state named $b_{s'_c}$ with a chosen action a :

$$b_{s'_c}(s'_c) = \int f_{s'_c}(s'_c|s_c, s_v, a) b_{s_c}(s_c) ds_c \tag{23}$$

The second is the probability of s'_v that is computed based on $b_{s'_c}$:

$$p(s'_v|b, a) = \sum_{i=0}^{|G|} p(s'_v|s'_c \in c_i) p(s'_c \in c_i|b, a) \tag{24}$$

where c_i corresponding to the i^{th} cell of the sensor availablity map and $|G|$ is the number of cells in the map. The third step corresponds to the "Belief state update" of Figure 2. $b'^{s'_v}_{s'_c,a}$ is computed in function of s'_v, the completely observable state.

$$b'^{s'_v}_{s'_c,a}(s'_c) = \frac{p(s'_v|s'_c)\int f_{s'_c|s_c,s_v,a}(s'_c|s_c,s_v,a)b_{s_c}(s_c)ds_c}{\sum\limits_{i=0}^{|G|} p(s'_v|s'_c \in c_i)p(s'_c \in c_i|b,a)} \tag{25}$$

Therefore the updated belief state is defined as $b'^{s'_v}_a$ and is derived as $b'^{s'_v}_a = (s'_v, b'^{s'_v}_{s'_c,a})$.

3.3.7 Gaussian Mixtures Learning

The representation of a belief state as a Gaussian mixture could simplify the calculation in the planning model. If we consider b^0 as a Gaussian (or a Gaussian mixture), the update of a belief state in Section 3.3.6 raises a problem. Indeed, in (25), the conditional probability $p(s'_v|s'_c)$ (given by the probability grid) make the belief non Gaussian and prevent us to use the initial Gaussian distribution given by (23) (considering b_{s_c} as a gaussian mixture).

A solution proposed in this paper to overcome the problem is to use a machine learning algorithm to approximate the non-Gaussian belief state by a Gaussian mixture model (GMM). More precisely, an "Expectation-Minimization" (EM) algorithm [10] is used to learn the GMM. The idea of the belief state update is unchanged, but the new probability distribution function of the belief state results in GMM learnt with the EM algorithm instead of (25).

Figure 7 illustrates this idea on a given path, the transition between two belief states in this example is just a simple propagation of the Gaussian (or the Gaussian mixture). The belief state update is done by an observation that the GPS is always available. Figure 7a is the GPS availability map used in this example. The graphic on the top of Figure 7b represents the belief states propagated along a given path without any update. The second represents the propagation of each belief states with the update by the observation, approximated by the Gaussian mixture learning.

The effect of the Gaussian mixture learning is more visible nearby an obstacle. The probability on the GPS availability is low, and the original belief states without update are partially in a collision. After using the Gaussian mixture learning, the belief states are updated supposing that an observation, GPS available, is received at each decision step. The belief state is intersected with the GPS availability map, and it results having a smaller state distribution.

3.3.8 Cost function of the model

Our objective is to find the safest and shortest path. To achieve this goal without prioritizing neither the uncertainty nor the length in an artificial way, the cost function is defined based on uncertainty corridor [9]. The corridor is created by a sequence of confidence ellipses, and its volume depends directly on the length of the path and on the dispersion of the uncertainty. Our cost function will be based on this corridor between the two consecutive states $s = (s_v, s_c)$ and $s' = (s'_v, s'_c)$ knowing that the uncertainty is characterized by P which is contained in s_v. Moreover, the cost function includes a direct cost K in case of collisions between states. This cost is needed for the safety of the path planned. If there is an obstacle that intersects the corridor on between two belief states, the second belief state is considered unreachable.

The cost function in MOMDPs needs to be defined over $\mathcal{B}$ instead of $\mathcal{S}$ such as :

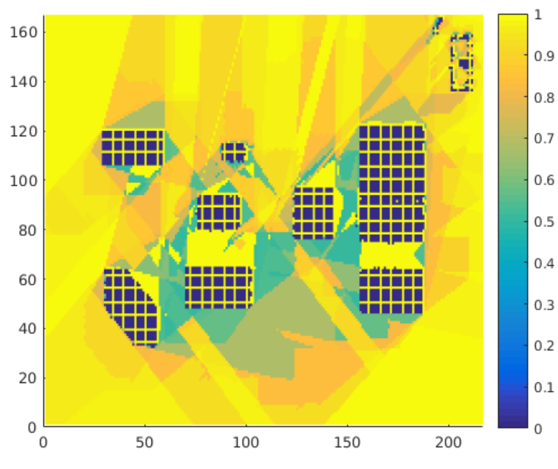
$$\begin{aligned}\mathcal{C}(b, b'^{s'_v}_a) = &\frac{1}{p(s'_v|b,a)}\sum_i p(s'_v|s'_c \in c_i)\\ &\int b_{s_c}\int_{c_i}\mathcal{C}(s,s')f_{s'_c|s_c,s_v,a}(s'_c|s_c,s_v,a)ds'_c ds_c\\ &+ K \times s'_v(Collision)\end{aligned} \tag{26}$$

This cost function is different from the regular reward function of the POMDP. Classically, a reward function corresponding to $R : S \times A \to \mathbb{R}$, where $R(s,a)$ depends directly on the current state s and the action a done. This way, the function $R(b,a)$ become the average of $R(b,a) = \sum\limits_s R(s,a)b(s)$. It is a linear average that approaches the value function of the POMDP with α-vectors based on its PWLC (Piecewise linear convex) property. In our model, the expected cost is no longer a linear average, and thus the value function is not PWLC.

