RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Doctorat de l'Université de Toulouse

délivré par l'ISAE-SUPAERO

Optimisation de trajectoire pour véhicules légers longue endurance dans des écoulements instationnaires en présence d'obstacles, de danger diffus et d'incertitude

Thèse présentée et soutenue, le 20 décembre 2024 par

Bastien SCHNITZLER

École doctorale
AA - Aéronautique, Astronautique

Spécialité
Mathématiques et Applications

Unité de recherche
ENAC-LAB - Laboratoire de Recherche ENAC

Thèse dirigée par
Jean-Marc MOSCHETTA et Antoine DROUIN

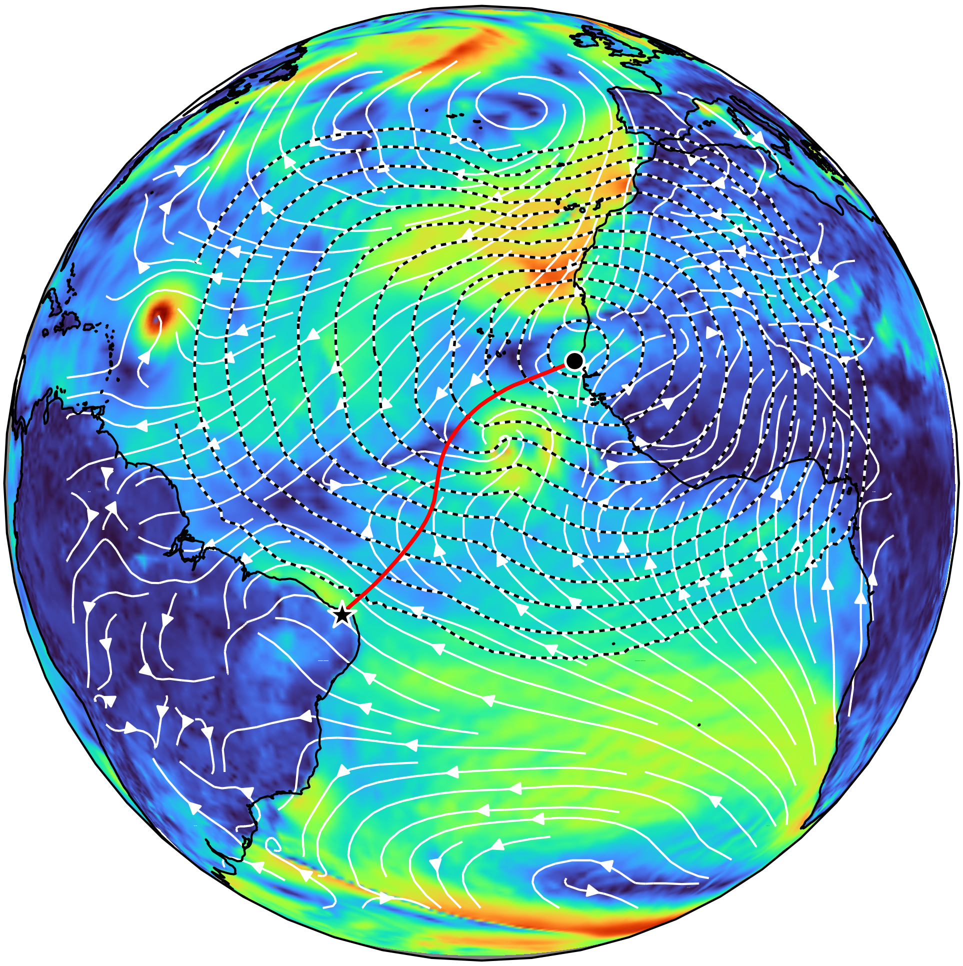
Composition du jury

M. Jean-Marie MIREBEAU, Rapporteur, Ecole Normale Supérieure Paris Saclay
M. Panagiotis TSIOTRAS, Rapporteur, Georgia Institute of Technology
M. Xavier PRATS, Examinateur, Universitat Politècnica de Catalunya (UPC)
M. Pierre LERMUSIAUX, Examinateur, Massachusetts Institute of Technology (MIT)
M. Jean-Marc MOSCHETTA, Directeur de thèse, Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-SUPAERO)
M. Antoine DROUIN, Co-directeur de thèse, Ecole Nationale de l'Aviation Civile (ENAC)

Membres invités

M. Daniel DELAHAYE, Ecole Nationale de l'Aviation Civile (ENAC)

Trajectory optimization for long-range light vehicles in unsteady flow fields with obstacles, diffuse hazard and uncertainty



Bastien Schnitzler

Dabry fit le point.
'Nous arriverons bientôt devant le Pot-au-Noir', alla-t-il crier à l'oreille de Mermoz.
Le Pot-au-Noir.
Le grand obstacle, l'éternelle et obscure barrière et dont tant de navigateurs à Dakar et à Rio avaient parlé à Mermoz. Les paquebots le contournaient à l'ordinaire. Mais, obligé de ménager sa provision d'essence et de couper au plus court, Mermoz devait passer à travers la zone redoutable. Comment franchirait-il ce domaine d'effroi et de sombres légendes ?

Mermoz, Joseph Kessel

Abstract

Continuous progress on developing ever better, safer and more autonomous cyber-physical systems has brought the need for efficient and optimal automatic decision making. Long-range mission drones especially, whether flying in atmospheric wind fields or diving in oceanic currents, are faced with the challenge to optimally plan the route they follow to fulfill their mission in a highly dynamic, unfavorable and uncertain environment, on space scales of hundreds or thousands of kilometers and on time windows spanning tenths of hours or several days.

In this thesis, solving such routing problems for long-range airborne or underwater vehicles is the main focus. The routing problems tackled consist in traveling optimally from a given point to a destination in a strong, unsteady and uncertain flow field, in the presence of diffuse hazard and strictly forbidden zones, with key metrics being travel time, spent energy or exposure to hazard. The considered environment geometries are either the planar 2D space or the Earth's 2D spherical space. The methods at stake are indirect methods, whether using extremals from Pontryagin's Maximum Principle or solving numerically a relevant Hamilton-Jacobi-Bellman equation.

First, the properties of an extremal-based algorithm to compute time-optimal trajectories in an unsteady and possibly strong flow field are studied. In given applications cases, similar existing algorithms from the literature are shown to reach their limit. Improvements are proposed for the latter and demonstrated to leverage the encountered caveats.

An extension of extremal-based algorithm is then proposed to handle hard obstacles, whether still or moving. The modified algorithm proves capable to compute time-optimal trajectories with obstacles but loses the ability to compute the optimal cost of the problem everywhere in space.

The navigation problem is then extended by adding the speed of the vehicle as a time variable and the total energy expense as an optimization metric. In this framework, the difference between energy-time-optimal trajectories and time-optimal trajectories is studied. On realistic examples, it is shown that an order of magnitude of 10% reduction in energy expense is possible when allowing the vehicle speed to adjust dynamically during the travel.

Hazard is added in the navigation problem as a dynamical and diffuse quantity. A Hamilton-Jacobi-Bellman partial differential equation is solved to get reachability sets for the vehicle in a hazard-physical space, from which hazard-time-optimal trajectories are computed. On realistic settings, it is shown that hazard-time-optimal trajectories are able to avoid a signif-

icant amount of hazard in the environment while increasing moderately the total travel time, thus proving the relevance of hazard-time-optimal planning in operational contexts.

Finally, uncertainty is tackled in the planning problem. A most important source of uncertainty comes from the flow field prediction. Weather ensemble predictions provide a collection of possible weather scenarios that help quantify uncertainty in the flow field data. On an airborne problem, the approach consisting in computing time-optimal trajectories in each scenario and simulating the variation in travel time incurred by following the trajectory in different scenarios is evaluated. The average travel time is overall constant over the possible paths, but there exist paths minimizing the dispersion in the travel duration. Next, a PMP formulation on ground paths rather than trajectories is proposed. It enables the writing of differential equations satisfied by extremals candidate to average travel time optimality. These average-time-optimal trajectories are shown to solve for the minimal average travel time in an example, however not with a significant reduction compared to classical time-optimal trajectories in the considered case.

Résumé de la thèse

Les systèmes cyber-physiques modernes tendent vers toujours plus de capacités, de sûreté de fonctionnement et d'autonomie. Dans ce contexte, la prise de décision automatique devient un enjeu central pour de tels systèmes. C'est d'autant plus vrai pour les drones aériens ou sous-marins de longue endurance, pour lesquels le routage dans un environnement dynamique, défavorable et incertain constitue un défi, avec des distances à franchir de plusieurs centaines à quelques milliers de kilomètres et des temps de trajet de plusieurs heures à plusieurs jours.

Dans cette thèse, on s'intéresse à de tels problèmes de routage optimal pour des véhicules à longue endurance, aériens ou sous-marins. Les problèmes considérés consistent à atteindre une destination dans un champ de flot instationnaire, fort et incertain, avec un danger diffus dans l'environnement et des obstacles infranchissables, et une métrique de performance pouvant être le temps de trajet, l'énergie dépensée ou l'exposition au danger. L'environnement considéré sera toujours soit plan soit sphérique. Les méthodes utilisées sont des méthodes indirectes, soit le calcul d'extrémales tirées du Principe du Maximum de Pontryagin, soit la résolution d'une équation aux dérivées partielles du type Hamilton-Jacobi-Bellman.

Dans un premier temps, un algorithme basé sur les extrémales pour le calcul des trajectoires temps-optimales dans un champ de flot instationnaire et possiblement fort est décrit puis analysé. Des exemples permettent de montrer les limites des algorithmes de la littérature basés sur des extrémales, et que l'algorithme proposé dans ce travail permet de dépasser ces dernières.

Une version modifiée de l'algorithme est proposée pour calculer les trajectoires temps-optimales en présence d'obstacles fixes ou mobiles. En revanche, cette version n'est plus capable de calculer la fonction valeur du problème partout dans l'espace.

Par la suite, la vitesse du véhicule est autorisée à varier durant le trajet et l'énergie totale dépensée est ajoutée en tant que métrique de performance. Dans ce cadre, la différence entre les trajectoires énergie-temps optimales et les trajectoires temps-optimales est étudiée. Des exemples réalistes montrent qu'un gain de l'ordre de 10% peut être espéré pour un véhicule avec vitesse variable comparé à une vitesse fixe.

Le problème de navigation est alors enrichi par l'ajout d'un champ de danger diffus et instationnaire dans l'environnement. Une équation aux dérivées partielles de type Hamilton-Jacobi-Bellman est résolue pour calculer l'ensemble atteignable pour le véhicule, duquel peuvent être déduites les trajectoires danger-temps optimales. Des exemples réels montrent que les trajec-

toires danger-temps optimales sont capables d'éviter une quantité de danger significative tout en augmentant modérément le temps de trajet, validant ainsi la pertinence d'utiliser l'approche dans un contexte opérationnel.

Enfin, l'incertitude est ajoutée au problème. L'accent est mis sur l'incertitude inhérente à la prédiction du champ de flot en considérant ce dernier donné comme un ensemble de prédictions météorologiques. Sur un cas aérien, des trajectoires temps-optimales sont calculées par scénario puis la variation de temps de trajet est évaluée pour chacune d'elles dans chaque autre scénario. La moyenne du temps de trajet varie peu en fonction du chemin choisi mais en revanche la variance du temps de trajet est moins forte pour certains chemins. Pour caractériser l'optimalité d'une trajectoire relativement au temps de trajet moyen tous scénarios, une application du principe du maximum de Pontryagin est proposée, basée sur chemin sol plutôt que sur trajectoire. Sur les exemples considérés, la méthode permet en effet de trouver les trajectoires optimales en moyenne de temps de trajet, mais sans écart significatif avec les trajectoires temps-optimales calculées par scénario.

Published content

A list of the published work related to the present thesis is given here.

Journal paper

- Schnitzler, B., Drouin, A., Moschetta, J.-M., and Delahaye, D. (2023a). General extremal field method for time-optimal trajectory planning in flow fields. *IEEE Control Systems Letters*

Conference presentations

- Schnitzler, B., Drouin, A., Moschetta, J.-M., and Delahaye, D. (2023b). General extremal field method for time-optimal trajectory planning in flow fields. In *IEEE Control and Decision Conference*, Singapore
- Schnitzler, B., Haley, Jr., P. J., Mirabito, C., Mule, E. M., Moschetta, J.-M., Delahaye, D., Drouin, A., and Lermusiaux, P. F. J. (2024). Hazard-Time Optimal Path Planning for Collaborative Air and Sea Drones. In *OCEANS 2024 IEEE/MTS Halifax*, Halifax. IEEE

Contents

Acronyms

AUV Autonomous Underwater Vehicle

BVP Boundary Value Problem

CAVI Closure-Aware Validity Index

CDS the Copernicus climate Data Store

ECMWF the European Center for Medium-range Weather Forecasts

ENP Energy-time-optimal navigation problem

FMM Fast Marching Method

FMT* Fast Marching Tree

GMTDS Global Maritime Traffic Density Service

HJ Hamilton-Jacobi

HJB Hamilton-Jacobi-Bellman

HNP Hazard-time-optimal Navigation Problem

ITCZ Inter-Tropical Convergence Zone

IVP Initial Value Problem

LSPOV Large-Scale Point Of View

MDP Markov Decision Process

MINLP Mixed-Integer Non-Linear Programming

NCO Necessary Conditions of Optimality

NLP Non-Linear Programming

ODE Ordinary Differential Equation

OVI Overall Validity Index

OUM Ordered Upwind Method

PDE Partial Derivatives Equation

PMP Pontryagin's Maximum Principle

PMP-G Pontryagin's Maximum Principle in non-degenerate Gamkrelidze's form

PRM Probabilistic Road Map

RL Reinforcement Learning

RRT Rapidly-exploring Random Tree

SDE Stochastic Differential Equation

SRF Speed Relative to the Flow

TNP Time-optimal Navigation Problem

UAV Uninhabitated Air Vehicle

VRF Velocity Relative to the Flow

ZNP Zermelo Navigation Problem

Mathematical notations

Throughout the whole thesis, we use the following mathematical conventions

- Symbols for vectors appear in bold, *i.e.* $\mathbf{x}$ is a vector while x is a real number.
- $\mathbf{x}^\intercal$ denotes the transpose of vector $\mathbf{x}$.
- By default, vectors are column vectors, noted either $\begin{pmatrix} x \\ y \end{pmatrix}$ or $(x\ y)^\intercal$ depending if they appear inside paragraphs or in equations.
- The vector scalar product is thus implicitly written as the usual matrix product, that is, $\mathbf{x}^\intercal\mathbf{y}$ is the scalar product of $\mathbf{x}$ and $\mathbf{y}$.
- The norm of vectors is noted $\|\mathbf{x}\|$. If not specified differently, it refers to the Euclidean norm $\|(x_1\ x_2\ \dots\ x_n)^\intercal\| = \sqrt{\sum_{i=1}^n x_i^2}$.
- If $f : x \mapsto f(x)$ is a function, we sometimes refer to the function using the notation $f(\cdot)$ to avoid confusing with particular images of the function.
- When $f : \mathbf{x} := (x_1\ x_2\ \dots\ x_n)^\intercal \mapsto f(\mathbf{x})$ is a differentiable multivariate function, its first derivative is rigorously the $1 \times n$ row vector $\frac{\mathrm{d}f}{\mathrm{d}\mathbf{x}} = (\frac{\partial f}{\partial x_1}\ \frac{\partial f}{\partial x_2}\ \dots\ \frac{\partial f}{\partial x_n})$.
- When $\mathbf{f} : \mathbf{x} := (x_1\ x_2\ \dots\ x_n)^\intercal \mapsto \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x})\ f_2(\mathbf{x})\ \dots\ f_m(\mathbf{x}))^\intercal$ is a differentiable multivariate vector function, its first derivative is the $m \times n$ Jacobian matrix

$$\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{x}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

- Total or partial derivative evaluations are noted using the bar notation. For instance, $\left.\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{x}}\right|_{x_1^\star,\dots,x_n^\star}$ is the evaluation of the Jacobian of $\mathbf{f}$ at point $(x_1^\star\ \dots\ x_n^\star)^\intercal$.

- For derivatives w.r.t. the time variable, the dot notation is often used, *e.g.* $\dot{\mathbf{x}}(t) = \left.\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\right|_t$.

We use the following list of symbols

List of symbols

$B\left(\mathbf{x}, r\right)$	Ball of center $\mathbf{x}$ and radius r
$\mathbb{R}$	Set of real numbers
$R\left(\theta\right)$	Rotation of angle θ
$\mathbb{S}^1$	Unit circle

Introduction

Context

1930. French aviator Jean Mermoz crosses the Atlantic between Saint-Louis, Senegal and Natal, Brazil and opens for the first time an air mail route for Aéropostale reaching South America. The crossing is 3000 km long and took 21 hours to complete. The crew struggled in the dark, the heat, the heavy rain showers and the gusts when flying in the doldrums, this zone around the equator where the wind circulation stops and large storms form.

1930. Mathematician Ernst Zermelo publishes a resolution for the problem of the optimal steering of an air vehicle in the wind (Zermelo, 1930), probably inspired by the first round-the-world airship navigation by the Graf Zeppelin in 1929.

The first story is that of a hard and risky challenge that was met. At ISAE-SUPAERO, it inspired the Mermoz challenge (Gavrilovic et al., 2023, 2019) consisting in the development of an Uninhabitated Air Vehicle (UAV) to cross the Atlantic on the same route as Jean Mermoz (Fig. 1) but with a significant reduction in greenhouse gas emissions. The UAV will have to cross 3000 km with a cruise speed of around 80 km/h, thus flying for about 36 h over the ocean. In such conditions, the effect of the wind can be significant on the trajectory: first, the great circle route to destination is not guaranteed to be the fastest; furthermore, in case of very strong wind, the reachable space for the UAV can be very limited. Additionally, over such a time scale, the wind field can evolve significantly, and this evolution can be difficult to predict, especially at the end of the time window, where weather forecast models show the most dispersion. As

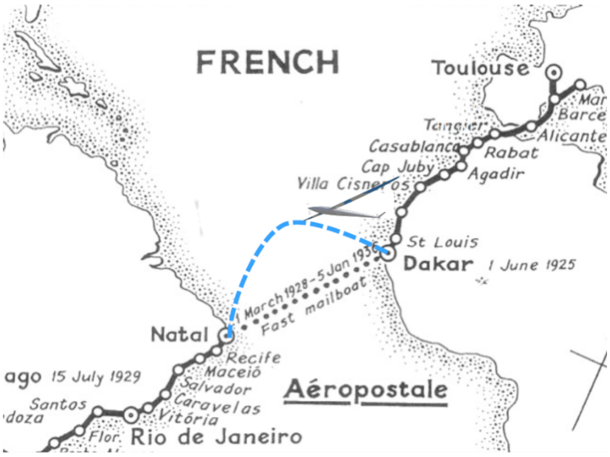


Figure 1: Map of French air mail company *Aéropostale*. The network spans to South America using a route over the Atlantic that was opened by Jean Mermoz on May 12th, 1930. From `websites.isae-supaero.fr/drone-for-earth`.

Jean Mermoz a century before, the UAV will also face zones with difficult flight conditions, especially in the Inter-Tropical Convergence Zone (ITCZ), where very active storms form.

The second story is the beginning of a century of work around the optimal routing of vehicles immersed in a flow field, both theoretically and numerically. Work which is now a fundamental basis for the resolution of complex trajectory planning problems, such as the one described in the previous paragraph, with all its different features.

The work presented in this thesis tries to connect the two stories. We will use the tools of optimal control to address trajectory planning problems: time-optimal navigation, obstacle avoidance, energy optimization, hazard avoidance or navigation in uncertain environments. To address this wide variety of problems, we first need to set the appropriate level of abstraction to model the situation, and review the sources of data available for trajectory planning.

The Mermoz drone

Before moving on to the trajectory planning problem itself, let us give some details about the Mermoz drone, the backbone application underlying the mathematical development to come.

The Mermoz drone development started at ISAE-SUPAERO in 2019 to fulfill the challenge

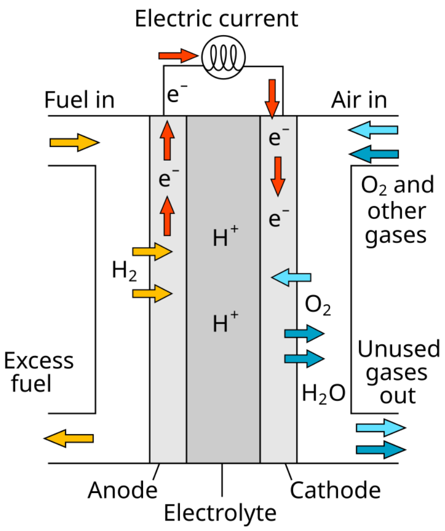


Figure 2: Fuel cell principle

of crossing the Atlantic between Dakar, Senegal and Natal, Brazil with a significant reduction in greenhouse gas emissions. While electric propulsion quickly appeared as a solution, electric power storage with lithium-based batteries did not provide a sufficient energy to mass ratio to cross the 3000 km. Thus, dihydrogen was selected as energy source, with a fuel cell to produce electric current from the combustion of dihydrogen with dioxygen. The fuel cell principle is sketched in Fig. 2 and recalled here. Dihydrogen H_2 flows continuously in one circuit and supplies protons H^+ to the electrolyte at the anode. The cathode is in contact with the oxydant (dioxygen O_2) which is reduced by the protons to form water H_2O, producing heat as byproduct. The stream of protons feeds a reverse stream of electrons, which creates an electric current.

The top-level requirements for the design of the Mermoz drone are the following ones:

1. able to cross the distance of 3000 km;
2. total mass of less than 25 kg;
3. hydrogen fuel cells as primary power source.

The design chosen for the Mermoz drone is a classical configuration of a fixed-wing, 4 m

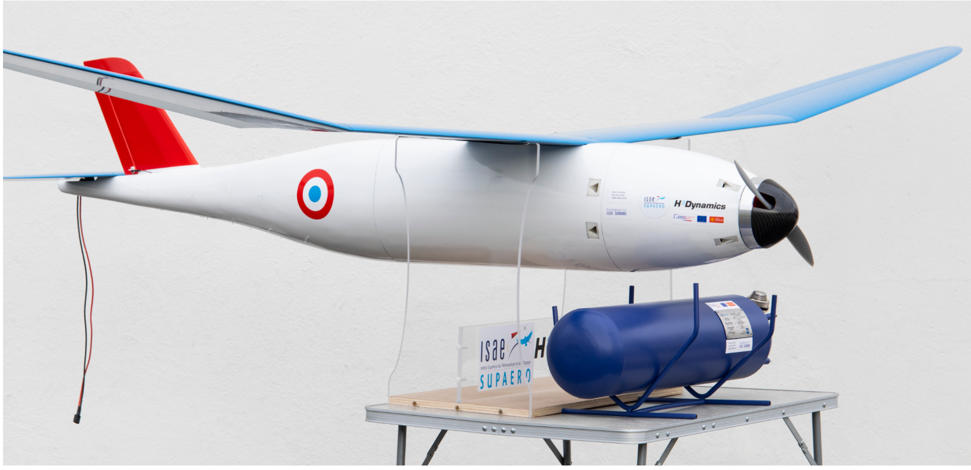


Figure 3: The Mermoz drone in the version integrating a 7L liquid dihydrogen tanK.

wingspan drone with a front propeller and vertical and horizontal stabilizers at the rear (Fig. 3). The fixed-wing configuration offers much more range capacity than the multi-rotor configuration, which fits to the mission. However, take-off and landing require much more care and constraints for fixed-wing configuration than for multi-rotors. The selected high aspect ratio wing (high length, low width) allows high aerodynamic efficiency for reduced drag in cruise. The fuselage shape and width is set to integrate a fuel cell and hydrogen storage. In a first version, that already flew at the time this thesis is written, hydrogen was stored in gaseous form in a 300 bar tank. But the energy density of a hydrogen-based power supply can be further improved by resorting to liquid dihydrogen, which can store more energy in the same volume than compressed dihydrogen in commonly used 300 bar or 700 bar tanks. Thus, a second version of the Mermoz drone is being developed to integrate a liquid dihydrogen tank. The challenge for liquid dihydrogen is to store it below its boiling point, *i.e.* at a temperature below 20.39 K. This requires a very efficient isolating tank to store the dihydrogen. Furthermore, as no tank is perfect, heat continuously flows into the tank, heating up liquid dihydrogen that turns into gas and accumulates in the tank. The tank is regulated to keep the pressure inside under a given threshold. If this threshold is reached, the tank jettisons dihydrogen into the atmosphere. The outgoing flow of gaseous dihydrogen that balances the constant gas supply from the tank

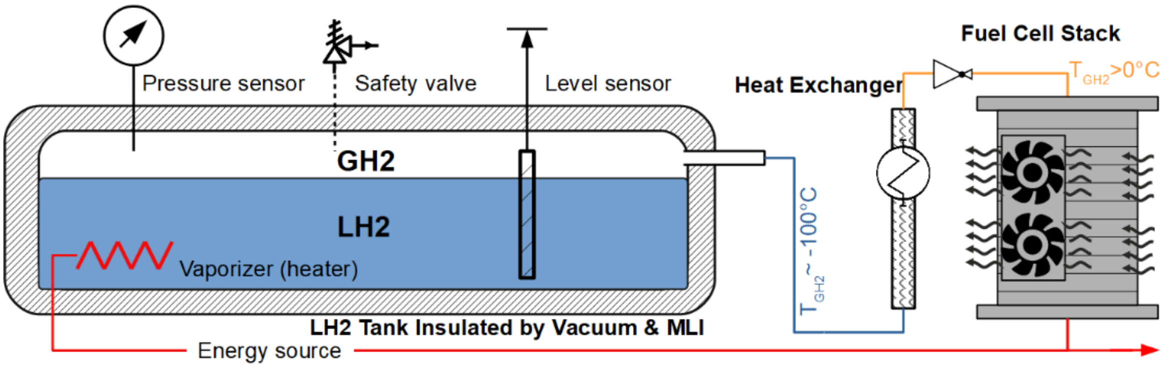


Figure 4: Diagram of a liquid dihydrogen power plant, producing electricity from the combustion of dihydrogen and dioxygen in a fuel cell. From Gavrilovic et al. (2023)

is called the *boil-off rate*. For a perfect operation of the tank, one would size its propulsive system to operate precisely at this boil-off rate, neither losing dihydrogen nor drawing too much on the tank (thus requiring a heating system to increase the boil-off). The general liquid H_2 power system is depicted in Fig. 4. More details about the sizing of the hydrogen power plant for the Mermoz drone can be found in Gavrilovic et al. (2023).

Models

In this section, the appropriate abstraction model for long-range path planning is chosen, and the different features of the navigation problem are discussed.

Vehicles

Advected or not? Different classes of vehicles are subject to the problem of long-range route planning in free space and dynamic environments. Aircraft, UAVs, ships, submarines, Autonomous Underwater Vehicles (AUVs) are examples of such vehicles (examples of autonomous marine vehicles are depicted in Fig. 5). There is an important difference between vehicles that are subject to *advection* by a flow field present in the environment and the ones that are not. Airborne and underwater vehicles are examples of vehicles that feel the push of

(a) (b)

Figure 5: (a) An example of an autonomous sailing vessel (Source: NOAA and Saildrone Inc.). (b) An example of an AUV (Source: Bluefin Robotics).

the wind or the ocean currents acting on their trajectories. But for ships, the advection by surface currents is generally negligible compared to the ship's speed, whether the ship is powered by an engine or is a sailship. In this thesis, we focus on the class of vehicles that are subject to advection.

Large Scale Point Of View In the literature of long-range route planning, the most common way to model the vehicle is to use a Large-Scale Point Of View (LSPOV) model of the situation. This means that the vehicle is assimilated to a single point with position vector $\mathbf{x}$. The trajectory of the vehicle is the function of time noted $\mathbf{x}(t)$. From the LSPOV, the forces acting on the vehicle are balanced and the vehicle has no acceleration. It thus evolves through the following kinematic model

$$\mathbf{v}_g = \mathbf{u} + \mathbf{v}_f$$

where

1. $\mathbf{v}_g := \frac{\mathrm{dx}}{\mathrm{d}t}$ is the *ground velocity* of the vehicle.

2. $\mathbf{u}$ is the vehicle's Velocity Relative to the Flow (VRF). Its norm $\|\mathbf{u}\|$ is called the Speed

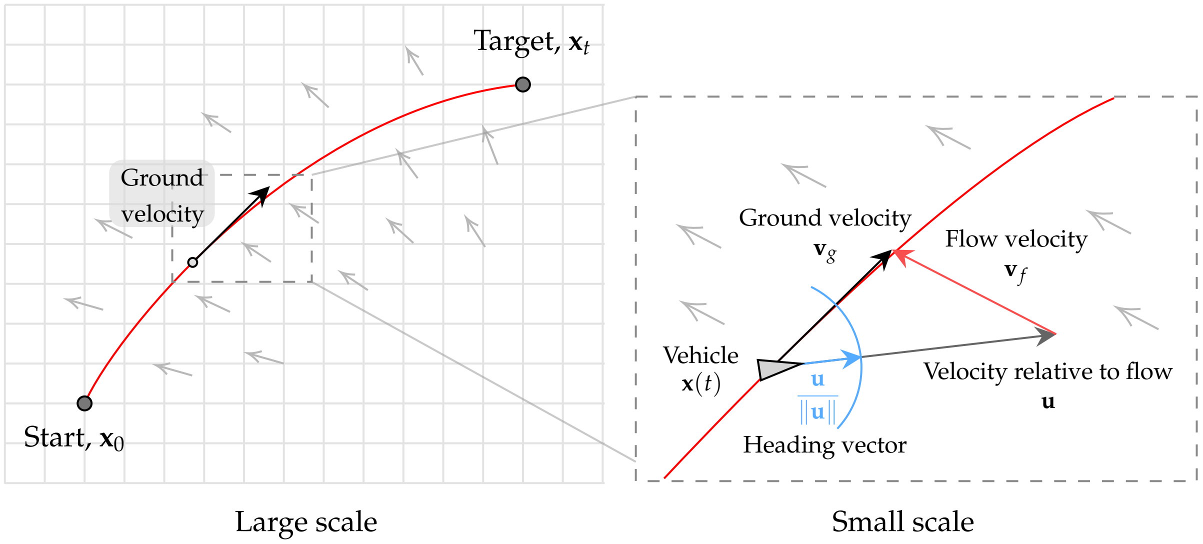


Figure 6: Left, the navigation problem: steering a vehicle from a start point to a destination in a flow field. Right, the velocity composition leading to the ground velocity of the vehicle.

Relative to the Flow (SRF). In the case of an airborne vehicle, it is the *airspeed*. The normalized VRF $\frac{\mathbf{u}}{\|\mathbf{u}\|}$ is called the *heading vector* and is noted $\mathbf{h}$.

3. $\mathbf{v}_f$ is the flow field vector. In the airborne case, it is the wind vector, and in the underwater case it is the ocean current.

The vector summation is detailed in Fig. 6.

Dimension of the state space

For long-range routing problems, there are different ways to model the free space depending what the routing is made for. The most common models are:

1. the 2D planar space, where $\mathbf{x} = (x\ y)^\intercal$;
2. the 2D spherical space, where $\mathbf{x} = (\varphi\ \lambda)^\intercal$ with φ the longitude and λ the latitude;
3. the 3D space, where $\mathbf{x} = (x\ y\ z)^\intercal$.

Of course, the 3D space is the most comprehensive one because it makes no particular assumptions to reduce the state space dimension, but many long-range navigation problems do not need the third dimension. It is the case for aircraft flying at a (almost) fixed altitude, given either by the optimum conditions of flight or by constraints (near the sea level for the Mermoz drone). It is also the case for autonomous underwater gliders that alternate upward and downward moving phases to produce thrust, and thus from the LSPOV it is as if they were at the average depth. Now, if the third dimension is useless, one has to choose between 2D planar space or 2D spherical space. The 2D planar space comes with the simplicity of equations: most results for optimal routing (formulas or open-source codes) are given in cartesian coordinates and can consequently be applied directly in this space. But the 2D spherical is the most accurate when moving on Earth, because it captures the sphericity. It appears that, for routing missions of up to three of four thousands of kilometers, projections of the Earth do not entail much distortion of the routing problem, so that the 2D planar model can be chosen. But for larger problems, the 2D spherical model may be needed. For instance, for the 3000 km length scale of the Dakar to Natal navigation problem, a Lambert projection can keep the relative error on distances under 1.6 %, so that trajectories optimized on the projected space are expected to be very close to what would be obtained by optimizing in the spherical space.

