Generation of RNP Approach Flight Procedures with an RRT* Path-Planning Algorithm

Raúl Sáez and Daichi Toratani
Air Traffic Management Department
Electronic Navigation Research Institute (ENRI)
Chofu, Tokyo, Japan
saezgarcia-r@mpat.go.jp, toratani-d@mpat.go.jp

Ryota Mori
Graduate School of Maritime Sciences
Kobe University
Kobe, Hyogo, Japan
r-mori@people.kobe-u.ac.jp

Xavier Prats
Department of Physics-Aerospace division
Technical University of Catalonia (UPC)
Castelldefels, Barcelona, Spain
xavier.prats@upc.edu

***Abstract*—We present a framework capable of generating required navigation performance authorization required approach (RNP AR APCH) procedures by using a combination of the optimal version of the path-planning rapidly-exploring random tree (RRT*) algorithm and Dubins paths. Procedures are generated by taking into account design constraints defined by the international civil aviation organization (ICAO) procedures for air navigation services - aircraft operations (PANS-OPS). The framework is used to compute several approach procedures for two airports in Japan, Kumamoto and Kitakyushu airports. Several feasible procedures are successfully obtained in a low amount of computational time, many of them resembling the actual procedures published in the selected airports. The output of our framework represents a valuable input for procedure designers, who could later refine the obtained results with specific flight-procedure-design software.**

Keywords—Path Planning; Required Navigation Performance; Performance Based Navigation; Route Design; Rapidly-Exploring Random Tree

I. Introduction

Optimal procedure design in civil aviation is one of the main factors affecting aircraft operations. It has a direct effect on, for instance, the flight time and fuel consumption, besides being a key element to airport accessibility for flights under instrumental flight rules (IFR), impacting as well airport capacity. Traditionally, the design of routes (i.e., the horizontal component of the procedure) was heavily constrained by the location of navigational aids on the ground. Thus, in the past, there was not much freedom in designing a route. Improvements in navigation systems, however, such as the introduction of global navigation satellite systems (GNSS), have enabled much more flexible route designs such as required navigation performance (RNP) approaches. RNP is a family of navigation specifications under performance based navigation (PBN) which enables the operation of aircraft along a precise flight path with the ability to determine aircraft position with a given level of both accuracy and integrity. Within RNP procedures, there is a special category of navigation specifications called RNP authorization required approach (RNP AR APCH) procedures, which require a lateral total system error lower than the standard RNP values on any segment of the approach procedure. An RNP AR APCH is a procedure that allows for narrow, linear obstacle clearance corridors in the procedure design, due to the assurance of specific navigation performance provided by aircraft on-board position monitoring and alerting systems. The regulatory authorities have labelled these procedures as AR, because of the monitoring and alerting systems required on the aircraft, as well as the pilot training required for the approaches. A formal approval in the form of a letter of authorization or operation specification is required for operators wanting to fly these approaches. RNP AR APCH was mainly developed to allow procedures to be implemented in challenging obstacle environments where conventional obstacle protection surfaces limit the possibility of implementation.

Depending on the scenario, it might be difficult to manually design an optimal IFR procedure. On the one hand, designing a procedure for an aircraft arriving at a runway faces complex constraints. For instance, in the case of an RNP AR APCH, the procedure can be designed using a combination of straight segments and curved flight paths, known as radius-to-fix (RF) legs. In addition, as in conventional flight procedures, the flight path must maintain a certain distance from the ground surface and obstacles. On top of that, the designed procedure must comply with the International Civil Aviation Organization (ICAO) procedure design rules [1], [2], which need to be considered when designing an optimal aircraft arrival. On the other hand, several potential procedures could comply with the constraints set by ICAO, making it difficult to choose the optimal one.

In previous publications [3], some of the authors of this work investigated the possibility of automatically generating optimal RNP AR APCH procedures by using genetic algorithms (GAs). Optimal procedures were obtained in several case studies conducted at several airports. Their applicability, however, was limited, because the proposed methodology required trial-and-error adjustments of several constraints and initial-guess settings to obtain optimal solutions. Furthermore, computational times were quite long, especially when the constraints imposed by ground obstacles were evaluated. This situation is undesirable from the perspective of procedure design, as flight-procedure designers prefer to obtain multiple feasible solutions—satisfying all the constraints required—in a small amount of time rather than obtaining one solution that is optimal but takes a long time to be computed.

Because of the problems aforementioned, in this work we are testing a new path-planning algorithm in order to automatically generate optimal RNP AR APCH procedures.

Several approaches can be found in the literature intending to find a solution to the path-planning problem. Depending on the application, the objective may be very different, from just finding a safe path to finding a path minimizing or maximizing a given criteria (e.g., distance, consumption, time, etc). Some of the most remarkable path-planning algorithms include optimal control, graph, or front-propagation approaches, which allow to find an optimal solution. More recently, methods based on graph generation called sampling-based-path-planning algorithms have been developed. First introduced in [4], these methods have been improved in [5] to be asymptotically optimal. Among them, probabilistic roadmap (PRM), rapidly-exploring random tree (RRT) and its optimal version (RRT*), and fast marching trees (FMT*) [6] can be mentioned. First used in robotics, they have recently been used in aviation.

In this paper, we present a framework that uses an adapted version of the RRT* algorithm to generate RNP AR APCH procedures, in which the radius of turn is considered when smoothing the path by using Dubins paths. This algorithm has a low time complexity, thus, leading to low computational times. It is also asymptotically optimal and it was recently used by some of the authors of this paper in previous publications [7] to successfully generate aircraft emergency routes. We reuse the procedure-design constraints proposed by ICAO and previously applied with the GAs approach [3] together with this path-planning algorithm, with the aim of obtaining, in a short amount of time, feasible RNP AR APCH procedures for some representative airports in Japan.

The work presented in this paper is expected to provide a valuable input for flight-procedure designers. Currently, flight-procedure-design software such as PANADES [8] is used to generate flight procedures for all stages of flight. This kind of software is compliant with the ICAO PANS-OPS criteria [1] and allows the user to design procedures, assess obstacles, manage data from the aeronautical information publication (AIP) or display data with high accuracy and in several layers (e.g., elevation information). We do not expect our results to replace such a kind of comprehensive software. However, after discussing the matter with flight-procedure designers, it was concluded that it would be very helpful for them to have such a kind of automated framework capable of generating a set of input trajectories that could be later refined with the flight-procedure-design software.

