Solving path planning problems in urban environments based on a priori sensor availability and execution error propagation

Jean-Alexis Delamer* and Yoko Watanabe†
ONERA - The French Aerospace Laboratory, Toulouse, France

Caroline P. Carvalho Chanel‡
ISAE-SUPAERO, Université de Toulouse, France

This paper addresses safe path planning problem in urban environments under onboard sensor availability uncertainty. In this context, an approach based on Mixed-Observability Markov Decision Process (MOMDP) is presented. Such a model enables the planner to deal with a priori probabilistic sensor availability and path execution error propagation, the which depends on the navigation solution. Due to modelling particularities of this safe path planning problem, such as bounded hidden and fully observable state variables, discrete actions and particular transition function form, the belief state update function becomes a complex step that cannot be ignored during planning. Recent advances in Partially Observable Markov Decision Process (POMDP) solving have proposed a planning algorithm called POMCP, which is based on Monte-Carlo Tree Search method. It allows the planner to work on the history of the action-observation pairs without the need to compute belief state updates. Thereby, this paper proposes to apply a POMCP-like algorithm to solve the addressed MOMDP safe path planning problem. The obtained results show the feasibility of the approach and the impact of considering different a priori probabilistic sensor availability on the result policy.

I. Introduction

Navigating through an urban environment with autonomous vehicles is a challenging problem, as safety and efficiency should be ensured [1, 2]. Navigation capability of such vehicles highly depends on their onboard sensor performances – both availability and precision – which can vary with the environment. For example, the widely-used GPS (Global Positionning System) has its precision depending on its satellite constellation visibility which depends on geo-localization and time. Availability and precision of vision sensor measurements are influenced by image textures, visibility of object-of-interest, lighting conditions, etc. However, fortunately, some of such sensor performances can be predicted from a priori knowledge on vehicle operation surroundings. As the GPS satellite orbit is known, it is possible to have some knowledge on GPS precision, represented as Dilution of Precision (DOP), given a 3D environment model, geo-localization and time window[3]. Such information could be very useful in safe path planning task, as it enable the planner to predict localization and path execution error propagation[1, 4].

In this context, this paper tackles such safe path planning problem for autonomous vehicles, especially focusing on flying ones (drones, UAVs), in an urban environment by exploiting probabilistic onboard sensor availability maps and path execution error propagation. Similar problems have already been addressed by [4], [1], [5] and [6], by considering vehicle localization uncertainty propagated along a planned path in function of the environment. For instance, [4] applies the A* algorithm and makes use of a concept of uncertainty corridor to evaluate a path plan for choosing the most efficient and safe path. [5] and [6] propagate the position uncertainty during path search using the Rapidly-exploring Random Belief Trees (RRBT) algorithm. However, any of these approaches considers a complete closed-loop vehicle motion model with GNC (Guidance, Navigation, and Control) functions into the decisional process.

The safe path planning problem addressed in this paper is modeled as a Mixed-Observability Markov Decision Process (MOMDP) [7]. MOMDP is an extension of the classical Partially Observable Markov Decision Process (POMDP) [8]. MOMDP allows the factorization of the state space into fully and partially observable state variables. It holds in a smaller belief state space dimension, and hence decreases the time of policy computation. In this work,

*PhD Student, Information Processing and Systems Departement (DTIS), jean-alexis.delamer@onera.fr
†Researcher, Information Processing and Systems Department (DTIS), Yoko.Watanabe@onera.fr
‡Researcher, Design and Control of Aerospace Vehicles Department (DCAS), caroline.chanel@isae-supaero.fr

the state transition and observation functions of the MOMDP are built on the vehicle GNC model, and on the a priori knowledge of the environment given as probability grid maps of obstacles and onboard sensor availabilities.

The MOMDP model built for our safe path planning problem has some particularities – bounded hidden and fully observable state variables, discrete actions and transition function form. These particularities cause a specific belief state transition function which makes the resulting belief state not easy to be handled during planning. One approach to deal with this difficulty is to learn and approximate the belief state (or state distribution) by a mixture of Gaussian functions [9]. Nevertheless, the computation of path cost, which was defined based on the execution error (i.e. on the belief state transition) as in [4], makes time-expensive value and policy optimization.

This paper proposes to solve the MOMDP-modeled safe path planning problem in a different way, by making use of a POMCP-like algorithm, and by proposing a simpler cost function. POMCP [10] extends UCT [11], an online Monte-Carlo tree search algorithm, to partially observable environments. POMCP, as UCT, applies the UCB1 (Upper Confidence Bounds) action selection strategy during value and policy optimization, what allows to deal with the explore-exploit trade-off while minimizing the regret of choosing a wrong action. Moreover, POMCP approximates the value (which defines the most promising action) of a belief state, by the average of costs evaluated during simulations departing from an initial state distribution. Each simulation sequentially samples a state, performs a selected action and samples an observation following the MOMDP model. This sequential mechanism allows us to generate a policy tree. In this tree, each belief node is represented by a history of action-observation pairs from the initial belief state (state distribution). Such tree representation and value computation avoid the need of an explicit belief state representation during planning.

To evaluate the proposed POMCP-like planning algorithm, policies were computed and simulated for: (i) different probabilistic sensor availability maps, which have an impact on the execution error propagation; (ii) different penalty costs for collisions, which have a direct impact on behavior of the computed policy and, consequently, on the mission success rate. The obtained results are promising in terms of policy time computation, simulated paths success rate and averaged path cost.

This paper is organized as follows: firstly the MOMDP model for this application case is presented. Then, a POMCP-like algorithm is proposed; the simulation test results show the impact of different probablilistic sensor availability maps and penalty costs on the policy. Finally, conclusion and future works are discussed.

II. UAV Safe Path Planning Problem

This paper addresses a safe path planning problem of autonomous vehicles by considering it as a problem of finding a navigation and guidance strategy for making vehicles reach a given destination safely and efficiently in a cluttered environment. This challenging problem considers a priori probabilistic availability of the vehicle onboard sensors and execution error propagation which depends on the navigation solution being used.

Let us suppose a vehicle equipped with N different onboard navigation sensors, such as inertial sensors, GPS and vision sensors, which are used by the GNC system to execute a path. The navigation filter estimates the vehicle state x and its error covariance matrix P by using measurements from a set of selected and available sensors, which defines a navigation mode. The guidance and control module executes a selected path segment (or action) by using the navigation solution. A priori knowledge on the environment is assumed to be given by a set of probability grid maps of obstacles and availability of each of the N onboard navigation sensors. These maps are used during planning task to propagate the path execution uncertainty given the probabilistic sensors' availability, and then to evaluate obstacle collision risk.

A. GNC transition model

The vehicle GNC model is described in this section (see [12] for more details). The transitional state of a vehicle $x = \begin{bmatrix} \mathcal{X}^T & \mathcal{V}^T & b_a^T \end{bmatrix}^T$ is defined respectively by its position, velocity and the accelerometer bias. The state transition is 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

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

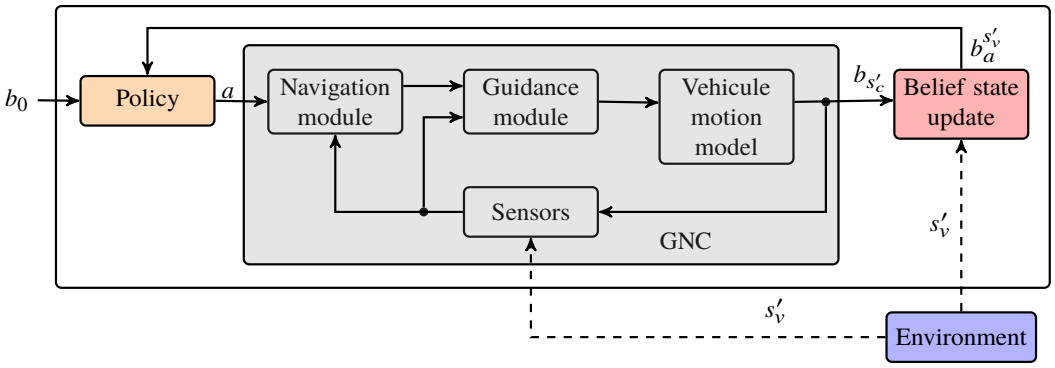


Fig. 1 System architecture diagram. The GNC closed-loop vehicle model is incorporated into the MOMDP transition function. A priori information forms a set of probability grid maps of obstacles and sensor availability.