It must be noted that the proposed parameterization of the spherical space in longitude/latitude is far from being the only possible one. While being canonical (a lot of applications use longitude/latitude to locate points on Earth), this coordinate system suffers from degeneration at the poles, which in practice can lead to bad numerical conditioning, or other errors. Other two-variable coordinate systems can shift the problem away from poles if needed, but there will always be a singularity somewhere in the representation.

Velocity Relative to the Flow

Depending on the dimension of the problem, the VRF $\mathbf{u}$ will be of dimension 2 or 3. If the problem is spherical, this velocity vector will belong to the tangent plane to Earth at the given position (and still be of dimension 2).

In dimension 2, taking the LSPOV has strong consequences on the sets of values $U \subset \mathbb{R}^2$

that the VRF can take. Let's focus on the small-scale vehicle behavior and note U_s the set of admissible VRF for the small scale. This VRF results from an equilibrium of forces between actuators and drag forces. Because the vehicle has limited onboard power, this VRF is *bounded* $\|\mathbf{u}\| \leq u_{\max}$. Furthermore, the set of admissible VRF U_s has to be *circularly symmetric*: when an SRF $u_1 = \|\mathbf{u}\|$ is reachable, the vehicle can use it with any heading to move around.

So the set U_s is either a disk or a collection of annulus contained in $B\left(0, u_{\max}\right)$. Vehicle that can idle ($\mathbf{u} = 0$) like AUVs indeed have $U_s = B\left(0, u_{\max}\right)$. But for vehicle relying on a lift force to sustain motion like aircraft, pure idling is impossible and U_s is indeed an annulus $U_s = \left\{\mathbf{u} \in \mathbb{R}^2 \,\middle|\, u_{\min} \leq \|\mathbf{u}\| \leq u_{\max}\right\}$.

But when using the LSPOV, the set of admissible controls U is really the convex hull of U_s. Indeed, if $\mathbf{u}_1$ and $\mathbf{u}_2$ are admissible controls in U_s, then consider actuating the vehicle at small-scale with the high frequency control

$$\mathbf{u}(t) = \begin{cases} \mathbf{u}_1 \text{ if } t \bmod T \in [0, \alpha T] \\ \mathbf{u}_2 \text{ if } t \bmod T \in [\alpha T, T] \end{cases}$$

with the period T being of the small-scale time magnitude (for instance minutes for an aircraft). Then from the LSPOV, the vehicle will appear to have VRF

$$\mathbf{u} = \alpha \mathbf{u}_1 + (1 - \alpha)\mathbf{u}_2$$

So any convex combination of the small-scale admissible VRF leads to an admissible VRF for the large scale. Thus, U is a convex, circularly-symmetric set, so it is a disk. To sum up, in dimension 2 and using the LSPOV on the routing problem, it makes sense for any vehicle to consider that the admissible set of VRFs is a disk

$$U = B\left(0, u_{\max}\right)$$

In particular, from the LSPOV, an aircraft can idle. This is easy to implement in practice by making the aircraft loop around a point.

Flow field

In the routing problem, the flow field (wind or oceanic current) is a time-space vector field $\mathbf{v}_f(t, \mathbf{x})$ of the same dimension as the state space (tangent to Earth in the spherical case). In some cases, it will be also useful to consider *steady* flow fields, *i.e.* flow fields that do not depend on time $\mathbf{v}_f(\mathbf{x})$.

Many methods for optimal routing require evaluations of the flow field at arbitrary times t and positions $\mathbf{x}$, which motivates the functional model for the flow field $t, \mathbf{x} \mapsto \mathbf{v}_f(t, \mathbf{x})$. But in practice, we don't have access to the flow field value at arbitrary times and positions. The flow field data always comes from a *weather model* or an *ocean model* (Fig. 7). These models use the laws of physics to compute on a 3D grid $(\mathbf{x}_{i,j,k})$ and for different times $t_0, t_1, \dots, t_n$, the state of the atmosphere (resp. ocean) $\mathbf{\Psi}^{(0)}_{i,j,k}, \mathbf{\Psi}^{(1)}_{i,j,k}, \dots, \mathbf{\Psi}^{(n)}_{i,j,k}$, where $\mathbf{\Psi}$ is a vector containing all the variables describing the state of the atmosphere (resp. ocean) at a given time and position. Usually, models can work in *forecast* mode, predicting future values for the state of atmosphere (or ocean), or in *reanalysis* mode, using past observed data to build a comprehensive gridded representation of the atmosphere (or ocean).

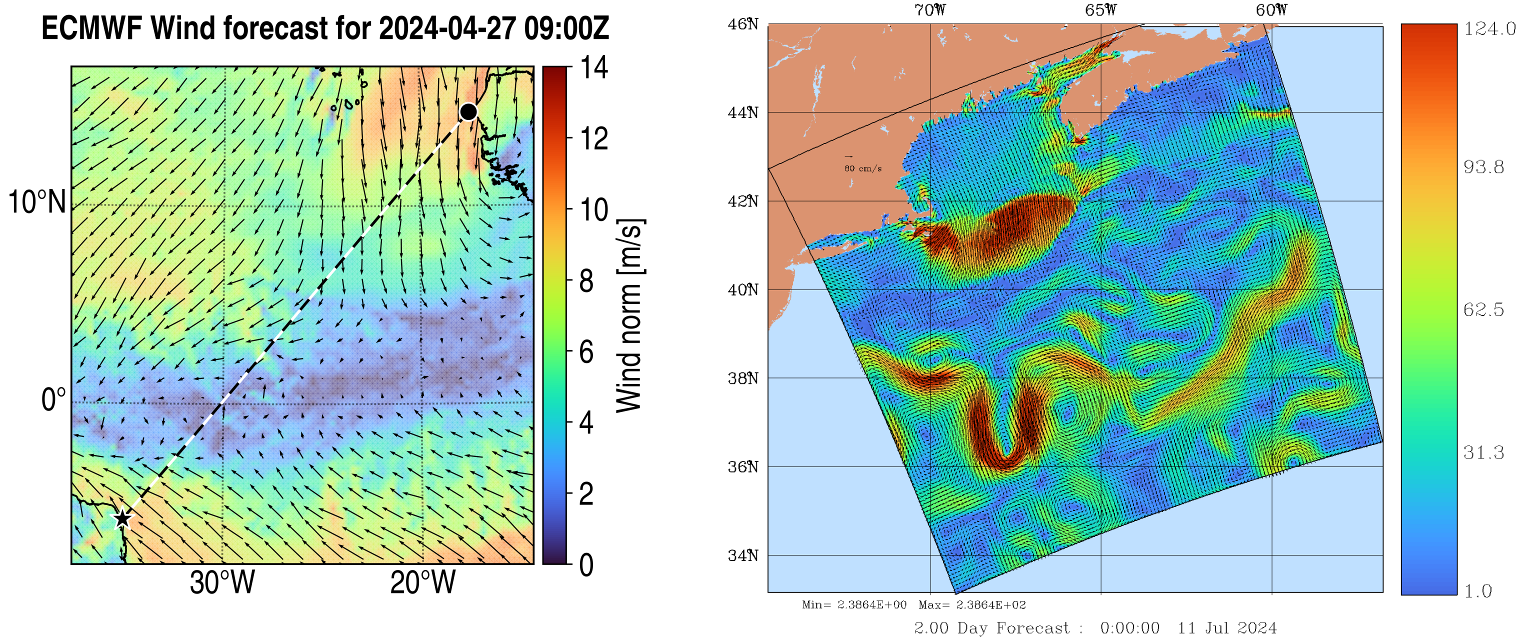
For global forecast of atmospheric data, models such as the Global Forecast System[1] (GFS), or the European Centre for Medium-Range Weather forecast model[2] (ECMWF) provide open-source access to weather forecasts. For reanalyses of atmosphere data, the Copernicus Data Store provides access to the open-source ECMWF Reanalysis in fifth version[3] (ERA5) that provides a wide variety of atmospheric parameters on multiple pressure levels from 1940 to present. For the ocean, the HYbrid Coordinate Ocean Model[4] (HYCOM) or the operational

[1] https://www.ncei.noaa.gov/products/weather-climate-models/global-forecast

[2] https://confluence.ecmwf.int/display/DAC/ECMWF+open+data%3A+real-time+forecasts+from+IFS+and+AIFS

[3] https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels

[4] https://www.hycom.org/data/glby0pt08/expt-93pt0


ECMWF Wind forecast for 2024-04-27 09:00Z
10°N
0°
30°W
20°W
14
12
10
8
6
4
2
0
Wind norm [m/s]
46N
44N
42N
40N
38N
36N
34N
70W
65W
60W
80 cm/s
124.0
93.8
62.5
31.3
1.0
Min= 2.3864E+00 Max= 2.3864E+02
2.00 Day Forecast : 0:00:00 11 Jul 2024

(a)

(b)

Figure 7: Examples of flow field data. (a) ECWMF 100 m wind forecast over the Atlantic ocean between Dakar, Senegal and Natal, Brazil. (b) 0-1000m averaged horizontal current velocity. Product from MSEAS, MIT.

Mercator Ocean model output[5] (Copernicus Marine Data Store) provide forecast data.

Weather models give a gridded prediction for the flow field $[\mathbf{v}_f]^{(s)}_{i,j,k}$. To get a continuous representation, one needs to resort to *interpolation*. Several methods are available, with different accuracy levels, as presented in Table 1.

Method	Pros	Cons
Nearest neighbor	Very fast, simple implementation	Discontinuous, not physically relevant between data points
Linear	Fast, continuous	Derivative discontinuous
Cubic splines	Smooth ($\mathcal{C}^1$ class)	More costly, not necessary comprehensively implemented in all code libraries[6]
Akima splines	Smooth ($\mathcal{C}^1$ class), reduces oscillations	More costly, not necessarily available in all libraries

Table 1: High-level comparison of interpolation methods.

The most widely used interpolation method is the linear one, because it achieves a trade-off between simplicity and consistency of the interpolation: it achieves continuity between grid points, which is what is expected for physical quantities, and it is available in a wide variety of code implementations.

Problems

Reachability

With all the previous elements defined, the first interesting problem is to compute the *reachable set* of the vehicle in some duration τ, from the start date t_0 and the start position $\mathbf{x}_0$, which we note $\mathcal{R}_{t_0,\mathbf{x}_0,\mathbf{v}_f}(\tau)$. It is the collection of all positions that can be reached departing from $(t_0, \mathbf{x}_0)$ in duration τ. A sketch is presented in Figure 8. The complexity of the reachable set is essentially

[5] `https://data.marine.copernicus.eu/product/GLOBAL_ANALYSISFORECAST_PHY_001_024/description`

[6] In Python, the package `scipy` implemented a cubic spline interpolator, but it is only recently, on April 3rd, 2024 with version 1.13 that the possibility to automatically compute derivatives of the interpolator was released. It is a key feature, intensively used in the algorithms presented in Chapters 3 and 4 for the derivatives of the flow field.

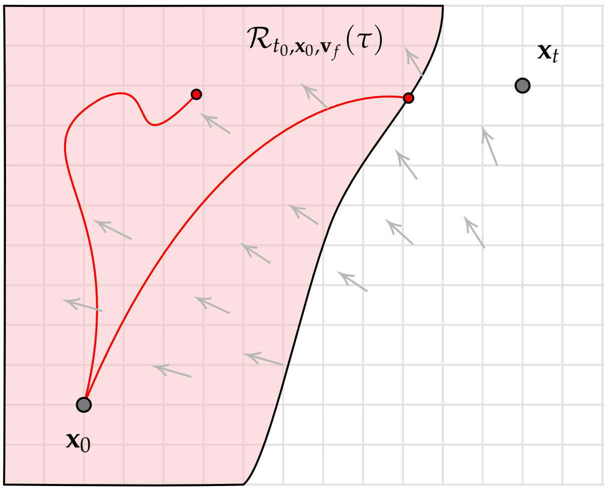


Figure 8: The reachable set from start point $\mathbf{x}_0$. Two example trajectories are depicted: one reaches the border of the reachable set while the second falls inside.

governed by the comparison of the vehicle's SRF to the flow field magnitude.

1. $v_r \gg \|\mathbf{v}_f\|$. When the SRF is way superior to the flow magnitude, the reachable set looks like a slightly deformed disk (in 3D, a sphere). It is the case, for instance, for commercial aircraft.

2. $v_r > \|\mathbf{v}_f\|$. When the SRF is comparable to the flow magnitude, the reachable set can be way deformed compared to the disk case. UAVs usually fall in this category.

3. $v_r < \|\mathbf{v}_f\|$. If the SRF is less than the flow magnitude, then the reachable set is not increasing anymore: positions that are reachable at some time may be out of reach at a later date, because the flow is pushing. AUVs are often confronted to this mode of operation.

4. $v_r \ll \|\mathbf{v}_f\|$. If the SRF is way smaller than the flow magnitude, then the trajectory is essentially the one of a free particle in the flow. In this case, control is not really useful.

When the SRF is always superior to the flow magnitude $v_r > \|\mathbf{v}_f\|$, the reachable set is *increasing*, and even better, all the state space is reachable, provided one has sufficient time. In this case, the control system (the vehicle) is said to be *controllable*: there exist a control (a

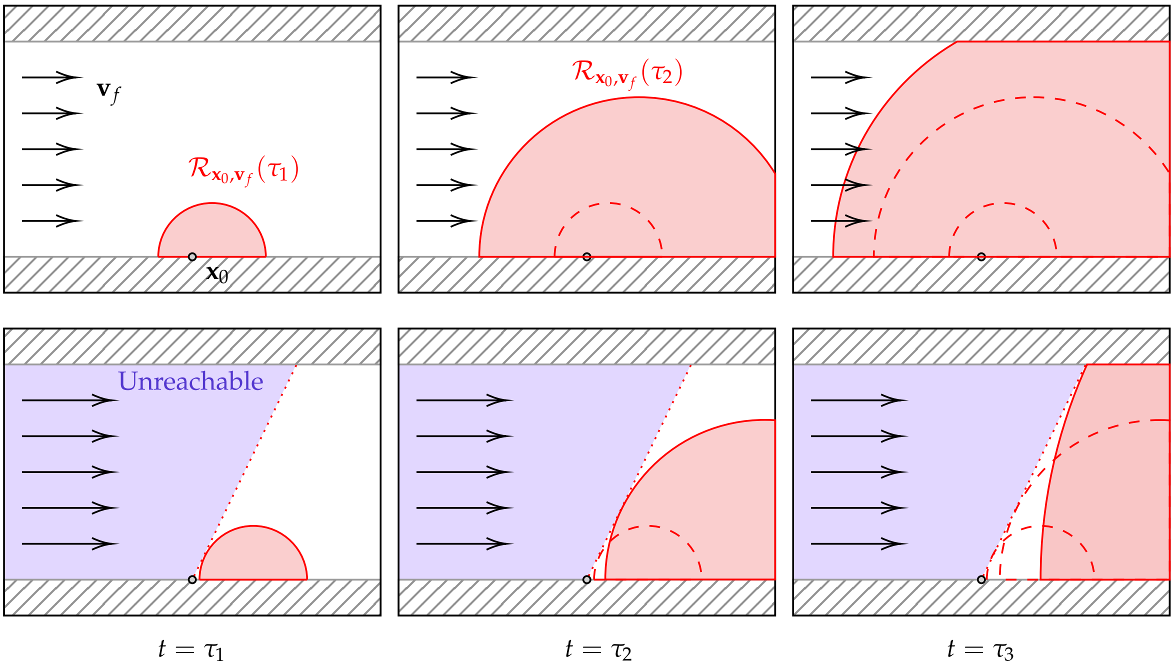

$\mathbf{v}_f$
$\mathcal{R}_{\mathbf{x}_0,\mathbf{v}_f}(\tau_1)$
$\mathbf{x}_0$
$\mathcal{R}_{\mathbf{x}_0,\mathbf{v}_f}(\tau_2)$
Unreachable
$t = \tau_1$
$t = \tau_2$
$t = \tau_3$

Figure 9: Example reachable sets in a stream flow. In the upper row, the flow field magnitude is inferior to the vehicle's SRF and the reachable set is increasing $\mathcal{R}_{\mathbf{x}_0,\mathbf{v}_f}(\tau_i) \subset \mathcal{R}_{\mathbf{x}_0,\mathbf{v}_f}(\tau_{i+1})$. In the lower row, the flow field magnitude is superior to the vehicle's SRF and the latter property does not hold.

sequence of heading vectors $\mathbf{h}(t)$) that steers the vehicle to any position in space. It is not necessarily the case when the flow is strong ($v_r < \|\mathbf{v}_f\|$). An example of this is given in Figure 9

Time optimality

When a location is reachable for the vehicle, an important question is to know how to steer the vehicle to this location in a minimal amount of time. This means finding a solution to the following *Time-optimal Navigation Problem* (TNP)

$$\text{(TNP)} \begin{cases} \min\limits_{\mathbf{u}(\cdot)} \ t_f \\ \dot{\mathbf{x}}(t) = \mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x}(t)) \\ \mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f \end{cases}$$

This problem has been entailing much research, starting with the work of mathematician

Ernst Zermelo in 1930 (Zermelo, 1930). It is known under the name of *Zermelo's Navigation Problem* (ZNP). A wide variety of optimization techniques exist to solve this problem and will be reviewed in Chapter 1. In Chapters 3 and 4, a specific technique will be detailed to solve this problem, first in its original formulation and then in the presence of obstacles.

Multi-objective optimality

Computing time-optimal trajectories is really the entry point in the trajectory optimization world. But of course, in routing problems, time is not the only relevant criteria.

Energy

Energy is an important matter in routing problems. Careful energy management is required for vehicles with low energy capacity but long missions. The energy expense depends on how the vehicle spends its power which, once the forces are balanced (assumption of the LSPOV) turns out to be a function of the VRF $\mathbf{u}$. Some vehicle can also gain energy, for example harnessing a time-space field such as the solar power. So in general the spent power $g(t, \mathbf{x}, \mathbf{u})$ is a function of time, space, and the VRF, and the total energy spent is its integral:

$$E_f = \int_{t=t_0}^{t_f} g(t, \mathbf{x}(t), \mathbf{u}(t)) \mathrm{d}t$$

The energy is a new objective that gets added to the travel time, so the routing problem becomes a multi-objective optimization problem that we call *Energy-time-optimal Navigation Problem* (ENP)

$$\text{(ENP)} \begin{cases} \min\limits_{\mathbf{u}(\cdot)} [t_f, E_f] \\ \dot{\mathbf{x}}(t) = \mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x}(t)) \\ \mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f \\ E_f = \int_{t=t_0}^{t_f} g(t, \mathbf{x}(t), \mathbf{u}(t)) \mathrm{d}t \end{cases}$$

In this case, an object of particular interest is the *Pareto front* of the energy-time graph, *i.e.*

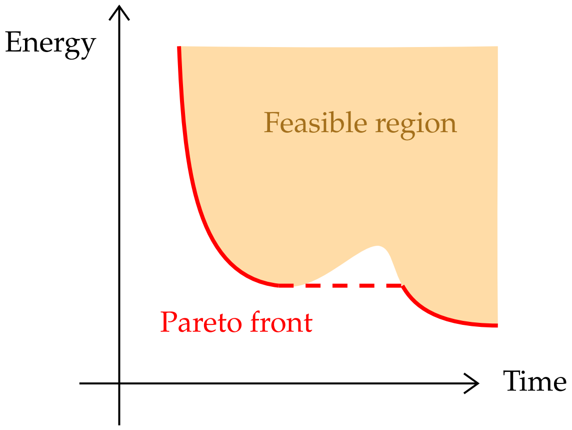


Figure 10: For a navigation problem, possible pairs of travel duration and energy expense, depicted as a yellow region. The solid red curve depicts the Pareto-optimal pairs.

the collection of pairs (t_f, E_f) such that no other pair exist with strictly less travel time and spent energy value at the same time.

Chapter 5 presents a discussion on assessing if the SRF has a significant impact on the energy expense, or not, for flying vehicles.

Hazard, collection

In routing problems, it may exist a time-space field $\varphi(t, \mathbf{x})$ (or several ones $\varphi_1(t, \mathbf{x}), \ldots, \varphi_n(t, \mathbf{x})$) that are beneficial or detrimental to the vehicle. For example, exposure to heavy rain and storms is detrimental to UAVs, in which case φ is called a *hazard* field. But if φ is the density of algae or fish for an AUV to collect, then it is beneficial and it is subsequently called a *collection* field. In any case, the total accumulated amount of exposure to the given field:

$$\Phi_f = \int_{t=t_0}^{t_f} \varphi(t, \mathbf{x}(t))\mathrm{d}t$$

is a quantity that one may want to minimize (hazard) or maximize (collection). Here, again, the routing problem becomes a multi-objective problem, with a trade-off between travel time and total accumulation of the field to balance: in the case of hazard, balance between high exposure to hazard but reduced travel time or longer travel but safer; in the case of collection, balance between short travel times and high collection.

In Chapter 6, we will focus on the *Hazard-time-optimal Navigation Problem* (HNP)

$$
\text{(HNP)}\begin{cases}
\min\limits_{\mathbf{u}(\cdot)} \; [t_f, \Phi_f] \\
\dot{\mathbf{x}}(t) = \mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x}(t)) \\
\mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f \\
\Phi_f = \displaystyle\int_{t=t_0}^{t_f} \varphi(t, \mathbf{x}(t))\mathrm{d}t
\end{cases}
$$

Obstacles

Reachability, time-optimality or multi-objective optimality problems can all be enriched with the addition of obstacles that the vehicle has to avoid. These obstacles can come from problem physical limits (*e.g.* coastlines for AUVs) but also from operational limits such as forbidden zones, very hazardous weather zones, other vehicles in space…

In general, an obstacle (or barrier) can be represented as a sublevel set $B(t) := \{\varphi(t, \mathbf{x}) \leq 0\}$ of a piecewise smooth function $\varphi \in \mathcal{C}^\infty_{\text{pwc}}$. The adaptation of a trajectory optimization technique to the presence of obstacles is discussed in Chapter 4, where the focus will be on the *Constrained Time-optimal Navigation Problem* (C-TNP)

$$
\text{(C-TNP)}\begin{cases}
\min\limits_{\mathbf{u}(\cdot)} \; t_f \\
\dot{\mathbf{x}}(t) = \mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x}(t)) \\
\mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f \\
\varphi_i(t, \mathbf{x}(t)) \leq 0, \quad i = 1, \ldots, n_{\text{obs}}
\end{cases} \tag{0.1}
$$

Uncertainty

In the routing problem, different sources of uncertainty can alter the planning process. There can be uncertainty in the vehicle's motion because of positioning errors, control imprecisions, wrong value of the flow field at the local scale compared to the global scale prediction…There can also be uncertainty in the flow field prediction, particularly in its evolution, as the atmosphere or the ocean are chaotic systems. In each case, taking into account that parameters are uncertain can be done theoretically, and the optimal planned trajectory in the presence of uncertainty can be adapted for better statistical or averaged performances. This will be the focus of Chapter 7.

Organization of the thesis

In this thesis, we will try to develop general theoretical tools and perform numerical applications to answer practical questions raised by the Mermoz drone crossing the Atlantic. In a first chapter (Chapter 1), the state of art on long-range optimal path planning in flow fields is reviewed. Then, a collection of chapters are proposed to provide answers to the following scientific questions:

1. What are the properties of the navigation problem in its long-range abstraction? (Chapter 2)

2. How to efficiently determine time-optimal trajectories for light vehicles on long-range missions when evolving in an unsteady and possibly strong flow field? Is it possible to have guarantees on the optimality? (Chapter 3)

3. How does optimal trajectories change when adding obstacles in the problem? How to adapt time-optimal computation methods in this case? (Chapter 4)

4. Can time-optimal methods be adapted to the case when the speed of the vehicle is variable and when the key metric is energy instead of time? What does the addition of this new variable change in the planning problem? Is there much to be gained by using energy-minimizing trajectories instead of time-minimizing ones? (Chapter 5)

5. How to handle diffuse hazard zones, that would not be correctly modelled by hard obstacles, into the trajectory planning procedure? What does it change in practice to follow

hazard-avoiding trajectories instead of time-optimal trajectories? (Chapter 6)

6. How can one decide best for a trajectory when the environment is uncertain? Can previous trajectory optimization methods be directly used in an uncertain context? (Chapter 7)

All these questions concur in making the strategic trajectory planning the most comprehensive possible. The general conclusions of this work will be drawn at the end of this dissertation in a dedicated chapter.

Code

A lot of numerical applications presented in this thesis are based on a companion module called[7] DABRY developed during the PhD by the author. A frozen version of the code is available at `https://doi.org/10.5281/zenodo.13939206`. It is the version used to generate the results presented in the present manuscript. The evolving Git repository for the code is at `github.com/dabry-route/dabry`.

Throughout the manuscript, some references to Python notebooks used to generate results are given in footnotes. They can be found in the Zenodo archive in the `thesis-notebooks` directory. The notebooks are provided 'as is' in an open-source effort, and may sometimes needs some manual adjustments to work correctly.

[7] Jean Dabry was the navigator onboard with Jean Mermoz for the crossing of the Atlantic on May 12th, 1930

Chapter 1

Literature review

This chapter is dedicated to reviewing the literature of work solving Zermelo's navigation problem. As explained in the introduction, references span from aerial vehicles in wind fields to submarines in ocean currents, sometimes also including ship routing. This review is focused on time-optimal navigation in a deterministic and possibly strong flow field, since it is the 'entry point' of the literature for vehicle routing in flow fields. To solve this problem, a wide variety of methods was proposed, as will be seen in this chapter.

References dealing with upgrades of Zermelo's problem such as obstacles, energy-optimal routing or routing in uncertain flow fields are kept for dedicated reviews in the corresponding chapters of this thesis.

Similar, well-documented literature reviews around trajectory optimization for vehicles in flow fields can be found in Lolla (2016), with an underwater flavor or in González Arribas (2019), with an aeronautics flavor.

Before reviewing contemporary work on trajectory optimization, we take a step back in history. Indeed, most tools that we manipulate in trajectory optimization have taken centuries to get mature, and it is useful to understand where the tools come from, to better understand how they work.

Trajectory optimization: Historical background

Before being the computer-based discipline that it is now, trajectory optimization went through centuries of evolution in mathematics and physics that gave it its own shape. In the end of the 17th century, Jean Bernoulli solved the canonical trajectory optimization problem called the *brachistochrone*, *i.e.* finding the optimal shape of a curve over which a freely moving heavy ball would reach its end the fastest possible. This work was contemporary with the development of the calculus of variations, which really set the first fundamental mathematical tools for optimization. In the wake of these developments, in the following centuries, physicists formalized the powerful stationary-action principle. This principle states that every physical system described by a finite amount of parameters tends to evolve so that the action - *i.e.* the integral of the difference between kinetic energy and potential energy - is minimized. Much of the terminology used in modern optimal control inherits from physics' terminology around this principle. Thus, to contextualize the optimal control notions that will be used throughout the whole thesis, we shall quickly review what the stationary action principle implies in physics.

If a physical system is described by a multivariate state vector $\mathbf{q} = (q_1, \ldots, q_n)$, let us note $\dot{\mathbf{q}}$ the instant variation of these parameters in time. Let $T(\mathbf{q}, \dot{\mathbf{q}}, t)$ be the kinetic energy of the system at time t, state $\mathbf{q}$ and variation of state $\dot{\mathbf{q}}$ and $V(\mathbf{q}, t)$ the potential energy (which by nature does not depend on the variations). Then, let

$$L(\mathbf{q}, \dot{\mathbf{q}}, t) := T(\mathbf{q}, \dot{\mathbf{q}}, t) - V(\mathbf{q}, t)$$

which we call the *Lagrangian* of the physical system. For a trajectory $t \mapsto \mathbf{q}(t)$ defined between t_0 and t_f, and with known values at these boundaries, the *action* is

$$S[\mathbf{q}] := \int_{t=t_0}^{t_f} L(\mathbf{q}(t), \dot{\mathbf{q}}(t), t) dt$$

It is a functional on the space of possible trajectories. The stationary-action principle states that the trajectory effectively followed by the system is the one verifying

$$\mathbf{q} = \underset{\tilde{\mathbf{q}} \text{ admissible traj.}}{\arg\min} \; S[\tilde{\mathbf{q}}]$$

This is very convenient for physicists since a lot of different physical systems (conservative ones) can be totally characterized by their energy in Lagrangian form. The system's trajectories are then consequences of the stationary-action principle applied to the corresponding Lagrangian. So, physical systems naturally draw optimal trajectories in the sense of the physical quantity of the action. This is what fostered the development of the calculus of variations to study physical systems.

Then, an important branching occurs on how to characterize trajectories according to this principle.

Euler-Lagrange One can derive Necessary Conditions of Optimality (NCO) that trajectories must verify. This is done by differentiating the action with respect to the state vector, and this leads to the Euler-Lagrange equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0 \text{ for } i = 1, \dots, n \tag{1.1}$$

Integrating these equations in time is usually sufficient to reconstruct the system's trajectories.

Hamilton-Jacobi There exist another formulation of these equations, called *Hamiltonian* formulation. For the latter, physicists define *conjugate moments* $\mathbf{p} = (p_1, p_2, \dots, p_n)$ as

$$p_i := \frac{\partial L}{\partial \dot{q}_i} \text{ for } i = 1, \dots, n$$

Usually, this formula is revertible in $\dot{q}_i$ so they become a function of q_i and p_i

$$\dot{q}_i = f_i(q_i, p_i)$$

Then, the Hamiltonian of the system is defined as follows:

$$H(\mathbf{q}, \mathbf{p}, t) = \mathbf{p}\, f(\mathbf{q}, \mathbf{p}) - L(\mathbf{q}, f(\mathbf{q}, \mathbf{p}), t)$$

It is homogeneous to an energy. Euler-Lagrange equations (1.1) are then equivalent to the Hamiltonian formulation

$$\dot{q}_i = \frac{\partial H}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial H}{\partial q_i} \quad \text{for } i = 1, \ldots, n \tag{1.2}$$

In Hamiltonian mechanics, to get the system's evolution in time, on uses canonical transformations of the coordinate system. Without going too much into the details, the stationary-action principle is equivalent to the existence of some function S called *Hamilton's principal function* satisfying the non-linear Partial Derivatives Equation (PDE) called *Hamilton-Jacobi* equation:

$$H\left(\mathbf{q}, \frac{\partial S}{\partial \mathbf{q}}, t\right) + \frac{\partial S}{\partial t} = 0 \tag{1.3}$$

There are n significant integration constants $\alpha_1, \ldots, \alpha_n$ in the solution of this equation ($n + 1$ in total but one of them is a non-significant summation constant). The solution depends on coordinate variables and on these constants and we note $S(q_1, \ldots, q_n, \alpha_1, \ldots, \alpha_n, t)$. Because of the nature of S, the new canonical coordinates:

$$q'_i := \frac{\partial S}{\partial \alpha_i} \quad i = 1, \ldots, n \tag{1.4}$$

are constant in time $\dot{\mathbf{q}}' = 0$. So, in (1.4), the left hand term is a constant and the right hand term makes $q_1, \ldots, q_n$ appear with time t and constants among $\alpha_1, \ldots, \alpha_n$. Thus, the inversion of this equation leads to the deduction of the system's trajectory.

These methods for computing trajectories of physical systems will be advantageously adapted in optimal control theory to solve trajectory optimization problems for any kind of dynamical system.

Brief introduction to Optimal Control

Control theory soared in the middle of the 20th century, benefiting from centuries of advances in calculus of variations and contemporary progress in numerical methods. As will appear in what follows, it inherits much from physics' terminology.

We give here some fundamental definitions. A *dynamical system* is defined by a smooth function $f : \mathbb{R}^n \to \mathbb{R}^n$, and its trajectories satisfy

$$\begin{gathered} \dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t)) \\ \mathbf{x}(0) \in X_0 \end{gathered} \tag{1.5}$$

where X_0 is a subset of $\mathbb{R}^n$ containing possible initial states, and the function f is called the *dynamics*. Many interesting properties can be studied for such a system (periodicity, stability, response time, oscillations, ...). But to get interaction with the system, one needs to consider control systems.