II. Literature Review

Previous works have dealt with the generation of routes by using different implementations of the RRT* algorithm. In [9], the authors used a combination of an RRT* algorithm and Dubins paths to generate emergency trajectories leading the aircraft to an emergency landing site, with very simple assumptions regarding the aircraft dynamics. A similar approach was applied in [10], where a missile application was tackled and several kind of obstacles were considered, together with a more comprehensive formulation of the aircraft dynamics.

In [11] and [12], applications of similar nature are found, where the dynamics of robots were considered when generating paths avoiding obstacles with the RRT* algorithm. An improved version of the RRT algorithm for a robot application was applied in [13], in which Gaussian mixture models were applied in order to reduce the search space of the RRT.

More recently, in [14], the authors proposed a new application involving the use of the RRT algorithm for robots path planning plus a combination of path pruning and path smoothing by using Bézier curves. Finally, a first approach towards the consideration of RNP requirements together with the use of an RRT algorithm was presented in [15], in which an A* algorithm was considered for path pruning and the smoothing of the path was achieved via B-spline curves, with no consideration of the aircraft turn dynamics.

There are many applications in the literature focusing on robot path planning with RRT or on the generation of aircraft trajectories (e.g., emergency trajectories) avoiding static or dynamic obstacles. However, to the best of the authors' knowledge, no other work has tackled the specific problem of flight-procedure design by using an RRT* algorithm, in which all the constraints defined by ICAO are considered [2], and in which the achievable radius of turn by the arriving aircraft is taken into account when smoothing the path with Dubins curves.

Other authors tried to tackle the flight-procedure design problem by using other methodologies. For instance, the seminal works conducted by Visser et al. [16] or Prats et al. [17] focused on the optimization of the nominal flight path using an optimal control strategy. Building on top of these works, [18], [19] successfully explored the application of evolutionary algorithms to address the same problem. All these works focused specifically on the design of noise abatement procedures and only the nominal flight path was considered, hence, disregarding the minimum obstacle clearance considerations and RNP corridors established by the ICAO. In this context, a survey comparing different optimization methods can be found in [20].

More recently, in [21], a mathematical and computational framework was proposed to automatically design instrument procedures, by combining a simulated annealing optimization with the well-known Bellman-Ford algorithm. Yet, this work mostly addressed the design of the nominal flight path, and not all ICAO defined constraints were considered.

III. RNP AR APCH Design Considerations

In this section, we describe some of the essential concepts of RNP AR procedure design, which were considered when applying the methodology used in this work. A series of constraints were taken into account when generating the approach procedures, which follow the guidelines regarding the design of RNP AR procedures defined by ICAO [1], [2]. In this section, we will only mention some of the most relevant constraints considered.

RNP is a family of navigation specifications under PBN which enable the operation of aircraft along a precise flight path with a high level of accuracy and the ability to determine aircraft position with both accuracy and integrity. This leads

to many benefits, such as an increase of airspace efficiency through reduced separation; a better use of multiple airport runway configurations for increased airport capacity; or reduced fuel burn/emissions from shorter flight paths via not being constrained to overfly navigational-aids on the ground.

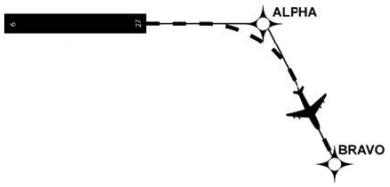
Among RNP procedures, RNP AR procedures are amongst the most modern and precise instrument approach options available nowadays, including unique capabilities that require special aircraft and aircrew authorization. They incorporate additional navigational accuracy, integrity and functional capabilities to permit operations using reduced obstacle clearance tolerances that enable approach and departure procedures to be implemented in circumstances where other types of approach and departure procedures are not operationally possible or satisfactory. RNP AR procedures require a lateral total system error (TSE) lower than the standard RNP values on any segment of the approach procedure. More specifically, a lateral TSE as low as ± 0.1 NM during 95 % of the flight time is required on any segment of the approach procedure.

In the following sections, we will describe some of the most relevant constraints for the problem tackled in this paper.

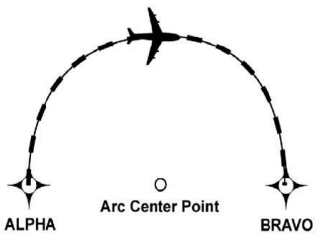
A. Segment and Leg Types

There are mainly 4 segments in an RNP AR APCH procedure: the initial, intermediate, final, and missed approach segments. Note, however, that in this work the missed approach segment is not considered.

For each segment, the ICAO procedure design rules [2] contemplate the use of only two leg types: track-to-fix (TF) and radius-to-fix (RF). TF legs are intercepted and acquired as the flight track to the following waypoint (Fig. 1(a)), while RF legs are defined as a constant radius circular path around a defined turn center that terminates at a fix (Fig. 1(b)). TF legs



(a) Track-to-fix leg



(b) Radius-to-fix leg

Fig. 1. RNP AR leg types [22]

are geodesic flight paths between two fixes and is the normal standard leg used in RNP AR procedures. TF legs are linked by fly-by waypoints or RF legs. More information regarding turns is given in Section III-B.

B. Turns

Depending on the leg type (i.e., TF or RF), the turn constraints will be different. For TF legs, fly-over waypoints are not permitted when designing RNP AR procedures, so only fly-by turns are used [1]. In addition, turn angles for TF legs are limited to a maximum of 70 degrees where aircraft are expected to fly-by the fix at altitudes above FL190, and to 90 degrees at and below FL190. An RF leg should be used if these constraints cannot be met.

In order to compute the turn radius applied at fly-by fixes, a standard bank angle (ϕ) of 18 degrees is used. The turn is assumed to be performed at a given indicated airspeed (IAS) for the fastest aircraft category for which the procedure is designed. The corresponding true airspeed (TAS) is obtained for the highest altitude allowed in the turn; then, a tailwind component which also depends on the altitude is added to it. In order to know the radius of turn (r), first the rate of turn (R) is computed:

$$R = \frac{3431 \ \tan \phi}{\pi V_g}, \tag{1}$$

where V_g is the TAS (in [kt]) plus the tailwind speed component. Once R is computed it is possible to derive r:

$$r = \frac{V_g}{20\pi R}. \tag{2}$$

Non-standard bank angles are allowed for smooth transitions, maintaining stabilized approaches, lower minima or to achieve specific leg lengths. Table I presents the allowed bank angle window. The heights shown are above threshold. When

TABLE I
BANK ANGLE WINDOW

Lowest above ground level height in RF segment	**Maximum bank angle [deg.]**
< 150 m (492 ft)	≤ 3
$\geq$ 150 m (492 ft)	≤ 20