The state estimator is based on an EKF (Extended Kalman Filter, [13]) which proceeds in two steps; Prediction by the IMU acceleration measurements, and Correction by the other navigation sensor measurements $S_n, n \in N$, if available.

INS Prediction: The IMU acceleration measurement is given as:

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

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

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

Then, the predicted state estimation error can be written as:

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

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

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

where, $\tilde{R}_{\mathrm{IMU}_k} = BR_{BI_k}^T R_{\mathrm{IMU}} R_{BI_k} B^T$. For simplicity, we 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.

Sensor correction: When the n-th onboard sensor measurement $z_{\mathrm{S}_{n_{k+1}}}$ is available at t_{k+1}, the predicted state (1) can be corrected by using it:

$$z_{\mathrm{S}_{n_{k+1}}} = H_{\mathrm{S}_n}x_{k+1} + \xi_{\mathrm{S}_{n_{k+1}}}$$

where, $\xi_{\mathrm{S}_n} \sim N(0, R_{\mathrm{S}_n})$ is a measurement noise of the n-th sensor. Then, the estimated state is corrected such as :

$$\hat{x}_{k+1} = \hat{x}_{k+1}^- + K_{\mathrm{S}_{n_{k+1}}}H_{\mathrm{S}_n}\left(z_{\mathrm{S}_{n_{k+1}}} - H_{\mathrm{S}_n}\hat{x}_{k+1}^-\right) \tag{6}$$

where, $K_{\mathrm{S}_{n_{k+1}}} = P_{k+1}^- H_{\mathrm{S}_n}^T\left(H_{\mathrm{S}_n}P_{k+1}^- H_{\mathrm{S}_n}^T + R_{\mathrm{S}_n}\right)^{-1}$ is the Kalman gain. Then, the estimation error and its covariance are updated as :

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

If there is no onboar sensor available or selected, the estimation error an its covariance remain as those from the prediction step.

Guidance law: Given a desired velocity $\mathcal{V}_{\text{ref}}$, the following linear guidance law is applied:

$$a_k = \hat{K}_p \mathcal{V}_{\text{ref}} - K_d(\hat{\mathcal{V}}_k - \mathcal{V}_{\text{ref}}) = K_p \mathcal{V}_{\text{ref}} - K_d \hat{\mathcal{V}}_k \tag{8}$$

where, K_p,$K_d > O$ are control gains and $\hat{\mathcal{V}}_k$ is the estimated vehicle velocity at instant t_k, i.e., $\hat{\mathcal{V}}_k = \begin{bmatrix} 0 & I & 0 \end{bmatrix} \hat{x}_k$. Then, x_{k+1} can be obtained by substituting this guidance law (8) into the discrete process model (1) :

$$x_{k+1} = (\Phi - \Delta\Phi^{\mathcal{V}})x_k + BK_p\mathcal{V}_{\text{ref}} + \Delta\Phi^{\mathcal{V}}\tilde{x}_k + v_{k+1} \tag{9}$$

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

$$\begin{aligned} x_{k+1} &\sim N((\Phi - \Delta\Phi^{\mathcal{V}})x_k + BK_p\mathcal{V}_{\text{ref}}, \Delta\Phi^{\mathcal{V}} P_k \Delta\Phi^{\mathcal{V}^T} + Q) \\ &= N(\bar{x}_{k+1|k}, \tilde{Q}^a_{k+1}) \end{aligned} \tag{10}$$

where, the covariance $\tilde{Q}^a_{k+1}$ becomes a function of the estimation error covariance P_k given by the navigation system (Eq. 7 or 5). Note that, this normal distribution defines the execution error of the path segment (or the action effect) being considered in the path planning problem.

III. MOMDP Model for efficient and safe path planning problem

In this paper, the safe and efficient path planning problem is modelled as a Mixed-Observability Markov Decision Process (MOMDP) ([14] and [7]), which is an extension of the POMDP (Partially Observable Markov Decision Process) [8]. In MOMDPs the state space is factorized into partially observable state variables and fully observable state variables. In this way, the belief state space (distribution probability over states) has smaller dimension compared to the classical POMDP framework. It decreases processing time for value and policy computation.

Applied to the path planning problem here addressed, one can assume that a vehicle always knows the current sensors' availability, i.e. at a given decision time step the embedded system knows if a given sensor can be used or not. Then, sensors' availability is considered as fully observable state variable of the model. On the other hand, the vehicle state vector x is considered as a hidden and non observable state from the planning model point of view. Given the GNC transition model described in Sec. II.A, the only output considered is the execution error distribution (bounded by the covariance matrix $\tilde{Q}$, see. Sec. II.A), and so, neither partial nor direct symbolic observation is possible for it. Figure 1 illustrates the system architecture with different modules.

Therefore, the MOMDP here addressed is defined as a tuple $\{\mathcal{S}_v, \mathcal{S}_h, \mathcal{A}, \Omega, \mathcal{T}, O, C, b_0\}$, such as:

- $\mathcal{S}_v$ is the bounded set of fully observable states;
- $\mathcal{S}_h$ is the bounded set of hidden continuous states;
- $\mathcal{A}$ is the bounded set of actions;
- Ω is the bounded set of observations;
- $\mathcal{T}$ is the state transition function;
- O is the observation function such as : $O(o, a, s'_h, s'_v) = p(o|s'_h, s'_v, a)$;
- $C : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the cost function;
- $b_0 = (s^0_v, b^0_{\mathcal{S}_h})$

where, $b^0_{\mathcal{S}_h} \in \mathcal{B}_h$ is the initial probability distribution over the initial hidden continuous state, conditioned to $s^0_v \in S_v$, the initial fully observable discrete state.

This model differs from the classical MOMDP approach presented in [14] or in [7]. It is important to note that, given the specificity of the model chosen by factorizing the state space into fully observable discrete state variables and hidden (and non observable) continuous state variables, the observation function in our MOMDP path planning model is defined as: $O(o, a, s'_h, s'_v) = p(o|s'_h, s'_v, a) = 1$ if $o = s'_v$, or 0 otherwise; and so, $\Omega = \mathcal{S}_v$. However, it can be shown that all developments of [14] and [7] still remain valid. It is because $\Omega = \mathcal{S}_v$ climbs into a particular case of factorization already presented in [14], where $\Omega : \Omega_v \times \Omega_h$ is the complete set of observations composed by the observation set for the fully observable state Ω_v and the observation set for the partially state Ω_h. Thereby, in our model $\Omega_v = \mathcal{S}_v$ and $\Omega_h = \emptyset$ (see Eq. (3) of [14] for more details).

A. State space

The visible state $s_v \in \mathcal{S}_v$ is defined as a tuple containing: the fully observable boolean state variables for sensors' availability [0; 1], a boolean variable for a collision flag, and the P the localization error covariance matrix propagated

by the navigation module in function of a selected navigation mode in a given decision step.

Thus, s_v is define such as $s_v = \left\{F_{S_1}, \ldots, F_{S_N}, F_{\text{Col}}, P\right\}$. It is assumed that the collision flag F_{Col} is also fully observable by measuring or estimating a force of contact.