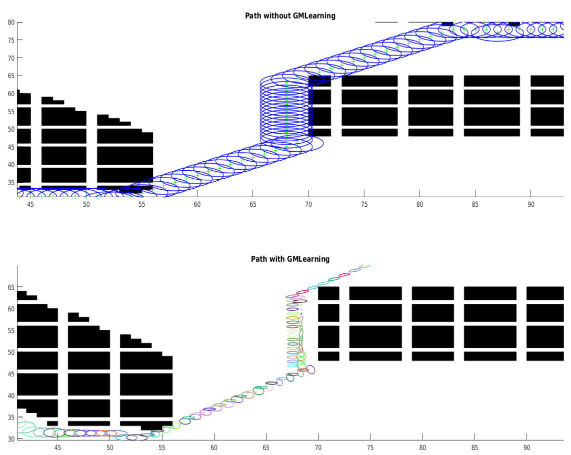
4 Discussion and perspective

There is an important state of the art and the theories behind PWLC in POMDP. It enables to represent the policy by a set of α-vectors which is easier and more intuitive, in contrary to the representation of the policy by a set of all the belief $b \in \mathcal{B}$. Moreover, the theoretical work on the POMDP by [11] ensures that if the value function is PWLC, then value function will be PWLC function at each epoch. The algorithms based on PWLC value function used this particularity to accelerate the computation of the optimum policy. In this work, it is not possible to use these algorithms. One solution is to use algorithms that are not based on PWLC value function. For example, *RTDP-bel* [12] does not work with α-vector, but keeps a hash-table that maps beliefs to values. This algorithm coincides with our problem, but the convergence is not proved. Consequently, the next priority of our work will be to research algorithms that could be adopted or to find a new algorithm that can help to solve the problem.

In this paper, we developed our planning model in perspective to calculate a safe policy in a continuous environment. However, for reducing complexity, our model defines discrete action set which leads to suboptimality in the path planning solution. Therefore in the future, we would like to



(a) Map of probability availability of the GPS



(b) Belief states propagation without (TOP) and with GMM (BOTTOM)

Figure 7: Example of the gaussian mixture learning

incorporate continuous actions, which will reduce the action constraints and then increase the optimality of the path.The planning results depend mainly on the model and the algorithms used. We are also interested in using reinforcement learning to solve the planning model.

References

[1] Markus W. Achtelik, Simon Lynen, Stephan Weiss, Margarita Chli, and Roland Siegwart. Motion- and uncertainty-aware path planning for micro aerial vehicles. *Journal of Field Robotics*, 31(4):676–698, 2014.

[2] Adam Bry and Nicholas Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 723–730. IEEE, 2011.

[3] Sachin Patil, Jur Van Den Berg, and Ron Alterovitz. Estimating probability of collision for safe motion planning under gaussian motion and sensing uncertainty. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3238–3244. IEEE, 2012.

[4] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. Solving continuous pomdps: Value iteration with incremental learning of an efficient space representation. In *In Proc. Int. Conf. on Machine Learning*, 2013.

[5] Richard Bellman. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957.

[6] Sylvie CW Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Planning under uncertainty for robotic tasks with mixed observability. *The International Journal of Robotics Research*, 29(8):1053–1068, 2010.

[7] Harold Wayne Sorenson. *Kalman filtering: theory and application*. IEEE, 1985.

[8] Mauricio Araya-López, Vincent Thomas, Olivier Buffet, and François Charpillet. A closer look at momdps. In *Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on*, volume 2, pages 197–204. IEEE, 2010.

[9] Yoko Watanabe, Sylvain Dessus, and Sylvain Fabiani. Safe path planning with localization uncertainty for urban operation of vtol uav. In *AHS Annual Forum*, 2014.

[10] Lei Xu and Michael I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, 8(1):129–151, 1996.

[11] Richard D Smallwood and Edward J Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

[12] Hector Geffner and Blai Bonet. High-level planning and control with incomplete information using pomdps. In *Proc. Fall AAAI Symposium on Cognitive Robotics*, 1998.

[13] Josep M Porta, Nikos Vlassis, Matthijs TJ Spaan, and Pascal Poupart. Point-based value iteration for continuous pomdps. *Journal of Machine Learning Research*, 7(Nov):2329–2367, 2006.