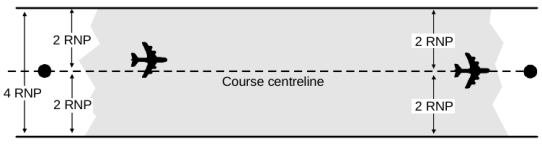
using RF legs, the bank angle required for a given TAS (in [kt]), tailwind speed w (in [kt]) and turn radius (in [deg]) is the following:

$$\phi = \frac{\arctan V_g^2}{68625 \ r}. \tag{3}$$

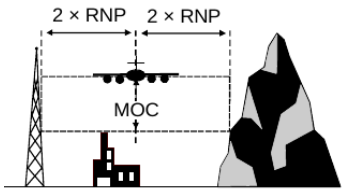
C. Segment Widths

Segments in RNP AR procedures have a protection-area width equal to 4 times the RNP navigation accuracy requirement and a semi-width equal to 2 times the RNP navigation accuracy requirement (Fig. 2). The minimum, standard and maximum values for RNP navigation accuracy requirements are presented in Table II. As shown in Fig. 2(b), for each

[1]Fly-by waypoints require turn anticipation to allow tangential interception of the next segment of a route or procedure, while flyover waypoints are waypoints at which a turn is initiated in order to join the next segment of a route or procedure, thus, overflying the waypoint.



(a) Plan view



(b) Cross-section view

Fig. 2. RNP AR segment widths [2]

TABLE II
RNP NAVIGATION ACCURACY REQUIREMENTS

Segment	Minimum	Standard	Maximum
Initial	0.1 NM	1 NM	1 NM
Intermediate	0.1 NM	1 NM	1 NM
Final	0.1 NM	0.3 NM	0.5 NM

segment, an obstacle clearance with respect to the obstacles inside the protection area needs to be ensured. This clearance depends on the minimum obstacle clearance (MOC) value, which depends on each segment and is detailed in Table III. For the final approach segment (FAS), there is not a fixed

TABLE III
MINIMUM OBSTACLE CLEARANCE (MOC)

Segment	MOC
Initial	984 ft
Intermediate	492 ft
Final	Evaluated using OAS

value for the MOC and the obstacle assessment surface (OAS) needs to be computed first, as detailed in [2]. Computing the MOC for the vertical error budget (VEB) is needed in order to derive the height of the OAS at any distance from the landing threshold point (LTP).

When merging two segments with different protection-area widths at a given fix, it is necessary to evaluate for both segments the area within ± 1 RNP navigation accuracy requirement of the fix. Regarding turns, for RF legs the protection-area width is equal to that of the straight segment, while for fly-by turns between two TF legs the procedure is more complex, as detailed in [2].

D. Descent Gradients

For each segment, a given standard and maximum descent gradient (γ) is considered, as detailed in Table IV. For the FAS, the maximum descent gradient depends on the aircraft category, as detailed in [2].

TABLE IV
DESCENT GRADIENTS

Segment	Standard	Maximum
Initial	4% (2.4 deg.)	8% (4.7 deg.)
Intermediate	$\leq$ 2.5% (1.4 deg.)	Equal to FAS
Final	5.2% (3 deg.)	See [2]

IV. METHODOLOGY

In this section, we present the framework developed in this work, capable of generating aircraft approach procedures by using a combination of the RRT* algorithm, path pruning and path smoothing by using Dubins paths. In Sections IV-A and IV-B we describe, respectively, the RRT* algorithm and Dubins paths technique. Then, in Section IV-C, we focus on the framework itself, describing all the process followed in order to generate the approach trajectories.

A. RRT* Path-Planning Algorithm

Thoroughly described in [5], the RRT* algorithm is an algorithm designed to efficiently search non-convex, high-dimensional spaces by randomly building a space-filling tree. The tree is constructed incrementally from samples drawn randomly from the search space and is inherently biased to grow towards large unexplored areas of the problem.

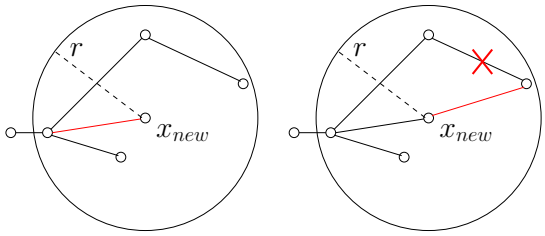
The RRT* algorithm incrementally builds a tree of feasible trajectories, rooted at the initial condition. The algorithm is initialized with a graph that includes the initial state as its single vertex, and no edges.

Let $\chi = (0,1)^d$ be the configuration space, where $d \in \mathbf{N}$ is the space dimension, ($d \geq 2$). Let χ_{obs} be an open set, which denotes the obstacle-free space as χ_{free} =cl($\chi \backslash \chi_{obs}$), where $cl(\chi)$ denotes the closure of a set χ. The initial condition is denoted by $x_{init} \in \chi_{free}$ and the goal region, χ_{goal}, is an open set of χ_{free}.

At each iteration of the algorithm, a point $x_{rand} \in \chi_{free}$ is sampled. Given a graph $G = (V, E)$—where V denotes the vertices set and E the edges set—the algorithm adds points to V as follows:

- An attempt is made to connect the nearest vertex $v \in V$ in the tree to the new sample.
- If such connection is feasible (i.e., no obstacles in the way), x_{rand} is steered to the nearest point x_{near}, obtaining x_{new}.
- Connections from the new vertex x_{new} to vertices that are within a certain distance from x_{new}—in a set denoted χ_{near}—are tested. Two elements should be considered when creating a new edge:
  - An edge is created from the vertex in χ_{near} that can be connected to x_{new} along a path with minimum cost (Fig. 3(a)).

– New edges are created from x_{new} to vertices in χ_{near} if the path through x_{new} has lower cost than the path through the current parent; in this case, the edge linking the vertex to its current parent is deleted in order to maintain the tree structure (Fig. 3(b)). This process is known as rewiring.