The hidden and non observable continuous state $s_h \in S_h$ is defined such as $s_h = x$, recalling that x is the continuous vehicle state vector (see Sec. II.A).

B. Action space

An action $a \in A$ is defined as a tuple $\{d, m_n\}$ where $d \in D$ is the desired direction of motion which specifies $\mathcal{V}_{\text{ref}}$ (see Eq. (8)). D defined here is a finite set of discretized directions. $m_n \in \{S_1 \ldots, S_N\}$ is the navigation mode to be considered depending on the sensors' availability – or navigation solution to be considered during planning depending on sensor selection.

C. Transition function

The transition function $T(s_v, s'_v, a, s'_h, s_h)$ is composed of two functions:

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

$$T_{\mathcal{S}_h}(s_h, s_v, a, s'_h) = f_{s'_h}(s'_h | s_h, s_v, a) \sim N(\bar{s'_h}, \tilde{Q}'(s_v)),$$

  which is based on the GNC closed-loop vehicle motion model, given that the probability distribution of a predicted state s'_h follows a normal distribution $N(\bar{s'_h}, \tilde{Q}'(s_v))$ (see Eq. 10), which in turn, is a function of the previous hidden state s_h, the previous visible state s_v and the action a.

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

$$T_{\mathcal{S}_v}(s'_h, s'_v) = p(s'_v | s'_h),$$

  which represents the transition function for s'_v and depends on the probabilistic sensors availability maps, and therefore, depends only on the next state s'_h. Concretely,

$$T_{\mathcal{S}_v}(s'_v | s'_h) = \prod_{i=1}^{N+1} p(s'_v(i) | s'_h) \tag{11}$$

  where, N is the number of sensors, thus $N + 1$ is the number of flags (booleans) in s_v, and $s'_v(i)$ the i-th flag.

Thus, the complete transition function becomes:

$$T(s_v, s'_v, a, s'_h, s_h) = T_{\mathcal{S}_h}(s_h, s_v, a, s'_h) \times T_{\mathcal{S}_v}(s'_h, s'_v) = p(s'_v | s'_h) f_{s'_h}(s'_h | s_h, s_v, a) \tag{12}$$

D. Cost function

The cost function to be minimized is defined as the vehicle travel (or flight) time plus a cost of collision. It is expected that by minimizing the cost, the algorithm will minimize the flight time (for efficiency) and the probability of collision (for safety) at the same time. More precisely the cost function is defined as :

$$\begin{cases} C(s_t, a_t) = f_t & \text{if } s_t \text{ not in collision} \\ C(s_t, a_t) = K - \sum_{k=0}^{t-1} C(s_k, a_k),\ \forall a_t \in \mathcal{A} & \text{otherwise} \end{cases} \tag{13}$$

where, f_t is the flight time for a given action a at decision step t, and K a fixed cost in case of collision. When a collision occurs, the cost of the any action is a fixed penalty minus the total flight time since the initial state until the collision state. This trick avoids to penalize more if the collision occur after a longer time flight or near the goal. In other words, if the cost of an entire path is defined by the sum of action costs, this cost function equally penalizes all the paths ended up with collision.

E. Belief State update

The belief state b represents the probability distribution over states. In this MOMDP model, the belief state can be factorized into a probability distribution over the hidden state space $\mathcal{S}_h$ conditioned on the fully observable state s_v, such as $b^{s_v}_{\mathcal{S}_c} = (b_{\mathcal{S}_c}, s_v)$.

The current belief state is updated after each action a and each perceived visible state $o' = s'_v$ the using the Bayes rule [14]. It allows to update the belief state distribution over the hidden state space. And so, choosing actions during planning based on a complete history information state [15].

In this MOMDP model, the belief state update is decomposed into two functions. The first function corresponds to the GNC closed-loop transition function (Eq. 11 defining a belief state transition related with the action execution error propagation:

$$b_{s'_h}(s'_h) = \int_{\mathcal{S}_h} f_{s'_h}(s'_h|s_h, s_v, a) b_{s_h}(s_h) ds_h \tag{14}$$

The second function is related to the probability of observing s'_v (given by the probability grid maps). This observation probability is computed based on $b_{s'_h}$:

$$p(s'_v|b, a) = \sum_{i=1}^{|G|} p(s'_v|s'_h \in c_i) p(s'_h \in c_i|b, a) \tag{15}$$

where, c_i corresponding to the i^{th} cell of the probability map and $|G|$ is the number of cells in the map.

Finally, the complete belief state update function can be write as :

$$b'^{s'_v}_{s'_h,a}(s'_h) = \frac{p(s'_v|s'_h) \int_{\mathcal{S}_h} f_{s'_h}(s'_h|s_h, s_v, a) b_{s_h}(s_h) ds_h}{\sum_{i=1}^{|G|} p(s'_v|s'_h \in c_i) p(s'_h \in c_i|b, a)} \tag{16}$$

F. Value function

The aim of solving a MOMDP consists in finding *a policy* $\pi : \mathcal{B} \rightarrow \mathcal{A}$, where $\mathcal{B}$ defines the belief state space, which minimizes a given criterion usually determined by a value function.

The value function $V^\pi(b_0)$ is defined as the expected total cost (weighed by time using γ) the agent will receive from b_0 when following a policy π [8].

$$V^\pi(b) = \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} \gamma^t \mathbb{E}\left[C(b_t, \pi(b_t))\right] | b_0 = b \right] \tag{17}$$

where, $C(b, \pi(b) = a) = \sum_{s \in \mathcal{S}} C(s, a) b(s)$ is the expected cost of an action a in the belief state b for the discrete state space case.

The optimal policy π^* is defined by the optimal value function $V^{\pi*}$, such as :

$$V^{\pi^*}(b) = \min_{\pi \in \Pi} \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \mathbb{E}\left[C(b_t, \pi(b_t))\right] | b_0 = b \right] \tag{18}$$

Opening the sum in (Eq. 18), it holds to a Bellman's equation, which allows the application of dynamic programming to find the optimal policy. For example:

$$\begin{aligned} V(b) &= \min_{a \in A} \mathbb{E}\left[C(b, a) + \gamma V(b_a^{s_v})\right] \\ &= \min_{a \in A} \left[C(b, a) + \gamma \sum_{s_v \in \mathcal{S}_v} p(s_v|b, a) V(b_a^{s_v}) \right] \end{aligned} \tag{19}$$

When the value (Eq. 19) converges for all belief states, it is possible to extract the optimal policy [8]. Such value and policy computation problem is known to be a hard decision problem (undecidable [16]), in particular given that the belief state space is a continuous space. Recent algorithms, such as SARSOP [17], RTDP-bel [18], or POMCP [10], approach the solution by searching a (partial-)policy only based on reachable belief states, following the MOMDP model dynamics. These algorithms are able to compute sub-optimal solutions in reasonable time.

G. Cost function and feasible policies

The MOMDP model presented in this work aims to compute a policy that will minimize the expected flight time and the collision risk, that is, maximize operation efficiency and safety respectively. This policy should be obtained by minimizing the value function defined in the model. Hereafter the value of the initial belief state is redefined in terms of

expected flight time and collision risk in the case where $\gamma = 1$. For simplicity, it is assumed that all the action duration coincides with the planning epoch which is constant. This gives a constant f_t for any action a_t, any state s_t at any depth t. It simplifies the collision cost in Eq. 13 to $C(s_t, a_t) = K - t \times f_t = K_t$. Also, this assumption removes the dependency of our cost function Eq. 13 on the action: $C(s_t, a_t) = C(s_t)$.

