POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Optimisation of UAV trajectories for target detection using a probabilistic approach

Laurea Magistrale in Space Engineering - Ingegneria Spaziale

Author: Giovanni Boscolo Boscoletto

Advisor: Prof. Davide Invernizzi

Co-advisors: Hélène Piet-Lahanier, Michel Kieffer

Academic year: 2024-2025

1. Introduction

The search for targets has been of practical interest for centuries and was rigorously analysed, for the first time, by Koopman during World War II, laying the foundation for *search theory* [3]. Due to advancements in UAV (Unmanned Aerial Vehicle) technology, search theory plays a critical role in several applications such as Search and Rescue (SAR) [5], infrastructure maintenance, precision agriculture, and transport [6].

A possible approach to this problem involves the use of a cooperative fleet of drones. Through information sharing, the aim is to optimise the search for target detection. More generally, the information to be taken into account may come from other agents or other sources, such as prior knowledge about a specific area of interest or directly from sensor measurements. The main challenge in carrying out such a mission is the generation of trajectories. To account for constantly changing information, an online or local problem formulation becomes necessary. These trajectories are updated periodically during the mission according to the information acquired. In addition to minimising an objective function for target detection, the trajectories must satisfy dynamic constraints and avoid obstacles in the environment. In this context of local re-planning, another key point is to exploit the updated information to select subsequent mission waypoints. For this reason, the term mission in this context refers to a set of waypoints and the trajectories connecting them.

The thesis can be divided into two main parts. The first is aimed at addressing the problem of trajectory generation. In particular, the decision was made to focus on the simplified problem in which the end point of the mission is already known before departure, during the planning phase. For this reason, in the thesis, we refer to offline or global trajectories. The future goal is to use this fundamental brick for the transition to local re-planning. The proposed framework generates optimal trajectories given a start point, an end point, prior target information, and an area of interest with known obstacles. Another aspect considered in the optimisation is the detection process. Depending on the sensors chosen for the mission, these have a direct influence on the trajectory selection. The second part of the thesis, on the other hand, concerns the method of updating the information as the UAV follows the previously calculated trajectory. This is done to address the future goal of local re-planning. It is desirable to update the existing information with new data coming from the sensors or through communications to select the next waypoints of the mission. The information about the presence or absence of targets is modelled probabilistically, and its update during the mission is incorporated using a Bayesian update method, as described in [2]. In the present work, we specifically consider a single UAV searching for a single fixed or mobile target. The obstacles are known and modelled as circles. Neither the UAV nor the target is allowed to fly over or enter any obstacles.

Bernstein polynomials (BPs) are chosen to represent the trajectories. As shown in [1], the characteristics and properties of these polynomials allow the development of a trajectory generation framework. The optimisation problem proposed by the authors produces trajectories that avoid obstacles while minimising a specified cost, such as mission time or path length. The contribution of this thesis is the integration of a target detection component into the framework proposed in [1]. Specifically, we incorporate prior information about the targets and the characteristics of the detection process into the optimisation procedure. For the detection component, the reference is the work in [4], which, however, addresses a different context, without obstacles and based on a different mathematical model.

2. Problem Description

The first assumption is that the drone in the mission moves at a constant altitude relative to the ground, which is assumed to be flat. The dynamics are therefore simplified, representing a planar movement with coordinates x_1^{u} and x_2^{u}. In addition, no dynamic relation is considered between the pitch angle and the horizontal velocity of the drone. A third assumption is that the sensor used to detect the targets is rigidly mounted and oriented in the UAV's motion direction. The equations governing the motion of the drones are:

$$\begin{cases} \dot{x_1}^{\mathrm{u}}(t) = V(t)\cos\psi(t) \\ \dot{x_2}^{\mathrm{u}}(t) = V(t)\sin\psi(t) \\ \dot{\psi}(t) = \omega(t), \end{cases} \quad (1)$$

where ψ represents the heading angle, positive counter-clockwise from the axis x_1 to the direction of the velocity, tangent to the trajectory. The control inputs are the velocity magnitude V and the angular rate ω.