(a) New edge created (b) Previous edge removed

Fig. 3. Edge generation in RRT* [7]

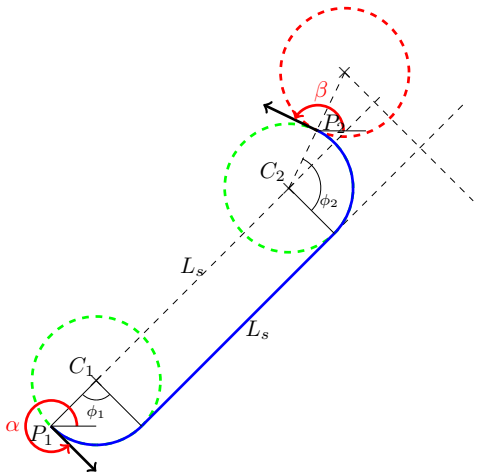


Fig. 4. Left-Segment-Left Dubins Curve [7]

B. Dubins Paths

Dubins paths were firstly introduced in [23], where it was showed that, for a forward-moving vehicle with a minimum turning radius r_{min}, the shortest path between a pair of coordinates with an associated orientation angle (i.e., (x, y, θ)) is composed entirely of no more than three circular arcs of radius r_{min} or straight lines. In the problem presented in this paper, the orientation angle θ is the track angle between two fixes.

Each segment type is categorized as follows: Right turn (R) , Left turn (L), and Straight (S). Thus, a path will always be at least one of these six types: RSR, RSL, LSR, LSL, RLR, LRL.

As an example, we show in this section the formulation for an LSL Dubins path [7], [24]. Let $P_1 = (x_1, x_2)$ and $P_2 = (x_2, y_2)$ be the initial and the final points, respectively. Let θ_1 and θ_2 be their track direction, respectively. An associated value of track is the orientation angle. Let α and β be the initial and final orientation angles, defined as follows:

$$\begin{cases} \alpha = & \mod\left(\frac{\pi}{2} - \theta_1, 2\pi\right) \\ \beta = & \mod\left(\frac{\pi}{2} - \theta_2, 2\pi\right). \end{cases} \tag{4}$$

Fig. 4 depicts an LSL Dubins curve. The curve connects a starting point with an orientation angle α and an end point with an orientation angle β. It is composed of a turn to the left around the first green circle, a segment of length L_s and a second turn to the left around the second green circle. The path ends on a given target circle (i.e., red circle in Fig. 4) with the heading in the tangential direction [24].

The length of the segment L_S and the two angles ϕ_1 and ϕ_2 are defined as follows :

$$L_S = \sqrt{(x_2 - r\sin\beta - x_1 + r\sin\alpha)^2 + (y_2 + r\cos\beta - y_1 - r\cos\alpha)^2}, \tag{5}$$

$$\phi_1 = \mod\left(\arctan\left(\frac{y_2 + r\cos\beta - y_1 - r\cos\alpha}{x_2 - r\sin\beta - x_1 + r\sin\alpha}\right), 2\pi\right), \tag{6}$$

$$\phi_2 = \mod\left(\beta - \phi_1, 2\pi\right). \tag{7}$$

C. RNP AR APCH Procedure-Generation Workflow

Our procedure-generation framework is mainly composed of three main modules: the RRT* module, the path-pruning module and the path-smoothing module. In addition, a series of inputs are necessary to run properly these modules. Furthermore, some auxiliary functions or modules are also needed when generating the approach procedures. A simple diagram showing the interactions between the different modules is shown in Fig. 5. Below, we detail the steps followed by the

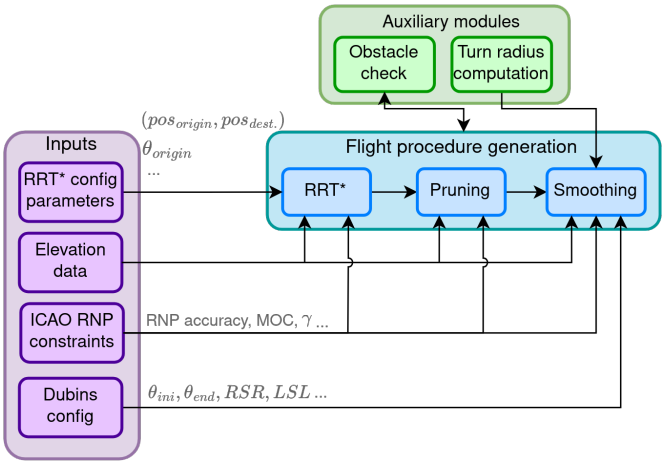


Fig. 5. RNP AR APCH procedure-generation framework workflow

framework in order to compute RNP AR APCH procedures.

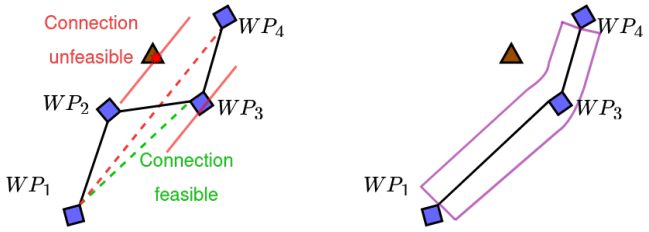
1) RTT path:* : the first step is to run the RRT* algorithm to compute a free-obstacle path from an origin to a destination, i.e., pos_{origin} and $pos_{dest.}$ in Fig. 5, respectively. Note that the

path is computed backwards, thus, starting from the runway. Not only the origin and destination positions are needed, but also the track angle required to properly land the aircraft, i.e., θ_{origin} in Fig. 5. In addition, as explained in Section IV-A, at some point of the process when running the RRT* algorithm, connections from a new vertex to vertices that are within a certain distance from that new vertex are tested. This distance is also one of the parameters that needs to be provided to the RRT* module, as well as the range at which random vertices are generated. Furthermore, the maximum number of iterations for the algorithm is also set. Finally, when attempting to add a new edge to the RRT* graph, the RNP AR constraints defined by ICAO [2] are taken into account. The process followed is described below:

- Given the final approach fix (FAF) and intermediate fix (IF) altitudes, it is possible to determine in which segment of the approach procedure (i.e., initial, intermediate or final) the new edge is going to be added to the RRT* graph.
- The descent gradient for each segment—γ in Fig. 5 and values described in Table IV—is used in order to compute the altitude of the new vertex that is being attempted to be added to the procedure (leading to a potential new edge in the RRT* graph).
- The RNP navigation accuracy requirement for the current segment (Table II) is used in order to determine the protection area width.
- The terrain-elevation data inside the protection area is used together with the MOC value for the current segment (Table III) in order to determine whether the required obstacle clearance is ensured. If that is the case, the new edge is added to the RRT* graph.