1. Redefining the value of the initial belief state in terms of collision risk and flight time

This subsection shows that, with the cost function defined in Eq. 13 and with $\gamma = 1$, the value of the initial belief state $V(b_0)$ can be written as a sum of the expected flight time to reach a goal and the constant collision penalty K weighed by the collision probability. From the definition Eq. 19, $V(b_0)$ can be expanded as follows, knowing the value of any belief state, knowing the value of the collision K, recalling the value of goal state is 0, knowing that the values fully observable state variable for collision are $F_{Col} = 1$ or $F_{Col} = 0$, and finally assuming $F_{Col_0} = 0$.

$$
\begin{aligned}
V(b_0) &= \min_{a \in A} \left[C(b_0) + \sum_{s_{v_1} \in \mathcal{S}_v} p(s_{v_1}|b_0, a) V(b_a^{s_{v_1}}) \right] \\
&= f_t + \sum_{s_{v_1} \in \mathcal{S}_v} p(s_{v_1}|b_0, a_0^*) V(b_{a_0^*}^{s_{v_1}}) = \sum_{s_{v_1} \in \mathcal{S}_v} p(s_{v_1}|b_0, a_0^*) \left(f_t + C(b_{a_0^*}^{s_{v_1}}) + \sum_{s_{v_2} \in \mathcal{S}_v} p(s_{v_2}|b_{a_0^*}^{s_{v_1}}, a_1^*) V(b_{a_1^*}^{s_{v_2}}) \right) \\
&= p(F_{Col_k} = 1|b_0, a_0^*)K + p(s_1 \in GOAL|b_0, a_0^*) f_t + \\
&\quad p(F_{Col_k} = 0 \cap s_1 \notin GOAL|b_0, a_0^*) \left(2f_t + \sum_{s_{v_2} \in (\mathcal{S}_v \cap F_{Col} = 0)} p(s_{v_2}|b_{a_0^*}^{s_{v_1}}, a_1^*, s_1 \in GOAL) V(b_{a_1^*}^{s_{v_2}}) \right) \\
&= p_{c_1} K + p_{g_1} f_t + \left(p_{c_2} K + p_{g_2} 2 f_t + \left(p_{c_3} K + p_{g_3} 3 f_t + \cdots \right) \right) \\
&= \cdots \\
&= \sum_{t=1}^{\infty} p_{c_t} K + \sum_{t=1}^{\infty} p_{g_t} t f_t = pcK + (1 - pc)T
\end{aligned}
$$

where p_{c_t} is the probability of being the collision state at the depth t when starting from the initial belief b_0 and following the optimal policy. Similarly, p_{g_t} is the probability in reaching at the goal state at the depth t. $pc = \sum_{t=1}^{\infty} p_{c_t}$ is the collision probability at the initial belief b_0 when following the optimal policy. $T =$ is the conditional expected total flight time to reach a goal starting from b_0 knowing that $F_{Col_k} = 0$ for $\forall k \geq 0$. Thanks to our cost function definition (see Eq. 13), the value of the initial belief $V(b_0)$ becomes a linear function of the constant collision penalty cost K with a slope of the collision probability pc. In the following section, this linear dependency will be used to determine the value of the collision penalty K from a given maximum allowable risk of collision.

2. Maximum allowable collision risk

Let us consider the following three extreme navigation policies;

- The most efficient (heuristic) policy (π_h): which minimizes the expected total flight time to the goal with considering neither the initial state uncertainty nor the collision risk due to the localization and path execution uncertainty. As it comes from the minimization of the total flight time only, the expected flight time T_h resulted from this heuristic policy is the shortest possible flight time of a given mission, and so $T_h \leq T$ is guaranteed. The collision probability p_h resulted from this policy could be high up to 1.
- The safest policy (π_s): which minimizes the expected flight time while not allowing any collision risk under the uncertainties. This policy does not necessarily exist for a given mission, but here we assume it does. Then the collision probability resulted from this policy should be 0. The resulting total flight time gives an upper-bound of T, because the optimal cost $V^{\pi^*}(b_0) = pcK + (1 - pc)T \leq V^{\pi}(b_0)$ for $\forall \pi$ and this safest policy gives $V^{\pi_s}(b_0) = T_{max}$.
- The collision policy (π_c): is a policy which always brings a vehicle in a collision state, i.e., the resulting collision probability is 1.

As derived in the previous section, the value of the initial belief state is linearly dependant on the collision penalty K and its slope is given by the collision probability. Figure 2 plots the values of these three extreme policies ($V^{\pi_h}(b_0)$, $V^{\pi_s}(b_0)$, $V^{\pi_c}(b_0)$ respectively) versus the collision penalty cost K. Since any policy line $V^{\pi}(b_0) = pcK + (1 - pc)T$

intersects with the collision policy line (Fig. 2 —■—) at $K = T$, the collision penalty should be chosen as $K > T_{max}$ so that the collision policy never becomes optimal.

Now, let us consider a maximum allowable collision probability p_{thd} given as a mission criteria. That is, the optimal navigation policy is required to have $pc \leq p_{thd}$ for being acceptable. From the condition $K > T_{max} \geq T$, for any feasible policy with $pc \leq p_{thd}$, the following is satisfied.

$$V^{\pi}(b_0) = pcK + (1 - pc)T \leq p_{thd}K + (1 - p_{thd})T$$

On the other hand, as the heuristic policy (π_h) gives the lower-bound of the expected total flight time, the right-hand side of the above inequality is lower-bounded by $p_{thd}K + (1-p_{thd})T_h$. Hence, it can be said that if $V^{\pi}(b_0) \leq p_{thd}K + (1-p_{thd})T_h$, the policy π is guaranteed to satisfy the maximum allowable collision risk. That is, the line corresponding to $p_{thd}K + (1 - p_{thd})T_h$ (Fig. 2 —○—) gives the upper threshold of the optimal policy line to be feasible.

At the same time, the safest policy gives the upper-bound of the value of the initial belief of any possible optimal policy (regardless of its feasibility), such as $V^{\pi^*}(b_0) \leq V^{\pi_s}(b_0) = T_{max}$. Therefore, if we choose the value of the collision penalty K as at which the safest policy line (Fig. 2 —●—) and the line of the upper threshold of the feasible policy (Fig. 2 —○—) intersect, it is guaranteed that any optimal policy satisfies

$$V^{\pi^*}(b_0) \leq p_{thd}K + (1 - p_{thd})T_h = T_{max} = V^{\pi_s}(b_0)$$

and hence $pc \leq p_{thd}$. This value of K is derived as follows.

$$K(T_{max}, T_h, p_{thd}) = T_h + \frac{\overbrace{(T_{max} - T_h)}^{\text{expected flight time gain}}}{\underbrace{p_{thd}}_{\text{maximum allowable collision probability}}} \tag{20}$$

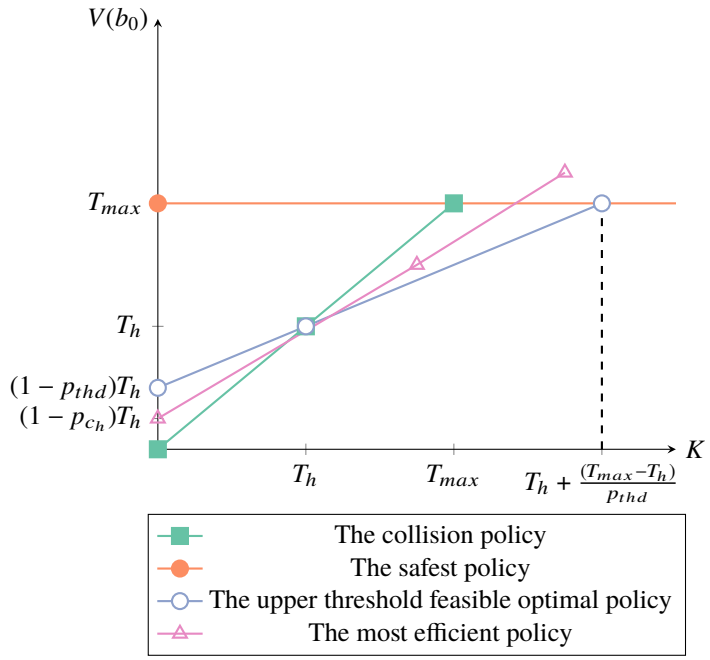


Fig. 2 Representation of $V(b_0)$ in terms of flight time and maximum allowable collision risk

IV. POMCP-like algorithm

Keeping and updating the belief states can be a challenging and computationally expensive step during problem resolution due to the potential complexity of the belief state update function (Eq. 16). In particular in the problem

addressed in this paper, as the hidden state space is continuous and the fully observable state space is discrete, the computation of the probability distribution over $\mathcal{S}_c$, and correction by s_v would require expensive computational effort that could be bypassed using approximations [9, 12].