The control inputs are subjected to the constraints $V_{min} \leq V(t) \leq V_{max}$ and $-\omega_{max} \leq \omega(t) \leq \omega_{max}$, with V_{min}, V_{max}, $\omega_{max} > 0$. We take advantage of the fact that the system 1 is differentially flat to rewrite the inputs as functions of the flat output $\boldsymbol{x}^{\mathrm{u}}(t) = [x_1^{\mathrm{u}}(t), x_2^{\mathrm{u}}(t)]^{\mathsf{T}}$ and its derivatives. In this way, the parameters to optimise are exclusively the control points of the flat output. Additionally, boundary conditions are included: initial position and velocity, and final position. The equations and inequalities of the problem are then rewritten using BPs.

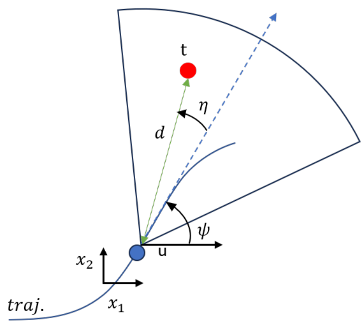

Figure 1: drone FOV

The drone's field of view (FOV) is represented by a circular section centred on the drone, with an angular width of FOV^{u} and a radius of d_{max}. A target can be observed only if it is within this area. Furthermore, depending on its position within the FOV, there will be varying probabilities of correctly identifying the target. The function used to describe the detection is a symmetric 2-dimensional Beta distribution, with parameters η and d, Figure 1. The function is normalized to ensure a maximum value of 1 at the centre, and it is denoted as the instantaneous detection rate $r(\eta, d)$.

$$r(\eta, d) = \frac{r_1(\eta) \cdot r_2(d)}{r_1(0) \cdot r_2((d_{max} - d_{min})/2)}, \quad (2)$$

where, for the general FOV domain $[-FOV^{\mathrm{u}}/2, FOV^{\mathrm{u}}/2] \times [d_{min}, d_{max}]$, we have the following equations, one for each variable:

$$r_1(\eta) = \left(\eta - (-FOV^{\mathrm{u}}/2)\right)^{\alpha-1} \cdot \left(FOV^{\mathrm{u}} - (\eta - (-FOV^{\mathrm{u}}/2))\right)^{\alpha-1} \quad (3)$$

$$r_2(d) = \left(d - d_{min}\right)^{\alpha-1} \cdot \left((d_{max} - d_{min}) - (d - d_{min})\right)^{\alpha-1} \quad (4)$$

In equations 3 and 4, we consider the family of symmetric Beta distributions, and therefore defined by a single parameter, α.

The targets are simplified and represented as points that lie in a 2-dimensional space. What specifically interests us is modelling the information the UAV has about the targets and the area of interest, which the optimisation algorithm uses to design the trajectory. Two scenarios regarding prior information are considered.

- *Scenario 1*: a fixed target with an imprecise position modelled using a 2-dimensional Beta distribution.
- *Scenario 2*: a potentially moving target with an unknown position.

For each of these scenarios, the mission area is discretised, and a probability value is assigned to each cell, representing the likelihood of the target being at that specific location. The collection of all these probabilities is called probability map. For *Scenario 1*, the probability is zero everywhere except within the domain of the Beta distribution, where the probability is defined by its probability density function. For *Scenario 2*, instead, the a priori distribution is uniform throughout the area and takes a value that is the inverse of the number of cells the target can reach, excluding obstacles where it is zero.