A *control system* is a dynamical system parameterized at each time step by a control law. For any possible control law $t \mapsto \mathbf{u}(t)$, the corresponding trajectory satisfies

$$\begin{gathered} \dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{x}(0) \in X_0 \end{gathered} \tag{1.6}$$

Different properties can be studied for control systems:

- What is the *reachable set* of the system *i.e.* the set of all possible values for the state when the control law varies?
- Can we stabilize the system around a given point using a specific control law?
- Can we steer the system using a minimum amount of a time/effort?

The last question is what optimal control studies. In general, one wants to minimize a given cost J that can have several forms:

$$\begin{array}{cl}\text{Lagrangian form} & J = \int_{t=0}^{T} L(t, \mathbf{x_u}(t), \mathbf{u}(t))\mathrm{d}t \\ \text{Mayer form} & J = \varphi(T, \mathbf{x_u}(T)) \\ \text{Bolza form} & J = \varphi(T, \mathbf{x_u}(T)) + \int_{t=0}^{T} L(t, \mathbf{x_u}(t), \mathbf{u}(t))\mathrm{d}t\end{array}$$

where L is a Lagrangian cost, φ is a terminal cost, and we write $\mathbf{x_u}$ to make clear that $\mathbf{x}$ depends on the control function $t \mapsto \mathbf{u}(t)$ (as it results from the integration of Eq. 1.6). The final time T may be fixed or free. For instance, with free terminal time, $L = 1$ and $\varphi = 0$ entails a minimum-time problem. To find the control which minimizes the functional J, control theory will make use of the same methods as physics did for physical systems.

Euler-Lagrange, Pontryagin's Maximum Principle The NCO for control systems sometimes appear in the form of Euler-Lagrange equations in textbooks (Hestenes, 1966; Bryson and Ho, 1975) by analogy with physics' Euler-Lagrange equations (Eq. 1.1). But a comprehensive, rigorous formulation of necessary conditions for optimality that is now a reference in optimal control is the one proposed by Soviet mathematician Lev Pontryagin (Boltyanskiy et al., 1962), known as Pontryagin's Maximum Principle (PMP).

We give here the general idea, and further explanations will be given in Chapter 3 where the PMP is used to build trajectory optimization algorithms. For control system:

$$\begin{cases} \dot{\mathbf{x}}_{\mathbf{u}}(t) = f(t, \mathbf{x_u}(t), \mathbf{u}(t)) \\ J(\tau, \xi; \mathbf{u}) = \varphi(\mathbf{x_u}(T), T) + \int_{t=0}^{T} L(t, \mathbf{x_u}(t), \mathbf{u}(t))\mathrm{d}t \end{cases}$$

we define a *Hamiltonian*[1]

$$H(t, \mathbf{x}, \mathbf{p}, \mathbf{u}) = \mathbf{p}^{\intercal} f(t, \mathbf{x}, \mathbf{u}) + L(t, \mathbf{x}, \mathbf{u})$$

It should not be mixed up with physics' Hamiltonian, despite having the same name, as in this

[1] We omitted the cost multiplier parameter in front of L for clarity in a first approach. It distinguishes between normal and abnormal extremals. This presentation focuses on the most common ones, normal extremals, for which the multiplier can be assumed equal to 1.

case it does not embody an energy but it is a mathematical tool to write optimality conditions. The PMP states that

- the $\mathbf{p}$ vector, called *costate vector* shall evolve[2] with

$$\dot{\mathbf{p}}(t) = -\frac{\partial H}{\partial \mathbf{x}}^{\intercal}$$

  (reminiscent of physics' Hamiltonian)

- the optimal control $t \mapsto \mathbf{u}^{\star}(t)$ minimizes the Hamiltonian for almost every time

$$\mathbf{u}^{\star}(t) \in \arg\min_{\mathbf{u}} H(t, \mathbf{x}(t), \mathbf{p}(t), \mathbf{u}) \ \text{ for a.e. } t \in [0, T]$$

and additional boundary conditions shall be satisfied whether there is a target set to reach or if the final time is free, for instance. This method can be very efficient, when the optimal control argmin extraction is cheap and when one has a good way to find solutions satisfying the boundary conditions. In particular, this method is often preferred when the dimension of the problem increases, as it suffers less from the 'curse of dimension' than its PDE-based counterpart that is detailed in the following section. For a comprehensive presentation of the PMP, the reader is referred to Trélat (2005) (French) or Vinter (2010) (English).

Hamilton-Jacobi-Bellman If we parametrize control problems using the start point $\boldsymbol{\xi}$ and the start date τ, we get a family of control systems

$$\begin{cases} \mathbf{x}_{\mathbf{u}}(\tau) = \boldsymbol{\xi} \\ \dot{\mathbf{x}}_{\mathbf{u}}(t) = f(t, \mathbf{x}_{\mathbf{u}}(t), \mathbf{u}(t)) \\ J(\tau, \xi; \mathbf{u}(\cdot)) = \varphi(\mathbf{x}_{\mathbf{u}}(T), T) + \int_{t=0}^{T} L(t, \mathbf{x}_{\mathbf{u}}(t), \mathbf{u}(t)) \mathrm{d}t \end{cases}$$

[2] The most natural definition of the costate vector is without the transpose on the Hamiltonian partial derivative, which makes $\mathbf{p}$ a row vector. However, in this thesis, we prefer to make the costate a column vector, like the state vector.

The only variable left in this family is the control $t \mapsto \mathbf{u}(t)$. The *value function* $V(\tau, \boldsymbol{\xi})$ of this family of control systems is

$$V(\tau, \boldsymbol{\xi}) = \inf_{\mathbf{u}(\cdot)} J(\tau, \boldsymbol{\xi}; \mathbf{u}(\cdot))$$

It can be shown that this value function satisfies a Bellman optimality criteria (Trélat, 2005)

$$V(\tau, \boldsymbol{\xi}) = \inf_{\mathbf{u}(\cdot)} \left\{ \int_{t=\tau}^{\tau'} L(t, \mathbf{x}_{\mathbf{u}}(t), \mathbf{u}(t)) \mathrm{d}t + V(\tau', \mathbf{x}_{\mathbf{u}}(\tau')) \right\}$$

for any τ' such that $\tau \leq \tau' \leq T$. Under sufficient assumptions, this optimality criteria is equivalent to the following PDE

$$\frac{\partial V}{\partial \tau} + \inf_{\mathbf{u}} \left[H\left(\tau, \boldsymbol{\xi}, \frac{\partial V}{\partial \boldsymbol{\xi}}, \mathbf{u} \right) \right] = 0 \tag{1.7}$$

where H is the Hamiltonian defined in the previous section. This PDE is called the Hamilton-Jacobi-Bellman (HJB) equation, as it is of the same form as physics' Hamilton-Jacobi PDE (Eq. 1.3) and found using the recursive general optimality principle attributed to Richard Bellman. Solving the HJB equation provides comprehensive information about how the optimal cost propagates in a problem and provides subsequently a collection of optimal trajectories to the target. However, HJB-based methods are known to scale badly with dimension (indeed, exponentially with dimension), so that they can only be used in practice on low dimension problems. Empirically, in strategic long-range navigation problems for which computation power is available (the reference being a modern laptop), depending on the spatial resolution, the problem dimension can be brought up to 3 or 4, but not really higher, if the computation time shall be kept under an hour. This is a rule-of-thumb assessment that of course depends on the computational power available, the resolution of the problem and the efficiency of the software implementation. Nevertheless, it helps assess the curse of dimension of HJB-based methods practically.

The PMP and the HJB PDE are strong tools from optimal control to build numerical optimization techniques. They will be the basis on *indirect* optimization methods, presented in the following section.

Zermelo's Navigation Problem

Zermelo Navigation Problem (ZNP) is an optimal control problem that received much attention in the literature since Zermelo presented it in 1930. A very large diversity of approaches have been proposed to solve it. For some problem settings, the explicit or semi-explicit resolution of the problem is possible.

Analytical solutions In only a very limited number of problem settings does the ZNP have analytical solutions. First of all, in the presence of a uniform flow field, it is well known that the optimal control is constant, as a result of optimal control theory. Then, in non-uniform flow fields, exact solution for the ZNP are scarce. In Girardet (2014) annex F (French), the ZNP is solved analytically for a linear wind gradient (the expression is recalled in Example 3.1.3). In Bonnard et al. (2021), optimal trajectories are studied in the presence of Helmholz-Kirchhoff vortices in the flow field, and in Techy (2011), in the presence of sources and sinks.

But for general flow fields, exact solutions cannot be exhibited. In the case of non-uniform but weak flow field compared to the speed of the vehicle, as is the case for commercial aircraft, Jardin and Bryson (2001) derived analytical near-optimal solutions for the ZNP with application to aircraft flights over the USA. The idea is that in these conditions, the time-optimal trajectory is close to the great circle between origin and destination, so they linearize the optimization problem around this great circle to get an explicit near-optimal control law.

It shall also be noted that work was proposed for vehicle navigation based on a so-called *Zermelo-Markov-Dubins* model. This model is an extension of the kinematic Zermelo model (Eq. 2.1) when the turning rate of the vehicle is constrained (combination of Zermelo's kinematics and the behavior of a Markov-Dubins vehicle). It would correspond to medium-scale-point-of-view routing problems, when the magnitude of the turns is not negligible compared to the distance crossed. In this case, analytical solutions have been proposed for a constant wind drift (Bakolas and Tsiotras, 2010a; Techy and Woolsey, 2009; McGee et al., 2005). For the medium-range point of view, the constant wind drift can be a valid approximation in many cases. However, when this drift varies in time, Bakolas and Tsiotras (2010b) gave a method to compute Voronoï diagrams, *i.e.* a partition of space based on a grid of points for which each partition is the closest (in the sense of travel time) to a corresponding point in the grid.

General case In the general case, one must resort to computer methods to build a numerical solution to the problem. So, by essence, even if the ZNP is formulated in a continuous space over continuous data, the result of the optimization will be a discrete object approximating the true optimum of the problem. Thus, an important choice is *when* to resort to discretization in the problem solving. This important question divides the methods into two categories called *direct* methods and *indirect* methods. In the former, discretization is performed directly (in the model of the state space or the flow field for instance), and optimization is performed over the discrete representation (‘*discretize first, optimize then*’). In the latter, some optimal criteria are applied prior to the numerical search for a solution (‘*optimize first, discretize then*’).

Overall, this review splits the literature into five main categories. They are listed below, with a (**D**) or a (**I**) whether the method family belongs to direct or indirect methods.

- Control parameterization methods (**D**)

  See a trajectory as a collection of waypoints linked together by an integration scheme

- Graph-based methods (**D**)

  See state space as a finite collection of points with connections

- Sampling-based methods (**D**)

  Build trajectories progressively by sampling new points in the state space

- Extremal methods (**I**)

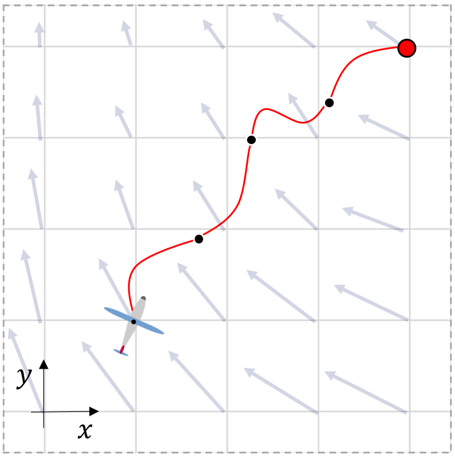


Figure 1.1: Sketch of collocation methods principle. The trajectory of the vehicle is discretized at given time stamps and interpolated in between, usually using a polynomial function.

Characterize candidate solutions to optimality (NCO)

- Front propagation methods (**I**)

Propagate iso-costs fronts in space

We now give important references of work belonging to each of the families and addressing routing optimization in flow fields. While some references indeed solve precisely the ZNP, we also include other references doing route optimization on more complex models of vehicles than the pure kinematic Zermelo vehicle, and also vessel routing references because of their proximity with the matter.

Direct methods

Control parameterization

When looking for discretizing the optimization problem, one idea may be to discretize the time axis. This is what *control parameterization* is doing. The principle is to replace the time-continuous representation of the trajectory and the control by a discrete one. This simplifies an infinite-dimensional optimization problem (the decision variable is the control

$t \mapsto \mathbf{u}(t)$) into a finite-dimensional one: the control values are chosen at discretization points $(\mathbf{u}(t_1), \mathbf{u}(t_2), \ldots, \mathbf{u}(t_n))$ and interpolated in between. This is sketched in Fig. 1.1. Since there are underlying dynamics governing the evolution of the state, the values of the control at the discretization nodes must satisfy constraints given by an integration scheme. One is then left with a Non-Linear Programming (NLP) problem with finitely many optimization variables and constraints, and general-purpose solvers can be used to minimize a cost on the whole trajectory (SNOPT, IPOPT, ...). This has made the method attractive since general-purpose solvers have become increasingly powerful thanks to advances in optimization and numerical methods.

Because discretization leaves empty choices for the values of the control and the state in the intervals between discretization nodes, there are different flavors of control parameterization depending on the interpolation. Transcription, collocation, pseudo-spectral methods are such flavors of control parameterization. Their differences are well documented in the literature review from González Arribas (2019), and we refer the reader to the latter for more details.

Overall, control parameterization methods have the main advantage that they can handle a complex model of the vehicle subject to trajectory optimization, definitely more complex than the kinematic abstraction presented in Eq. 2.1. It has been successfully applied to refined models of commercial aircraft (Hargraves and Paris, 1987) even with flight regulation constraints (Betts and Cramer, 1995). In Wang et al. (2019), AUV path planning problems are solved with state constraints as well as constraints on the acceleration of the vehicle. Control parameterization is also a very useful tool to investigate special trajectory optimization problems such as the energy minimization of flight in the presence of a wind gradient, mimicking the behavior if the albatross for efficient long-range flight, such as in Bonnin (2015).

The main limit of control parameterization is that most efficient methods solving the underlying NLPs are made to find *local* minima, thus there is no guarantee that the solution found is the best *globally* for the problem. This may or may not be a problem depending on the application case.

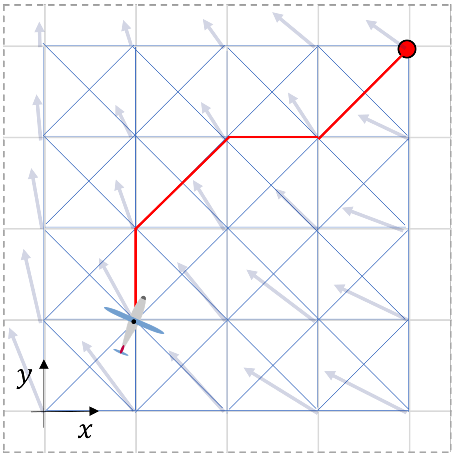


Figure 1.2: Sketch of graph-based methods principle. The motion of the vehicle is abstracted to a motion on a grid, with commutation time adjusted for the flow field advection.

Graph-based methods

Another useful way to discretize the trajectory optimization problem is to cast it to the problem of finding a shortest path in a *graph*. This enables the usage of an exhaustive, well established literature on graphs: for undirected, non-negative weights graphs, Dijkstra's algorithm can compute a shortest path between two points; for the same problem in a directed graph with possibly negative weights, there is the Bellman-Ford algorithm.

Turning the navigation problem into a graph can be done, for instance, by sampling points in the environment (vertices) and creating connections between neighboring points (edges) and assigning the elementary travel time from the kinematic model (depending on the flow magnitude and direction) as cost on edges, as sketched in Fig. 1.2. In such a graph, shortest paths are time-optimal paths for the navigation problem.

Dijkstra's algorithm was really though to solve shortest path problems efficiently in any kind of graph. But for trajectory optimization problem, the underlying graph is embedded in a metric space. Thus, a very useful upgrade of Dijkstra's algorithm is the A* (A star) algorithm. The principle of the A* algorithm is to bias cost propagation in priority in a direction provided by a heuristic function. For instance, in the 2D planar space for robot motion planning in the presence of obstacles, the Euclidean distance to the target is a valid heuristic (the heuristic value always has to be an underestimation of the real cost to destination for the algorithm to converge to the true optimum of the problem). The A* algorithm is an extension of Dijkstra's algorithm in the sense that if the heuristic function is chosen identically null, then the algorithm is Dijkstra's algorithm.

The approach proved efficient to solve trajectory optimization problems for AUVs in steady but non uniform flow fields (Garau et al., 2005; Kularatne et al., 2016). The graph construction can also be advantageously biased in the flow direction for reduced computation cost, as shown in Li et al. (2020). The A* approach efficiency relies on a proper choice of the heuristic function. This choice can be simple in the case of steady and weak flow fields. But for unsteady, strong flow fields, it may be less obvious if not unsuited. Still, the A* star principle, which is to prioritize computation where it is most likely needed, can often be included as refinement of an existing algorithm, as done in the graph-based method presented in Eichhorn (2015).

For vessel route optimization, Mannarini et al. (2016) proposed a graph-based method valid in time-varying currents. The authors compared their approach to a level-set method in Mannarini et al. (2020). It appeared that to have similar resolution in the optimal trajectories, the graph-based method was computationally more intensive than the level-set method.

Other approaches such as the Bellman-Ford algorithm proved efficient, for instance for aircraft routing in atmospheric wind fields (Legrand et al., 2018).

Sampling-based methods

Discretization can also be performed progressively in the exploration of the state space. It is how *sampling-based* methods operate. In this class of methods, the optimization is performed *asymptotically*, meaning that when the number of state space sampled points increases, the solution trajectory built converges to the optimal trajectory. The principle is sketched in Fig. 1.3.

The first renown general sampling-based approach is that of Probabilistic Road Maps (PRMs). The PRM approach samples points in the environment, creates connections when-

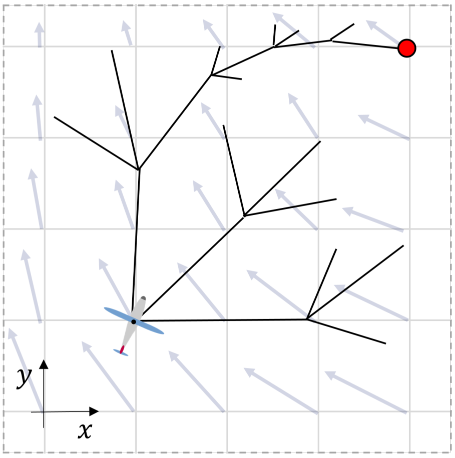


Figure 1.3: Sketch of sampling-based methods principle. The space of possible trajectories for the vehicle is sampled progressively as the algorithm moves on.

ever possible (out of obstacles) and then performs graph-based shortest path computation of the created graph. So, PRMs are more of a graph generation procedure than a new optimization method. But we mention them as they are the entry point to sampling methods.

The more modern Rapidly-exploring Random Trees (RRTs) sampling-based methods are now very popular. They proved efficient for robot path planning in the presence of obstacles. The principle relies on sampling the state space randomly and connecting new sampled points to previous ones according to a distance criteria, thus building a tree. The method is especially attractive for problems in a high-dimensional state space since this search heuristic can build elementary paths between two states given constraints in a computationally efficient way. But the raw method is not exact (neither does it find the global optimal path nor a local optimum), so a popular refinement is RRT*, which can achieve asymptotic optimality.

For vehicles in flow fields, the issue is to take advection into account. In Rao and Williams (2009), a modification of RRT* is proposed so that the tree expansion is biased by the flow field, thus converging faster to the time-optimal path. In Oettershagen et al. (2017), a RRT* search combined with Dubins-path is used to create optimal paths in a non-uniform flow field for a UAV. Their application puts a primer on the real-time computation feature since the algorithm is supposed to run on an onboard computer. Thus, the method needs not be exact but qualitatively approximate optimal paths while being fast. The vast majority of RRT applications occur in such a context.

Combining ideas from RRT and fast marching methods (presented in Section 'Front propagation') led to the construction of the Fast Marching Tree (FMT*) algorithm by Janson et al. (2015) (the star highlighting its asymptotic optimality). The principle is similar to RRT in the sense that one samples the environment and iteratively builds a tree. But FMT* does it following hypersurfaces of constant cost, as fast marching methods do or even Dijkstra's algorithm in a graph sense. It has been shown to outperform RRT* and PRMs in several applications. Once again the method seduces because of its efficiency to compute near-optimal trajectories with a satisfactory trade-off between computation time and error to the global optimum. It has proven efficient for real-time decision problems such as emergency trajectory computation (Guitart et al., 2022) or UAV path planning in the presence of obstacles (Schneider, 2016). However, both previous applications do not consider wind advection.

In Chakrabarty and Langelaan (2013), an original sampling strategy was developed to take into account the kinematics of a glider through the use of so-called *motion primitives*. The resulting method, called kinematic tree, can then deal with time-varying environments, because the discretization occurs on control variables and not on the position. The method has proven efficient for the computation of time-optimal paths for a glider in a 3D dynamic environment featuring thermal updrafts.

Intersections

Of course, the current method classification attempt fails at some point. Methods are not incompatible with one another, and some authors choose to harness the advantages of several methods at the same time. It is the case of Borndörfer et al. (2020), where the authors use a coarse graph-based method to initialize a finer control parameterization method for aircraft trajectory optimization. They harness the exhaustiveness of the graph search on a coarse discretization with the efficiency of the NLP solver to find a smooth solution, thus less suffering

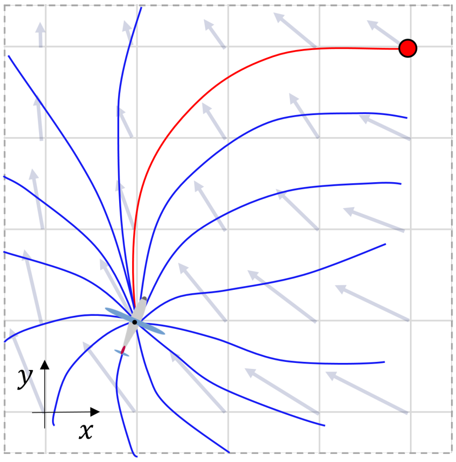


Figure 1.4: Sketch of extremal-based methods principle. Trajectories candidate for optimality are integrated forward in time from the origin of the problem and a search is performed to find the ones satisfying terminal conditions.

from the problem of falling in local minima.

Out of categories

Out of the previous categories, the approach presented in Soulignac (2011) is also worth noting. While the proposed method does not really fit in the previous families, it belongs nevertheless to the class of direct methods. The proposed approach computes regions of approximately constant flow field and makes the discretization of the trajectory happen on the borders of these regions. This simplification of the flow field is valid when the latter does not exhibit strong local variations. In the regions of constant wind, control theory states that the control value must be constant. Thus, the search for optimality results in adjusting the state at the border of the constant-wind regions, creating a piecewise linear approximation of the optimal trajectory.

Indirect methods

Extremal methods

The ZNP appearing in Eq. 2.6 exhibits some mathematical structure that can be exploited before casting to discrete representation for numerical resolution. In particular, one can *characterize* the only possible candidate solutions to optimality, in other words, one can apply necessary conditions for optimality to the optimization problem. But here the decision variable at stake (the control function $t \mapsto \mathbf{u}(t)$) is infinite-dimensional, so a special tool must be used. Thus tool is Pontryagin's Maximum Principle (PMP) (Boltyanskiy et al., 1962). The PMP reduces the search space to a collection of trajectories candidate to optimality called *extremal trajectories* or simply *extremals*. They will be the basis of the discussion presented in Chapters 3, 4. When applying the PMP, a virtual additional state called the *costate* $t \mapsto \mathbf{p}(t)$ is added to the problem and satisfies conditions at the final time, as opposed to the state which satisfies initial conditions. So in general, one has to solve a Two-Point Boundary Value Problem that can be solved using a shooting method, as sketched in Fig. 1.4.

Ship routing was addressed with extremals in Bijlsma (1975) and then extended to aircraft in Bijlsma (2009). In 2D spherical space, which is required for long-range commercial aircraft flights to account correctly for the Earth's sphericity, methods based on the shooting of extremals on the sphere were proposed in Jardin and Bryson (2010); Marchidan and Bakolas (2016). For AUV navigation, Rhoads (2013) proposed a general algorithm that will be discussed and compared to what is proposed in Chapter 3.

A special mention shall be added to the *chord length parameterization* point of view on the navigation problem, which leads a different writing of the NCO. Indeed, it is possible to reverse the role of space and time in the trajectory optimization problem, considering a trajectory parameterized by its chord length $s \mapsto \mathbf{x}(s)$ rather than time. Thus, the time evolution is deduced from the *slowness* of the vehicle, which depends on the location and on the direction $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s}$ of the path (compared to the direction of the flow field). This point of view is adopted in Davis

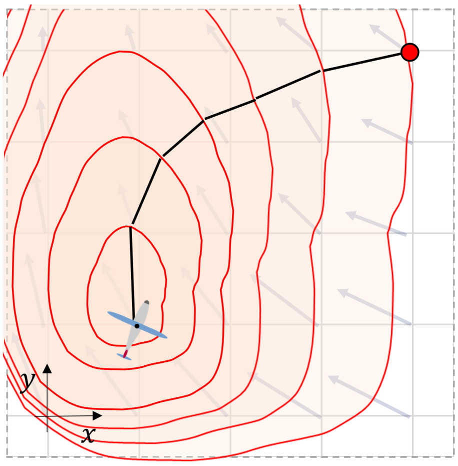


Figure 1.5: Sketch of front propagation methods principle. Iso-cost level-set are compute forward in time. Optimal trajectories can be deduced by backtracking the gradient of the level-set function.

et al. (2009) to shoot extremals, and proved efficient in the case of weak flow field for the computation of optimal trajectories. Chord length parameterization will be the basis of an approach proposed in Chapter 7.

In an aerospace context, apart from the long-range navigation problem in wind fields, the shooting of extremals was applied to many different other problems. Of course to spacecraft optimal routing and orbit transfers, but this is beyond the scope of this thesis. For aircraft applications we can find the optimization of aircraft vertical flight profile (Nguyen, 2006), the optimal decision of aborting landing given wind shears (Bulirsch et al., 1991a,b) or contrail avoidance (Sridhar et al., 2011).

Finally, for a review of extremal methods for the ZNP in the presence of state constraints, the reader is referred to Chapter 4.

Front propagation methods

Instead of focusing on trajectories, one can try to solve for the vehicle's reachable set. If a function $\phi(t, \mathbf{x})$ is such that $\phi(t, \mathbf{x}) < 0$ if $\mathbf{x}$ is reachable in time $t' < t$, $\phi(t, \mathbf{x}) = 0$ if $\mathbf{x}$ is reachable in time t and $\phi(t, \mathbf{x}) > 0$ if $\mathbf{x}$ is not reachable in time less than t, then it can be shown (Lolla, 2016, Theorem 1) that ϕ satisfies the following PDE

$$\frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial \mathbf{x}} \mathbf{v}_f + u_{\max} \left\| \frac{\partial \phi}{\partial \mathbf{x}} \right\| = 0 \tag{1.8}$$

with initial condition $\phi(t_0, \mathbf{x}) = \|\mathbf{x} - \mathbf{x}_0\|$. It is well known that the solution of this PDE can have irregularities that make it non-smooth (shocks). So, ϕ cannot be searched as a classical solution of the PDE. Instead, it should be searched as a *viscosity solution* of the PDE, which is an extension of the definition of a solution to a PDE that allows cusps, as proposed in Crandall and Lions (1983).

In the front propagation method family, optimality is thus characterized through a PDE, and the discretization occurs when solving the equation. Optimal trajectories can be deduced from ϕ by *backtracking* the gradient of ϕ, as depicted in Fig. 1.5. For instance, if $\mathbf{x}_f$ is reachable in optimal time $t^\star$, then the optimal trajectory from $(t_0, \mathbf{x}_0)$ to $(t^\star, \mathbf{x}_f)$ is found by integrating the backward Ordinary Differential Equation (ODE) (Lolla, 2016)

$$\dot{\mathbf{x}}(t) = -u_{\max} \left. \frac{\partial \phi}{\partial \mathbf{x}} \right|_{t, \mathbf{x}(t)} - \mathbf{v}_f(t, \mathbf{x}(t))$$

from $\mathbf{x}(t^\star) = \mathbf{x}_f$.

Equation 1.8 belongs to the general class of Hamilton-Jacobi-Bellman PDEs, presented in the previous section.

To find solutions of the ZNP, it is also possible to solve a PDE directly on the value function of the problem (as Eq. 1.7) as done in Doshi et al. (2022). This presents the advantage of computing simultaneously all the backward reachable sets reaching the target in some window $[T - \Delta T, T]$.

In general, *fast marching* methods to solve this kind of PDEs were presented by Sethian (1996). It shall be noted that a strong simplification occurs when the flow field is weak, *i.e.* smaller than the vehicle's SRF $\left\|\mathbf{v}_f(t, \mathbf{x})\right\| < u_{\max}$ for all t and $\mathbf{x}$. In this case, the solutions of the

ZNP can be found by solving the *eikonal* equation (see e.g. Lolla (2016))

$$u_{\max}\left\|\frac{\mathrm{d}\psi}{\mathrm{d}\mathbf{x}}\right\| + \frac{\mathrm{d}\psi}{\mathrm{d}\mathbf{x}}\mathbf{v}_f - 1 = 0$$

where the function $\mathbf{x} \mapsto \psi(\mathbf{x})$ is the function mapping the optimal time to come to each position $\mathbf{x}$. For eikonal equations, ordered upwind methods were proposed for improved performance (Sethian and Vladimirsky, 2003). For the medium-range point of view where the vehicle's trajectory cannot take arbitrary curvature, efficient fast marching methods were proposed in the eikonal regime such as Mirebeau (2018). Practical resolutions of the ZNP in the eikonal regime using fast marching methods can be found in Petres et al. (2007) for AUVs or in Girardet et al. (2014) for aircraft trajectory planning.

Extensions of this review

The literature review focused on references dealing with the ZNP, which is about time-optimal trajectory computation in free space and deterministic environment. Chapter 4 will briefly review references for the application of the PMP in the presence of state constraints. In Chapter 5, references dealing with energy-time trajectory optimization will be provided. Lastly, Chapter 7 will review the references dealing with uncertainty in the flow field data.

Conclusion of the review

The literature on trajectory optimization is very rich and features many different families of methods able to solve trajectory planning problems. Whether graph-based, sampling-based or based on Non-Linear Programming, the class of direct methods features many successful resolutions of trajectory planning problems. The indirect family with extremals and front propagation also proved successful, with comparatively less references than the direct group, which could be explained by the increased complexity in the theory underlying the methods. The latter family of methods does not need the weak flow assumption and can provide guarantees of optimality by computing the cost map in a given navigation problem, which is the reason why it is the family that will give the building blocks of what comes in the following chapters.

Research gaps

In general, complex navigation problems featuring flow fields with possibly strong magnitude, obstacles, whether still or moving, diffuse hazard in the environment or energy optimization have been mostly addressed using direct methods in the literature. Indeed, the latter can be easily implemented on top of an arbitrarily complex model of the navigation problem. However, the approximations usually done by direct methods and the local optimum property (for some of direct methods), if acceptable to reduce the computation cost for fast resolution or onboard applications, are not desirable when the trajectory planning problem is solved at the strategic level, *i.e.* when high computation power is available and computation time up to an hour is acceptable. Indirect methods, while being computationally intensive, can provide the appropriate precision level for strategic trajectory planning. Nevertheless, references implementing indirect methods with complex problem feature are not numerous. In particular, extremal methods have received comparatively less attention than the other, despite their potential for efficiency even in high-dimensional problems. These research gaps motivated the work presented in this thesis.

Chapter 2

Properties of the navigation problem

Abstract

In this chapter, the mathematical properties of the navigation problem when adopting Zermelo's kinematic model are under study. The mathematical framework required to build a rigorous basis for considering a continuous model for trajectories is presented. It is highlighted that in strong flows, some directions of motion are not possible for the vehicle. When a specific direction is indeed possible, an expression for the control steering the vehicle accordingly is given. The equivalence between forward reachability and backward reachability is stated, so that the problem of finding possible positions of the vehicle after some time is mathematically the same as knowing from which positions the vehicle can reach a target in some given time. The notion of ground paths is dissociated from the notion of trajectory, as it can be the basis of a different model for the navigation problem. The notion of value function for optimal control problems is introduced, in particular for time-optimal navigation problems. Lastly, the kinematic model is adapted to spherical coordinates.