An interesting solution is to use algorithms that do not need to maintain and update the belief state in each decision step. In this sense, resolution approaches like POMCP algorithm [10] are promising. POMCP is a Monte-Carlo Tree Search algorithm for partially observable environments. POMCP works by sampling a state s from the initial belief state b_0, and simulating sequences of action-observation (by a trial procedure) to construct a tree of belief nodes. Each tree node h represents an history of action-observation pairs since the initial belief state (being a belief node h). POMCP calculates for a given node h of the tree the average cost observed for all trials that have started from this node. POMCP does not update the belief state after each action-observation pair but update the average cost for each node of the tree called, and keeps in memory the number of times a node was explored $N(h)$, and the number of times a given action was chosen $N(ha)$ in this node.

This procedure (trials) allows to POMCP to approach the value function and policy for a given node (a belief state represented by an history). During value and policy computation, trials are performed in a greedy way. The most promising action, following a given heuristic is chosen in each tree node. More precisely, in each node the greedy action selection strategy is based on the same heuristic choice as UCT [11], the UCB1 (Upper Confidence Bound 1) action selection strategy.

This action selection strategy is based on a combination of two characteristics, the action Q-value and a measure of how well explored an action is (how well estimated is its value).

The action's Q-value can be defined as:

$$Q(b,a) = \left[C(b,a) + \gamma \sum_{s_v \in S_v} p(s_v|s,a) V(b_a^{s_v}) \right], \tag{21}$$

being the value of performing an action a in the belief state $b = (s_v, b_{\mathcal{S}_h})$ – one-step lookahead computation of the value –, supposing that the optimal policy will be performed after. Note V is the true expected cost to reach the goal [19]. Because POMCP works by simulating sequences of action-observation each action Q-value is regularly updated, which allows to estimate the value of $V(b) \leftarrow \min_{a \in A} Q(b,a)$.

The second characteristic of this greedy action selection strategy used during policy computation is a measure (given by $c\sqrt{\frac{\log N(h)}{N(ha)}}$) of how well-explored the action a is, given the history h (or belief node). This measure is used to balance the action choice in the algorithm, which will select the action that minimizes $Q(h,a) - c\sqrt{\frac{\log N(h)}{N(ha)}}$. In other words, the algorithm will select the action that minimizes the regret of choosing a wrong action [11].

The exploration coefficient c is used to force the algorithm to try actions that seem a priori less interesting for avoiding to falling into a local optimum policy. If c is high the algorithm will try more often the actions that seem less interesting, on the contrary, if c is low the algorithm will rarely explore different actions.

However, due to the particularities of the problem addressed in this work, the POMCP algorithm was modified, and is presented on Alg. 1. As in the classical POMCP algorithm, it is starting with an initial belief state b_0 and an empty history h (line 5) that will be initialized to b_0, then the algorithm will expand the tree for a given number of trials. The principal differences between the classical POMCP and the algorithm here presented are hereafter discussed.

The classical POMCP algorithm estimates the value of a tree node by selecting an action using the UCB1 greedy action selection until a new node is reached. When a new node is created a *rollout* method is used to estimate the value of the new node, that simulates and evaluates sequences of random action-observation pairs starting from this new node. And, then POMCP backtracks this value until the initial belief node and starts a new trial.

The algorithm here presented estimates the value of a tree node by selecting an action using the UCB1 greedy action selection until *a final state is reached* (goal G or collision C). If a new node is discovered (this history is not yet in the tree), its Q-value and value are estimated using an initial heuristic value hereafter presented, and not a *rollout* policy.

The belief state value initialization considered for this work explores the A* shortest path solution considering only the obstacles grid map and computes the estimated flight time. This can be pre-calculated and hence save time during value and policy optimization process. Note that this value initialization gives a more informative value approximation in this goal-oriented path planning problem (for a given state in a given grid cell) compared to a *rollout* policy being, in this sense, preferable.

Figure 3 shows the different steps of the algorithm. Figure 3a is the initializing of the algorithm with an empty history and the estimated value of the two possible actions. Figure 3b shows an example after the first trial where the

Algorithm 1: POMPC-like

Function *POMPC*$(h,\ b_0,\ c,\ \gamma)$
  $h \leftarrow b_0$
  while $nbTrial < Nb_{max}$ **do**
    $s_h \leftarrow$ sampling s_h from b_0
    Trial$(h,\ s_h,\ c,\ \gamma)$
  $a^* \leftarrow \underset{a \in A}{\operatorname{argmin}}\, V(b_0)$
Function *Trial*$(h,\ s_h,\ c,\ \gamma)$
  if $s_h \in Goal$ **then**
    return 0
  if $h \notin T$ **then**
    for $a \in A$ **do**
      for $s_v \in S_v$ **do**
        $hao \leftarrow h + a + s_v$
        $T(hao) \leftarrow (N_{\text{init}}(hao),\ V_{\text{init}}(hao),\ \emptyset)$
  $\bar{a} \leftarrow \underset{a \in A}{\operatorname{argmin}}\, Q(s_h, a) - c\sqrt{\frac{\log N(h)}{N(ha)}}$
  $ha \leftarrow h + \bar{a}$
  $s'_h,\ s_v \sim \mathcal{G}(s_h, \bar{a})$
  $hao \leftarrow ha + s_v$
  $Trial(hao,\ s'_h,\ c,\ \gamma)$
  $N(h) \leftarrow N(h) + 1$
  $N(ha) \leftarrow N(ha) + 1$
  $Q(h, \bar{a})' \leftarrow C(s_h, a) + \gamma \sum_{s_v \in S_v} p(s_v \,|s_h, \bar{a}) V(hao)$
  $Q(h, \bar{a}) \leftarrow Q(h, \bar{a}) + \frac{Q(h,\bar{a})' - Q(h,\bar{a})}{N(ha)}$
  $V(h) \leftarrow \min_{a \in A} Q(h, a)$

action a_2 has been selected at the beginning, and stopping only when the goal G has been reached. The policy after the first trial is represented in red. And, figure 3c shows what could happen after a number Nb of trials with the new policy that starts with the action a_2 that was not interesting at the beginning of the example.

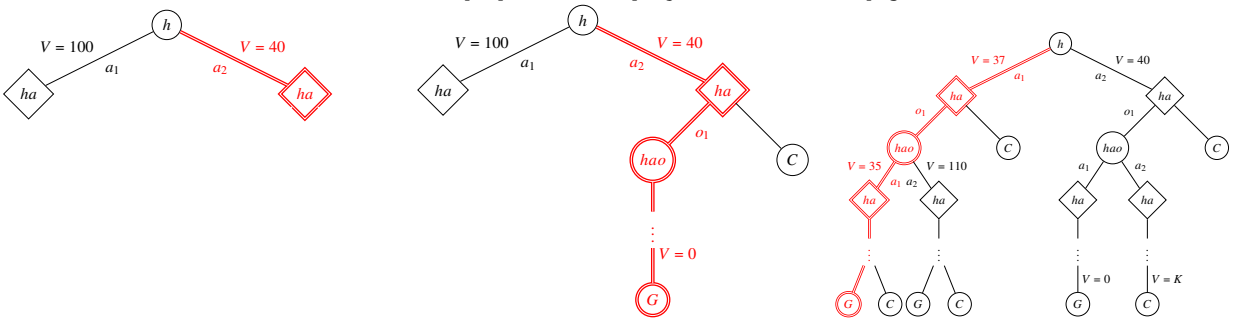
V. Simulation results

A. Configuration

POMDP has been tested on a benchmarking framework for UAV obstacle field navigation* proposed in [20], which provides environments with different obstacle configurations. The selected benchmark "Cube baffle" contains two cube obstacles with a grid size of $100 \times 100 \times 20$ cells, where each grid cell has the size of $2\text{m} \times 2\text{m} \times 2\text{m}$.