2) Path pruning: : when the number of vertices approaches infinity, the RRT* algorithm will deliver the shortest possible path to the goal. While realistically unfeasible, this statement suggests that the algorithm does work to develop a shortest path. However, in reality, although the path is straighter and shorter than the path obtained with the non-optimal version of the algorithm (i.e., RRT), there are still some randomly-generated vertices that would not allow to fly neither the straightest nor the shortest path to the arriving aircraft flying the approach procedure. Moreover, most of the time the paths obtained with the RRT* algorithm are not operationally sound, as they usually involve many turns which are undesirable from an aircraft operations point of view.

For all the reasons aforementioned, we perform a path-pruning operation to the path obtained with the RRT* algorithm. A simple diagram depicting the process followed is shown in Fig. 6. We attempt to establish connections in an iterative way for each of the waypoints of the resulting path generated by the RRT* algorithm. Usually, the RRT* path is quite straight, so the number of connections to be tested is not very high. However, for each of the connections, the ICAO RNP AR constraints need to be taken into account again. In Fig. 6(a), we can observe how the connection between the first waypoint



(a) Testing new connections (b) Path after pruning

Fig. 6. Path-pruning process

(WP_1) and the third waypoint (WP_3) is possible, as there are no obstacles in between. However, the connection between WP_1 and the forth waypoint (WP_4) is not possible, as there is an obstacle—the brown triangle in Fig. 6(a)—-inside the segment protection area. In this case, the RRT* algorithm correctly added WP_3 in order to avoid the obstacle. The resulting path after the path-pruning process is shown in Fig. 6(b). More advanced techniques could have been used to prune the path (such as an A* algorithm), but, due to the already straight nature of the path generated by the RRT* algorithm, the "manual" process used in this work has proven to be enough.

3) Path smoothing: : the path obtained after the path-pruning process could already be the final path proposed by the procedure-generation framework if only TF legs were used. However, in order to generate a path where RF legs are present, we need to smooth the path. In this work, we use Dubins paths to perform the path-smoothing operation, being one of the inputs for their computation the radius of turn (computed as shown in Section III-B). Once again, the ICAO RNP AR constraints are taken into account when smoothing the path.

V. Results

In this section, we present the results obtained in this work. In Section V-A, we describe the experimental setup, while in Section V-B, we focus on the different generated procedures for each of the scenarios tackled in this paper.

A. Experimental Setup

Two airports in southern Japan will be considered in this paper: Kitakyushu airport (RJFR) and Kumamoto airport (RJFT), both located in the island of Kyushu. The RNP AR approach chart for RJFR is shown in Fig. 7. As it can be observed, the approach route is located entirely over the sea, which greatly facilitates the task of the generation of the approach procedure with our framework. On the other hand, RJFT is located in quite a challenging location, with high mountains to the east of the airport. The RNP AR approach charts are shown in Fig. 8.

Regarding the RRT* configuration parameters, we used a maximum number of iterations of 500 for both scenarios. Then, random vertices were generated at a distance of 1 NM from the last vertex of the RRT* graph. Furthermore, the

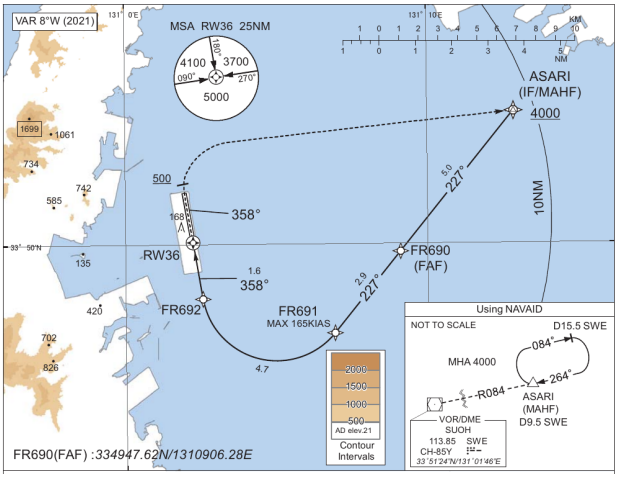


Fig. 7. Kitakyushu Airport (RJFR) RNP AR approach chart for RWY36 (source: Japanese AIP)

distance to check for less-costly connections once a new vertex is generated (Section IV-A) was also set to 1 NM.

The runway location was used as the starting point for the RRT* algorithm. Additionally, the runway direction was also considered. For the final point, the initial approach fix (IAF) location was used. It is also important to highlight the fact that for both scenarios we initially forced the RRT* algorithm to generate a straight segment (within the FAS) with an orientation matching the runway direction, and with the same length as the one in the published charts. This is something which is needed because, if not enforced, the RRT* algorithm would freely generate random points with any given direction from the origin (i.e., runway). Thus, by adding this constraint, we ensure that there will always be a segment aligned with the runway in order to stabilize the arriving aircraft before landing. The minimum length of this segment can be computed by following the ICAO RNP AR APCH procedure design rules [2], but, as aforementioned, in this paper we used the length of the stabilization segment located in the published charts.

Regarding the descent gradient and RNP navigation accuracy values, standard values were applied to all scenarios, except for a 3.1-degree descent angle for the final segment in the south case study in RJFT (as detailed at the end of this section). For both scenarios, we also used the FAF and IF altitudes of the published charts to determine the start and end altitudes of each approach segment. Note, however, that the FAF and IF locations (i.e., latitude and longitude) were not enforced. In future work, we will improve our framework in order to automatically choose these altitudes. Finally, only RF legs were considered for both scenarios.

For the terrain-elevation data, we used the digital elevation model (DEM) data provided by the Geospatial Information Authority of Japan [25]. This data has a gridded format with 5-meter intervals, leading to a total number of points—including latitude, longitude and elevation information—of 43,605,000 for the RJFT scenario. No DEM data was used for the RJFR scenario, as the approach route is located entirely over the sea.