Résumé en français

Dans ce chapitre, on s'intéresse aux propriétés mathématiques du problème de navigation basé sur le modèle cinématique de Zermelo. Un cadre mathématique rigoureux nous permettant de travailler sur un modèle continu pour les trajectoires est introduit. Le manque de contrôlabilité pour le véhicule dans le cas d'un champ de flot fort est étudié, avec des directions de mouvement qui deviennent inaccessibles. Pour les directions de mouvement possibles, une expression du contrôle suivant cette direction à vitesse maximale est donnée. L'équivalence entre ensemble atteignable en temps croissant et ensemble atteignable en temps décroissant est établie, de sorte que le problème de trouver les états atteignables pour le véhicule en un certain temps est mathématiquement équivalent à trouver les états depuis lesquels on peut rejoindre une cible en un temps donné. La notion de chemin sol est dissociée de la notion de trajectoire, l'utilisation d'un objet ou de l'autre amenant à des modèles différents du problème de navigation. Enfin, le modèle cinématique est adapté aux coordonnées sphériques pour les applications nécessitant la prise en compte de la courbure de la Terre.

Contents

The simple kinematic vehicle model

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x}(t)) \tag{2.1}$$

coming from the LSPOV has some interesting properties that should be discussed before addressing any routing optimization problem. This chapter discusses these properties. The environment is assumed to be the 2D planar space $\mathbb{R}^2$ (or a subspace of it) for the simplicity of definitions and properties.

Chapter's main questions

- Mathematically, what are the properties of the large-scale point of view model adopted for long-range trajectory planning?
- Is the large-scale point of view adaptable to spherical coordinates?

2.1 Mathematical basis

For now, the trajectory of the vehicle $t \mapsto \mathbf{x}(t)$ was implicitly defined as a function of time, without additional precisions. Intuitively, if the following are fixed:

- The starting date t_0
- The starting position $\mathbf{x}_0$
- The flow field $t, \mathbf{x} \mapsto \mathbf{v}_f(t, \mathbf{x})$
- The control law $t \mapsto \mathbf{u}(t)$

then the vehicle's trajectory $t \mapsto \mathbf{x}(t)$ should be unambiguously defined.

Mathematically speaking, it is the Cauchy-Lipschitz theorem which ensures the previous intuitive fact. In what follows, the dot notation is used for time derivation $\dot{\mathbf{x}} = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}$. We consider the Cauchy problem:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x}(t)) \\ \mathbf{x}(t_0) = \mathbf{x}_0 \end{cases} \tag{2.2}$$

We call $\mathcal{D}$ the domain of the state space allowed for the vehicle. Under the following assumptions:

(H1) $\forall \mathbf{x} \in \mathcal{D}, \quad t \mapsto \mathbf{v}_f(t, \mathbf{x})$ is continuous

(H2) $\forall t \in I, \quad \mathbf{x} \mapsto \mathbf{v}_f(t, \mathbf{x})$ is piece-wise Lipschitz continuous, *i.e.* there exists a finite collection of pairwise disjoint open sets $(\Omega_j)_{j \in J}$ such that $\mathcal{D} = \cup_{j \in J} \overline{\Omega_j}$ and for all $t \in I$, $\mathbf{x} \mapsto \mathbf{v}_f(t, \mathbf{x})$ is Lipschitz continuous on Ω_j

(H3) $[t_0, +\infty[\ \ni t \mapsto \mathbf{u}(t)$ is bounded

problem 2.2 admits a solution $\mathbf{x}(\cdot)$ which is:

- defined either over $[t_0, +\infty[$ or $[t_0, t_u]$ with t_u the time upper bound and $\mathbf{x}(t_u) \in \partial\mathcal{D}$;
- absolutely continuous (thus continuous and differentiable almost everywhere);
- unique over its definition window.

In particular, the uniqueness ensures that defining the vehicle's evolution using its velocity is unambiguous, which fits to intuition.

> **Remark**
> The common Cauchy-Lipschitz theorem is stated for Lipschitz continuous dynamics w.r.t. the state variable. However, hypothesis (H2) differs from this. A rigorous justification of this extension is proposed in Appendix A. The extension is needed since we want to be able to consider simplified analytical problems that may violate the pure Lipschitz hypothesis. The ideal stream flow appearing in Subramani and Lermusiaux (2016) (see Fig. 5 in the article) is an example of flow field that is only piece-wise Lipschitz continuous (it has discontinuities).

The set of control functions is

$$\mathcal{U} := U^{[t_0,+\infty[} = B\left(0, u_{\max}\right)^{[t_0,+\infty[}$$

Flow of equation (2.2) In navigation settings $(t_0, \mathbf{x}_0, \mathbf{v}_f)$ with a given control function $\mathbf{u}(\cdot) \in \mathcal{U}$, if the evolution of the vehicle given by equation (2.2) is noted $\mathbf{x}(\cdot)$, we define the *output application* (also called the flow of 2.2), for $\tau \geq 0$

$$\phi^{\mathbf{u}(\cdot)}_{t_0,\mathbf{x}_0,\mathbf{v}_f}(\tau) = \{\mathbf{x}(t_0 + \tau)\}$$

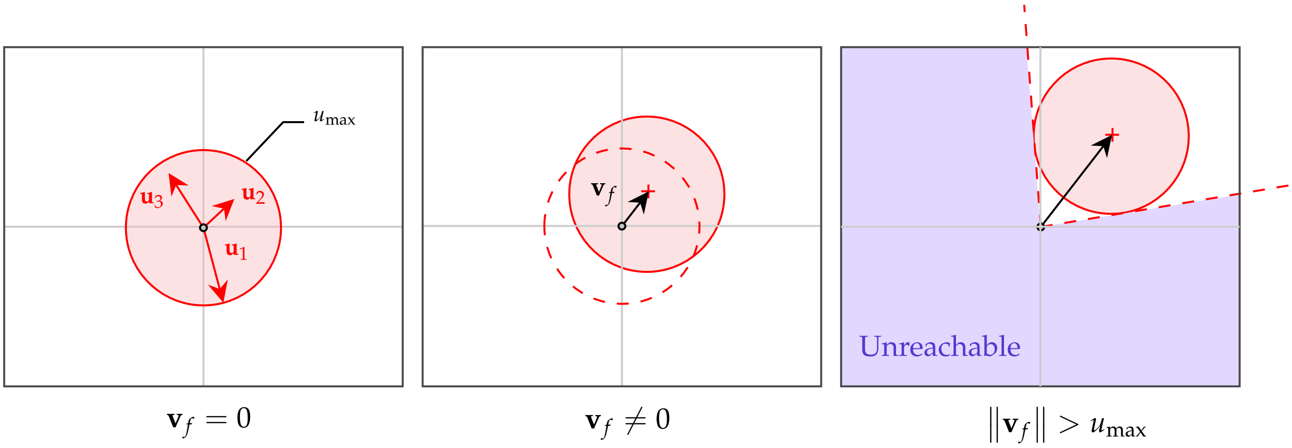


Figure 2.1: Sets of possible ground velocities (in red) depending on the flow magnitude. Left, three example control vectors are shown among the possible $\|\mathbf{u}\| \leq u_{\max}$. In this case the ground velocity is equal to the control (no flow). Middle, a weak flow pushes the vehicle. All directions of motion are still possible for the ground velocity, but the speed is higher with the flow in the back than in the front. Right, strong flow case where some directions are not possible anymore for the ground velocity.

If $t_0 + \tau$ is superior to the maximum definition date t_u for the trajectory, then we set $\phi^{\mathbf{u}(\cdot)}_{t_0,\mathbf{x}_0,\mathbf{v}_f}(\tau) = \varnothing$. It is an application that maps a start time and position $(t_0, \mathbf{x}_0)$ in a flow field $\mathbf{v}_f$ to the vehicle's future position after duration τ by following control $\mathbf{u}(\cdot)$. Because the solution to equation (2.2) is unique, $\phi^{\mathbf{u}(\cdot)}_{t_0,\mathbf{x}_0,\mathbf{v}_f}(\tau)$ always contains at most one element.

2.2 Controllability

The advection of the vehicle by the flow field changes the way the vehicle can evolve in space. In particular, as stated in the introduction, the vehicle can reach the whole state space if $u_{\max} > \|\mathbf{v}_f(t, \mathbf{x})\|$ for all t and all $\mathbf{x}$. It is not necessarily the case if the latter condition is not satisfied. This because at a given time and position, the ground velocity of the vehicle is the summation of the flow field and the possible VRFs. The available ground velocity vectors are depicted in Figure 2.1 depending on the flow magnitude.

Since some directions of movement are sometimes not allowed for the vehicle, it is useful to put a name on possible directions.

Definition 1 (Valid direction)**.** *Direction* $\mathbf{d} \in \mathbb{R}^2$ *is said* valid *for* $(t, \mathbf{x})$ *if there exists a control steering the vehicle in the given direction,* i.e. $\exists \mathbf{u} \in B\left(0, u_{max}\right),\ \mathbf{u} + \mathbf{v}_f(t, \mathbf{x}) = \lambda\, \mathbf{d}$ *for some* $\lambda > 0$*.*

> **Remark**
>
> The previous definition is independent of the norm of the direction vector **d**

Directional time-optimal control If the problem is to find the fastest way to follow a given valid direction **d**, we prove that the control achieving this goal is unique.

> **Property 2.1**
>
> If $\mathbf{d} \in \mathbb{R}^2$ is a valid direction at point $(t, \mathbf{x})$, then there exists a unique control maximizing ground speed $\left\|\mathbf{u} + \mathbf{v}_f(t, \mathbf{x})\right\|$ and such that the ground velocity vector $\mathbf{u} + \mathbf{v}_f(t, \mathbf{x})$ is colinear and of the same direction as **d**.

Proof. Since the direction **d** is assumed valid, the set L of λ such that it exists a control $\mathbf{u} \in B\left(0, u_{\max}\right)$ satisfying $\mathbf{u} + \mathbf{v}_f(t, \mathbf{x}) = \lambda\, \mathbf{d}$ is not empty. It is bounded because controls are bounded. Thus, it has an upper bound $\lambda_{\max}$, for which the ground speed vector $\lambda_{\max}\, \mathbf{d}$ has maximal norm. The corresponding control is unique, indeed it is $\mathbf{u}^\star = \lambda_{\max}\, \mathbf{d} - \mathbf{v}_f(t, \mathbf{x})$. □

Since the best directional control is unique, we give it a name and a symbol.

Definition 2 (Directional time-optimal control)**.** *For* $\mathbf{d} \in \mathbb{R}^2$ *a valid direction at* $(t, \mathbf{x})$*, the unique control achieving highest speed in the given direction is called the* directional time-optimal control*. It is a function of the direction and of the flow field value and we note it* $\mathbf{u}(\mathbf{d}, \mathbf{v}_f)$*.*

The directional time-optimal control has an expression given by the following property.

Property 2.2: Directional time-optimal control

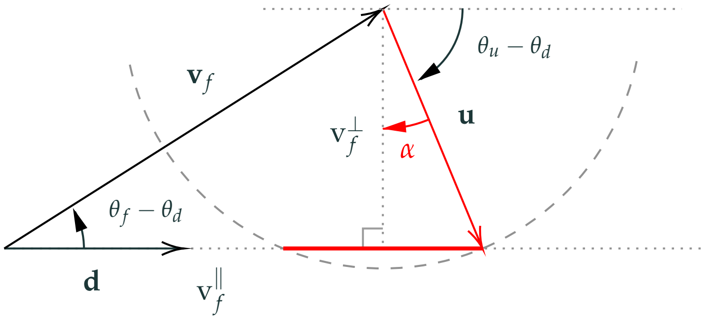
For any valid direction $\mathbf{d} \in \mathbb{R}^2$, define the components of the flow field $\mathrm{v}_f^{\parallel} := \mathbf{v}_f(t, \mathbf{x})^\intercal \mathbf{d}$ and $\mathrm{v}_f^{\perp} := \mathbf{v}_f(t, \mathbf{x})^\intercal R\left(\frac{\pi}{2}\right) \mathbf{d}$.

The directional time-optimal control is:

$$\mathbf{u}(\mathbf{d}, \mathbf{v}_f) = u_{\max} R\left(\alpha - \frac{\pi}{2}\right) \frac{\mathbf{d}}{\|\mathbf{d}\|}$$
$$\text{with } \alpha = \operatorname{arctan2}\left(\sqrt{u_{\max}^2 - (\mathrm{v}_f^{\perp})^2}, \mathrm{v}_f^{\perp}\right)$$

The resulting motion has ground speed of $\mathrm{v}_f^{\parallel} + \sqrt{u_{\max}^2 - (\mathrm{v}_f^{\perp})^2}$.

Proof. On the following geometric construction



the solid red line indicates the possible values for $\mathbf{u}$ to have a ground speed in the direction $\mathbf{d}$. Among all these control values, the one achieving the maximum ground speed in the direction $\mathbf{d}$ is obtained at the right edge of the solid red line. This edge belongs to the circle $\|\mathbf{u}\| = u_{\max}$. With α defined in the figure, we have

$$\alpha = -\operatorname{arctan2}\left(\sqrt{u_{\max}^2 - (\mathrm{v}_f^{\perp})^2},\ \mathrm{v}_f^{\perp}\right)$$

the minus sign coming from the fact that $\sqrt{u_{\max}^2 - (\mathrm{v}_f^{\perp})^2}$ is positive and $\mathrm{v}_f^{\perp}$ is positive in this situation, but α is negative here. And furthermore

$$\theta_u - \theta_d + \alpha = -\frac{\pi}{2}$$

thus

$$\mathbf{u}(\mathbf{d}, \mathbf{v}_f) = u_{\max} R \left(\theta_s - \theta_d\right) \frac{\mathbf{d}}{\|\mathbf{d}\|} = u_{\max} R \left(\alpha - \frac{\pi}{2}\right) \frac{\mathbf{d}}{\|\mathbf{d}\|}$$

□

2.3 Forward and backward reachable sets

In the introduction, we reckoned reachability as a first key problem in navigation. With navigation settings (start time, start position, flow field) $(t_0, \mathbf{x}_0, \mathbf{v}_f)$, the *reachable set*

$$\mathcal{R}_{t_0,\mathbf{x}_0,\mathbf{v}_f}(\tau) := \bigcup_{\mathbf{u}(\cdot)\in\mathcal{U}} \phi^{\mathbf{u}(\cdot)}_{t_0,\mathbf{x}_0,\mathbf{v}_f}(\tau) \tag{2.3}$$

is the collection of all positions that are reachable in some duration τ. It can also be called *forward* reachable set, as it collects the propagation of possible trajectories of the system. We note the total reachable set

$$\mathcal{R}_{t_0,\mathbf{x}_0,\mathbf{v}_f} := \bigcup_{\tau\geq 0} \mathcal{R}_{t_0,\mathbf{x}_0,\mathbf{v}_f}(\tau)$$

the collection of all reachable sets. The subscript in the flow field is dropped whenever there is no ambiguity.

The dual object to the forward reachable set is the *backward* reachable set

$$\begin{aligned} &\mathcal{BR}_{t_f,\mathbf{x}_f,\mathbf{v}_f}(\tau) := \\ &\left\{\mathbf{x} \in \mathbb{R}^2 \mid \exists \mathbf{u}(\cdot) \in \mathcal{U},\ \phi^{\mathbf{u}(\cdot)}_{t_f-\tau,\mathbf{x},\mathbf{v}_f}(\tau) = \{\mathbf{x}_f\}\right\} \end{aligned} \tag{2.4}$$

It is the collection of all positions that can be steered to the destination in duration τ, arriving at a prescribed time t_f

An important remark is that if one knows how to compute forward reachable sets, then one knows how to compute backward reachable sets, as explained by the following property.

Property 2.3

The backward reachable set is a forward reachable set for a mirror problem in which the wind field is reversed, *i.e.*

$$\mathcal{BR}_{t_f,\mathbf{x}_f,\mathbf{v}_f}(\tau) = \mathcal{R}_{0,\mathbf{x}_f,\widetilde{\mathbf{v}_f}}(\tau) \quad \text{with } \widetilde{\mathbf{v}}_f(t,\mathbf{x}) := -\mathbf{v}_f(t_f - t, \mathbf{x})$$

Proof.

- We first show that $\mathcal{BR}_{t_f,\mathbf{x}_f,\mathbf{v}_f}(\tau) \subset \mathcal{R}_{0,\mathbf{x}_f,\widetilde{\mathbf{v}_f}}$. Let $\mathbf{y} \in \mathcal{BR}_{t_f,\mathbf{x}_f,\mathbf{v}_f}(\tau)$. Then let $\mathbf{u}(\cdot) \in \mathcal{U}$ be the control and $\mathbf{x}(\cdot)$ the trajectory satisfying

  - $\dot{\mathbf{x}}(t) = \mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x}(t))$
  - $\mathbf{x}(t_f - \tau) = \mathbf{y}$
  - $\mathbf{x}(t_f) = \mathbf{x}_f$

  We show that

  $$\widetilde{\mathbf{u}}(t) := -\mathbf{u}(t_f - t) \tag{2.5}$$

  steers the mirror problem from $\mathbf{x}_f$ to $\mathbf{y}$ in duration τ.

  Define $\mathbf{z} : [t_f - \tau, t_f] \ni t \mapsto \mathbf{x}(t_f - t)$. Thus $\mathbf{z}(0) = \mathbf{x}(t_f) = \mathbf{x}_f$. Then:

  $$\begin{aligned} \dot{\mathbf{z}}(t) &= -(\mathbf{u}(t_f - t) + \mathbf{v}_f(t_f - t, \mathbf{x}(t_f - t)) \\ &= -\mathbf{u}(t_f - t) - \mathbf{v}_f(t_f - t, \mathbf{z}(t)) \\ &= \widetilde{\mathbf{u}}(t) + \widetilde{\mathbf{v}}_f(t, \mathbf{z}(t)) \end{aligned}$$

So $\mathbf{z}$ is solution of the Cauchy problem

$$\begin{cases} \dot{\mathbf{z}}(t) = \widetilde{\mathbf{u}}(t) + \widetilde{\mathbf{v}_f}(t, \mathbf{z}(t)) \\ \mathbf{z}(0) = \mathbf{x}_f \end{cases}$$

Furthermore $\mathbf{z}(\tau) = \mathbf{x}(t_f - \tau) = \mathbf{y}$ which proves $\phi^{\widetilde{\mathbf{u}}(\cdot)}_{0,x_t,\widetilde{\mathbf{v}_f}}(\tau) = \{\mathbf{y}\}$ so $\mathbf{y} \in \mathcal{R}_{0,\mathbf{x}_f,\widetilde{\mathbf{v}_f}}$.

- The reverse inclusion is shown in a same manner by using the mirror operation 2.5 on the control.

□

The latter property ensures that if a method is able to compute forward reachable sets, then the same method can be used to compute backward reachable set with only a simple transformation of the problem data.

2.4 Ground paths

A very intuitive object in navigation problem is the ground path. It corresponds literally to a curve drawn from a given point to a destination on the 'map' of the state space. While ground paths are drawn by the vehicle when it moves, one can also draw a ground path from scratch and ask the vehicle to follow it. But because of the advection of the flow field, any ground path is not necessarily followable by the vehicle. In this section, we define mathematically ground paths, show their space exhibit an equivalence relation (quotient), and then define valid paths for a given navigation problem.

First of all, we give a precise mathematical definition of what we call a ground path

Definition 3 (Ground path). *A* ground path $\mathbf{q}$ *is a continuous mapping* $\mathbf{q} : [a, b] \to \mathbb{R}^2$.

> **Remark**
>
> The intrinsic parameterization of a ground path does not matter. That is, if $\gamma : [a,b] \to [a,b]$ is a warping function[a] then $\mathbf{q}' : h \mapsto \mathbf{q}(\gamma(h))$ describes the same ground path as $\mathbf{q}$.
>
> [a] A *warping function* is a mapping $[a,b] \to [c,d]$ such that $\gamma(a) = c$, $\gamma(b) = d$, γ invertible and γ, γ^{-1} continuous.

The previous remark provides a way to quotient the space of ground paths

Definition 4. *For $\mathbf{q} : [a,b] \to \mathbb{R}^2$, $\mathbf{r} : [c,d] \to \mathbb{R}^2$, we say that $\mathbf{q}(\cdot)$ and $\mathbf{r}(\cdot)$ are equivalent in the sense of ground paths and we note $\mathbf{q}(\cdot) \equiv_\gamma \mathbf{r}(\cdot)$ if there exists $\gamma : [a,b] \to [c,d]$ a warping function such that*

$$\forall t \in [a,b],\ \mathbf{q}(\gamma(t)) = \mathbf{r}(t)$$

Definition 5 (Valid ground path). *A ground path $\mathbf{q} : [0,1] \to \mathbb{R}^2$ is said* valid *for navigation settings $(t_0, \mathbf{x}_0, \mathbf{v}_f)$ if a vehicle starting at $\mathbf{q}(0)$ at time t_0 can entirely follow the path up to $\mathbf{q}(1)$ in some amount of time. That is*

- $\mathbf{q}(0) = \mathbf{x}_0$
- $\exists \tau_u > 0$, $\exists \mathbf{u}(\cdot) \in \mathcal{U}$, $\exists \gamma$ *warping function* $[t_0, t_0 + \tau_u] \to [0,1]$ *such that*

$$\forall \tau \in [0, \tau_u],\ \phi^{\mathbf{u}(\cdot)}_{t_0, \mathbf{x}_0, \mathbf{v}_f}(\tau) = \{\mathbf{q}(\gamma(t_0 + \tau))\}$$

Ground paths will be intensively used in Chapter 7 to develop a performance evaluation for trajectories when the weather scenario changes.

2.5 Time optimal navigation

In the introduction, we identified the Time-optimal Navigation Problem (TNP) problem or Zermelo's navigation problem as being a key problem in long-range routing. When a given location is reachable by the vehicle, the problem consists in finding a trajectory achieving minimal travel time between origin and this location. The TNP is an optimal control problem and we recall its formulation

$$\text{(TNP)}\begin{cases} \min\limits_{\mathbf{u}(\cdot)\in\mathcal{U}} t_f \\ \forall t \geq t_0,\ \dot{\mathbf{x}}(t) = \mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x}(t)) \\ \mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f \end{cases} \tag{2.6}$$

A strong result is that there exist time minimal trajectories to every point $\mathbf{x}$ that is reachable, *i.e.* for all $\mathbf{x} \in \mathcal{R}_{t_0,\mathbf{x}_0}$. This is ensured by a control theory theorem called Filippov's existence theorem because the control law $\mathbf{u}(\cdot)$ takes its values in the closed and bounded set $U = B\left(0, u_{\max}\right)$.

Value functions An important object for optimal control problems is the *value function*, *i.e.* the mapping giving the minimal cost to reach a point in space, when the point is reachable. It is the object at the basis of dynamic programming or Hamilton-Jacobi-Bellman methods. We specify some definitions for varying ways to consider value functions for problem 2.6. In what follows, MTD stands for 'Minimum travel duration'.

Definition 6 (Origin value function)**.** *The origin value function $J : \mathbf{x} \mapsto J(\mathbf{x})$ is defined as the minimum duration to reach a given point in space:*

$$J(\mathbf{x}) := \begin{cases} \textit{MTD from } (t_0, \mathbf{x}_0) \textit{ to } \mathbf{x}, \textit{ if } \mathbf{x} \in \mathcal{R}_{t_0,\mathbf{x}_0} \\ +\infty \textit{ else} \end{cases}$$

Similarly, we define *destination value functions*

Definition 7 (Destination value function)**.** *The destination value function $V : \mathbf{x} \mapsto V(\mathbf{x})$ is defined*

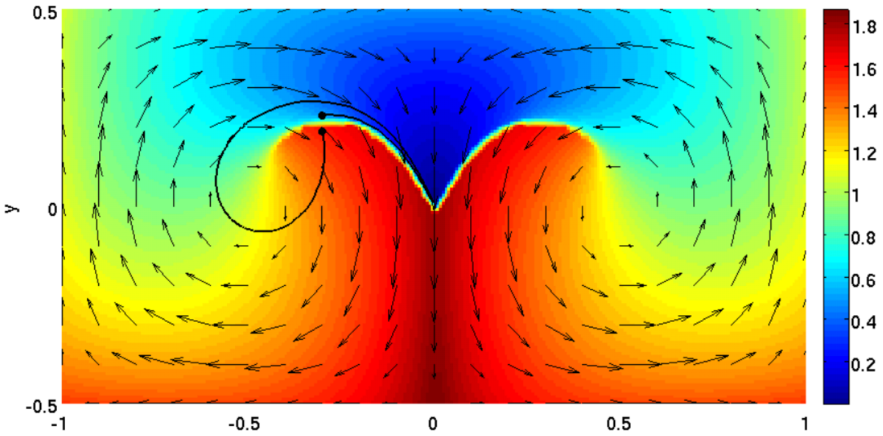


Figure 2.2: Gyre flow field with the destination value function (minimum time to come) from any point in space to the center of the image indicated with the color scale. Two example time-optimal trajectories reaching the center point from two different starting locations are depicted in black. The sharp change of color between red and blue in the center of the image is a numerical approximation of a discontinuity in the value function. The figure is from Rhoads et al. (2010)

.

as the minimum duration to reach the destination at t_f from a given point in space

$$V(\mathbf{x}) := \begin{cases} \textit{MTD from } \mathbf{x} \textit{ to } (t_f, \mathbf{x}_f), \textit{ if } \mathbf{x} \in \mathcal{BR}_{t_f, \mathbf{x}_f} \\ +\infty \textit{ else} \end{cases}$$

For the destination value function, it is also of interest to study the possibility to reach the destination from several time-position pairs; hence, the following definition:

Definition 8 (Multi-time destination value function). *The multi-time destination value function $V : t, \mathbf{x} \mapsto V(t, \mathbf{x})$ is defined as the minimum duration to reach the destination from a given point in space and a given starting time, with a time limit at destination:*

$$V(t, \mathbf{x}) := \begin{cases} \textit{MTD from } (t, \mathbf{x}) \textit{ to } \mathbf{x}_f, \textit{ if } \exists \tilde{t} \leq t_{max},\ \mathbf{x}_f \in \mathcal{R}_{t,\mathbf{x}}(\tilde{t} - t) \\ +\infty \textit{ else} \end{cases}$$

> **Remark**
> We use the same symbol for both functions as they can be discriminated against each other by the number of arguments.

The origin value function answers the question of what duration is required to reach any point in space when starting from a given location and a given time, which is the canonical time-optimal problem answer. In ship navigation, level sets of the origin value function $\{\mathbf{x} \mid J(\mathbf{x}) = c\}$ are called *isochrons* and can be used for optimized routing.

Even if the problem data is smooth, for instance a $\mathcal{C}^\infty$ flow field function, the origin value function can be non-differentiable or even discontinuous. For instance, in Fig. 2.2, a strong gyre flow entails a discontinuity in the destination value function. A consequence is that for very close starting locations, time-optimal trajectories to the center point can be very different near the discontinuity.

Similar to the origin value function, the destination value function gives the minimal travel duration from a point in space to reach the destination at a given date. Since imposing a terminal date can be restrictive in the interpretation of the value function, the multi-time destination value function offers more flexibility in the data it holds because it provides the minimum time to reach the destination from any pair of time-position. This value function is the basis for the development of Hamilton-Jacobi multi-time reachability as presented in Doshi et al. (2022).

2.6 Navigation on the sphere

To finish this chapter, we explain how to write the equations of motions for a LSPOV vehicle in 2D spherical space, and the corresponding TNP in spherical coordinates.

The Earth is approximated by a sphere of radius $R_E = 6.371\,\text{km}$. In this setting, using the

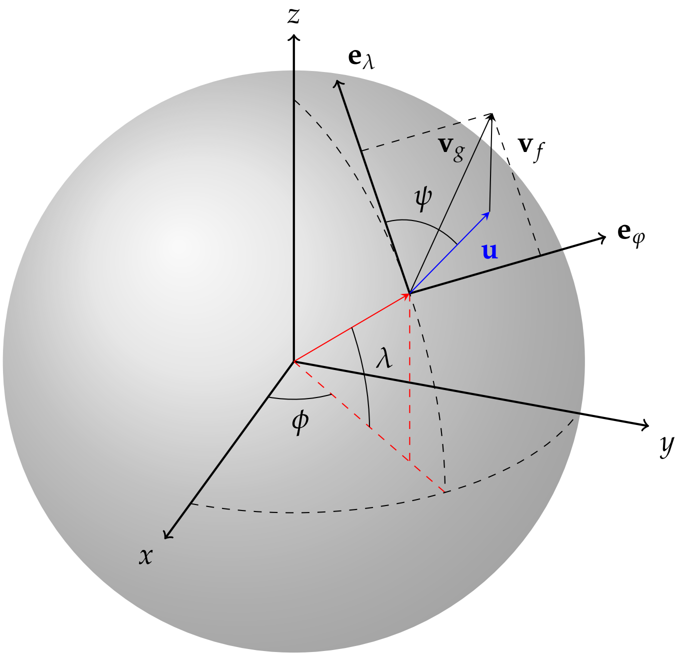


Figure 2.3: Spherical coordinate system

longitude φ and latitude λ variables, the mapping of the lon/lat space to the physical space is

$$\Phi : \ ]-\pi,\pi] \times \left]-\frac{\pi}{2},\frac{\pi}{2}\right[\ \ni \begin{pmatrix} \varphi \\ \lambda \end{pmatrix} \mapsto \begin{pmatrix} R_E \cos\lambda \cos\varphi \\ R_E \cos\lambda \sin\varphi \\ R_E \sin\lambda \end{pmatrix} \in \mathbb{R}^3$$

Note that this corresponds implicitly to projecting the Earth using a plate-carrée (or equirectangular) projection. We exclude the poles so that this mapping is a $\mathcal{C}^\infty$-diffeomorphism. This choice of coordinate system is thus inappropriate to study phenomena happening around the poles. However, for most real world trajectory planning problems, it provides an easy and sufficient way to take the Earth's curvature into account.

At every point on the sphere (φ, λ), we have a local frame tangent to the sphere (see figure 2.3)

$$\left(\mathbf{e}_{\varphi},\,\mathbf{e}_{\lambda}\right) := \left(\left\|\frac{\partial\Phi}{\partial\varphi}\right\|^{-1}\frac{\partial\Phi}{\partial\varphi}, \left\|\frac{\partial\Phi}{\partial\lambda}\right\|^{-1}\frac{\partial\Phi}{\partial\lambda}\right) = \left(\begin{pmatrix} -\sin\varphi \\ \cos\varphi \\ 0 \end{pmatrix}, \begin{pmatrix} -\sin\lambda\cos\varphi \\ -\sin\lambda\sin\varphi \\ \cos\lambda \end{pmatrix}\right).$$

The flow field vector map $\mathbf{v}_f(t,\varphi,\lambda)$ is tangent to the sphere at each point (φ,λ), and it is also the case for the control vector $\mathbf{u}$. The ground velocity is still the sum of the two vectors

$$\mathbf{v}_g = \mathbf{u} + \mathbf{v}_f(t,\varphi,\lambda)$$

> **Remark**
>
> If we note $u_n = \|\mathbf{u}\|$ and write the decomposition $\mathbf{u} = u_n\left(\sin\psi\,\mathbf{e}_{\varphi} + \cos\psi\,\mathbf{e}_{\lambda}\right)$, then ψ corresponds to the usual definition of the heading angle, having $\mathbf{u}$ pointing towards the geographical North when $\psi = 0$ and turning clock-wise when ψ increases.

The velocities $\mathbf{v}_g$, $\mathbf{u}$ and $\mathbf{v}_f$ can be decomposed on the local frame at every point

$$\mathbf{v}_g = v_{g,\varphi}\,\mathbf{e}_{\varphi} + v_{g,\lambda}\,\mathbf{e}_{\lambda}$$

$$\mathbf{u} = u_{\varphi}\,\mathbf{e}_{\varphi} + u_{\lambda}\,\mathbf{e}_{\lambda}$$

$$\mathbf{v}_f = v_{f,\varphi}\,\mathbf{e}_{\varphi} + v_{f,\lambda}\,\mathbf{e}_{\lambda}$$

The components on the φ-axis are commonly called *zonal* components while the ones on the λ-axis are called *meridian* components.