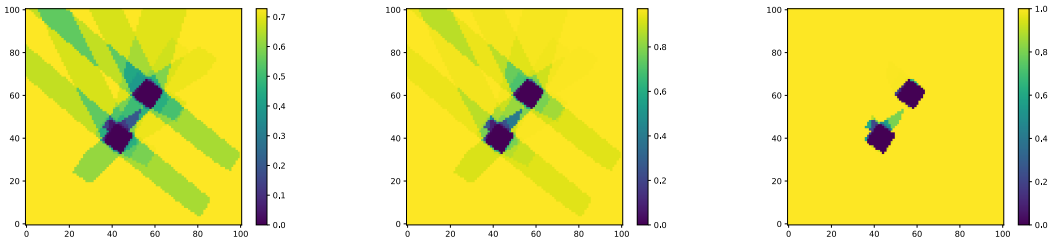
In the following tests, two onboard sensors are considered: INS and GPS. While INS is known to be available anywhere, GPS is not. However, a priori knowledge on the GPS availability is supposed. Probability grid maps for GPS

*benckmark framework from: `www.aem.umn.edu/people/mettler/projects/AFDD/AFFDwebpage.htm`



(a) History tree at the beginning of POMCP. (b) History tree after the first trial. (c) History tree after N trials.

Fig. 3 Evolution of the history tree during the value optimization.



(a) 1 meter precision. (b) 2 meters precision. (c) 5 meters precision.

Fig. 4 Availability maps of the GPS in function of the minimum precision required.

availability were created based on a DOP map calculated with a GPS simulator for a given geolocation and time, for different GPS precision thresholds (1, 2 and 5 meters, Fig. 4). These thresholds give different probabilistic availability maps. For instance, if the 1-meter GPS availability map indicates 60% probability in a given cell, it means that there is 60% of chance that GPS will be available with 1-meter of precision in this cell. Note that, these precision thresholds were only used to generate different GPS availability maps, but not used in the Kalman filter. These maps were created with the aim of comparing the effect of considering the GPS availability probability during planning on the resulting navigation policy for different environments.

In the following experiments, the planning objective was to find a policy allowing at most 10% of collision ($p_{thd} = 0.1$). For each of the three GPS probabilistic availability maps, the first set of optimization has been run to compute the safest policy with 0% collision risk (considering a prohibitive collision cost $\bar{K}$). This computation allows to approximate the lowest (i.e. the tightest) expected flight time T without collision. Thus, this expected value is used as T_{max} to define the collision penalty by Eq. 20. Then, a new set of optimization has been launched to compute another policy using this new collision penalty so that the resulting policy will respect the given admissible collision risk threshold.

Moreover, to compare the performance of policies obtained in these stochastic problems being solved with POMCP-like algorithm, five distinct policy optimization processes, with 100000 trials each (see Alg. 1), have been run for each probabilistic GPS availability map. In addition to that, for each run, the policy being optimized was evaluated for 1000 simulations after each 5000 trials. For these experiments γ was set to 1. Note that this γ value does not prevent value and policy convergence, given that the stochastic shortest path problem here addressed respects all convergence assumptions (see [19] and [11] for more details).

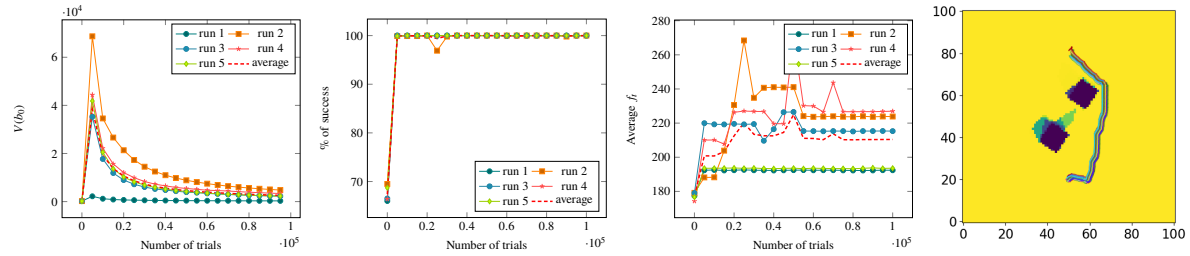
B. Results

To highlight the contribution of using a path planning model like a MOMDP in an environment with uncertainty on the availability of the onboard sensors, the well-known A^* algorithm has been used to compute the shortest path ignoring the uncertainties in environment (sensor availability) as well as in the vehicle state transition. This *heuristic* policy chooses the best action to follow only based on the A^* shortest path from the current position. Then 1000 simulations have been run on each GPS availability map. Figure 8 shows among other things, the results of these simulations. It shows that even for the easiest environment with highest probability of GPS being available(i.e. the 5-meter GPS availability map), the simulations have only a success rate of 66%. This is because the uncertainties in the environment and in the vehicle state transition are ignored in the A^* planning task. In the following, we'll show that this success rate can be improved by the proposed path planner and hence the vehicle operation safety can be enhanced.

Figures 5, 6 and 7 present the averaged results for five runs of the POMCP algorithm (offline runs) for the three different probabilistic GPS availability maps respectively. The first row figures are the results of the safest policy with $\bar{K} = 10^6$ allowing to defining the tightest T_{max}. The second row are those of the policy calculated for $K^* = K(T_{max}, T_h, p_{thd})$, where T_h is the heuristic flight time given by the A^* algorithm and $p_{thd} = 0.1$.

The figure 5 shows the results with the 5-meter precision GPS availability map (Fig. 4c). This is the map where GPS is the most likely available: when less GPS precision is required more likely GPS availability is.

It is reminded that T_{max} value is obtained by the result from the firstly calculated safest policy. Figures 5a and 5e show the evolution of the initial belief state value. In this case, the results with $\bar{K}$ and $K^* = K(T_{max} = 195, T_h, p_{thd} = 0.1)$ show that after 100000 trials $V(b_0)$ has almost converged - the final initial belief state value is afterward the same for the 5 runs in both cases. Figures 5b and 5f show success rate for each computed policy, reaching 100%, and respecting the collision risk threshold of 10% required.

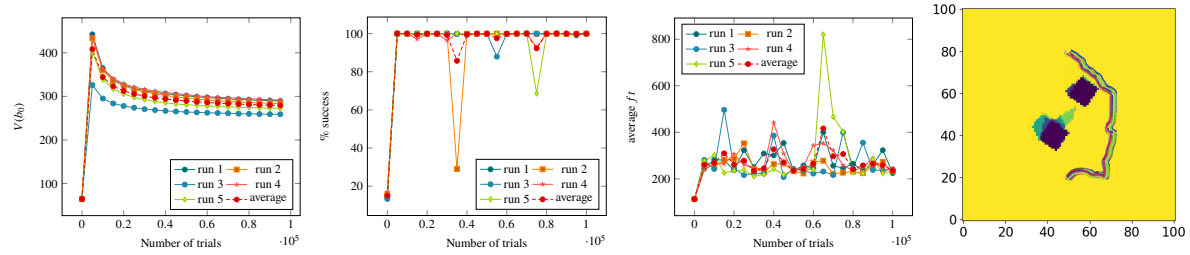


(a) Initial belief state value evolution with $\bar{K}$.