Because of the random nature of the RRT* algorithm, every time the framework is executed, a different route is obtained. Still, depending on the particular scenario, more or less options are available. For instance, in the RJFR scenario, as there are no mountains in the surroundings of the airport, multiple routes could be generated by the RRT* algorithm from the origin to the destination. In this paper, the aim is to show that our framework is capable of generating approach procedures resembling those published in the approach charts (Fig. 7 and Fig. 8). In order to do that, we ran the following case studies for each scenario:

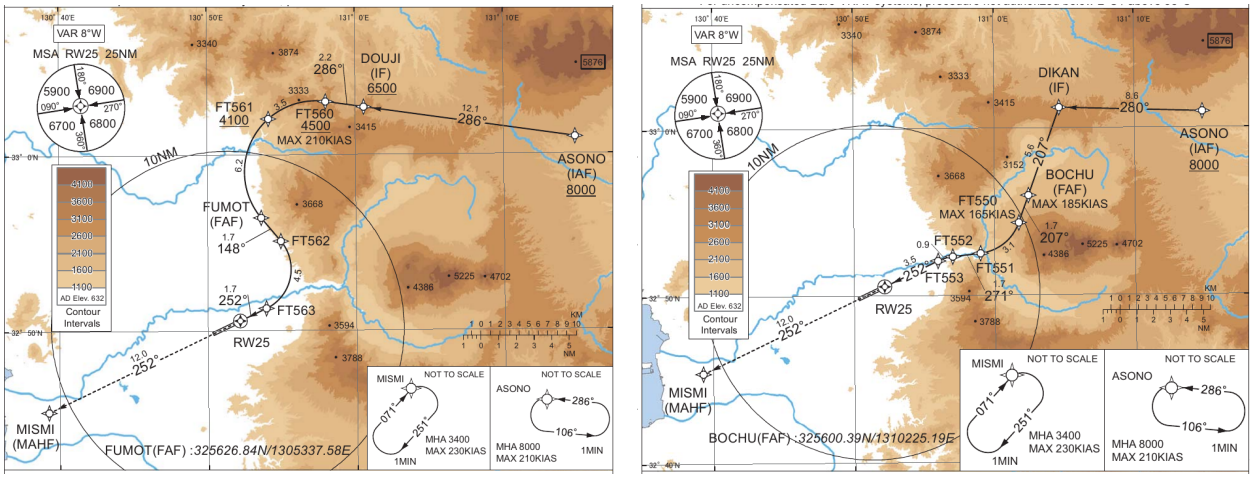
- **RJFR case study**: for the RJFR scenario, only one case study was run. In this case, as shown in Fig. 7, only the final and intermediate segment are required. We used the altitude of the FAF of the published chart, 3,000 ft, to determine the final altitude of the final segment. Additionally, the runway altitude and direction, 21 ft and 178 degrees (recall that the origin for our algorithm is the runway threshold), respectively, were given as inputs for our framework as well. This scenario is quite simple, but it allowed us to perform some initial tests for our framework.
- **RJFT north case study**: in this case study, the objective is to generate a route resembling the approach route shown in Fig. 8(a). A FAF altitude of 3,200 ft and an IF altitude of 6,500 ft were used in this case (i.e., the minimum altitudes in the published chart). The origin altitude and direction were, respectively, 632 ft and 70 degrees.
- **RJFT south case study**: in this case, the aim is to generate a route resembling the one shown in Fig. 8(b). As it was aforementioned, a 3.1-degree descent angle was used in the final segment. This was mainly done in order to avoid the high terrain right to the east of RJFT airport. The FAF altitude used was 4,300 ft and the IF altitude, 6,000 ft (i.e., the minimum values in the published chart). Same values for the origin altitude and direction as in the RJFT north case study were applied.

Table V summarizes the source of the parameters used by our framework (note the list of parameters is not exhaustive), namely, the ICAO design rules for RNP AR APCH, the published charts or as a result of our framework route-optimization process.

B. Proposed RNP AR APCH Procedures

In this section, we discuss the results for the three case studies presented in Section V-A.

In Fig. 9, we show the approach route generated for the RJFR case study. The published approach route is depicted in red, while the route generated by our framework is depicted in black. As it can be observed, the route generated by our framework matches quite well the published route, including as well an RF leg before the stabilization segment. Regarding the total length of the route, the one generated by our framework is 13.7 NM, while the one published in the approach charts is 14.2 NM.



(a) RNP Y

(b) RNP Z

Fig. 8. Kumamoto Airport (RJFT) RNP AR approach charts for RWY25 (source: Japanese AIP)

TABLE V
SOURCE OF THE FRAMEWORK PARAMETERS

Parameter	Source
MOC	ICAO
Segment width	ICAO
Descent gradient	ICAO
Bank angle	ICAO
FAS Descent gradient (RJFT south)	Chart
Stabilization segment length	Chart
FAF & IF altitude	Chart
Origin	Chart (i.e., RWY)
Destination	Chart (i.e., IF/IAF)
FAF & IF location	Optimized
Route (i.e., lateral path)	Optimized

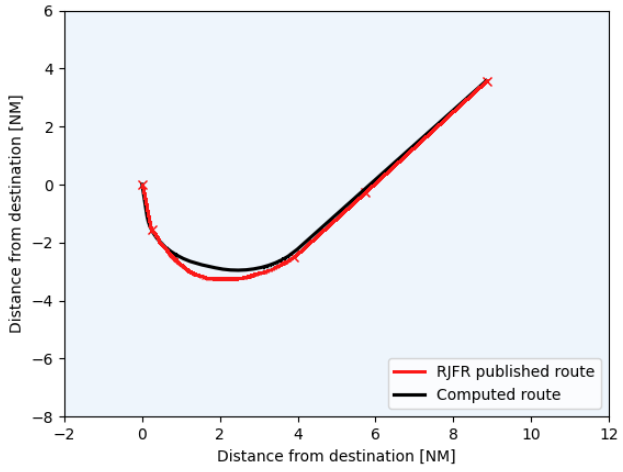


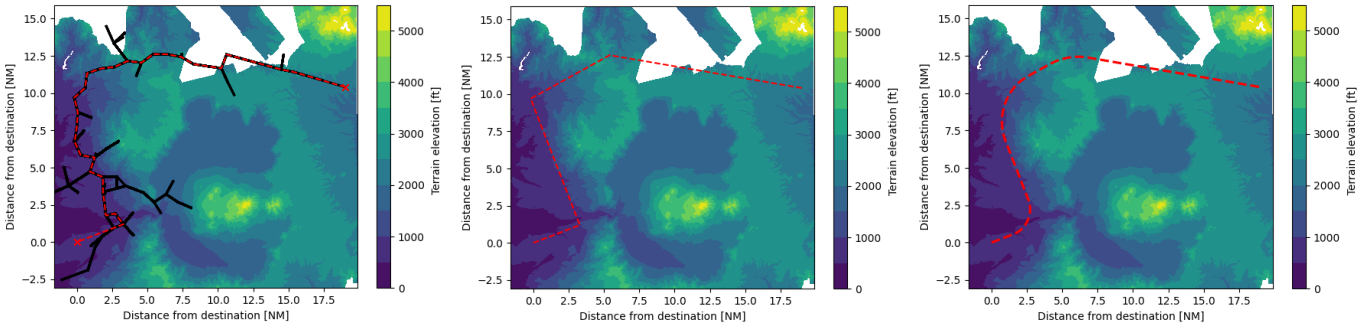
Fig. 9. Comparison of computed trajectory and published chart for Kitakyushu Airport (RJFR)

For the RJFT case studies, the results obtained are more remarkable, as the terrain in the vicinity of the airport makes the scenario more challenging. First, in Fig. 10, we show the process followed by the framework to generate the approach route for the RJFT north case study. We depict the resulting path generated by the RRT* algorithm, the path-pruning process and the path-smoothing process in Fig. 10(a), Fig. 10(b) and Fig. 10(c), respectively. As it can be observed, the path generated by the RRT* algorithm is already quite straight; however, there are still some unnecessary waypoints that can be skipped when pruning the path, as shown in Fig. 10(b).