The previous velocity vectors having only two degrees of freedom, we will now represent them using the 2D angular velocity vectors

$$\begin{aligned}
\boldsymbol{\omega}_g &= \left(\omega_{g,\varphi}\ \omega_{g,\lambda}\right)^\intercal := \tfrac{1}{R_E}\left(v_{g,\varphi}\ v_{g,\lambda}\right)^\intercal \\
\boldsymbol{\chi} &= \left(\chi_\varphi\ \chi_\lambda\right)^\intercal := \tfrac{1}{R_E}\left(u_\varphi\ u_\lambda\right)^\intercal \\
\boldsymbol{\omega}_f &= \left(\omega_{f,\varphi}\ \omega_{f,\lambda}\right)^\intercal := \tfrac{1}{R_E}\left(v_{f,\varphi}\ v_{f,\lambda}\right)^\intercal
\end{aligned}$$

If the velocities are provided in m/s, then angular velocities are in rad/s.

To write the dynamics of the vehicle in spherical coordinates, one simply has to remark that $\dot{\lambda}$ is the angular velocity of the vehicle over the current meridian, so:

$$\dot{\lambda} = \omega_{g,\lambda}$$

Similarly, the angular velocity $\dot{\varphi}$ is the velocity of the vehicle around the South-North axis and the vehicle is rotating on a shrunk circle of radius $R_E \cos\lambda$. So:

$$\dot{\varphi} = \frac{\mathbf{v}_g^\intercal\, \mathbf{e}_\varphi}{R_E \cos\lambda} = \frac{\omega_{g,\varphi}}{\cos\lambda}$$

And with the new state vector $\boldsymbol{\xi} := (\varphi\ \ \lambda)^\intercal$, we have the following dynamics:

$$\begin{aligned}
\dot{\boldsymbol{\xi}} &= M(\lambda)\left(\boldsymbol{\chi} + \boldsymbol{\omega}_f\right) \\
M(\lambda) &:= \begin{pmatrix} \frac{1}{\cos\lambda} & 0 \\ 0 & 1 \end{pmatrix}
\end{aligned} \tag{2.7}$$

which is very similar to equation (2.1).

The model is now stated in closed form in spherical coordinates and we can write the *Spherical Time-optimal Navigation Problem* (S-TNP)

$$
\text{(S-TNP)} \begin{cases} \min\limits_{\chi(\cdot)\in\mathcal{U}} \; t_f \\ \dot{\xi} = M(\lambda)\left(\chi + \omega_f\right) \\ \xi(t_0) = \xi_0, \quad \xi(t_f) = \xi_f \end{cases} \tag{2.8}
$$

The methods applied to solve problem (2.6) can also be applied to the latter problem to solve for exact trajectories in spherical coordinates.

Conclusion

This chapter defined tools to study trajectory optimization or reachability for vehicles in flow fields with the LSPOV abstraction model. With these tools defined, we are now ready to address the problem of computing time-optimal trajectories for navigation problems in arbitrary flow fields, which is the focus of the following chapter.

Chapter's main questions – Answers

- Mathematically, what are the properties of the large-scale point of view model adopted for long-range trajectory planning?

  The LSPOV model captures the kinematic behavior of a vehicle immersed in a flow field. It enables a time- and space- continuous modelling of trajectories. It captures the local loss of controllability in strong flow, with excluded directions of motion.

- Is the large-scale point of view adaptable to spherical coordinates?

  In longitude/latitude coordinates, the LSPOV model differs from the planar model only by a multiplicative matrix in the kinematics.

Chapter 3

Time-optimal navigation problem: Algorithms

Abstract

In this chapter, necessary conditions of optimality for time-optimal trajectories solving Zermelo's navigation problem in the presence of an unsteady and possible strong flow field are given by using Pontryagin's Maximum Principle. Three equivalent augmented ordinary differential equations defining the trajectories candidate to optimality (extremals) are given and commented against their use in the literature. Similar optimality conditions are written in spherical coordinates, showing few differences with the planar case. An algorithm based on the resampling of extremals and similar to what is found in the literature is detailed and studied. The origin resampling scheme is shown to reach its limit in a real world example featuring high winds because it requires too much precision on the initial condition. An improvement of the latter algorithm is proposed, with resampling happening along extremals rather than on the initial condition. It manages to solve the real world case where the first algorithm failed. Finally, some possible improvements for the algorithms are suggested to improve the computation efficiency.

Résumé en français

Dans ce chapitre, les conditions nécessaires d'optimalité pour les trajectoires optimales en temps résolvant le problème de navigation de Zermelo dans un écoulement fort et instationnaire sont données en utilisant le principe du maximum de Pontryagin. Trois équations différentielles ordinaires augmentées équivalentes définissant les trajectoires candidates à l'optimalité (extrémales) sont données et commentées par rapport à leur utilisation dans la littérature. Des conditions d'optimalité similaires sont écrites en coordonnées sphériques, montrant peu de différences avec le cas plan. Un algorithme basé sur le rééchantillonnage des extrémales et similaire à ce que l'on trouve dans la littérature est détaillé et étudié. Il est démontré que le principe de rééchantillonnage à l'origine atteint ses limites dans un exemple réel comportant des vents fort, parce qu'il exige trop de précision sur la condition initiale. Une amélioration de ce dernier algorithme est proposée, le rééchantillonnage se faisant le long des extrémités plutôt que sur la condition initiale. Il parvient à résoudre le cas réel où le premier algorithme a échoué. Enfin, des pistes d'amélioration sont suggérées afin d'améliorer l'efficacité des algorithmes décrits.

Contents

The previous chapter described the tools to study optimal navigation problems both in planar space and spherical space. We now provide computer methods to actually *solve* the navigation problem and compute the different quantities of interest such as trajectories and reachability sets.

The following development will be made in planar space for simplicity. But the results easily generalize to spherical coordinates.

The previous chapter elaboration on a continuous mathematical model for navigation problems paved the way for the use of indirect methods. Indeed, in this chapter, we present algorithms belonging to the 'Extremal methods' category of the literature. Extremal computation for the time-optimal navigation of vehicles in flow fields has been studied in the past, whether for underwater vehicles (Rhoads et al., 2013; Wang et al., 2016; Bijlsma, 1975) or for aircraft (Bijlsma, 2009; Sridhar et al., 2011; Marchidan and Bakolas, 2016), but it is in general less popular than other methods such as front propagation or direct methods, because screening extremals can be difficult in strong flow field environment. Nevertheless, these extremals are helpful to get a fine understanding of why some trajectories are optimal, as their construction results from applying necessary conditions of optimality.

Chapter's main questions

- How to compute efficiently trajectories that minimize the travel time in the long-range navigation problem?
- Can we provide guarantees that a given trajectory is indeed a global optimum of the problem?
- What are the typical shapes of time-minimizing trajectories?

3.1 Extremal trajectories

Pontryagin's Maximum Principle (PMP) (Boltyanskiy et al., 1962) provides necessary conditions of optimality that a trajectory must satisfy to be a solution of an optimal control problem. Trajectories satisfying the PMP conditions are subsequently called *extremals*. They are candidates to optimality, in the sense that if a solution trajectory exists for the optimal control at stake, then it is a particular extremal of the problem. We are going to build extremals (shooting method) for the TNP (2.6) and look for the solution to our problem in the set of all extremals.

We first define the Hamiltonian of problem (2.6) as

$$\begin{aligned} H \ : \ \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R} \quad &\to \quad \mathbb{R} \\ t, \mathbf{x}, \mathbf{p}, \mathbf{u}, \lambda \qquad\qquad &\mapsto \quad \mathbf{p}^\intercal \left(\mathbf{u} + \mathbf{v}_f(t, \mathbf{x})\right) + \lambda \end{aligned} \tag{3.1}$$

In the previous equation, a new vector $\mathbf{p}$ called the *costate* vector was introduced. It is a virtual state, a mathematical tool that will help write the characterization of optimality. $\lambda = \lambda \times 1$ is a multiplier for the marginal cost of (2.6), which is 1 as we do time optimality. It does not vary with time. In the canonical PMP, it comes with a minus sign in the Hamiltonian. But in our case, we are *minimizing* a cost, so it comes with a plus sign (it is equivalent to say that we maximize the opposite of total travel time).

To apply the PMP, we need to assume that the flow field is piece-wise $\mathcal{C}^1$.

The costate $\mathbf{p}$ is an absolutely continuous function of time which is characterized by its evolution in time, given as:

$$\frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -\left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t, \mathbf{x}(t)}^\intercal \mathbf{p} \tag{3.2}$$

The PMP states NCOs for the solution trajectory of control problems. It states that if $t \mapsto (\mathbf{x}^*(t), \mathbf{p}^*(t), \mathbf{u}^*(t), \lambda^*)$ is an optimal trajectory, then:

1. (*Point-wise minimization*)

   $$\text{For a.e. } t \in [0, T], \quad \mathbf{u}^*(t) \in \operatorname{argmin}_{\mathbf{u} \in U} H(t, \mathbf{x}^*(t), \mathbf{p}^*(t), \mathbf{u}, \lambda^*)$$

2. (*Costate transversality condition*) $\mathbf{p}(T)$ is orthonormal to the tangent space to the target set at $\mathbf{x}(T)$

3. (*Free final time transversality condition*)

   $$\min_{\mathbf{u} \in U} H(T, \mathbf{x}^*(T), \mathbf{p}^*(T), \mathbf{u}, \lambda^*) = 0$$

The first important result is that in our case, (1) leads to an explicit link between the optimal control and the value of the costate.

Property 3.1: NCOs for the TNP

The necessary optimality conditions for the solutions of the TNP write

$$\text{for a.e. } t \in [0,T], \quad \mathbf{u}^*(t) = -u_{\max} \tfrac{\mathbf{p}^*(t)}{\|\mathbf{p}^*(t)\|} \tag{3.3}$$

Proof. $H(t, \mathbf{x}, \mathbf{p}, \mathbf{u}, \lambda) = \mathbf{p}^\intercal \mathbf{u} + \mathbf{p}^\intercal \mathbf{v}_f(t, \mathbf{x}) + \lambda$ is minimal in $\mathbf{u}$ when $\mathbf{u}$ is opposite to the costate $\mathbf{p}$ with maximal norm. □

Property 3.1 is interesting in the sense that the argmin extraction operation is expected to be computationally intensive in non-linear optimal control problems. But in this setting, we have an analytical link between costate and optimal control which paves the way for computational efficiency. If we consider the system to be defined not only by its position vector $\mathbf{x}$ but also by its costate vector $\mathbf{p}$, then the previous property turns the control system into a closed-loop system, which one we will study in Section 3.1.1. But before moving on, we need to see what conditions (2) and (3) imply for extremals.

As regards the transversality condition (2), it is void since the TNP 2.6 has a point as target set, so the tangent space is $\{0\}$ which admits $\mathbb{R}^2$ as orthogonal space.

Finally, the final time transversality condition shall be considered. We have:

$$\min_{\mathbf{u} \in U} H(T, \mathbf{x}^*(T), \mathbf{p}^*(T), \mathbf{u}, \lambda^*) = \mathbf{p}^*(T)^\intercal \left(-v_r \frac{\mathbf{p}^*(T)}{\|\mathbf{p}^*(T)\|} + \mathbf{v}_f(T, \mathbf{x}_f) \right) + \lambda^*$$

Because the PMP is invariant by scaling of $(\mathbf{p}^*(\cdot), \lambda^*)$, we can suppose that $\lambda^* = 1$. Condition (3) thus writes:

$$\mathbf{p}^*(T)^\intercal \left(-v_r \frac{\mathbf{p}^*(T)}{\|\mathbf{p}^*(T)\|} + \mathbf{v}_f(T, \mathbf{x}_f) \right) = -1$$

which we can rewrite:

$$\|\mathbf{p}^*(T)\| \left(\frac{\mathbf{p}^*(T)^\intercal}{\|\mathbf{p}^*(T)\|} \mathbf{v}_f(T, \mathbf{x}_f) - v_r \right) = -1$$

This necessary condition links the costate norm and direction at final time. Because it satisfies the ODE 3.2, under sufficient smoothness assumption, the costate is entirely defined by specifying a particular value $\mathbf{p}(t_1) = \mathbf{p}_1$. So, we could try to use the information of condition (3) to set the costate vector at final time and integrate backwards to the origin the extremal of interest. But this means we have to pick a final time T arbitrarily to do so. So, in practice we would have to integrate from several final times to try to find the optimum one. This is what is done in Rhoads et al. (2010).

In what follows, we will prefer to set the costate vector using an initial condition. Because it will be the basis to build efficient algorithms, and also because Property 3.2 shows that the costate magnitude does not matter, *i.e.* $\|\mathbf{p}\|$ can be scaled by any positive factor without changing the trajectory in the physical space. We will ensure the trajectory we find is globally optimal by screening exhaustively into the set of all extremals.

3.1.1 Extremal system, costate form

We define an augmented state by stacking the state and the costate vector $\mathbf{y} = (\mathbf{x}\ \mathbf{p})^\intercal$. We gather ODEs in a single extremal system that writes:

$$\left\{\begin{aligned} \dot{\mathbf{y}}(t) &= \begin{pmatrix} -u_{\max}\frac{\mathbf{p}(t)}{\|\mathbf{p}(t)\|} + \mathbf{v}_f(t, \mathbf{x}(t)) \\ -\left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}(t)}^\intercal \mathbf{p}(t) \end{pmatrix} \\ \mathbf{y}(0) = \mathbf{y}_0 &= \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{p}_0 \end{pmatrix} \end{aligned}\right. \tag{3.4}$$

We will call trajectories $\mathbf{y}(\cdot)$ *extremals* as well, in a small abuse of terminology. Then, we notice that this system exhibits a special property about the costate vector.

Property 3.2

Extremal system 3.4 produces the same $\mathbf{x}$-trajectories if the initial condition $\mathbf{p}_0$ is replaced by $\alpha \mathbf{p}_0$ with $\alpha > 0$.

Proof. Let $\alpha > 0$ and let $\mathbf{y}(\cdot) = (\mathbf{x}(\cdot), \mathbf{p}(\cdot))$ be the trajectory integrated from 3.4 with initial condition $(\mathbf{x}_0, \mathbf{p}_0)$. Define $\mathbf{p}'(\cdot) := \alpha \mathbf{p}(\cdot)$ and $\mathbf{y}'(\cdot) := (\mathbf{x}(\cdot), \mathbf{p}'(\cdot))$. Observe that $\frac{\mathbf{p}(t)}{\|\mathbf{p}(t)\|} = \frac{\alpha \mathbf{p}(t)}{\|\alpha \mathbf{p}(t)\|}$, so

$$\dot{\mathbf{x}}(t) = -u_{\max} \frac{\mathbf{p}(t)}{\|\mathbf{p}(t)\|} + \mathbf{v}_f(t, \mathbf{x}(t)) = -u_{\max} \frac{\mathbf{p}'(t)}{\|\mathbf{p}'(t)\|} + \mathbf{v}_f(t, \mathbf{x}(t))$$

Then

$$\dot{\mathbf{p}'}(t) = \alpha \dot{\mathbf{p}}(t) = -\alpha \frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\bigg|_{t, \mathbf{x}(t)}^{\intercal} \mathbf{p}(t) = -\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\bigg|_{t, \mathbf{x}(t)}^{\intercal} \mathbf{p}'(t)$$

So $\mathbf{y}'$ satisfies the ODE of system 3.4, and furthermore we have

$$\mathbf{y}'(0) = (\mathbf{x}_0, \alpha \mathbf{p}_0)$$

The Cauchy problem 3.4 being well posed (Lipschitz dynamics), solutions are unique for given initial conditions and we deduce that $\mathbf{y}'$ is the trajectory from system 3.4 obtained for initial conditions $(\mathbf{x}_0, \mathbf{p}_0')$ where $\mathbf{p}_0' = \alpha \mathbf{p}_0$. This proves that the $\mathbf{x}$ part of $\mathbf{y}$ trajectories integrated from system 3.4 does not change when the initial condition on the costate is multiplied by a positive factor.

□

The previous property shows that the extremal system 3.4 is over-determined, in the sense that there are too many dimensions among the four dimensions. Hence, other equivalent, lower-dimensional systems of ODE can capture the dynamics of extremals in the physical space, as will be shown in the following sections.

3.1.2 Extremal system, heading vector form

The extremal system 3.4 can also be formulated using the heading vector $\mathbf{h}$ that belongs to the unit circle and thus has only one degree of freedom as opposed to the costate vector.

$$\left\{\begin{array}{l}\begin{pmatrix}\dot{\mathbf{x}}(t)\\ \dot{\mathbf{h}}(t)\end{pmatrix}=\begin{pmatrix}u_{\max}\,\mathbf{h}(t)+\mathbf{v}_f(t,\mathbf{x}(t))\\ -\left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}(t)}^{\intercal}\mathbf{h}(t)+\left(\mathbf{h}(t)^{\intercal}\left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}(t)}\mathbf{h}(t)\right)\mathbf{h}(t)\end{pmatrix}\\ \begin{pmatrix}\mathbf{x}(0)\\ \mathbf{h}(0)\end{pmatrix}=\begin{pmatrix}\mathbf{x}_0\\ \mathbf{h}_0\end{pmatrix}\end{array}\right. \tag{3.5}$$

If we note $\mathbf{h} := -\frac{\mathbf{p}(t)}{\|\mathbf{p}(t)\|}$ the heading vector, then Cauchy problem 3.4 is equivalent to the problem 3.5. We prove this in Property 3.3.

3.1.3 Extremal system, heading angle form

The heading vector $\mathbf{h}$ always belongs to $\mathbb{S}^1$, so in reality its only degree of freedom is the heading angle θ such that:

$$\mathbf{h} := \begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix}$$

It is possible to write the extremal system only using the heading angle θ variable:

$$\left\{\begin{array}{l}\begin{pmatrix}\dot{\mathbf{x}}(t)\\ \dot{\theta}(t)\end{pmatrix}=\begin{pmatrix}u_{\max}\,\mathbf{h}(t)+\mathbf{v}_f(t,\mathbf{x}(t))\\ -\frac{\partial v_{f,1}}{\partial x_2}\cos^2\theta+\left(\frac{\partial v_{f,1}}{\partial x_1}-\frac{\partial v_{f,2}}{\partial x_2}\right)\cos\theta\sin\theta+\frac{\partial v_{f,2}}{\partial x_1}\sin^2\theta\end{pmatrix}\\ \begin{pmatrix}\mathbf{x}(0)\\ \theta(0)\end{pmatrix}=\begin{pmatrix}\mathbf{x}_0\\ \theta_0\end{pmatrix}\end{array}\right. \tag{3.6}$$

This characterization of extremals using the heading angle appears in Rhoads et al. 2013, Eq. 5 or Sridhar et al. 2011, Eq. 15 but also in chemistry for reaction fronts Megson et al. 2015, Eq. 1b or Mahoney et al. 2012, Eq. 2b.

Property 3.3

The $\mathbf{x}$-trajectories of 3.4, 3.5 and 3.6 are the same when values θ_0, $\mathbf{h}_0$, $\mathbf{p}_0$ satisfy

$$\mathbf{h}_0 = -\frac{\mathbf{p}_0}{\|\mathbf{p}_0\|} = \begin{pmatrix} \cos\theta_0 \\ \sin\theta_0 \end{pmatrix}$$

Proof.

- We first show that the ODE on $\mathbf{p}(\cdot)$ and $\mathbf{h}(\cdot)$ are equivalent.

We have

$$\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{p}\| = \frac{\dot{\mathbf{p}}^\intercal \mathbf{p}}{\|\mathbf{p}\|}$$

so

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathbf{p}}{\|\mathbf{p}\|}\right) &= \frac{\|\mathbf{p}\|\,\dot{\mathbf{p}} - \frac{\dot{\mathbf{p}}^\intercal \mathbf{p}}{\|\mathbf{p}\|}\mathbf{p}}{\|\mathbf{p}\|^2} \\
&= \frac{\dot{\mathbf{p}}}{\|\mathbf{p}\|} - \left[\left(\frac{\dot{\mathbf{p}}}{\|\mathbf{p}\|}\right)^\intercal \frac{\mathbf{p}}{\|\mathbf{p}\|}\right]\frac{\mathbf{p}}{\|\mathbf{p}\|} \\
&= \frac{-\left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}(t)}^\intercal \mathbf{p}}{\|\mathbf{p}\|} - \left[\left(\frac{-\left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}(t)}^\intercal \mathbf{p}}{\|\mathbf{p}\|}\right)^\intercal \frac{\mathbf{p}}{\|\mathbf{p}\|}\right]\frac{\mathbf{p}}{\|\mathbf{p}\|}
\end{aligned}$$

with $\mathbf{h} = -\frac{\mathbf{p}}{\|\mathbf{p}\|}$ we then have:

$$\dot{\mathbf{h}}(t) = -\left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}(t)}^\intercal \mathbf{h} + \left(\mathbf{h}^\intercal \left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}(t)} \mathbf{h}\right)\mathbf{h} \tag{A}$$

which shows that the new ODE on $\mathbf{h}(\cdot)$ is just a reformulation of the ODE on $\mathbf{p}(\cdot)$.

- Then, writing $\mathbf{h} = (\cos\theta \;\; \sin\theta)^\intercal$, and using $k = \mathbf{h}^\intercal \frac{\partial \mathbf{v}_f}{\partial \mathbf{x}} \mathbf{h}$, equation (A) now writes

$$\begin{cases} -\dot{\theta}\sin\theta &= -\dfrac{\partial v_{f,1}}{\partial x_1}\cos\theta - \dfrac{\partial v_{f,2}}{\partial x_1}\sin\theta - k\cos\theta \\ \dot{\theta}\cos\theta &= -\dfrac{\partial v_{f,1}}{\partial x_2}\cos\theta - \dfrac{\partial v_{f,2}}{\partial x_2}\sin\theta - k\sin\theta \end{cases}$$

Thus, multiplying line 1 with $(-\sin(\theta))$ and line 2 with $\cos\theta$, and adding it together, the left-hand side is just $\dot{\theta}$ and the term in factor of k vanishes, so that:

$$\dot{\theta} = -\frac{\partial v_{f,1}}{\partial x_2}\cos^2\theta + \left(\frac{\partial v_{f,1}}{\partial x_1} - \frac{\partial v_{f,2}}{\partial x_2}\right)\cos\theta\sin\theta + \frac{\partial v_{f,2}}{\partial x_1}\sin^2\theta$$

□

Linear flow field

One of the most canonical example of a Zermelo problem is navigating in a linear flow field that is aligned with the axes. The flow field writes:

$$\mathbf{v}_f(\mathbf{x}) := \begin{pmatrix} 0 & a \\ 0 & 0 \end{pmatrix} \mathbf{x} \tag{3.7}$$

with $a = 1$ in what follows. The navigation problem is:

$$\begin{cases} \min\ \tau \\ \dot{\mathbf{x}} = \mathbf{u} + \mathbf{v}_f(\mathbf{x}) \\ \mathbf{x}(0) = (0, 0) \\ \mathbf{x}(\tau) = \mathbf{x}_f \end{cases}$$

We display some extremal trajectories for this problem in Fig. 3.1, with the one reaching the target $\mathbf{x}_f = (1, 0)$ in red[1].

[1] The code can be found in the notebook `linear_extremals.ipynb` at DOI 10.5281/zenodo.13939206.

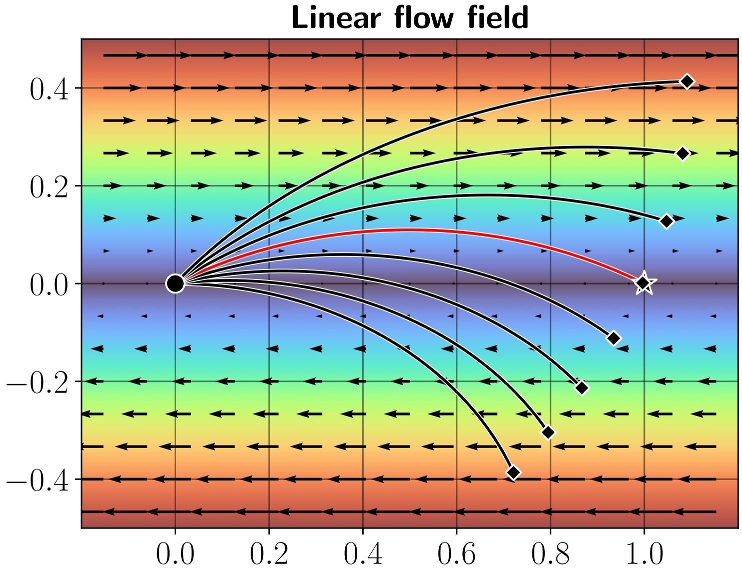


Figure 3.1: Time-optimal extremals in the linear flow field. The extremals are integrated for different initial heading values.

For the considered problem, there exist an analytical solution for extremals that is given for instance in Girardet (2014), Appendix F. We write the expression and adapt it to the fact that in our case we start from $(0,0)$ and want to reach a specified target $\mathbf{x}_f$ (the opposite in Girardet's thesis).

$$\boldsymbol{\xi}_{\theta_0}(\theta) := \begin{pmatrix} \frac{1}{2}\frac{v_r}{a}\left(\frac{-1}{\cos\theta_0}(\tan\theta_0 - \tan\theta) + \tan\theta\left(\frac{1}{\cos\theta_0} - \frac{1}{\cos\theta}\right) - \ln\frac{\tan\theta_0 + \frac{1}{\cos\theta_0}}{\tan\theta + \frac{1}{\cos\theta}}\right) \\ \frac{v_r}{a}\left(\frac{1}{\cos\theta} - \frac{1}{\cos\theta_0}\right) \end{pmatrix}$$

where θ is both a parameter of the path and the heading angle of the vehicle. This expression satisfies $\boldsymbol{\xi}_{\theta_0}(\theta_0) = (0,0)$. To find the optimal trajectory to $\mathbf{x}_f$, one has to find a θ_0 for which equation $\boldsymbol{\xi}_{\theta_0}(\theta) = \mathbf{x}_f$ admits a solution $\theta_f \leq \theta_0$. When found, say $\theta_0^\star$, the optimal path is then $\{\boldsymbol{\xi}_{\theta_0}(\theta) \mid \theta \in [\theta_f, \theta_0]\}$. For other analytical resolutions of the TNP, see Techy (2011); Bonnard et al. (2021).

3.2 Extremals in spherical coordinates

We can write a Hamiltonian for the optimal control problem 2.8 where we note $\boldsymbol{\pi} := (\pi_\varphi \;\; \pi_\lambda)^\intercal$ the costate vector and π_0 the cost multiplier:

$$H(\boldsymbol{\xi}, \boldsymbol{\pi}, \boldsymbol{\chi}, \pi_0) = \boldsymbol{\pi}^\intercal M(\lambda) \left(\boldsymbol{\chi} + \boldsymbol{\omega}_f\right) + \pi_0$$

To write the costate evolution, we first need to differentiate the dynamics of the problem. We set the dynamics:

$$f(t, \varphi, \lambda) := M(\lambda) \left(\boldsymbol{\chi} + \boldsymbol{\omega}_f(t, \varphi, \lambda)\right)$$

We have:

$$\begin{aligned}
\frac{\partial f}{\partial \varphi} &= M(\lambda) \frac{\partial \boldsymbol{\omega}_f}{\partial \varphi} = \begin{pmatrix} \frac{1}{\cos\lambda} \frac{\partial \omega_{f,\varphi}}{\partial \varphi} \\ \frac{\partial \omega_{f,\lambda}}{\partial \varphi} \end{pmatrix} \\
\frac{\partial f}{\partial \lambda} &= M(\lambda) \frac{\partial \boldsymbol{\omega}_f}{\partial \lambda} + \frac{\mathrm{d}M}{\mathrm{d}\lambda}(\lambda) \begin{pmatrix} \chi_\varphi + \omega_{f,\varphi} \\ \chi_\lambda + \omega_{f,\lambda} \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{\cos\lambda} \frac{\partial \omega_{f,\varphi}}{\partial \lambda} \\ \frac{\partial \omega_{f,\lambda}}{\partial \lambda} \end{pmatrix} + \begin{pmatrix} \frac{\tan\lambda}{\cos\lambda}(\chi_\varphi + \omega_{f,\varphi}) \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{\cos\lambda} \frac{\partial \omega_{f,\varphi}}{\partial \lambda} + \frac{\tan\lambda}{\cos\lambda}(\chi_\varphi + \omega_{f,\varphi}) \\ \frac{\partial \omega_{f,\lambda}}{\partial \lambda} \end{pmatrix}
\end{aligned}$$

The evolution $\dot{\boldsymbol{\pi}} = -\begin{pmatrix} \frac{\partial f}{\partial \varphi} & \frac{\partial f}{\partial \lambda} \end{pmatrix}^\intercal \boldsymbol{\pi}$ thus writes:

$$\begin{pmatrix} \dot{\pi}_\varphi \\ \dot{\pi}_\lambda \end{pmatrix} = -\begin{pmatrix} \frac{1}{\cos\lambda} \frac{\partial \omega_{f,\varphi}}{\partial \varphi} & \frac{\partial \omega_{f,\lambda}}{\partial \varphi} \\ \frac{1}{\cos\lambda} \frac{\partial \omega_{f,\varphi}}{\partial \lambda} + \frac{\tan\lambda}{\cos\lambda}(\chi_\varphi + \omega_{f,\varphi}) & \frac{\partial \omega_{f,\lambda}}{\partial \lambda} \end{pmatrix} \begin{pmatrix} \pi_\varphi \\ \pi_\lambda \end{pmatrix}$$

The Hamiltonian is differentiable w.r.t. the control variable ψ and:

$$\frac{\partial H}{\partial \psi} = \boldsymbol{\pi}^\intercal M(\lambda) \begin{pmatrix} -v_a \cos\psi \\ v_a \sin\psi \end{pmatrix}$$

$$\frac{\partial^2 H}{\partial \psi^2} = \boldsymbol{\pi}^\intercal M(\lambda) \begin{pmatrix} -v_a \sin\psi \\ -v_a \cos\psi \end{pmatrix}$$

Thus, to have a minimum in ψ, *i.e.* $\frac{\partial H}{\partial \psi} = 0$ and $\frac{\partial^2 H}{\partial \psi^2} \geq 0$, the control $\mathbf{s} = \begin{pmatrix} \sin\psi \\ \cos\psi \end{pmatrix}$ shall be:

$$\mathbf{s} = -\frac{M(\lambda)\boldsymbol{\pi}}{\|M(\lambda)\boldsymbol{\pi}\|}$$

which is very similar to the planar case, with a correction matrix to take spherical geometry into account. We thus have a closed form for the optimal control and as in the planar case, we can shoot extremals to solve the OCP.

3.3 Algorithms

In this section, we assume the problem is set so that $t_0 = 0$ for simplicity. We formalize algorithm ideas to solve the TNP (2.6). More precisely, we want to build the optimal trajectory from the construction of extremals, since Pontryagin's Maximum Principle states that the optimal trajectory is an extremal.