(b) Success rate with $\bar{K}$.

(c) Average value for 1000 simulation with $\bar{K}$.

(d) Paths simulated with a result policy.



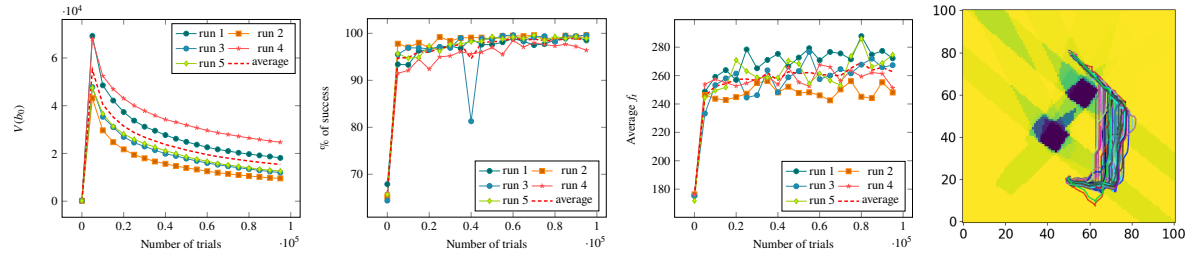
(e) Initial belief state value evolution with $K^* = K(195, T_h, 0.1)$.

(f) Success rate with $K^* = K(195, T_h, 0.1)$.

(g) Average value for 1000 simulations with $K^* = K(195, T_h, 0.1)$.

(h) Paths simulated with a result policy.

Fig. 5 Obtained results in function of the number of trials for 5-meters GPS precision probabilistic availability map.

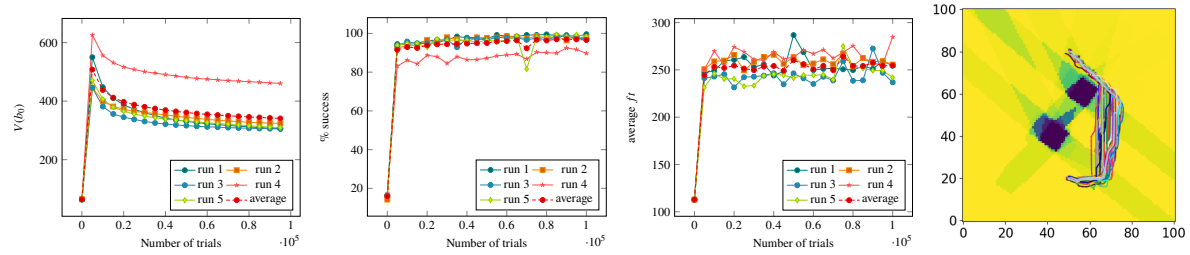


(a) Initial belief state value evolution with $\bar{K}$.

(b) Success rate with $\bar{K}$.

(c) Average value of the 1000 simulations with $\bar{K}$.

(d) Paths simulated with a result policy.



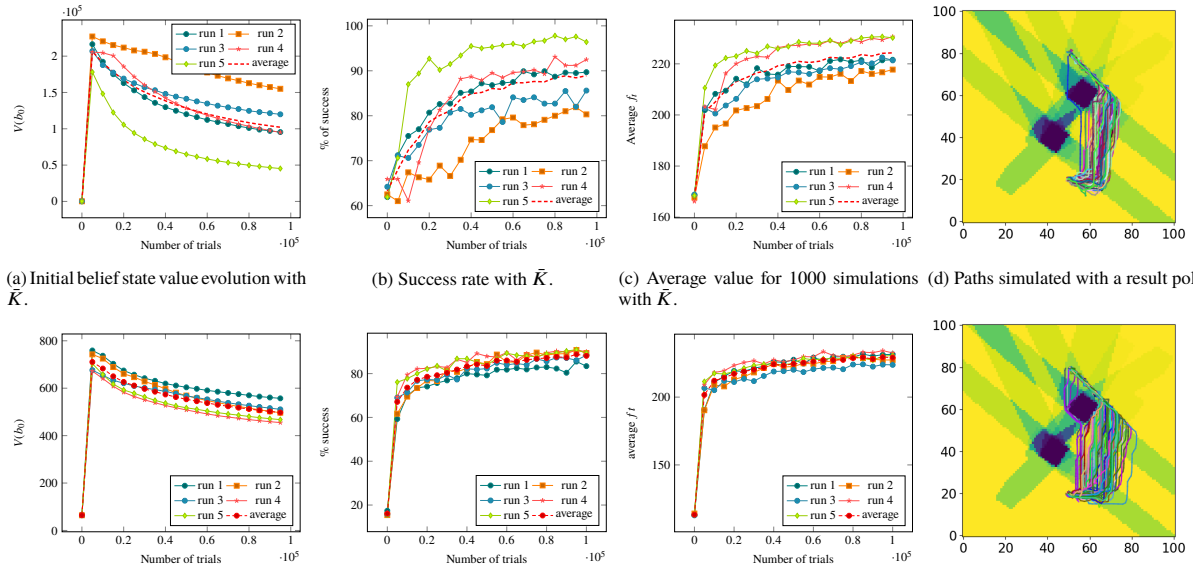
(e) Initial belief state value evolution with $K^* = K(245, T_h, 0.1)$.

(f) Success rate with $K^* = K(245, T_h, 0.1)$.

(g) Average value of 1000 simulation with $K^* = K(245, T_h, 0.1)$.

(h) Paths simulated with a result policy.

Fig. 6 Obtained results in function of the number of trials for 2-meters GPS precision probabilistic availability map.



(a) Initial belief state value evolution with $\bar{K}$.

(b) Success rate with $\bar{K}$.

(c) Average value for 1000 simulations with $\bar{K}$.

(d) Paths simulated with a result policy.

(e) Initial belief state value evolution with $K^* = K(230, T_h, 0.1)$.

(f) Success rate with $K^* = K(230, T_h, 0.1)$.

(g) Average value for 1000 simulations with $K^* = K(230, T_h, 0.1)$.

(h) Paths simulated with one policy.

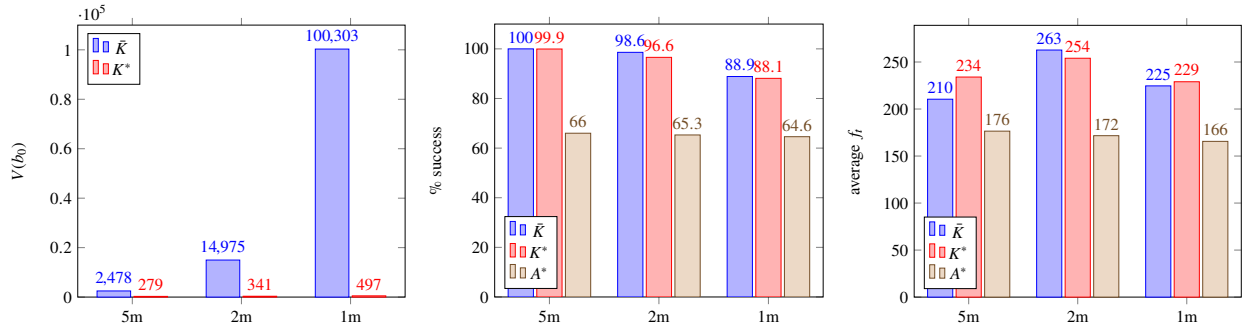
Fig. 7 Obtained results in function of the number of trials for 1-meter GPS precision probabilistic availability map.