In Fig. 11, we show the final trajectories (i.e., after smoothing the path) generated for both RJFT case studies. Similarly to the RJFR case study, we also plotted the published route in order to compare it with the route obtained with our framework. As it can be observed, the routes are not identical, but quite similar.

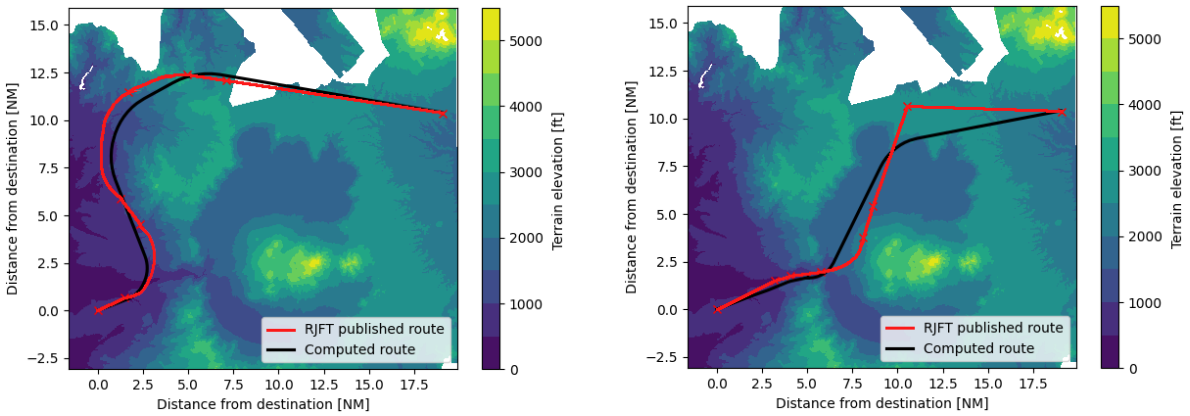
The challenging terrain to the east of RJFT is known as the Aso Volcano. A small corridor to the east of RJFT—with an altitude of 656 ft surrounded by mountains of 2,600 ft—allows access to the caldera, which measures 13.5 NM north to south, 9.7 NM east to west and has a circumference of 54 NM. In the central part of the caldera there is a group of volcanoes ranging from 4,334 ft to 5,225 ft. This central part is surrounded by a lower-altitude terrain ranging from 1,300ft to 1,800 ft. Finally, the caldera rim mountains range from 2,600 ft to 4,100 ft. By observing the published charts in Fig. 8, it is quite clear that there are not that many options to reach the runway from the IAF: either the approach route needs to surround the caldera by following the northernmost part of the rim (Fig. 8(a)) or, instead, the approach route could also enter the caldera by staying above its lower-altitude region and avoiding its central part. After that, the approach route could reach the airport by flying along the small corridor to the west of the caldera, as shown in Fig. 8(b).

Although every time the RRT* algorithm was run we ob-



(a) RRT* raw output (b) Trajectory after the pruning process (c) Trajectory smoothed with Dubins paths

Fig. 10. Route generation process; north case study in Kumamoto Airport (RJFT)



(a) North case study (b) South case study

Fig. 11. Comparison of computed trajectories and published charts for Kumamoto Airport (RJFT)

tained different approach routes, the high amount of restrictive constraints imposed by the mountainous nature of the scenario led to the generation of very similar trajectories: either by surrounding the caldera or by reaching the airport through the small corridor to the west of the caldera.

For the north case study (Fig. 11(a)), starting from the runway, the approach route first diverts to the north in order to avoid the terrain to the east of the airport, thus, surrounding the caldera. The route proposed by the framework is located a bit closer to the mountains than the route published. Still, all the constraints regarding the obstacle clearance are met. The approach route computed by our framework is composed of three RF legs and four TF legs, and its total length is 30.7 NM; the length for the published route is 31.9 NM.

For the south case study (Fig. 11(b)), both the route generated by our framework and the published route enter the caldera through the small corridor between the mountains located to the east of the airport, with an orientation almost matching the runway direction. Then, both routes follow a north-east direction in order to avoid the high volcanoes in the center of the caldera. For the route generated by our framework, two RF legs are needed. The total length for the approach route obtained with our framework is 23.6 NM, while the length for the published route is 25.1 NM.

All experiments were conducted in a laptop with Ubuntu 22.04 LTS, with an Intel Core i7-1280P processor and 32 GB of memory. Computational times for the RJFR case study were very low, less than 5 seconds. However, for both RJFT case studies, the computational time was between 5 and 15 minutes, depending on each run. Due to examining neighboring vertices and rewiring the graph, an implementation of RRT* can take up to 8 times longer to complete a single path on average than the non-optimal version of the algorithm (i.e., RRT). Most of the computing effort comes from obstacle avoidance. Obstacle avoidance must be checked when a new vertex is placed, when a vertex is connected to the nearest vertex in the graph, and for each vertex that is to be rewired. In addition, the fact of having a DEM data with such a high granularity further increases the computational time. Due to the random nature of the algorithm, sometimes the new generated vertices allow to easily avoid the obstacles, which drastically reduces the computational time. However, sometimes many iterations are needed in order to successfully find a free-obstacle path, specially in a challenging scenario such as the one in RJFT.

VI. Conclusions

In this paper, we presented an automated framework capable of generating aircraft RNP AR APCH procedures with an RRT* algorithm. The free-obstacle paths obtained were later

refined with a path-pruning technique and smoothed with Dubins curves in order to take into account the turn radius constraints of the aircraft. The set of constraints defined by ICAO for RNP AR procedure design were also considered when computing the approach procedures.