We will consider the dynamics function:

$$\begin{array}{cccc} F: & \mathbb{R} \times \mathbb{R}^4 & \to & \mathbb{R}^4 \\ & (t, \mathbf{y}) = (t, \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix}) & \mapsto & \begin{pmatrix} -u_{\max} \frac{\mathbf{p}}{\|\mathbf{p}\|} + \mathbf{v}_f(t, \mathbf{x}) \\ -\left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}}^\intercal \mathbf{p} \end{pmatrix} \end{array}$$

which comes from the extremal system in costate form 3.4, and solve Initial Value Problems

(IVPs) of the form:

$$\begin{cases} \dot{\mathbf{y}} = F(t, \mathbf{y}) \\ \mathbf{y}(0) = \mathbf{y}_0 \end{cases} \tag{3.8}$$

The function F is not defined for $\mathbf{p} = 0$. But fortunately, solutions of Eq. 3.8 will never have a vanishing costate. Indeed, if one has for some t_1 that $\mathbf{p}(t_1) = 0$, then by forward and backward integration of the costate ODE, one has $\forall t,\ \mathbf{p}(t) = 0$. So, this never happens if the costate is initialized to a non-zero value. Thus, we will assume in what follows that we always work with costate values away from zero by a safety margin $p_{\min}$:

$$\|\mathbf{p}\| \geq p_{\min} > 0$$

Similarly, we are always going to work on a finite time horizon. We will assume the costate vector never diverges in finite time, *i.e.* there exists $p_{\max}$ such that:

$$\|\mathbf{p}\| \leq p_{\max}$$

We assume the flow field is $\mathcal{C}^2$ over the domain $\mathcal{D}$. Because the domain is compact (closed and bounded), the flow field's Lipschitz constant is well defined and we note:

$$L_1 = \sup_{\substack{t \in [0,T] \\ \mathbf{x} \in \mathcal{D}}} \left\| \left. \frac{\partial \mathbf{v}_f}{\partial \mathbf{x}} \right|_{t,\mathbf{x}} \right\|_{\text{Frob}}$$

with the Frobenius norm:

$$\left\| \left. \frac{\partial \mathbf{v}_f}{\partial \mathbf{x}} \right|_{t,\mathbf{x}} \right\|_{\text{Frob}} = \sqrt{\sum_{\substack{i=1,2 \\ j=1,2}} \left(\left. \frac{\partial v_{f,i}}{\partial x_j} \right|_{t,\mathbf{x}} \right)^2}$$

And similarly for the second derivative, the following Lipschitz constant is well defined:

$$L_2 = \sup_{\substack{t\in[0,T]\\ \mathbf{x}\in\mathcal{D}}} \left\| \left.\frac{\partial^2 \mathbf{v}_f}{\partial \mathbf{x}^2}\right|_{t,\mathbf{x}} \right\|_{\mathrm{Frob}_2}$$

with

$$\left\| \left.\frac{\partial^2 \mathbf{v}_f}{\partial \mathbf{x}^2}\right|_{t,\mathbf{x}} \right\|_{\mathrm{Frob}_2} = \sqrt{\sum_{\substack{i=1,2\\ j=1,2\\ k=1,2}} \left(\left.\frac{\partial^2 v_{f,i}}{\partial x_j \partial x_k}\right|_{t,\mathbf{x}} \right)^2}$$

With these assumptions, we can prove IVP 3.8 is well posed in the sense that function F is Lipschitz continuous.

Property 3.4: Augmented dynamics Lipschitz continuity

F is Lipschitz with constant

$$L_F = \sqrt{\left(L_1 + u_{\max} p_{\min}^{-1}\right)^2 + \left(L_1 + L_2\, p_{\max}\right)^2}$$

Proof. We first recall that

$$\|\mathbf{y}\| = \left\| \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix} \right\| = \sqrt{\|\mathbf{x}\|^2 + \|\mathbf{p}\|^2}$$

and

$$\max\left\{\|\mathbf{x}\|,\ \|\mathbf{p}\|\right\} \leq \|\mathbf{y}\|$$

Using the following notation for the normalizing function:

$$\nu\ :\ \mathbf{p} \mapsto \frac{\mathbf{p}}{\|\mathbf{p}\|}$$

we have:

$$F(t,\mathbf{y}_2) - F(t,\mathbf{y}_1) = \begin{pmatrix} \mathbf{v}_f(t,\mathbf{x}_2) - \mathbf{v}_f(t,\mathbf{x}_1) - u_{\max}\left(\nu(\mathbf{p}_2) - \nu(\mathbf{p}_1)\right) \\ -\left.\dfrac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}_2}^{\intercal} \mathbf{p}_2 + \left.\dfrac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}_1}^{\intercal} \mathbf{p}_1 \end{pmatrix}$$

Then,

$$\begin{aligned} \left\|\mathbf{v}_f(t,\mathbf{x}_2) - \mathbf{v}_f(t,\mathbf{x}_1) - u_{\max}\left(\nu(\mathbf{p}_2) - \nu(\mathbf{p}_1)\right)\right\| &\leq L_1 \left\|\mathbf{x}_2 - \mathbf{x}_1\right\| + u_{\max} p_{\min}^{-1} \left\|\mathbf{p}_2 - \mathbf{p}_1\right\| \\ &\leq \left(L_1 + u_{\max} p_{\min}^{-1}\right) \left\|\mathbf{y}_2 - \mathbf{y}_1\right\| \end{aligned}$$

and

$$\begin{aligned} \left\| -\left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}_2}^{\intercal} \mathbf{p}_2 + \left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}_1}^{\intercal} \mathbf{p}_1 \right\| &= \left\| \left(\left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}_1}^{\intercal} - \left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}_2}^{\intercal} \right) \mathbf{p}_2 + \left.\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}_1}^{\intercal} (\mathbf{p}_1 - \mathbf{p}_2) \right\| \\ &\leq L_2 \left\|\mathbf{x}_2 - \mathbf{x}_1\right\| \left\|\mathbf{p}_2\right\| + L_1 \left\|\mathbf{p}_2 - \mathbf{p}_1\right\| \\ &\leq \left(L_2\, p_{\max} + L_1\right) \left\|\mathbf{y}_2 - \mathbf{y}_1\right\| \end{aligned}$$

Thus,

$$\begin{aligned} \left\|F(t,\mathbf{y}_2) - F(t,\mathbf{y}_1)\right\| &\leq \sqrt{\left(\left(L_1 + u_{\max} p_{\min}^{-1}\right) \left\|\mathbf{y}_2 - \mathbf{y}_1\right\|\right)^2 + \left(\left(L_2\, p_{\max} + L_1\right) \left\|\mathbf{y}_2 - \mathbf{y}_1\right\|\right)^2} \\ &\leq L_F \left\|\mathbf{y}_2 - \mathbf{y}_1\right\| \end{aligned}$$

with

$$L_F = \sqrt{\left(L_1 + u_{\max} p_{\min}^{-1}\right)^2 + \left(L_1 + L_2\, p_{\max}\right)^2}$$

□

Notation 1. *If* $\mathbf{y}(t) = \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{p}(t) \end{pmatrix}$ *is a solution of extremal system 3.4 with initial condition* $\mathbf{y}(0) = \mathbf{y}_0$*, we note:*

- $\phi_t\left(\mathbf{y}_0\right) := \mathbf{y}(t)$ *the flow of system 3.4*
- $\phi_t^{\mathbf{x}}\left(\mathbf{y}_0\right) := \mathbf{x}(t)$ *the* $\mathbf{x}$*-component (first two components) of the flow of system 3.4*

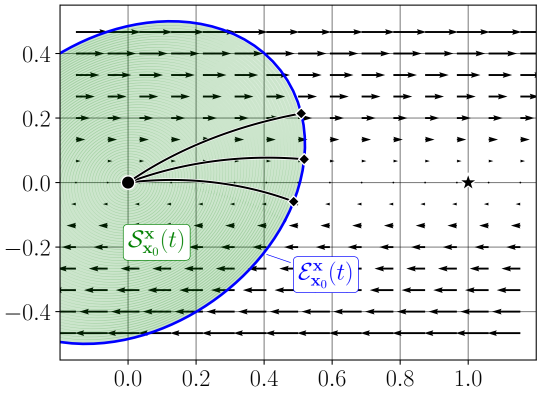


Figure 3.2: An example of an extremal set in blue and an extremal surface in green for the linear flow field case 3.7. Three extremals are also depicted.

Notation 2. *When $\mathbf{x}_0$ is fixed in a given context, we may note $\phi_t(\mathbf{y}_0) = \phi_t(\mathbf{p}_0)$ to emphasize that the only dependency left is on the initial costate value.*

Definition 9 (Extremal set)**.** *The set of all extremals initialized in the state space at $\mathbf{x}_0$ and evaluated at time t is noted:*

$$\mathcal{E}_{\mathbf{x}_0}(t) := \bigcup_{\mathbf{p}_0 \in \mathbb{R}^2} \{\phi_t(\mathbf{p}_0)\}$$

Definition 10 (Extremal surface)**.** *The set of all extremals initialized in the state space at $\mathbf{x}_0$ and evaluated over time window $[0, \tau]$ is noted:*

$$\mathcal{S}_{\mathbf{x}_0}(\tau) := \bigcup_{t \in [0,\tau]} \mathcal{E}_{\mathbf{x}_0}(t)$$

For the case of a linear flow field as presented in page 88, we depict[2] an extremal set and an extremal surface in Fig. 3.2.

[2] The code can be found in the notebook `linear_extremals.ipynb` at DOI 10.5281/zenodo.13939206.

Notation 3. *To only consider the state component of extremals, we use the notation:*

$$\mathcal{E}_{\mathbf{x}_0}^{\mathbf{x}}(t) := \bigcup_{\mathbf{p}_0 \in \mathbb{R}^2} \left\{ \phi_t^{\mathbf{x}}(\mathbf{p}_0) \right\}$$

$$\mathcal{S}_{\mathbf{x}_0}^{\mathbf{x}}(\tau) := \bigcup_{t \in [0,\tau]} \mathcal{E}_{\mathbf{x}_0}^{\mathbf{x}}(t)$$

Property 3.5

In the TNP, it is sufficient to sample the circle to get the entire extremal set in the physical space

$$\mathcal{E}_{\mathbf{x}_0}^{\mathbf{x}}(t) := \bigcup_{\mathbf{p}_0 \in \mathbb{S}^1} \left\{ \phi_t^{\mathbf{x}}(\mathbf{p}_0) \right\}$$

Proof. It is a consequence of property 3.2. □

The following property will bound the behavior of two neighboring extremals integrated forward in time.

Property 3.6: Grönwall's inequality

If a state initial condition $\mathbf{x}_0$ and two costate initial conditions $\mathbf{p}_0^{(1)}$ and $\mathbf{p}_0^{(2)}$ are given, the following bound on the two extremals holds:

$$\left\| \phi_t\left(\mathbf{p}_0^{(1)}\right) - \phi_t\left(\mathbf{p}_0^{(2)}\right) \right\| \leq \left\| \mathbf{p}_0^{(2)} - \mathbf{p}_0^{(1)} \right\| e^{L_F t} \tag{3.9}$$

Proof. This results from the application of Grönwall's inequality to 3.4. □

> **Remark**
>
> The previous property shows that $\mathbf{p}_0 \mapsto \phi_t(\mathbf{p}_0)$ is a continuous application.

Then, we establish the structure of the extremal set with the following property.

Property 3.7

$\mathcal{E}_{\mathbf{x}_0}^{\mathbf{x}}(t)$ is a loop.

Proof. $\mathcal{E}_{\mathbf{x}_0}^{\mathbf{x}}(t)$ is the image of the loop $\mathbb{S}^1$ by the continuous application $\phi_t^{\mathbf{x}}(\cdot)$ composed with the projection (continuous) on the $\mathbf{x}$-space. □

Finally, we highlight the link between extremals and the optimal travel time in a navigation problem.

Property 3.8

If $\tau^\star(\mathbf{x}_f)$ is the minimum travel duration from $\mathbf{x}_0$ to $\mathbf{x}_f$, then

$$\tau^\star(\mathbf{x}_f) = \min\left\{\tau \,\middle|\, \{\mathbf{x}_f\} \cap \mathcal{E}_{\mathbf{x}_0}^{\mathbf{x}}(\tau) \neq \varnothing\right\}$$

Proof. This is just a reformulation of Pontryagin's minimum principle. Extremals are candidates to optimality which means the optimal trajectory of the problem is to be found among the set of extremals. When the extremal set hits the target for several duration values, the smallest one is indeed the minimum travel time. □

> **Remark**
> $\tau^\star$ is the origin value function J from Definition 6.

3.3.1 Piecewise linear approximation of extremal surface

Since the extremal surface is a continuous object, we have to define a numerical way to approximate it. We build a piecewise linear approximation of the extremal surface for a given finite number of extremals and a number of time steps[3]. This is illustrated in figure 3.3.

[3] The code can be found in the notebook `linear_extremals.ipynb` at DOI 10.5281/zenodo.13939206.

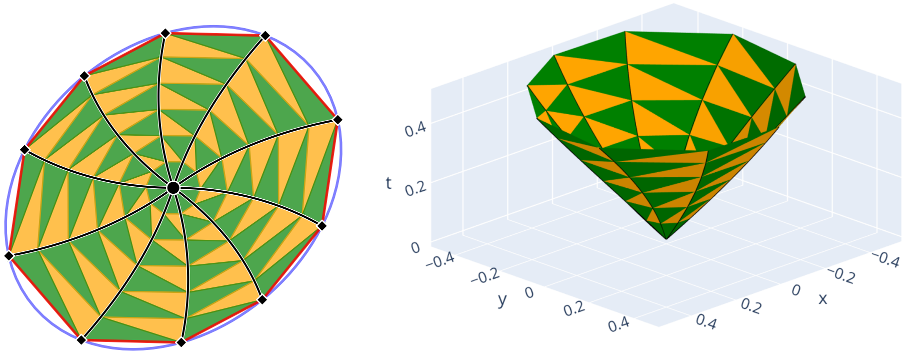


Figure 3.3: Left: Piecewise linear approximation of the exact extremal surface using a finite collection of extremals for the linear flow field 3.7. Right: The same object represented in 3D with time as third dimension. It is the approximate value function $J_{\text{app}}(\mathbf{x})$.

For any neighboring pair $\left(\mathbf{p}_0^-,\ \mathbf{p}_0^+\right)$, we want to use the approximation:

$$\forall \mathbf{p}_0 \in [\mathbf{p}_0^-, \mathbf{p}_0^+], \alpha \in [0,1], \phi_t^{\mathbf{x}}\left((1-\alpha)\mathbf{p}_0^- + \alpha\mathbf{p}_0^+\right) \approx (1-\alpha)\phi_t^{\mathbf{x}}\left(\mathbf{p}_0^-\right) + \alpha\phi_t^{\mathbf{x}}\left(\mathbf{p}_0^+\right)$$

in order to build the extremal set partial approximation:

$$E_{\mathbf{p}_0^-,\mathbf{p}_0^+}(t) := \left\{(1-\alpha)\phi_t^{\mathbf{x}}\left(\mathbf{p}_0^-\right) + \alpha\phi_t^{\mathbf{x}}\left(\mathbf{p}_0^+\right) \,\middle|\, \alpha \in [0,1]\right\} \approx \bigcup_{\mathbf{p}_0 \in [\mathbf{p}_0^-,\mathbf{p}_0^+]} \left\{\phi_t^{\mathbf{x}}\left(\mathbf{p}_0\right)\right\}$$

Then, the collection of all these segments in time provides an approximation of the extremal surface

$$\bigcup_{t\in[0,\tau]} \left\{(1-\alpha)\phi_t^{\mathbf{x}}\left(\mathbf{p}_0^-\right) + \alpha\phi_t^{\mathbf{x}}\left(\mathbf{p}_0^+\right) \,\middle|\, \alpha \in [0,1]\right\} \approx \mathcal{S}_{\mathbf{x}_0}(\tau)$$

The previous collection depends on the knowledge of the extremal values at any time step t in the working time window. But we will solve the IVP for extremals on a given time discretization $(t_i)_{i\in\{0,\dots,n_t-1\}}$. Thus, we define the extremal surface partial approximations by the triangles:

$$S^{(i,-)}_{\mathbf{p}_0^-,\mathbf{p}_0^+} := \operatorname{conv}\left(\phi_{t_i}^{\mathbf{x}}\left(\mathbf{p}_0^-\right), \phi_{t_{i+1}}^{\mathbf{x}}\left(\mathbf{p}_0^-\right), \phi_{t_i}^{\mathbf{x}}\left(\mathbf{p}_0^+\right)\right)$$

and

$$S^{(i,+)}_{\mathbf{p}_0^-,\mathbf{p}_0^+} := \operatorname{conv}\left(\phi^{\mathbf{x}}_{t_i}\left(\mathbf{p}_0^-\right), \phi^{\mathbf{x}}_{t_i}\left(\mathbf{p}_0^+\right), \phi^{\mathbf{x}}_{t_{i+1}}\left(\mathbf{p}_0^+\right)\right)$$

We can then build an approximation of the origin value function (Definition 6) from the the linear interpolation of travel times over the triangles forming the approximate extremal surface. First let $\operatorname{lin}(\mathbf{x};\mathbf{a},\mathbf{b},\mathbf{c},\alpha,\beta,\gamma)$ be the value at position $\mathbf{x}$ of the linear function of $\mathbb{R}^2$ that is uniquely defined by:

$$\begin{aligned}\operatorname{lin}(\mathbf{a};\mathbf{a},\mathbf{b},\mathbf{c},\alpha,\beta,\gamma) &= \alpha\\ \operatorname{lin}(\mathbf{b};\mathbf{a},\mathbf{b},\mathbf{c},\alpha,\beta,\gamma) &= \beta\\ \operatorname{lin}(\mathbf{c};\mathbf{a},\mathbf{b},\mathbf{c},\alpha,\beta,\gamma) &= \gamma\end{aligned}$$

We then define partial value functions as:

$$J^{(i,-)}_{\mathbf{p}_0^-,\mathbf{p}_0^+}(\mathbf{x}) := \begin{cases} \operatorname{lin}\left(\mathbf{x};\phi^{\mathbf{x}}_{t_i}\left(\mathbf{p}_0^-\right), \phi^{\mathbf{x}}_{t_{i+1}}\left(\mathbf{p}_0^-\right), \phi^{\mathbf{x}}_{t_i}\left(\mathbf{p}_0^+\right), t_i, t_{i+1}, t_i\right) \text{ if } \mathbf{x} \in S^{(i,-)}_{\mathbf{p}_0^-,\mathbf{p}_0^+} \\ +\infty \text{ else} \end{cases}$$

and

$$J^{(i,+)}_{\mathbf{p}_0^-,\mathbf{p}_0^+}(\mathbf{x}) := \begin{cases} \operatorname{lin}\left(\mathbf{x};\phi^{\mathbf{x}}_{t_i}\left(\mathbf{p}_0^-\right), \phi^{\mathbf{x}}_{t_i}\left(\mathbf{p}_0^+\right), \phi^{\mathbf{x}}_{t_{i+1}}\left(\mathbf{p}_0^+\right), t_i, t_i, t_{i+1}\right) \text{ if } \mathbf{x} \in S^{(i,+)}_{\mathbf{p}_0^-,\mathbf{p}_0^+} \\ +\infty \text{ else} \end{cases}$$

And then we have an approximation for the value function over the time window $[t_0, t_{n_t-1}]$:

$$J_{\text{app}}(\mathbf{x}) := \min_{\substack{i\in\{0,\dots,n_t-1\}\\ (\mathbf{p}_0^+,\mathbf{p}_0^-)\text{ neighboring pair}}} \min\left\{J^{(i,-)}_{\mathbf{p}_0^-,\mathbf{p}_0^+}(\mathbf{x}),\ J^{(i,+)}_{\mathbf{p}_0^-,\mathbf{p}_0^+}(\mathbf{x})\right\} \tag{3.10}$$

In the time direction, the error made on the computation of the augmented state is controlled and it depends on the integration scheme. In the extremal initial parameter $\mathbf{p}_0$ space though, it is more difficult to provide a bound between the linear approximation and the actual extremal set. But keeping in mind practical applications, we are going to work under the assumption that if all pairs of neighboring extremals are close enough, say distant by a maximum of ε, then the approximate value function $J_{\text{app}}(\mathbf{x})$ is sufficiently representative of the true value function.

Assumption 1 (Extremal set approximation). *For every neighboring pair $\left(\mathbf{p}_0^-,\ \mathbf{p}_0^+\right)$, if the following property holds:*

$$\forall t \in [0, \tau], \quad \left\|\phi_t^{\mathbf{x}}\left(\mathbf{p}_0^-\right) - \phi_t^{\mathbf{x}}\left(\mathbf{p}_0^+\right)\right\| \leq \varepsilon \tag{3.11}$$

then the value function approximation is close to the real value function:

$$J_{app}(\mathbf{x}) \approx J(\mathbf{x})$$

Working under Assumption 1 implicitly states that the true value function does not exhibits features under the scale of ε. This is limiting the variety of value functions that we can correctly represent with the extremal scheme. Still, it seems a reasonable assumption as regards the LSPOV, because we previously chose to study long-range navigation problem disregarding the small-scale behavior of the vehicle. Thus, it does not seem to make sense to try to resolve small features of the value function.

Similarly, trying the solve the TNP by exactly reaching the target state $\mathbf{x}_f$ does not make sense numerically. So, in practice we are going to look for solutions as trajectories reaching a ball around the target $B(\mathbf{x}_f, \rho)$, with ρ small compared to the problem's characteristic length, and set to an operational value for which we can consider the target is reached. It would then make sense to take the precision threshold ε equal to the tolerance radius ρ around the target.

Now that we are working under Assumption 1, we define the validity index for a pair of neighboring extremals.

Definition 11 (Validity index). *For any neighboring pair $\left(\mathbf{p}_0^-,\ \mathbf{p}_0^+\right)$, the validity index i_{VI} is*

$$i_{VI} := \max\left\{ i \;\middle|\; \forall j \leq i, \quad \left\|\phi_{t_j}\left(\mathbf{p}_0^-\right) - \phi_{t_j}\left(\mathbf{p}_0^+\right)\right\| \leq \varepsilon \right\}$$

In other words it is the largest time index up to which inequality 3.11 holds for the considered pair.

With this definition of proximity between extremals, it is now possible to design algorithms that sample extremals close to one another to approximate the extremal set and thus solve navigation problems.

3.3.2 Equisampling algorithm

When an initial costate vector $\mathbf{p}_0$ is given, the computation of the associated extremal $\phi_t(\mathbf{p}_0)$ can be done in a efficient way using any IVP solver, for instance `scipy`'s `solve_ivp` function. A first idea of algorithm is to shoot a large number of extremals this way. But how do we ensure 3.11 is satisfied for all pairs of neighboring initial costates? One way could be trial and error: if the selected number of extremals is insufficient for all pairs of extremals to be close enough, then rerun the computation with more extremals. We can also try to *predict* the required number of extremals. For this, using the Grönwall inequality 3.9, we predict for a pair of neighboring extremals $(\mathbf{p}_0^-, \mathbf{p}_0^+)$ that it will satisfy 3.11 at least up to a certain time:

$$t_G := \frac{1}{L_F} \ln\left(\frac{\varepsilon}{\|\mathbf{p}_0^+ - \mathbf{p}_0^-\|}\right)$$

So, if we sample costate vectors:

$$\mathbf{p}_0^{(k)} = \left(\cos\left(\frac{2\pi k}{n}\right) \;\; \sin\left(\frac{2\pi k}{n}\right)\right)^\mathsf{T}, \quad k \in \{0, 1, \ldots, n-1\}$$

each pair $\left(\mathbf{p}_0^{(k)}, \mathbf{p}_0^{(k')}\right)$ with $k' - k \equiv 1[n]$ will be valid on the whole time window $[0, T]$ if

$$n \geq N_{\text{cons}} := \left\lceil 2\pi\, \varepsilon^{-1}\, e^{L_F T} \right\rceil \tag{3.12}$$

where we used the approximation, valid for $n \gg 1$, that $\left\|\mathbf{p}_0^{(k')} - \mathbf{p}_0^{(k)}\right\| \approx \frac{2\pi}{n}$. This is the basis of a first algorithm, which is presented in Algorithm 1.

> **Remark**
>
> The presence of the term $L_F T$ in the exponential suggests that the "complexity" of the behavior of extremal trajectories in a navigation problem is mostly influenced by the magnitude of this product. Indeed it shall increase exponentially with it.

```
Data: $\mathbf{x}_0$, $\mathbf{x}_t$, $\mathbf{v}_w$, $T$   ▷ Start, target, flow field, time window upper bound
Parameters: $\varepsilon$, $L$   ▷ Max. dist. between extremals, Lipschitz constant of aug. dynamics
Result: $l_{\mathrm{extr}}$   ▷ List of extremals covering reachable set
$n \leftarrow \lceil 2\pi\, \varepsilon^{-1}\, e^{LT} \rceil$;
$l_{\mathrm{extr}} \leftarrow []$;
for $k \in \{0, 1, \dots, n-1\}$ do
    $\mathbf{p}_0 \leftarrow \left(\cos\left(\frac{2\pi k}{n}\right)\ \sin\left(\frac{2\pi k}{n}\right)\right)^\intercal$;   ▷ Sample init. costates on the unit circle
    $\mathbf{y}_0 \leftarrow (\mathbf{x}_0\ \mathbf{p}_0)^\intercal$;   ▷ Set initial condition for IVP solver
    $\mathbf{y} \leftarrow \mathrm{SolveIVP}(F, \mathbf{y}_0, 0, T)$;   ▷ Dyn. $F$, init. cond. $\mathbf{y}_0$ at $t = 0$, integr. up to $t = T$
    $l_{\mathrm{extr}} \leftarrow l_{\mathrm{extr}} :: [\mathbf{y}]$;   ▷ Add extremal to list
end
```

Algorithm 1: Equisampling extremal algorithm for the TNP.

If $C(t)$ is an upper bound on the computation cost of $\phi_t(\mathbf{p}_0)$ for all $\mathbf{p}_0 \in \mathbb{S}^1$, then the described algorithm has cost bounded by $C(T)\,\lceil 2\pi\, \varepsilon^{-1}\, e^{L_F T} \rceil$. We depict[4] an example of extremals sampled by the equisampling algorithm in the case of a linear flow field in Fig. 3.4. The figure shows that the spacing between extremals can vary highly depending on the neighboring pair considered.

3.3.3 In-depth sampling algorithm

The exponential cost on the problem complexity $L_F T$ is a major downside of the previous approach. But in the preceding analysis, we used a conservative bound to compute the level of

[4] The code can be found in the notebook `gyre_extremals.ipynb` at DOI 10.5281/zenodo.13939206.

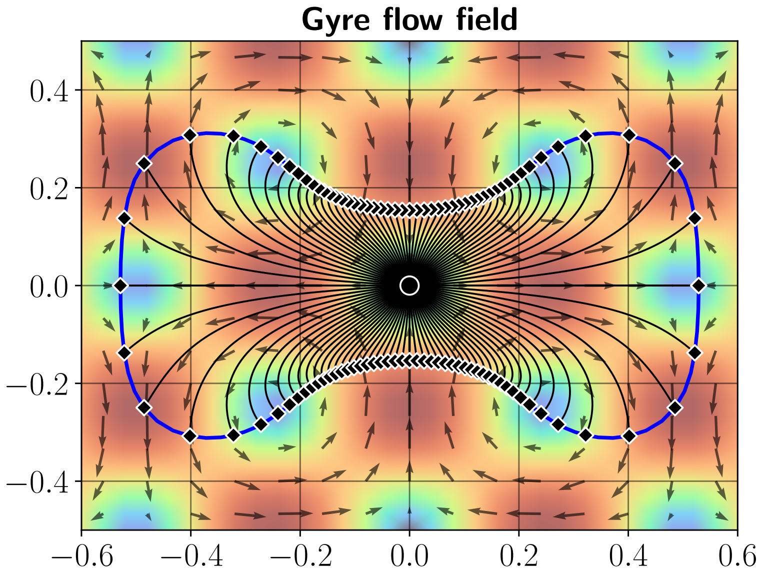


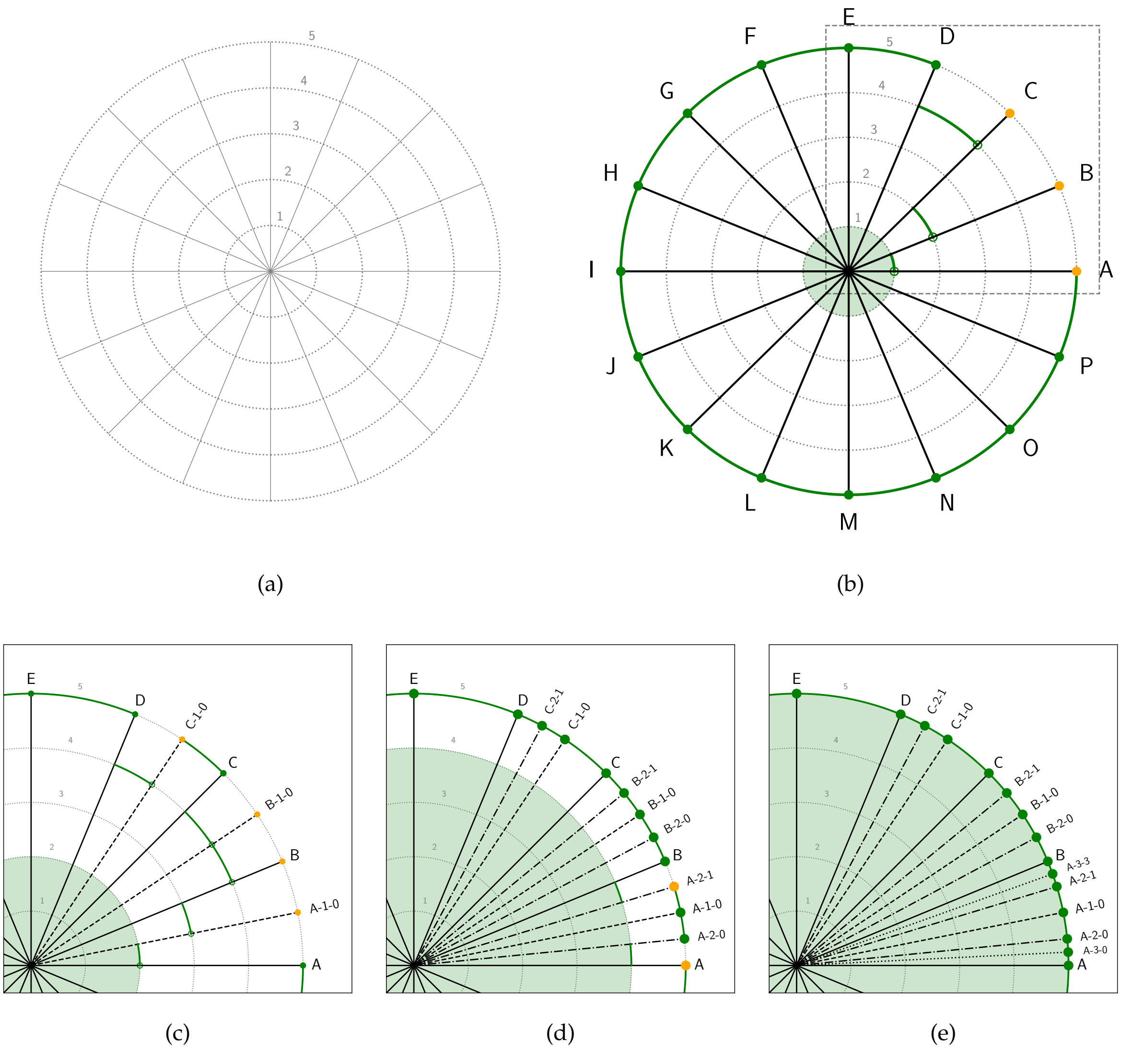
Figure 3.4: Extremals in the gyre flow. Some extremals shot from the center of the figure are depicted in black with their endpoints as diamonds. The blue curve is a high resolution approximation of the exact extremal set.

precision needed for the costate initialization so that condition 3.11 holds. What may happen in practice is that this precision level is indeed too much for most portions of the extremal front, and only required for some of them (this is shown in page 107). This leads to the idea of *resampling* trajectories from the origin only in sectors where the invariant is violated.

A similar idea for the resampling of extremals is given in (Rhoads et al., 2010), but with a heuristic choice of resampling only between a maximum amount of k neighbors leading to the largest distance between endpoints. A real systemic initial heading angle bisection idea appears in Marchidan and Bakolas (2016), but the check for bisection occurs on the endpoint of trajectories as opposed to the synchronous pairwise distance that will be described below.

The resampling principle is illustrated in figure 3.5 and formalized in Algorithm 2.

First (Fig. 3.5a), we choose an initial discretization of $[0, 2\pi]$, here with 16 points. We choose an arbitrary discretization of time, and our goal is to compute a collection of extremals that is valid up to time step number 5. The different radiuses depict time, precisely time steps 1, 2, 3, 4 and 5.



(a)

(b)

(c)

(d)

(e)

Figure 3.5: Extremal polar plot. The angle represents the initial heading angle. The radius corresponds to time, with increasing time for increasing radius. The green disk shows the maximum time for which the extremal field is considered valid, *i.e.* a correct approximation of the value function.

Then (Fig. 3.5b), we integrate forward in time (Algorithm 2, line 8) the initial set of 16 extremals and give them letters. The solid dots show the time step at which integration stopped. Here, integration stopped for all extremals at the last time steps, but it will be different in the following sections.

For every pair of extremals, we remove from the list of neighbors all pairs with a validity index having reached the end of the time window (function `checkNeighbors`, Algorithm 2, line 16, detailed in Algorithm 3). For the remaining pairs, we depict an unfilled dot at the validity index to the member of the pair that is at the clock wise extremity. We make solid points orange to depict that these points do not guarantee the proximity to their counter clockwise neighbor.