3. Trajectory Representation

Bernstein polynomials are chosen to represent the trajectories. The general definition of a n-dimensional, N^{th} order BP is:

$$\mathbf{C}_N(t) = \sum_{i=0}^{N} \mathbf{P}_{i,N} B_{i,N}(t), \quad (5)$$

with $t \in [0, t_f]$. $\mathbf{P}_{i,N} \in \mathbb{R}^n$ are the control points, $i = 0, \dots, N$. $B_{i,N}(t)$ are the BP bases and are defined as $B_{i,N}(t) = \binom{N}{i} \frac{t^i (t_f - t)^{N-i}}{t_f^N}$ for all $i = 0, \dots, N$. An example of a 2-dimensional, 3^{rd} order BP with its control points is represented in Figure 2.

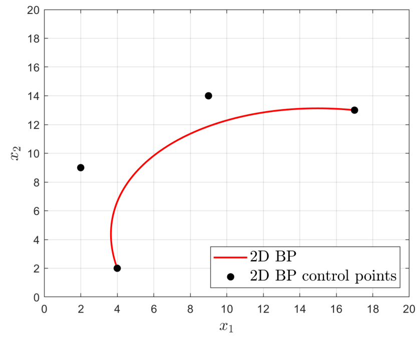

Figure 2: 2-dimensional, 3^{rd} order BP

As shown in Figure 2, Bernstein polynomials allow for the generation of smooth and continuous trajectories using a small number of parameters, namely the control points. These control points are the variables used in the optimisation process. Some of the properties used in the implementation are reported below. The *Convex Hull* property states that a BP lies within the convex hull defined by its control points. This allows certain trajectory constraints to be imposed by verifying them only at the control points. The *de Casteljau* algorithm instead allows for the evaluation of a BP at any point in time along the trajectory, not just at the control points. This step is necessary during discretisation, which must be carried out in the numerical implementation. Finally, other properties enable the implementation of efficient algorithms for trajectory planning, such as the distance between BP curves and obstacles.

4. Trajectory Optimisation

The cost function utilises the instantaneous detection rate function $r(\eta, d)$. In particular, it relies on the exponential detection model. Minimising the equation 6 is equivalent to minimising the risk that a UAV fails to detect the target in a zone of interest $\boldsymbol{\Delta}$ by the end t_f of a search mission. $\boldsymbol{X}^{\mathrm{t}}$ is the random variable for the target's position and $f_{\boldsymbol{X}^{\mathrm{t}}}(\boldsymbol{X}^{\mathrm{t}})$ is its probability density function, in our case the probability map. The cost function is written as:

$$J = \int_{\boldsymbol{\Delta}} e^{-\int_0^{t_f} r(\boldsymbol{P}^{\mathrm{u}}_{i,N}, \boldsymbol{X}^{\mathrm{t}}, t) dt} f_{\boldsymbol{X}^{\mathrm{t}}}(\boldsymbol{X}^{\mathrm{t}}) d\boldsymbol{X}^{\mathrm{t}}. \quad (6)$$

To prioritise detection in the trajectory design, an initial optimisation is performed that considers only the information about the target. Obstacle avoidance constraint is added only in the subsequent optimisation. The initial guess of the first optimisation is the set of control points that generates a straight-line trajectory from the starting point to the final point. The result of the first optimisation is used as the initial guess for the second one.

5. Target Location Estimation

As mentioned earlier, when referring to the target we consider the information that the UAV possesses about it, the whole area of interest. For this reason, by estimation of target location, we refer to the process used to update the probability map as the mission progresses, *i.e.*, traversing the optimal trajectory previously calculated given the initial information. The process of estimation is divided into two parts: correction and prediction steps. The correction step is used to update the existing information $P_{j,k-1}$ (the probability that the target is in cell j at time instant $k-1$), based on new measurements taken $y_{j,k}$ (measurement of cell j at time instant k. 1 if the target is detected in that cell, 0 if not, and $\emptyset$ if the cell is not observed). This is done using Bayes' formula:

$$P_{j,k} = \begin{cases} \frac{p_D P_{j,k-1}}{p_D P_{j,k-1} + p_F(1-P_{j,k-1})} & \text{if } y_{j,k} = 1 \\ \frac{(1-p_D)P_{j,k-1}}{(1-p_D)P_{j,k-1} + (1-p_F)(1-P_{j,k-1})} & \text{if } y_{j,k} = 0 \\ P_{j,k-1} & \text{if } y_{j,k} = \emptyset. \end{cases}$$