However, with the 2-meters probabilistic GPS availability map (Fig. 4b), the results are different (see Fig. 6). Figure 6a shows that the value of the initial belief state for $\bar{K}$ has not converged to a same value on these five runs, even after 100000 trials. Thereby, the result of the run 2, who has the highest success rate is the success, was used to define $T_{max} = 245$ for computing K^*. For this $K^*(T_{max} = 245, T_h, p_{thd} = 0.1)$, the value of b_0, shown in Figure 6e, decreases with 100000 trials reaching almost the same value among runs (except for run 4). The corresponding evolution of the success rate is shown in Figure 6f. It reaches 90%, respecting the maximum allowable collision risk fixed to 10% (except for run 4). The exception of the run 4 can be explained by the fact that even 100000 trials of POMCP-like search could be not enough to converge to a minimum value, and so, not respecting the expected success rate.

The results with the 1-meter precision probabilistic GPS availability map (Fig. 4a) reflect a more difficult optimization problem due to the increase in the uncertainty on sensors availability (Fig. 7). Figure 7a shows that, except for run 5, it needs more than 100000 trials to decrease (minimize) the initial belief state value when using $\bar{K}$. The success rates of these runs do not reach 100%. Indeed, the result of run 5 reaches the highest success rate, and so $T_{max} = 230$ from run 5 was used for the collision penalty computation, as a rough estimation of the expected flight time of the safest policy. Figure 7e shows that the majority of the runs are going to converge to the same expect initial belief state value. Also, the smooth increase in the success rates is shown in Fig. 7f. Note that, even if the value, and the related policy, have not converged, the maximum collision risk of 10% is near to be respected. In this sense, if more trials are performed during policy optimization, one could expect to reach the minimum value and to respect the collision risk defined.

Figure 8 summarizes the results presented with three bar plots. The first one (Fig. 8a) shows the average initial belief state value $V(b_0)$ after 100000 trials. It shows that less likely the GPS is available, higher is $V(b_0)$. The second bar plot (Fig. 8b) presents the average success rate. With 5-meters probabilistic GPS availability map, the initial belief state value has almost converged for both cases, thus both policies respect the success rates imposed (e.g. 100% for $\bar{K}$ and, for K^*, 99.9% being superior to 90%). For K^* in the 2-meter probabilistic GPS availability map, the success rate is above 96% which also respects the given collision risk threshold. However, for K^* in 5 and 2-meters cases, the average flight time is still nearly T_{max} (Fig. 8c). It is expected to be further improved (at a price of taking more collision risk), if we perform more trials in the optimization process. Similarly, for the 1-meter probabilistic GPS availability map, $V(b_0)$ did not converge and consequently, the success rate is below the expected. Moreover, the average flight time is lower than the 2-meters GPS availability map, because the algorithm did not have enough trials to optimize the success rate, taking the more uncertain (risky) but shorter path.

Some conclusions can be extracted from these results. Firstly, the algorithm needs more time to converge when the uncertainty on the GPS availability grows. Indeed, when the GPS availability is less likely the UAV need to rely more on the INS measurement which leads to more risk of collision. Thus, to optimize the policy the algorithm needs more



(a) Comparison of the initial belief state value in function of collision cost.

(b) Comparison of the success in function of the collision cost and the GPS availability maps.

(c) Comparison of the average flight time.

Fig. 8 Summary of the initial belief state value and success rates in function of the collision cost.

trials (more exploration) to find a policy with a minimum collision risk. Secondly, even when the GPS availability is less likely, the algorithm can compute a policy with a good success rate (above 88, 1%) for this benchmark environment.

VI. Conclusion and future work

This paper presents a MOMDP framework to model and solve the safe path planning problem in urban environments. Comparative results show the impact of the probabilistic GPS availability and collision costs on the safety of the path. Moreover, this paper proposes a simple method to impose a maximum allowable flight time and a collision risk threshold in order to compute a feasible policy. Indeed the number of trials necessary to value and policy convergence in the more uncertainty probabilistic sensor maps is important. However, the results show that it is possible to take into account the probabilistic availability of the onboard sensors in the planning process. Moreover, the results also show that in the considered benchmark map with important uncertainty on the sensors availability, significant solutions can be found even if the policy has not converged. Thereby, more evaluations are necessary, in particular to evaluate the proposed approach in real urban environments [20]. Further work will study an online optimization process given that the sensors probabilistic availability map evolves with time.

References

[1] Prentice, S., and Roy, N., "The belief roadmap: Efficient planning in linear POMDPs by factoring the covariance," *Robotics Research*, Springer, 2010, pp. 293–305.

[2] Dolgov, D., Thrun, S., Montemerlo, M., and Diebel, J., "Path planning for autonomous vehicles in unknown semi-structured environments," *The International Journal of Robotics Research*, 2010.

[3] Kleijer, F., Odijk, D., and Verbree, E., "Prediction of GNSS availability and accuracy in urban environments case study schiphol airport," *Location Based Services and TeleCartography II*, Springer, 2009.

[4] Watanabe, Y., Dessus, S., and Fabiani, S., "Safe Path Planning with Localization Uncertainty for Urban Operation of VTOL UAV," *AHS Annual Forum*, 2014.

[5] Bry, A., and Roy, N., "Rapidly-exploring random belief trees for motion planning under uncertainty," *2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[6] Achtelik, M. W., Lynen, S., Weiss, S., Chli, M., and Siegwart, R., "Motion- and Uncertainty-aware Path Planning for Micro Aerial Vehicles," *Journal of Field Robotics*, 2014.

[7] Ong, S. C., Png, S. W., Hsu, D., and Lee, W. S., "Planning under uncertainty for robotic tasks with mixed observability," *The International Journal of Robotics Research*, 2010.

[8] Kaelbling, L. P., Littman, M. L., and Cassandra, A. R., "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, 1998.

[9] Delamer, J.-A., Watanabe, Y., and Chanel, C. P. C., "MOMDP solving algorithms comparison for safe path planning problems in urban environments," *9th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, 2017.

[10] Silver, D., and Veness, J., "Monte-Carlo planning in large POMDPs," *Advances in neural information processing systems*, 2010.

[11] Kocsis, L., and Szepesvári, C., "Bandit based monte-carlo planning," *ECML*, 2006.

[12] Delamer, J.-A., Watanabe, Y., and Ponzoni Carvalho Chanel, C., "Towards a MOMDP model for UAV safe path planning in urban environment," *International Micro-Air Vehicle Competition (IMAV)*, 2017.

[13] Sorenson, H. W., *Kalman filtering: theory and application*, IEEE, 1985.

[14] Araya-López, M., Thomas, V., Buffet, O., and Charpillet, F., "A closer look at MOMDPs," *22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 2010.

[15] Buffet, O., and Sigaud, O., "Processus décisionnels de Markov en intelligence artificielle," , 2008.

[16] Madani, O., Hanks, S., and Condon, A., "On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems," *AAAI/IAAI*, 1999, pp. 541–548.

[17] Kurniawati, H., Hsu, D., and Lee, W. S., "SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces." *Robotics: Science and Systems*, 2008.

[18] Bonet, B., and Geffner, H., "Solving POMDPs: RTDP-Bel vs. Point-based Algorithms." *IJCAI*, 2009.

[19] Kolobov, A., *Planning with Markov decision processes: An AI perspective*, Vol. 6, Morgan & Claypool Publishers, 2012.

[20] Mettler, B., Kong, Z., Goerzen, C., and Whalley, M., "Benchmarking of obstacle field navigation algorithms for autonomous helicopters," *66th Forum of the American Helicopter Society:" Rising to New Heights in Vertical Lift Technology", AHS Forum 66*, 2010. URL `www.aem.umn.edu/people/mettler/projects/AFDD/AFFDwebpage.htm`.