Thus (c), when validity fails to be proven up to the final time step, we add new extremals for shooting as "children" of previous ones (Algorithm 2, line 20), with initial costate vector being the average of the parents' initial costate vectors. A new child break its parents' pair to form two new pairs: the first parent and the child, the second parent and the child (Algorithm 2, lines 24 and 25).

We then continue the process up to the point where every pair of neighbors satisfies 3.11 up to the final time step.

Definition 12 (Overall validity index)**.** *The Overall Validity Index (OVI) is defined as the smallest validity index among all pairs of neighbors.*

> **Remark**
> The green circle in figure 3.5 is a graphical representation of the OVI.

```
Data: $\mathbf{x}_0$, $\mathbf{x}_t$, $\mathbf{v}_w$, $T$   ▷ Start, target, flow field, time window upper bound
Parameters: $\varepsilon$, $n_{init}$   ▷ Max. dist. between extremals, init. number of extremals
Result: $d_{\text{extr}}$   ▷ Dictionary of extremals covering reachable set
$d_{\text{extr}} \leftarrow$ emptyDictionary();
$l_{\text{neigh}} \leftarrow []$;
for $k \in \{0, 1, \ldots, n_{init} - 1\}$   ▷ Shoot initial set of extremals
do
    $\mathbf{p}_0^- \leftarrow \left(\cos\left(\frac{2\pi k}{n_{\text{init}}}\right)\ \sin\left(\frac{2\pi k}{n_{\text{init}}}\right)\right)^\intercal$;
    $\mathbf{y}_0 \leftarrow \left(\mathbf{x}_0\ \mathbf{p}_0^-\right)^\intercal$;
    $\mathbf{y} \leftarrow \text{SolveIVP}(F, \mathbf{y}_0, 0, T)$;   ▷ Dyn. $F$, init. cond. $\mathbf{y}_0$ at $t = 0$, int. up to $t = T$
    $d_{\text{extr}}[\mathbf{p}_0^-] \leftarrow \mathbf{y}$;
    $\mathbf{p}_0^+ \leftarrow \left(\cos\left(\frac{2\pi(k+1)}{n_{\text{init}}}\right)\ \sin\left(\frac{2\pi(k+1)}{n_{\text{init}}}\right)\right)^\intercal$;
    $l_{\text{neigh}} \leftarrow l_{\text{neigh}} :: \left[\left(\mathbf{p}_0^-, \mathbf{p}_0^+\right)\right]$;   ▷ Keep track of neighboring relations
end
$l'_{\text{neigh}} \leftarrow l_{\text{neigh}}$;
while $l'_{neigh} \neq \varnothing$   ▷ Resample extremals until precision reached
do
    $l_{\text{neigh}} \leftarrow$ checkNeighbors($l'_{\text{neigh}}, \varepsilon, T, d_{\text{extr}}$);   ▷ Remove pairs of extremals that are
      close enough. See Algorithm 3
    $l'_{\text{neigh}} \leftarrow []$;
    for $\left(\mathbf{p}_0^-, \mathbf{p}_0^+\right) \in l_{neigh}$   ▷ For all pairs too far from one another
    do
        $\mathbf{p}_0 \leftarrow \frac{1}{2}\left(\mathbf{p}_0^- + \mathbf{p}_0^+\right)$;   ▷ Init. on mean costate value
        $\mathbf{y}_0 \leftarrow (\mathbf{x}_0\ \mathbf{p}_0)^\intercal$;
        $\mathbf{y} \leftarrow \text{SolveIVP}(F, \mathbf{y}_0, 0, T)$;   ▷ Integrate extremal
        $d_{\text{extr}}[\mathbf{p}_0] \leftarrow \mathbf{y}$;   ▷ Register extremal
        $l'_{\text{neigh}} \leftarrow l'_{\text{neigh}} :: \left[\left(\mathbf{p}_0^-, \mathbf{p}_0\right)\right]$;   ▷ Register new pairs of neighbors
        $l'_{\text{neigh}} \leftarrow l'_{\text{neigh}} :: \left[\left(\mathbf{p}_0, \mathbf{p}_0^+\right)\right]$;
    end
end
```

Algorithm 2: In-depth sampling extremal algorithm for the TNP.

```
Input: $l_{\text{neigh}}$, $\varepsilon$, $T$, $d_{\text{extr}}$   ▷ List of pairs of neighb. extremals, max. dist. between
    extremals, time window up. bound, dictionary matching init. costate to extremal
Result: $l'_{\text{neigh}}$   ▷ List of neighboring extremals that are too far from one another
$l'_{\text{neigh}} \leftarrow []$;
for $(\mathbf{p}_0^-, \mathbf{p}_0^+) \in l_{neigh}$ do
    $\mathbf{y}^- \leftarrow d_{\text{extr}}[\mathbf{p}_0^-]$;   ▷ Get extremal corresponding to init. costate
    $\mathbf{y}^+ \leftarrow d_{\text{extr}}[\mathbf{p}_0^+]$;
    if $\exists t \in [0, T], \|\mathbf{y}^-(t) - \mathbf{y}^+(t)\| > \varepsilon$   ▷ Extremals violate Assumption 1
     then
        $l'_{\text{neigh}} \leftarrow l'_{\text{neigh}} :: [(\mathbf{p}_0^-, \mathbf{p}_0^+)]$   ▷ Add pair to returned list
    end
end
```

Algorithm 3: `checkNeighbors` routine. Keep only the pairs of initial costates for which extremals are too far from one another.

Comparison of equisampling to in-depth sampling

We compare the equisampling algorithm to the in-depth sampling algorithm on a case with a multi-gyre flow, *i.e.* the flow field is form:

$$\mathbf{v}_f(\mathbf{x}) := a \begin{pmatrix} -\sin(x_1')\cos(x_2') \\ \cos(x_2')\sin(x_1') \end{pmatrix}; \quad x_1' = \frac{2\pi}{\lambda_1}(x_1 - c); \quad x_2' = \frac{2\pi}{\lambda_2}(x_2 - c) \tag{3.13}$$

We set $\lambda_1 = \lambda_2 = 2$, $a = 1$, $c = (0.5\ 0.5)^\intercal$ so that we work in that same settings as in Kularatne et al. (2016).

Starting from point $(0.6\ 0.6)^\intercal$ we want to reach $(2.4\ 2.4)^\intercal$ with a precision $\varepsilon = 0.1$. The precision means that at the time the target is reachable, we want that 3.11 holds for all pairs of initial costate vectors. We display a target circle of radius ε to visually account for this precision requirement in Figure 3.6.

We run the equisampling algorithm and empirically adjust the number of extremals to ap-

Gyre			
T	0.8	1.2	1.6
Equisampling			
Number of trajectories	500	2000	8000
Max. dist. extremals at T	0.1113	0.1013	0.1126
In depth			
Number of trajectories	128	264	501
Max. dist. extremals at T	0.09888	0.09931	0.09987

Table 3.1: Comparison of equisampling to in-depth sampling. For different time window upper bounds T, the number of extremals required to achieve a distance between neighbors of approximately 0.1 is given for the equisampling algorithm or the in-depth sampling algorithm.

proximately match the ε precision requirement on the extremal set spacing. We allow the spacing to be slightly over the ε requirement because what matters is the order of magnitude for the number of extremals. We let both algorithms run their computation in free space *i.e.* we do not set boundaries for the problem. We want to understand how the number of extremals evolves with the time upper bound.

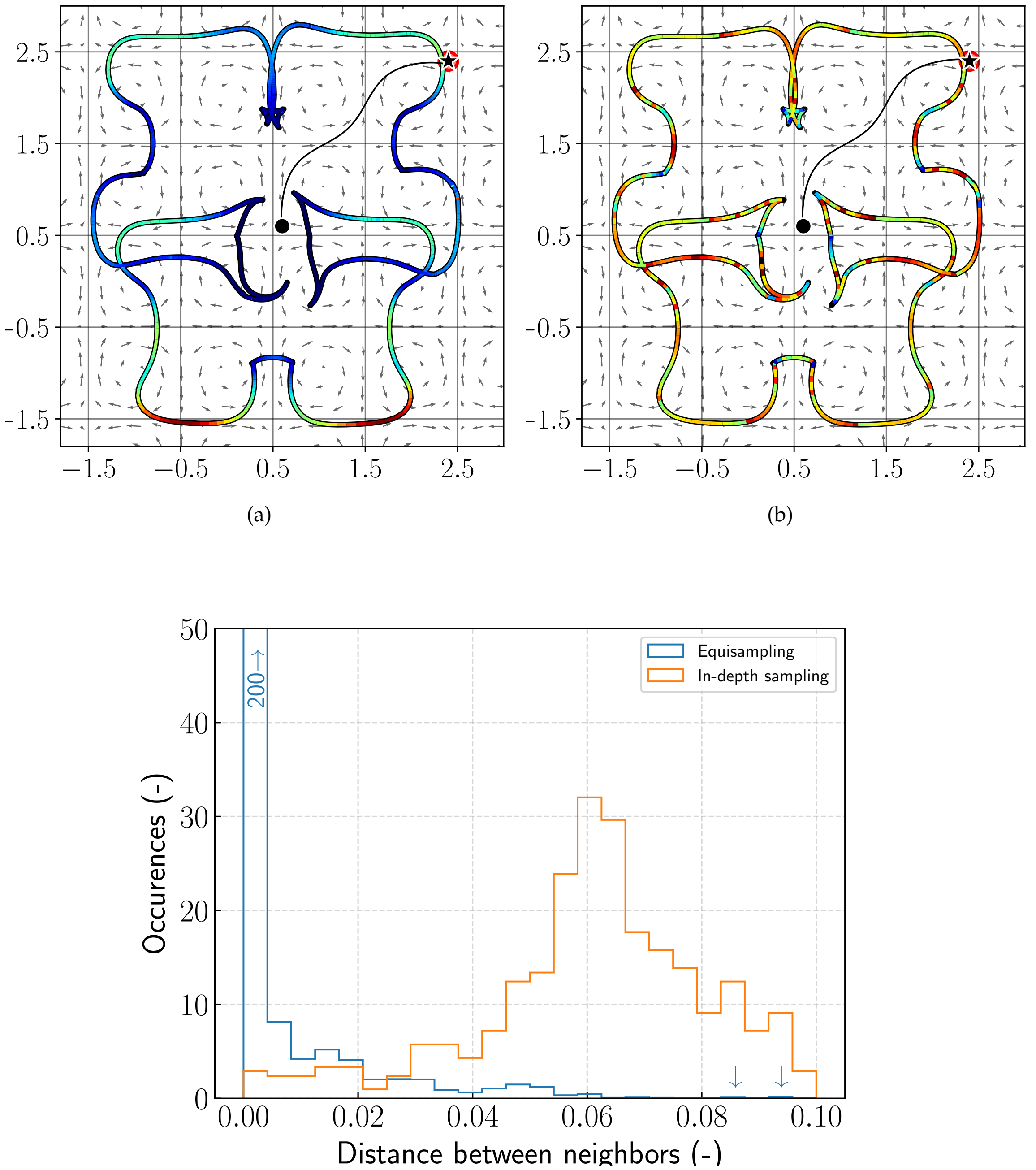
We also run the in-depth sampling algorithm and note its number of extremals[5]. Results are shown in Table 3.1.

In each case, the in-depth principle shows a major reduction in the required number of extremals to achieve comparable precision on the front. As depicted in Fig. 3.6, the spacing between neighboring extremals is also better balanced.

3.3.4 Closure

For now, we have considered the case where integration is always possible on the whole time window. But as can be noticed on previous figures, the extremal set grows in every direction, some of them being useless for the computation of the optimal path under study, *i.e.* parts of the front going away front the target point. For this reason, we may want to *frame* the problem *i.e.* add borders to the problem which make the integration stop. We may stress that on a first

[5] The code can be found in the notebook `algo_comparison.ipynb` at DOI 10.5281/zenodo.13939206.



(a)

(b)

(c)

Figure 3.6: Comparison of equisampling to in-depth sampling of extremals. On top, fronts of extremals computed through (a) equisampling or (b) in-depth sampling. Color indicates the distance between neighboring extremals. The more red, the further away they are. Color is scaled by the maximum distance in each case. (c) Distribution of distances among all pairs of neighboring extremals.

approach, these borders are really integration stoppers and *not* obstacles. Obstacles will be discussed in Chapter 4. We also note that by adding a frame we *choose* to look for a solution within the defined frame of the problem.

We now have to derive a modified version of the in-depth resampling algorithm that accounts for the fact that some extremals can fail to integrate up to the end of the time window. We call these extremals *closed* and depict their endpoint as a cross. Because these extremals cannot proceed further in time, the OVI will be stuck forever by closed trajectories. We thus need to define a new validity index which goal is to track the progression of the computation validity over the full time window, taking closed trajectories into account.

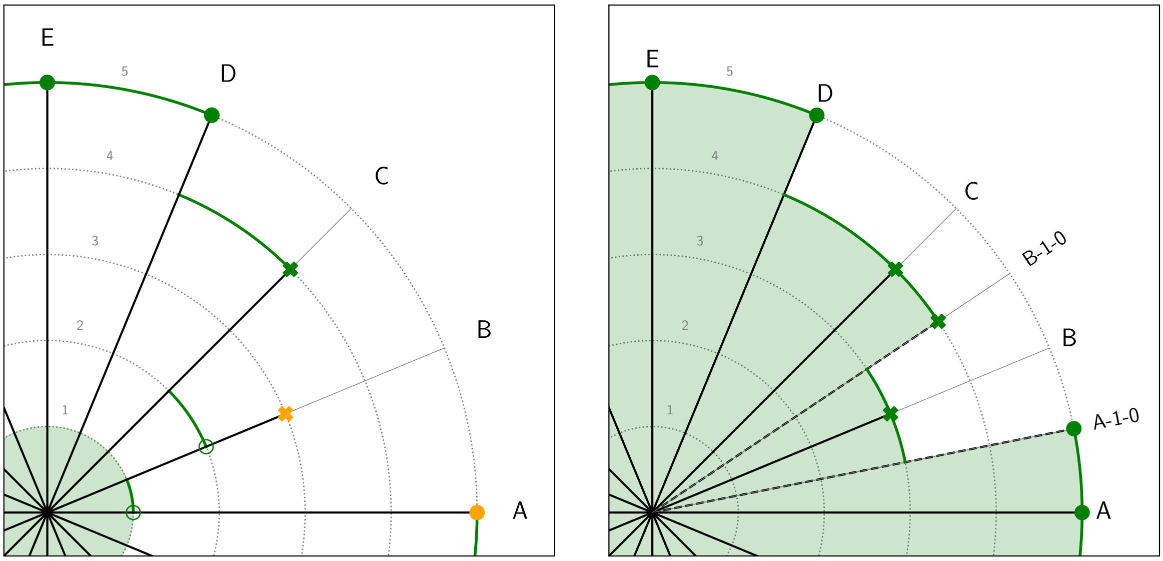
Definition 13 (Closure-aware validity index)**.** *The Closure-Aware Validity Index (CAVI) is the smallest validity index among all neighboring pairs of extremals excepting pairs where one extremal at least is closed and the pair's validity index equals its closure index.*

> **Remark**
> Because it is a minimum taken on a smaller set of values, the CAVI is always greater than the OVI.

The CAVI ensures that all pairs of neighboring extremals either satisfy 3.11 up to it or were closed before the index and valid at time of closure. The closure and CAVI update mechanism is displayed in Fig. 3.7. We do not display the algorithm updated with the closure mechanism as it is almost the same as the in-depth algorithm, the difference being that the `checkNeighbors` function does not feed pairs of neighbors for shooting if one of the members of the pair is closed and the pair's validity index equals its closure index.

Limit of the in-depth algorithm

We document a real wind field case in which the in-depth sampling algorithm fails to converge. The failure is in the sense that the algorithm indeed find extremals reaching target, say at time



(a)

(b)

Figure 3.7: Extremal polar plot depicting the closure of some extremals. (a) Extremals A to E were integrated, but the integration of B and C was stopped (these are closures). B is not close enough to its neighbor C at the time of closure so it is invalid at time of closure and consequently depicted in yellow. But C is, so it is depicted in green. (b) In-depth sampling is performed: A-1-0 and B-1-0 are integrated. A-1-0 is close enough to A at time step 5 and B at time step 3 (which is the closure index of B). B-1-0 is close enough to B at time step 3 and C at time step 4. So, each pair of neighbors satisfy the precision criteria 3.11 and there is no further integration to be done. The algorithm terminates.

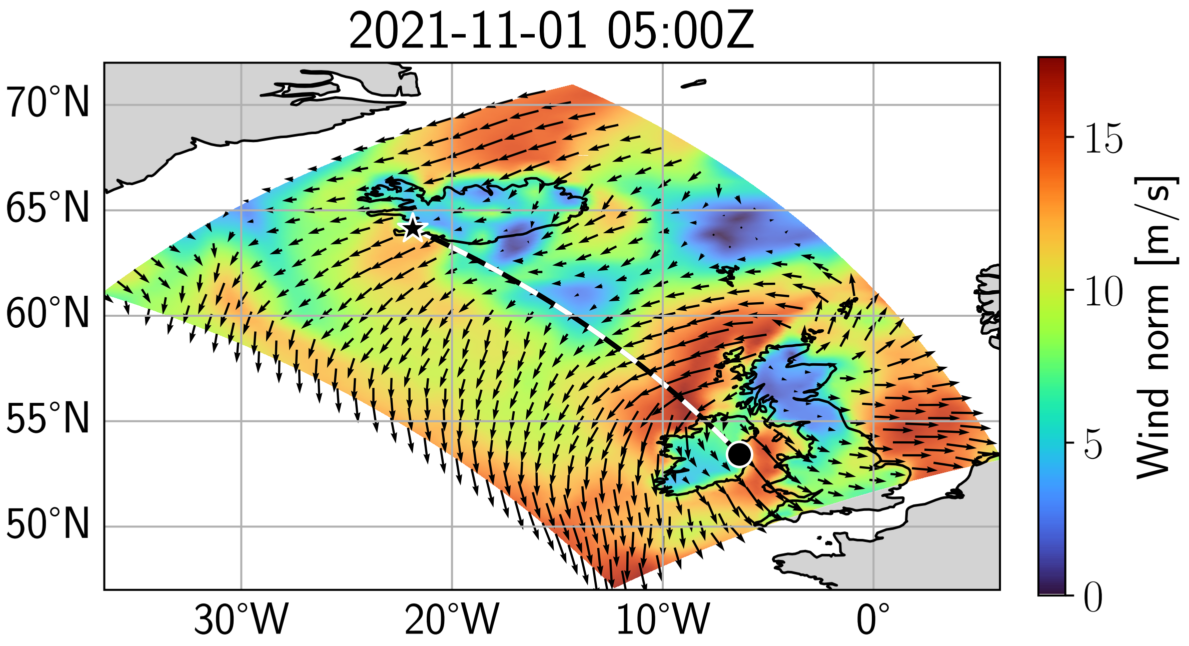


Figure 3.8: Navigation problem from Dublin, Ireland (black dot) to Reykjavik, Iceland (black star) near sea level. The wind field is the ERA5 reanalysis surface wind field for 2021-11-01 05:00Z. The figure is a plate-carrée projection of Earth, and the wind data is sliced and resampled in an oblique Mercator band adjusted between origin and destination. The black and white line is the geodesic path from origin to destination.

index i_2, but the CAVI gets stuck at some time index i_1 way below i_2. Consequently, the algorithm cannot guarantee to have the correct value function at time index i_2. Thus, it cannot guarantee the solution found is a global minimum of the problem.

The problem setting[6] is that of taking off from Dublin, Ireland on 2021-11-01 at 05:00 UTC and reaching Reykjavik, Iceland the fastest possible at 23 m/s. The wind data is the ERA5 Reanalysis surface wind field extracted from the the Copernicus climate Data Store (CDS), with a time resolution of 3 hours and a spatial discretization of 0.5 deg both in longitude and latitude. The general situation is depicted in Fig. 3.8.

We choose to cast the problem to planar 2D space for simplicity and to make the problem non-dimensionalized. The casting to 2D space is performed using an oblique Mercator projection: it is achieved by rotating the equator to the great circle passing through origin and

[6] The code can be found in the notebook `reykjavik_dublin.ipynb` at DOI 10.5281/zenodo.13939206.

destination. Thus, in the projected space, no error is done on distances computed along the great circle between origin and destination, and moderate error is done when computing distances near the borders of the domain. For the considered case, the accuracy of the projection is not a concern as the main focus is to work with wind fields representative of real-world data but which are set up on a planar space. For a discussion on the difference between working in projected space or spherical coordinates, the reader is referred to Appendix D.

In the projected space, Dublin is at $(1857 \times 10^3 \quad 743 \times 10^3\,)$ (in meters) and Reykjavik at $(371 \times 10^3 \quad 743 \times 10^3\,)$.

We make the problem non-dimensionalized by building a rectangular frame of aspect ratio 1.5 aligned with the origin-destination axis and centered at the midpoint between them, with a width of 3/2 the great circle distance between Dublin and Reykjavik. We then set the height of the rectangle as the reference length, *i.e.* we divide all distances by a factor $k_l = 1486 \times 10^3\,\mathrm{m}$. We then build a time scale by dividing the length scale by the vehicle's airspeed. Hence, the time scale $k_t = 64.6 \times 10^3\,\mathrm{s} \approx 17\,\mathrm{h}\,57\,\mathrm{min}$. By construction, in the non-dimensionalized problem, the vehicle has an SRF of 1. We depict snapshots of the wind field in Figure 3.9.

We run the in-depth algorithm for this problem. The maximum distance between extremals is set to the tolerance radius around the target, which is set to 1/80th of the rectangle diagonal, that is 2.25×10^{-2} in the non-dimensionalized problem. The numerical integrator is the 'RK45' method implemented in `scipy 1.13`. This integrator is free to choose its time step but we set an upper bound of $\Delta t_{\text{max}} = 0.05$ for the latter. We run the algorithm with a manual control of the 'while' loop (Algorithm 2). We observe that the CAVI stops increasing with entries in the while loop. We thus stop the algorithm after 13 iterations of the while loop. There is a total of 8382 extremals shot. They lead to the construction of a candidate value function for the problem. It is depicted as color contours in Fig. 3.10. To validate the value function, we compute reachability fronts for the vehicle using the MSEAS Path Planning Level Set software. They are depicted in dashed lines in Fig. 3.10.

We see that, even though the value function is in accordance with the reachability fronts, the

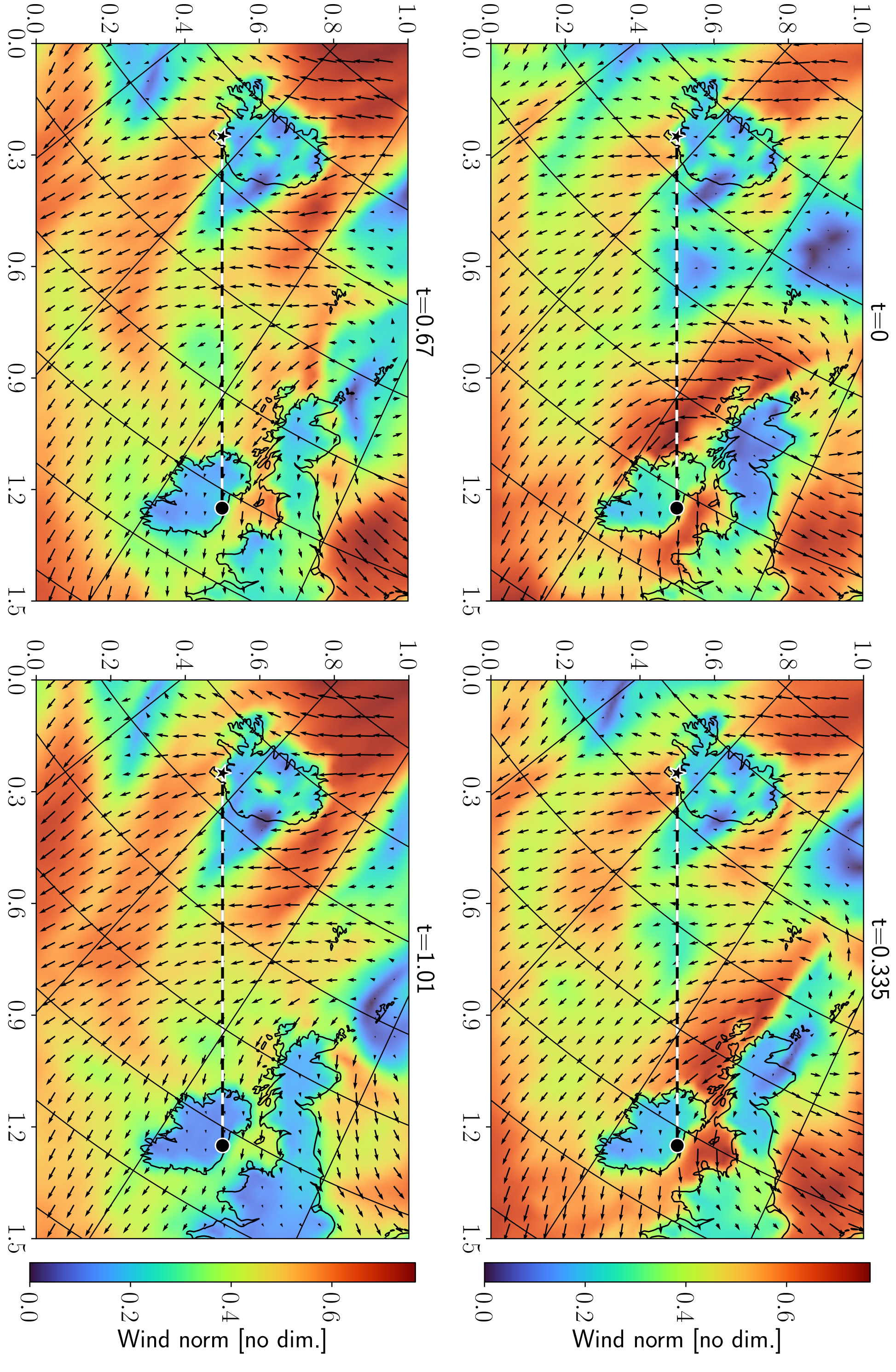


Figure 3.9: Wind field between Reykjavik (star, left) and Dublin (dot, right) from 2021-11-01 05:00 UTC (first frame) to (approximately) 2021-11-01 23:00 UTC (last frame) in the non-dimensionalized space. Parallels and meridians are depicted in black.

validity principle in the sense of closeness between extremals (Assumption 1) fails to guarantee that the value function is valid for the problem. This is because the CAVI did not reach the end of the time window (or at least, the candidate time to optimality), and we depict this using hatches where the value function is considered safe.

To understand why the value function seems correct but fails to be proven valid, we display in Fig. 3.11 a behavior occurring for many pairs of neighboring extremals in the previous computation. We identify three extremals: B, R and Y associated to colors red, blue and yellow. B and R and neighbors up to a given index after which they diverge. Y is the children of B and R, shot in the while loop of Algorithm 2 with mean costate value as initial condition. Y is expected to be between B and R because it has the mean costate initial condition. But as we see in the figure, it ends up staying close to B. What is very bad is that it is not closer to B than R was, so the shooting of the yellow trajectory does not move the validity forward as expected.

We explain this phenomenon by chaos occurring between extremals too close to one another. For instance, B and R have very close initial costate vectors $\mathbf{p}_0^{(B)}$ and $\mathbf{p}_0^{(R)}$, with $\left\|\mathbf{p}_0^{(B)} - \mathbf{p}_0^{(R)}\right\| = 5.10 \times 10^{-5}$. At the first discretization time step t_1, we have a very small difference in the positions of extremals $\mathbf{x}^{(B)}(t_1)$ and $\mathbf{x}^{(R)}(t_1)$, with $\left\|\mathbf{x}^{(B)}(t_1) - \mathbf{x}^{(R)}(t_1)\right\| = 7.48 \times 10^{-7}$.

One fix for this may be to ask for more precision from the numerical solver by lowering $\Delta t_{\max}$. But this much increases the cost of the method. And furthermore, the fact that the approximated, unproven value function (color contours in Fig. 3.10) is a good approximation of the real value function makes one feel that more precision is not really what is needed. In the following section, we propose a way to work with interpolation of extremals instead of always going back to initial conditions.

3.3.5 In-depth interpolated algorithm

In Fig. 3.4, we can see that the principle of sampling extremals from the origin leads to a set of extremals that will be very dense around the origin and sparse at the border of the reachability front. This property is shared by both the equisampling algorithm and the in-depth

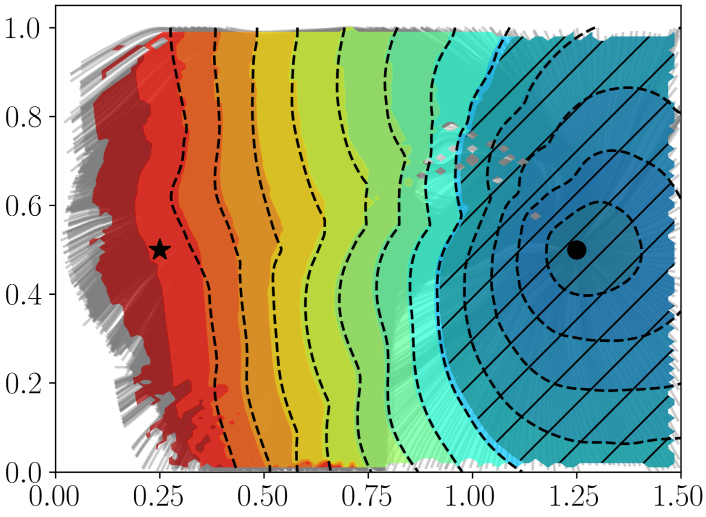


Figure 3.10: Result of running the in-depth sampling algorithm on the Dublin-Reykjavik case. Color contours are inter-level-sets of the approximate value function build from extremals $J_{\text{app}}(x)$ between times that are multiples of 0.1. Dashed lines are level-sets of the true value function for times multiples of 0.1, computed using an external tool. The hatched portion is the part of the approximate value function that is proven valid by the algorithm. Grey lines in the background are the extremals shot by the algorithm.

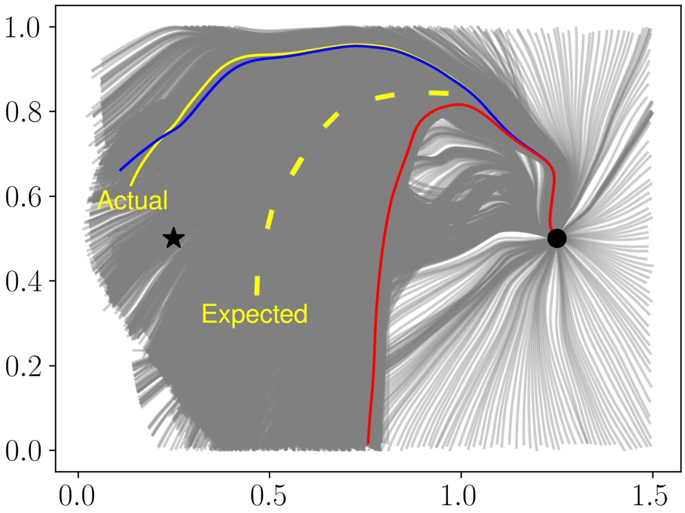


Figure 3.11: Triplet of extremals B (blue), R (red) and Y (yellow) that are integrated forward from different initial angles. Y is the children extremal of B and R, which means its initial costate vector is the mean of B and R's costate vectors. It is expected to follow the dashed yellow line, but actually follows the plain line.