The terms p_D and p_F are defined as the probability of detection and the probability of false alarm, respectively. In the case where it is assumed that the target can move, it is necessary to add the prediction step. The probability map is propagated over time. The process is carried out using the state transition matrix A with the following equation:

$$P_{k+1|k} = A \cdot P_{k|k}. \tag{7}$$

Given a certain environment and under the assumption of a maximum velocity reachable by the target, the matrix is constructed once and for all. It is assumed that the target cannot enter obstacles or leave the area of interest.

6. Results

The results are obtained using MATLAB built-in function *fmincon*, in particular using the Sequential Quadratic Programming (SQP) algorithm. The final time chosen is $t_f = 30$ s.

6.1. Scenario 1

Regarding *Scenario 1*, Figure 3 shows the trajectory from the first optimisation in red. This trajectory serves as the initial guess for the second optimisation, which produces the final trajectory shown in blue. It can be observed that the latter tries to exploit the available probabilistic information as much as possible by passing through the centre of the Beta distribution (whose domain is outlined with a black frame).

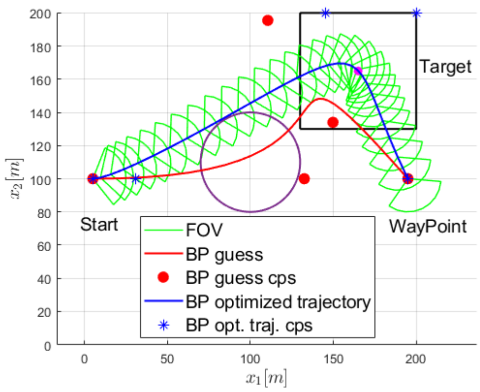

Figure 3: *Scenario 1* optimised trajectory

6.2. Scenario 2

For *Scenario 2*, the final trajectory obtained is the red one in Figure 5. Once this trajectory is obtained, it is simulated and two cases are analysed with respect to the estimation of target location. In the first case, no target is inserted in the simulation, and the parameters p_D and p_F are set to 1 and 0, respectively, representing the case of perfect detection. In Figure 4, the probability map at the final time is shown. In this first case, we observe that the trajectory obtained using the probabilistic information fully utilises the FOV without passing near the obstacle. Moreover, since we assumed a possibly moving target, we observe how the probability 'propagates' over time in the absence of measurements in a given area.

In the second case, the trajectory is simulated 500 times. A fixed target is added at position

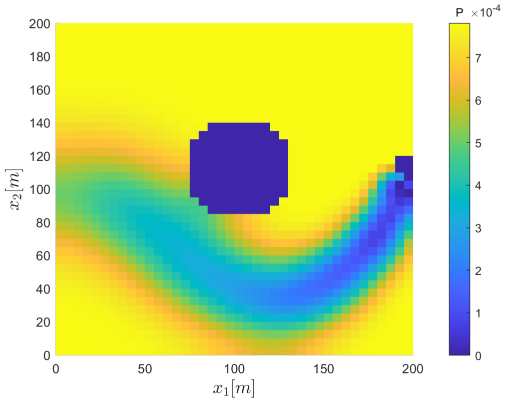

Figure 4: *Scenario 2* - case 1

(100, 40) m, and measurement randomness is introduced using $p_D = r(\eta, d)$ and $p_F = 0.3p_D$. After each simulation, the maximum value of the probability map is stored. The 500 maxima are reported in Figure 5 and are mainly organised into two groups: one is in the vicinity of the true position of the target, and the other is around the *FOV* of the most recent measurements. When the same point is observed several times, the accuracy of measurements increases. For this reason, the latest measurements are more prone to false alarms.