Computational times were low for scenarios with no challenging terrain; however, for more mountainous regions, the computational time drastically increased, mainly due to the obstacle-avoidance-check-function of the RRT* algorithm. More advanced techniques could be used to accelerate the computations and optimize the RRT* search, such as the Gaussian mixture model presented in [13], which aimed at reducing the search space to reach convergence faster. In addition, more advanced implementations of the RRT* algorithm could be applied, such as a bi-directional RRT* (i.e., generating a path from both the origin and destination).

Several improvements could be implemented to our framework in order to obtain more operational approach procedures. Not only terrain elevation constraints could be taken into account, but also the presence of populated areas. In addition, it would be interesting to automatize the decision of using TF or RF legs, a process which is currently done manually. Furthermore, in future work, we are planning to explore the configuration parameters of the RRT* algorithm, such as the distance at which new vertices are generated; tuning these parameters could potentially lead to better results. In this paper, the aim was to present a proof of concept of our framework; in future work, we are planning to test our approach in other scenarios in order to properly validate the viability of the framework for RNP AR APCH procedure design.

It is worth highlighting the fact that having such an automated tool would be specially useful for challenging scenarios—such as the Kumamoto airport scenario presented in this paper—in which it is very difficult to define an approach procedure manually. In such cases, the procedure generated by our framework would be a valuable input for procedure designers, who could later refine it by using a flight-procedure design software. Indeed, in future work, we are planning to work closely with procedure designers with the aim of validating our framework by generating several procedures in such challenging scenarios. Finally, it is also worth noting the fact that the framework developed in this work could be easily extended to generate not only RNP AR APCH procedures, but also other kind of procedures involving other kinds of constraints.

REFERENCES

[1] ICAO, *Procedures for air navigation services: Aircraft Operations. Doc 8168, Sixth edition*. ICAO, 2018.

[2] ——, *Required Navigation Performance Authorization Required (RNP AR) Procedure Design Manual. Doc 9905, Third edition*. ICAO, 2021.

[3] D. Toratani and R. Mori, "RNP AR Approach Route Optimization Using a Genetic Algorithm," in *2022 Integrated Communication, Navigation and Surveillance Conference (ICNS)*. Dulles, VA, USA: IEEE, Apr 2022.

[4] S. M. LaValle, "Rapidly-exploring random trees : a new tool for path planning," *The annual research report*, 1998.

[5] S. Karaman and E. Frazzoli, "Sampling-based Algorithms for Optimal Motion Planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846—894, 2011.

[6] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast Marching Tree: a Fast Marching Sampling-Based Method for Optimal Motion Planning in Many Dimensions," *arXiv:1306.3532 [cs]*, Feb. 2015, arXiv: 1306.3532. [Online]. Available: http://arxiv.org/abs/1306.3532

[7] R. Sáez, H. Khaledian, X. Prats, A. Guitart, D. Delahaye, and E. Feron, "A Fast and Flexible Emergency Trajectory Generator: Enhancing Emergency Geometric Planning with Aircraft Dynamics," in *Proceedings of the 14th USA/Europe Air Traffic Management Research and Development Seminar*. Virtual: Eurocontrol and FAA, Sep 2021.

[8] NTT Data, "PANADES," http://www.airpalette.net/panades, [Accessed: May 18, 2023].

[9] A. Fallast and B. Messnarz, "Automated trajectory generation and airport selection for an emergency landing procedure of a CS23 aircraft," *CEAS Aeronautical Journal*, vol. 8, no. 3, pp. 481–492, Sep. 2017.

[10] P. Pharpatara, B. Hérissé, and Y. Bestaoui, "3-D Trajectory Planning of Aerial Vehicles Using RRT*," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 1116–1123, 2017.

[11] D. J. Webb and J. van den Berg, "Kinodynamic RRT*: Optimal Motion Planning for Systems with Linear Differential Constraints," 2012, arXiv:1205.5088[cs.RO].

[12] A. Khanal, "RRT and RRT* Using Vehicle Dynamics," 2022, arXiv:2206.10533[cs.RO].

[13] P. Sharma, A. Gupta, D. Ghosh, V. Honkote, G. Nandakumar, and D. Ghose, "PG-RRT: A Gaussian Mixture Model Driven, Kinematically Constrained Bi-directional RRT for Robot Path Planning," in *Procceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Prague, Czech Republic: IEEE, Sep 2021.

[14] S. Yang and Y. Lin, "Development of an Improved Rapidly Exploring Random Trees Algorithm for Static Obstacle Avoidance in Autonomous Vehicles," *Sensors*, vol. 21, no. 6, p. 2244, 2021.

[15] B. Shi, P. Cheng, and N. Cheng, "3D Flight Path Planning based on RRTs for RNP Requirements," in *Procceedings of the IEEE International Conference on Information and Automation*. Shenyang, China: IEEE, June 2012.

[16] H. Visser, "Generic and site-specific criteria in the optimization of noise abatement trajectories," *Transportation Research Part D: Transport and Environment*, vol. 10, no. 5, pp. 405–419, 2005.

[17] X. Prats, V. Puig, and J. Quevedo, "A multi-objective optimization strategy for designing aircraft noise abatement procedures: case study at Girona airport," *Transportation Research Part D: Transport and Environment*, vol. 16, no. 1, pp. 31–41, 2011.

[18] S. Hartjes and H. Visser, "Efficient trajectory parameterization for environmental optimization of departure flight paths using a genetic algorithm," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 231, no. 6, pp. 1115–1123, 2017.

[19] V. Ho-Huu, S. Hartjes, H. Visser, and R. Curran, "An Efficient Application of the MOEA/D Algorithm for Designing Noise Abatement Departure Trajectories," *Aerospace*, vol. 4, no. 4, p. 54, 2017.

[20] S. Khardi and L. Abdallah, "Optimization approaches of aircraft flight path reducing noise: comparison of modeling methods," *Applied Acoustics*, vol. 73, no. 4, pp. 291–301, 2012.

[21] J. Chevalier, "Departure and arrival route optimisation approaching big airports," Ph.D. dissertation, Université Paul Sabatier, 2020.

[22] Federal Aviation Administration (FAA), *United States of America Aeronautical Information Publication (AIP)*, 2023.

[23] L. E. Dubins, "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497—516, 1957.

[24] G. Satyanarayana, D. C. Manyam, A. L. Von Moll, and Z. Fuchs, "Shortest Dubins Path to a circle," in *AIAA Scitech 2019 Forum*, San Diego, California, USA, 2019.

[25] Geospatial Information Authority of Japan, "Base Geospatial Information Download Service," https://fgd.gsi.go.jp/download/menu.php, [Accessed: Feb 15, 2020].