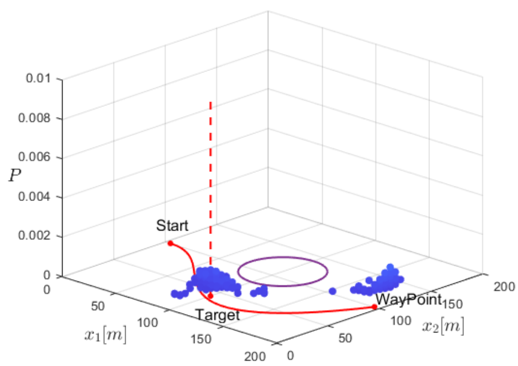

Figure 5: *Scenario 2* - case 2

Regarding the computational cost, it takes approximately 32 seconds to compute the optimal trajectory for *Scenario 2*. This time is currently too high to allow an online implementation. The time required to obtain the trajectory depends not only on the optimizer itself and how it is configured but also on the following factors: spatial discretisation of the area and temporal discretisation of the trajectory, number and arrangement of obstacles, and initial guess in terms of control points. While the first two factors are relatively constrained, further investigation and studies can be conducted on the third aspect. Experience from other research groups working on similar problems shows that the choice of optimiser and programming language is crucial to reducing computation time. For example, switching from MATLAB to C++ resulted in performance improvements of up to a factor of 100.

7. Conclusions

We can state that an efficient framework for the offline generation of optimal trajectories has been developed. This optimisation not only produces trajectories that comply with the system dynamics and the imposed constraints but also considers probabilistic information regarding the presence or absence of the targets. Using Bernstein polynomials has proven effective for describing smooth and dynamically feasible trajectories. However, the choice of the degree N, i.e., the number of control points, is non-trivial and strongly depends on the considered environment. Due to the presence of obstacles, the problem is non-convex. This characteristic means that the initial guess has a significant influence on both the success of the optimisation and the global optimality of the final result. Moreover, with the current problem formulation, computation times remain high, which limits the feasibility of a local online replanning. Additionally, the estimation of the target position is dependent on the performance of the detection process. Nonetheless, consistent results and an improvement in the initial information have been generally observed.

For future work, two main directions are promising. The first concerns the enhancement of the detection model and the investigation of alternative methods for choosing the initial guess. Once this is achieved, the main focus should shift toward developing a framework capable of online planning. To support this transition, it is recommended, as previously mentioned, to implement the framework in a more efficient programming language that is directly applicable to embedded software. To fully exploit the potential of local replanning during a mission, it is also advisable to introduce multiple agents and extend the approach to a 3-dimensional environment. Finally, it is necessary to implement a logic that takes advantage of the probabilistic information shared across the UAV fleet to continuously update subsequent waypoints.

References

[1] Venanzio Cichella, Isaac Kaminer, Claire Walton, and Naira Hovakimyan. Optimal motion planning for differentially flat systems using bernstein approximation. *IEEE Control Systems Letters*, 2(1):181–186, 2017.

[2] Jinwen Hu, Lihua Xie, Kai-Yew Lum, and Jun Xu. Multiagent information fusion and cooperative control in target search. *IEEE Transactions on Control Systems Technology*, 21(4):1223–1235, 2012.

[3] BO Koopman. Search and screening. operations evaluation group, office of the chief of naval operations, washington dc. Technical report, OEG Report, 1946.

[4] Sean Kragelund, Claire Walton, Isaac Kaminer, and Vladimir Dobrokhodov. Generalized optimal control for autonomous mine countermeasures missions. *IEEE Journal of Oceanic Engineering*, 46(2):466–496, 2020.

[5] Mingyang Lyu, Yibo Zhao, Chao Huang, and Hailong Huang. Unmanned aerial vehicles for search and rescue: A survey. *Remote Sensing*, 15(13):3266, 2023.

[6] Alena Otto, Niels Agatz, James Campbell, Bruce Golden, and Erwin Pesch. Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. *Networks*, 72(4):411–458, 2018.