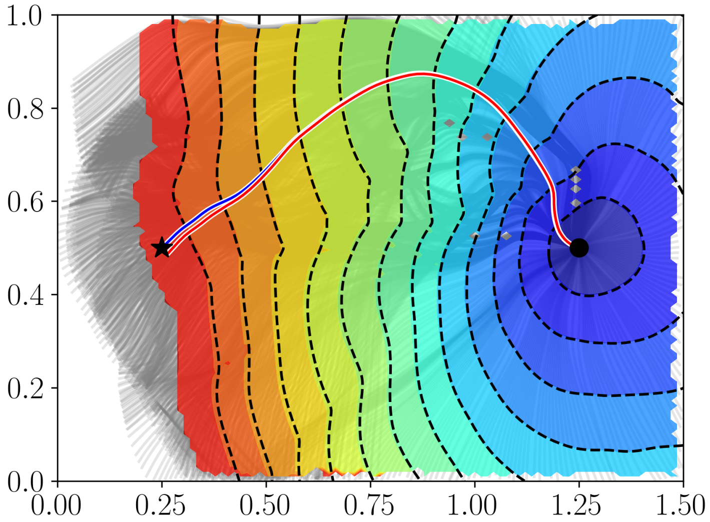


Figure 3.12: Result of running the in-depth interpolated sampling algorithm on the Dublin-Reykjavik case in the same setting as for the in-depth sampling algorithm. Color contours are inter-level-sets of the approximate value function build from extremals $J_{\text{app}}(x)$ between times that are multiples of 0.1. Dashed lines are level-sets of the true value function for times multiples of 0.1, computed using an external tool. Grey lines in the background are the extremals shot by the algorithm. The red trajectory is the solution found by the in-depth interpolated sampling algorithm. The blue trajectory is the solution found by the external level-set solver.

algorithm. Intuitively, one may think that trajectory integration performed at the beginning of the time window are redundant with one another since trajectories are very close to their neighbors. Furthermore, it was shown in page 110 that abusive dichotomy on the initial angle for extremals falls into the caveat of numerical errors and chaos. This motivates the idea not to create new trajectories from the origin but from the interpolation of neighbors (in the sense of the augmented state of 3.4).

To validate this new algorithm, we run it on the Dublin Reykjavik test case presented at page 110. We run the algorithm with the same parameters as the in-depth sampling. The algorithm terminates using a maximum depth of 19 (number of passes in the while loop) and we have a total of 2178 extremals shot. The value function is proven valid up to the time the best extremal reaches the target. So, the problem is solved. We depict the result in Fig. 3.12

The computed value function is in accordance with the external level set solver. The total

```
   Data: $\mathbf{x}_0$, $\mathbf{x}_t$, $\mathbf{v}_w$, $T$   ▷ Start, target, flow field, time window upper bound
1  Parameters: $\varepsilon$, $n_{init}$   ▷ Max. dist. between extremals, init. number of extremals
   Result: $d_{\text{extr}}$   ▷ Dictionary of extremals covering reachable set
2  $d_{\text{extr}} \leftarrow$ emptyDictionary();
3  $l_{\text{neigh}} \leftarrow []$;
4  for $k \in \{0, 1, \ldots, n_{init} - 1\}$   ▷ Shoot initial set of extremals
5  do
6      $\mathbf{p}_0^- \leftarrow \left(\cos\left(\frac{2\pi k}{n_{\text{init}}}\right) \ \sin\left(\frac{2\pi k}{n_{\text{init}}}\right)\right)^\intercal$;
7      $\mathbf{y}_0 \leftarrow (\mathbf{x}_0 \ \mathbf{p}_0^-)^\intercal$;
8      $\mathbf{y} \leftarrow \text{SolveIVP}(F, \mathbf{y}_0, 0, T)$;   ▷ Dyn. $F$, init. cond. $\mathbf{y}_0$ at $t = 0$, int. up to $t = T$
9      $d_{\text{extr}}[\mathbf{p}_0^-] \leftarrow \mathbf{y}$;
10     $\mathbf{p}_0^+ \leftarrow \left(\cos\left(\frac{2\pi(k+1)}{n_{\text{init}}}\right) \ \sin\left(\frac{2\pi(k+1)}{n_{\text{init}}}\right)\right)^\intercal$;
11     $l_{\text{neigh}} \leftarrow l_{\text{neigh}} :: [(\mathbf{p}_0^-, \mathbf{p}_0^+)]$;   ▷ Keep track of neighboring relations
12 end
13 $l'_{\text{neigh}} \leftarrow l_{\text{neigh}}$;
14 while $l'_{neigh} \neq \varnothing$   ▷ Resample extremals until precision reached
15 do
16     $l_{\text{neigh}} \leftarrow$ checkNeighbors($l'_{\text{neigh}}, \varepsilon, T, d_{\text{extr}}$);   ▷ Remove pairs of extremals that are
         close enough. See Algorithm 3
17     $l'_{\text{neigh}} \leftarrow []$;
18     for $(\mathbf{p}_0^-, \mathbf{p}_0^+) \in l_{neigh}$   ▷ For all pairs too far from one another
19     do
20         $\mathbf{y}^- \leftarrow d_{\text{extr}}[\mathbf{p}_0^-]$;   ▷ Get extremal from init. costate
21         $\mathbf{y}^+ \leftarrow d_{\text{extr}}[\mathbf{p}_0^+]$;
22         $t_\varepsilon \leftarrow \min\{t \mid \|\mathbf{y}^-(t) - \mathbf{y}^-(t)\| > \varepsilon\}$;   ▷ Time of distance violation
23         $\mathbf{p}_{\text{interp}} \leftarrow \frac{1}{2}(\mathbf{p}^-(t_\varepsilon) + \mathbf{p}^+(t_\varepsilon))$;   ▷ Mean init. cond.
24         $\mathbf{x}_{\text{interp}} \leftarrow \frac{1}{2}(\mathbf{x}^-(t_\varepsilon) + \mathbf{x}^+(t_\varepsilon))$;
25         $\mathbf{y}_{\text{interp}} \leftarrow (\mathbf{x}_{\text{interp}} \ \mathbf{p}_{\text{interp}})^\intercal$;
26         $\mathbf{y} \leftarrow \text{SolveIVP}(F, \mathbf{y}_{\text{interp}}, t_\varepsilon, T)$;   ▷ Integrate from $t_\varepsilon$ to $T$
27         $\mathbf{p}_0 \leftarrow \frac{1}{2}(\mathbf{p}_0^- + \mathbf{p}_0^+)$   ▷ Virtual init. cond., only used for dictionary key
28         $d_{\text{extr}}[\mathbf{p}_0] \leftarrow \mathbf{y}$;   ▷ Register extremal
29         $l'_{\text{neigh}} \leftarrow l'_{\text{neigh}} :: [(\mathbf{p}_0^-, \mathbf{p}_0)]$;   ▷ Register new pairs of neighbors
30         $l'_{\text{neigh}} \leftarrow l'_{\text{neigh}} :: [(\mathbf{p}_0, \mathbf{p}_0^+)]$;
31     end
32 end
```

Algorithm 4: In-depth interpolated sampling extremal algorithm for the TNP. The lines are the same as for Algorithm 2 except for the lines in color (16-23).

number of extremals is reduced by a factor of 4 compared to the in-depth sampling algorithm, and the convergence is proven.

3.3.6 Trimming

As can be seen for instance on figure 3.6, some parts of the extremal set form loops that fall inside the reachable set. These parts of the extremal set do not propagate the optimality information and become a burden for the IVP solver still integrating them. This is why we may want to include *trimming* in the previous algorithm. This means finding a criteria to close suboptimal trajectories.

Target-based trimming A first method to trim trajectories is to close every trajectory too far away from target when little time is left. More precisely, if the target $\mathbf{x}_f$ shall be reached no later than T, and $v_{f,\max}$ is the maximum flow field magnitude, then close an extremal located at $\mathbf{x}(t)$ at time t if the distance to the target is greater than the distance that would be achieved with maximum tailwind over the given duration, that is, if:

$$\left(v_{f,\max} + v_r\right)(T - t) < \left\|\mathbf{x}(t) - \mathbf{x}_f\right\| \tag{3.14}$$

This is reminiscent of the heuristic part of the cost in the A star algorithm that guides the search to the target based on the distance to target.

Cycles-based trimming A trimming method based on cycles detection and eviction is described in Rhoads et al. 2013. The principle is to build the line collection formed by all segments joining neighboring extremals and compute its self-intersections. Intersection points cut the closed line collection into sub-pieces and one has to find out which pieces fall inside the reachability front to trim them away.

Alpha-shape trimming To try to be exhaustive, we mention that it is also possible to try to trim the extremal front through geometric procedures. While the 2D case enable us to keep track of the extremal relative ordering for trimming, in the 3D case, the latter is much more difficult. To overcome the difficulty, Sharp and Ross proposed the use of alpha shapes in the trimming procedure (Sharp and Ross, 2015). Alpha shapes are a tool to reconstruct a shape from a set of points, allowing concavities down to a given radius α (see Edelsbrunner and Mücke 1994).

Conclusion of the chapter

This chapter built algorithms on top of extremals to compute time-optimal trajectories for navigation problems with guarantees of optimality. We showed finding time-optimal trajectories in our framework reduces to building the appropriate extremal trajectory by a shooting method, with a single parameter to adjust in the initial condition. We showed the uniform sampling of test parameters is highly inefficient to find the optimal solution even on simple analytical problems. The dynamically adjusted sampling of initial parameter values performs much better. However, it reaches a limit when the precision on the initial parameter becomes so small that numerical errors propagate in the shooting method and falsify the computed extremal trajectory. To overcome this difficulty, we implemented a dynamical sampling where the initial condition is set between neighboring extremals rather than at the start position. This dynamical sampling technique proved efficient to compute time-optimal trajectories in real-world flow field data, when benchmarked against a level-set based solver. We finished by proposing heuristics to reduce the search space for extremals.

In our development, we always considered that the space was free of obstacles. We sometimes closed extremals reaching the borders of the problem, but it is not equivalent as considering the borders of the problem as an obstacle since we did not implement a way for the trajectories to follow the borders. Thus, in the next chapter, we will see how to extend our algorithms to the presence of hard obstacles, whether still or moving.

Chapter's main questions – Answers

- How to compute efficiently trajectories that minimize the travel time in the long-range navigation problem?

  Necessary conditions for time-optimality characterize trajectories in an infinitesimal way, i.e. *enable the writing of a system of ordinary differential equations that must be satisfied by time-optimal trajectories. Sampling candidate trajectories from such a system of ODEs with a shooting method is an efficient way to find the optimum of a problem, if an appropriate sampling procedure is used.*

- Can we provide guarantees that a given trajectory is indeed a global optimum of the problem?

  By sampling a sufficiently representative set of extremal trajectories, one can build the optimal cost function (or value function) of the navigation problem and prove global optimality of solutions.

- What are the typical shapes of time-minimizing trajectories?

  For weak flow fields, time-optimal trajectories stay close to the straight line between origin and destination. When the flow field magnitude increases, depending on its space-time structure, time-optimal trajectories can distort significantly away from the straight line.

Chapter 4

Time-optimal navigation problem: Obstacles

Abstract

In this chapter, the changes brought by the addition of obstacles in Zermelo's navigation problem in unsteady and possibly strong flow fields are studied. In the presence of obstacles, the optimality conditions satisfied by the trajectories candidate to optimality (extremals) are derived from Pontryagin's Maximum Principle in non-degenerate Gamkrelidze form. The best algorithm from the previous chapter is modified to handle obstacles. The resulting algorithm is validated on analytical flow fields and on real wind fields. The algorithm proves able to find time-optimal trajectories in all the considered cases. However, the algorithm cannot build the optimal cost map everywhere in space due to the shadowing effect of moving obstacles.

Résumé en français

Dans ce chapitre, les changements apportés par l'ajout d'obstacles au problème de navigation de Zermelo dans des écoulements instationnaires et possiblement forts sont étudiés. En présence d'obstacles, les conditions d'optimalité satisfaites par les trajectoires candidates à l'optimalité (extrémales) sont dérivées du principe du maximum de Pontryagin sous la forme de Gamkrelidze non dégénérée. Le meilleur algorithme du chapitre précédent est modifié pour prendre en compte les obstacles. L'algorithme résultant est validé sur des champs d'écoulement analytiques et sur des champs de vent réels. L'algorithme s'avère capable de trouver des trajectoires optimales en temps dans tous les cas considérés. Cependant, l'algorithme ne peut pas construire la carte de coût optimal partout dans l'espace en raison de l'effet d'ombrage des obstacles en mouvement.

Contents

Real world problems usually come with obstacles that need to be avoided in the planning of trajectories. Problem borders, land coastlines for AUVs, hazardous zones, forbidden zones are examples of obstacles that one may want to consider. These obstacles can be still or time-varying.

In what follows, we will explain how to adapt the algorithms from Chapter 3 to the presence of obstacles.

Chapter's main questions

- How to compute efficiently trajectories that minimize the travel time in the long-range navigation problem, when there are strictly forbidden zones in the environment?
- Can we provide guarantees that a given trajectory is indeed a global optimum of the problem in this case?
- How do extremals behave for the time-optimal navigation problem in the presence of state constraints?

4.1 Maximum principle

Obstacles are very challenging for the definition of extremals. When constraints on the state variable are present in an optimal control problem, one cannot use the standard form of the maximum principle anymore. To overcome the difficulty, other statements of the maximum principle have been proposed, for instance the Gamkrelidze or the Dubovitskii–Milyutin forms. A comparison of these two approaches is proposed in Arutyunov and Karamzin (2020). As regards the addition of state constraints in the Zermelo problem, Chertovskih et al. have proposed a semi-analytical resolution for planar problems (Chertovskih et al., 2021) as well as 3D problems (Chertovskih et al., 2020) that makes use of the Gamkrelidze form in the resolution. The Gamkrelidze form is the one we are going to use in what follows, because it relies on smoothness assumptions on the obstacles that we can afford to make, and it has a simpler formulation that the Dubovitskii–Milyutin.

Let $(B_i(t))_{1\leq i\leq n_{\text{obs}}} := \{\mathbf{x} \mid g_i(t,\mathbf{x}) \geq 0\}$ be the collection[1] of time-varying obstacles with g_i

[1] The references, in particular (Arutyunov and Karamzin, 2020), unfortunately use a reverse convention for obstacles $g_i(t,\mathbf{x}) \geq 0$ as the one we stated in the introduction with $\varphi(t,\mathbf{x}) \leq 0$. In this chapter, we will stick to the convention $g_i(t,\mathbf{x}) \geq 0$ so that the transition with the references is straightforward. In DABRY, the convention is the contrary.

smooth and let the constrained TNP be

$$\text{(C-TNP)}\begin{cases} \min\limits_{\mathbf{u}(\cdot)\in\mathcal{U}} t_f \\ \dot{\mathbf{x}}(t) = \mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x}(t)) \\ \mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f \\ \varphi_i(t, \mathbf{x}(t)) \leq 0, \quad i = 1, \dots, n_{\text{obs}} \end{cases} \tag{4.1}$$

Following Arutyunov and Karamzin 2020, Definition 2.10, we are going to consider trajectories satisfying Pontryagin's Maximum Principle in non-degenerate Gamkrelidze's form (PMP-G). They are ensured to be candidate solutions of Problem 4.1 thanks to Arutyunov and Karamzin 2020, Theorem 2.5, in the sense that any solution of Problem 4.1 satisfies the PMP-G. We continue to call them *extremals*.

We introduce as many real multipliers $\boldsymbol{\mu} := (\mu_1, \dots, \mu_{n_{\text{obs}}})$ as obstacle functions and we write an augmented Hamiltonian for this problem (Arutyunov and Karamzin 2020, p. 707):

$$H(t, \mathbf{x}, \mathbf{p}, \mathbf{u}, \boldsymbol{\mu}, \lambda) := \left(\mathbf{p} - \sum_i \mu_i \frac{\partial g_i}{\partial \mathbf{x}}^\intercal\right)^\intercal (\mathbf{u} + \mathbf{v}_f) - \sum_i \mu_i \frac{\partial g_i}{\partial t} + \lambda$$

The costate evolution $\dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{x}}^\intercal$ is also changed:

$$\dot{\mathbf{p}} = -\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}^\intercal \left(\mathbf{p} - \sum_i \mu_i \frac{\partial g_i}{\partial \mathbf{x}}^\intercal\right) + \sum_i \mu_i \left(\frac{\partial^2 g_i}{\partial \mathbf{x}^2}(\mathbf{u} + \mathbf{v}_f) + \frac{\partial^2 g_i}{\partial t \partial \mathbf{x}}^\intercal\right) \tag{4.2}$$

The Hamiltonian minimization w.r.t. the control variable $\mathbf{u}$ still leads to a closed form for the optimal control:

$$\mathbf{u}^\star = -\frac{\mathbf{p} - \sum_i \mu_i \frac{\partial g_i}{\partial \mathbf{x}}^\intercal}{\left\|\mathbf{p} - \sum_i \mu_i \frac{\partial g_i}{\partial \mathbf{x}}^\intercal\right\|} \tag{4.3}$$

For $(\mathbf{x}(\cdot), \mathbf{p}(\cdot), \mathbf{u}(\cdot), \boldsymbol{\mu}(\cdot), \lambda)$ to satisfy the PMP-G, $\boldsymbol{\mu}(\cdot)$ has to be a decreasing function (meaning each component decreases). Because of Arutyunov and Karamzin 2020, Eq. 24, μ_i does not vary where $g_i(t, \mathbf{x}) < 0$. This means that the only time windows where μ_i is allowed to decrease

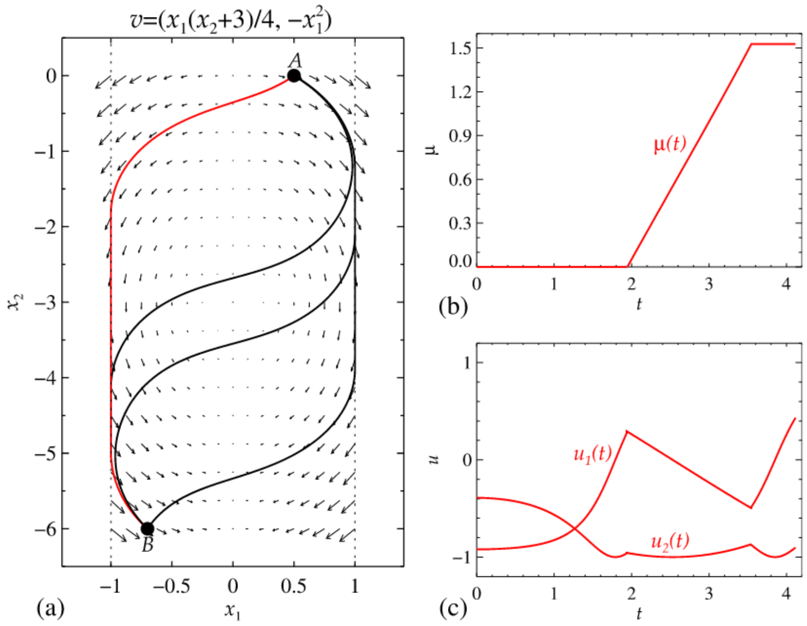


Figure 4.1: Extremals from the PMP-G with a problem constrained on the x_1 axis. (a) The flow field is represented as well as extremals in black and red. (b) For the red extremal, the difference of multipliers $\mu := \mu_1 - \mu_2$ is plotted, where μ_1 is associated to the right border and μ_2 to the left border. (c) For the red extremal, control values are displayed. From Chertovskih et al. (2021).

are when the trajectory touches the border of an obstacle, in the sense that $g_i(t, \mathbf{x}) = 0$. This is illustrated in Fig. 4.1. In the figure, different extremals are plotted with their corresponding control values. The difference in the μ_i multipliers is noted $\mu := \mu_1 - \mu_2$. Multiplier μ_1 is associated to the right border and μ_2 to the left border, hence μ increases on the graph even though the individual μ_i can only decrease in the PMP-G.

Thus, we will distinguish the behavior of extremals between the *free* mode and the *constrained* mode, whether the extremal lies out of obstacle boundaries or on it.

This PMP-G formulation encapsulates symmetries. Indeed (see Arutyunov and Karamzin 2020, p. 708), the PMP-G is invariant by the transformation:

$$\begin{aligned} &\mathbf{p} \leftarrow \mathbf{p} + \sum_i a_i \frac{\partial g_i}{\partial \mathbf{x}}^{\top} \\ &\forall i, \quad \mu_i \leftarrow \mu_i + a_i \end{aligned} \tag{4.4}$$

The previous remark motivate the definition of an alternate costate vector by:

$$\mathbf{q} := \mathbf{p} - \sum_i \mu_i \frac{\partial g_i}{\partial \mathbf{x}}^\intercal$$

This alternate costate vector is now insensitive to the previous symmetry. And because of the symmetry, we can always suppose that the μ_i coefficients start at 0, $\forall i, \;\; \mu_i(t_0) = 0$.

At times where the alternate costate expression is differentiable w.r.t. time, we have:

$$\dot{\mathbf{q}} = \dot{\mathbf{p}} - \sum \mu_i \left(\frac{\partial^2 g_i}{\partial t \partial \mathbf{x}}^\intercal + \frac{\partial^2 g_i}{\partial \mathbf{x}^2}(\mathbf{u} + \mathbf{v}_f) \right) - \sum \dot{\mu}_i \frac{\partial g_i}{\partial \mathbf{x}}^\intercal \tag{4.5}$$

The non-differentiable time stamps will typically be the non-differentiability time stamps of the μ_i (visible in 4.1), where a jump in the alternate costate will occur to satisfy the kinematic constraint of evolving on the border of obstacles (jump of the heading vector).

So, combining 4.5 with 4.2, we have:

$$\dot{\mathbf{q}} = -\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}^\intercal \mathbf{q} - \sum \dot{\mu}_i \frac{\partial g_i}{\partial \mathbf{x}}^\intercal \tag{4.6}$$

Out of obstacles, this is equation is simply the usual costate evolution Eq. 3.2 since $\dot{\mu}_i = 0$ when the extremal is not on the boundary of obstacle i. The optimal control now also has a simple expression:

$$\mathbf{u}^\star = -u_{\max} \frac{\mathbf{q}}{\|\mathbf{q}\|} \tag{4.7}$$

reminiscent of Eq. 3.3.

The application of the PMP-G gave similar building blocks for an algorithm as the standard PMP gave in the previous chapter. Thus, it is now possible to start describing a general procedure to find time-optimal trajectories out of extremals in the presence of obstacles.

4.2 Still obstacles

To derive an algorithm in the presence of obstacle, we will first suppose that the obstacles are *still* for the simplicity of explanations. So, in this section, the collection of obstacles is $(B_i)_{1\leq i\leq n_{\text{obs}}} := \{\mathbf{x} \mid g_i(\mathbf{x}) \geq 0\}$. In the following section (Sec. 4.3), we will explain how to adapt the algorithm to moving obstacles.

4.2.1 Hybrid integration

We will build an algorithm taking obstacles into account by modifying algorithms from Chapter 3.

We first notice that extremal legs in free mode are computed in a same fashion as in Chapter 3. With $\mathbf{y} = (\mathbf{x}\ \mathbf{q})^\intercal$, the free evolution is solution of:

$$\text{(FREE)} \quad \begin{cases} \dot{\mathbf{y}}(t) = \begin{pmatrix} -u_{\max} \frac{\mathbf{q}(t)}{\|\mathbf{q}(t)\|} + \mathbf{v}_f(t, \mathbf{x}(t)) \\ -\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\Big|_{t,\mathbf{x}(t)}^\intercal \mathbf{q}(t) \end{pmatrix} \\ \mathbf{y}(0) = \mathbf{y}_0 = \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{q}_0 \end{pmatrix} \end{cases} \tag{4.8}$$

Now, when an extremal hits a boundary for some t_{obs} at $\mathbf{x}(t_{\text{obs}}) = \mathbf{x}_{\text{obs}}$, say $g_i(\mathbf{x}_{\text{obs}}) = 0$ for some i, we have to change the time evolution of the extremal so that it follows the boundary of obstacle i. Rigorously, we shall use Eq. 4.6 and solve for the value of μ_i that leads to a ground speed colinear to the obstacle border.

But instead, we are going to make use of the directional time-optimal control $\mathrm{u}(\mathbf{d}, \mathbf{v}_f)$, defined in Prop. 2.2 to integrate trajectories on the border of obstacles. The evolution of a constrained leg is solution of:

$$
\text{(OBS)} \quad \begin{cases} \dot{\mathbf{x}}(t) = \mathrm{u}\left(R\left(\sigma\frac{\pi}{2}\right) \dfrac{\left.\dfrac{\partial g_i}{\partial \mathbf{x}}\right|_{\mathbf{x}(t)}^{\mathsf{T}}}{\left\| \left.\dfrac{\partial g_i}{\partial \mathbf{x}}\right|_{\mathbf{x}(t)}^{\mathsf{T}} \right\|}, \mathbf{v}_f(t, \mathbf{x}(t)) \right) + \mathbf{v}_f(t, \mathbf{x}(t)) \\ \mathbf{x}(t_{\text{obs}}) = \mathbf{x}_{\text{obs}} \end{cases} \tag{4.9}
$$

with the orientation given by the sign of σ:

$$
\sigma = \begin{cases} +1 \text{ if } \left(R\left(\frac{\pi}{2}\right) \frac{\partial g_i}{\partial \mathbf{x}}(\mathbf{x}_{\text{obs}})^{\mathsf{T}} \right)^{\mathsf{T}} \mathbf{u}(t_{\text{obs}}^-) > 0 \\ -1 \text{ else.} \end{cases}
$$

where $\mathbf{u}(t_{\text{obs}}^-) = \lim_{\substack{t \to t_{\text{obs}} \\ t < t_{\text{obs}}}} \mathbf{u}(t)$. The orientation constant σ is chosen at the contact with the obstacle depending on the control direction. This choice of orientation is a consequence of the link between the control, the costate, and the $(\mu_i)_{i \in \{1,\dots,n_{\text{obs}}\}}$ multipliers, as in Eq. 4.3, and the fact that each μ_i can only decrease. This is explained and depicted in Fig. 4.2. The alternate costate update at t_{obs}^+ is done remembering that the costate $\mathbf{p}$ is absolutely continuous thus continuous so $\mathbf{p}(t_{\text{obs}}^+) = \mathbf{p}(t_{\text{obs}}^-)$.

Rigorously, the σ parameter shall be allowed to change in time, depending on the sign of $\left(R\left(\frac{\pi}{2}\right) \left.\dfrac{\partial g_i}{\partial \mathbf{x}}\right|_{\mathbf{x}(t)}^{\mathsf{T}} \right)^{\mathsf{T}} \mathbf{u}(t)$. This corresponds to extremals doing a turn-around on the obstacle border. While they may exist in the general case, we choose from now on not to include them in the resolution, as it seems they are more the exception than the norm. Hence, σ does not depend on time in what follows.

Costate tracking in obstacle mode

What is noticeable in IVP 4.9 is that there is no alternate costate vector left. Indeed, the control is now entirely defined by the fact that the vehicle follows the obstacle border with a given orientation σ, so there is no need to track the alternate costate values. Nevertheless, it would

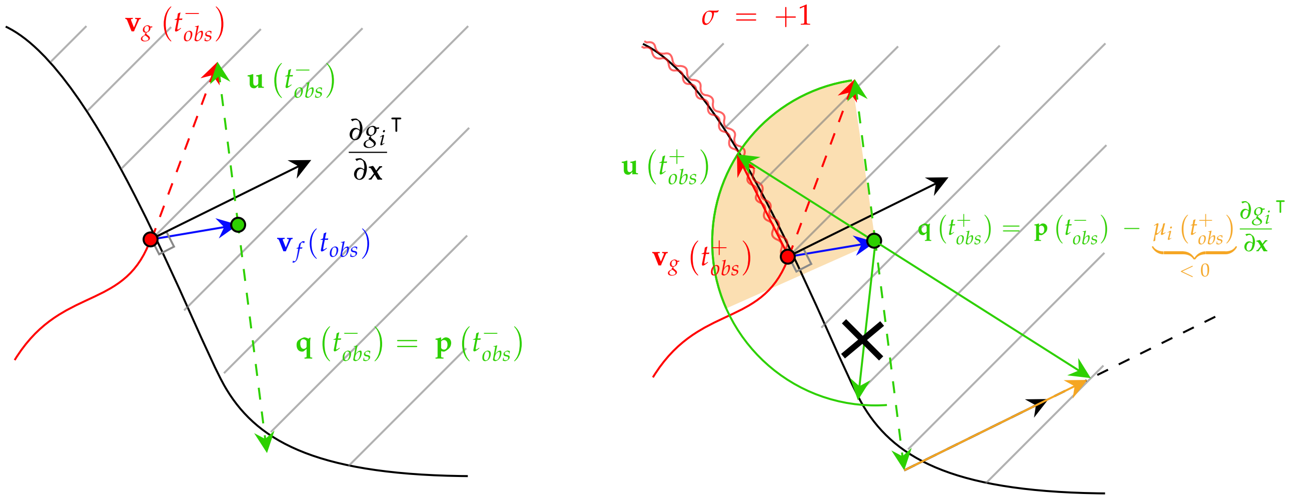

$\mathbf{v}_g\left(t_{obs}^-\right)$
$\mathbf{u}\left(t_{obs}^-\right)$
$\frac{\partial g_i}{\partial \mathbf{x}}^\mathsf{T}$
$\mathbf{v}_f\left(t_{obs}\right)$
$\mathbf{q}\left(t_{obs}^-\right) = \mathbf{p}\left(t_{obs}^-\right)$
$\sigma = +1$
$\mathbf{u}\left(t_{obs}^+\right)$
$\mathbf{v}_g\left(t_{obs}^+\right)$
$\mathbf{q}\left(t_{obs}^+\right) = \mathbf{p}\left(t_{obs}^-\right) - \underbrace{\mu_i\left(t_{obs}^+\right)}_{<0} \frac{\partial g_i}{\partial \mathbf{x}}^\mathsf{T}$

Figure 4.2: Left, obstacle collision. The control vector, the ground velocity vector and the alternate costate vector are displayed at a time t_{obs}^- infinitesimally close to t_{obs} and inferior to it. Right, the collision is resolved for a time t_{obs}^+ infinitesimally close to t_{obs} and superior to it. There are two possible controls to follow the obstacle. One of them is barred by a black cross because it is incompatible with the values that the μ_i multiplier can take. The yellow sectors shows the possible values for the control when μ_i varies from 0 to $-\infty$.

be possible to do so. On the obstacle border, the equation:

$$-u_{\max} \frac{\mathbf{p} - \mu_i \left.\frac{\partial g_i}{\partial \mathbf{x}}\right|_{\mathbf{x}}^\mathsf{T}}{\left\| \mathbf{p} - \mu_i \left.\frac{\partial g_i}{\partial \mathbf{x}}\right|_{\mathbf{x}}^\mathsf{T} \right\|} = \mathrm{u}\left(R\left(\sigma \frac{\pi}{2}\right) \frac{\left.\frac{\partial g_i}{\partial \mathbf{x}}\right|_{\mathbf{x}(t)}^\mathsf{T}}{\left\| \left.\frac{\partial g_i}{\partial \mathbf{x}}\right|_{\mathbf{x}(t)}^\mathsf{T} \right\|}, \mathbf{v}_f(t, \mathbf{x}(t)) \right) \tag{4.10}$$

has at most one solution in $\mu_i \in]-\infty, 0]$ (see Fig. 4.2). It has exactly one solution when the border is followable, *i.e.* the vehicle is neither forced inwards or outwards of the obstacle. In this case, μ_i is implicitly defined by System 4.10 and we note $\mu_i := m_i(t, \mathbf{x}, \mathbf{p})$ its solution. So μ_i must be initialized at value $m_i(t_{\text{obs}}, \mathbf{x}(t_{\text{obs}}), \mathbf{p}(t_{\text{obs}}))$. Then, dividing Eq. 4.10 by $(-u_{\max})$ leads to an equation of type:

$$\nu(\mathbf{q}(t)) = f(t, \mathbf{x}(t))$$

where $\nu(\xi) = \frac{\xi}{\|\xi\|}$ is the normalization function.

By differentiation w.r.t. time and using Eq. 4.6 with $\mu_j = 0,\ j \neq i$ one has:

$$\nu'(\mathbf{q}(t))\left(-\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}^{\mathsf{T}}\mathbf{q} - \dot{\mu}_i \frac{\partial g_i}{\partial \mathbf{x}}^{\mathsf{T}}\right) = \frac{\mathrm{d}}{\mathrm{d}t} f(t, \mathbf{x}(t))$$

where ν' is the jacobian matrix of the normalization function.

Finally:

$$\nu'(\mathbf{q}(t))\frac{\partial g_i}{\partial \mathbf{x}}^{\mathsf{T}}\dot{\mu}_i = -\nu'(\mathbf{q}(t))\frac{\partial \mathbf{v}_f}{\partial \mathbf{x}}^{\mathsf{T}}\mathbf{q} - \frac{\mathrm{d}}{\mathrm{d}t} f(t, \mathbf{x}(t)) \tag{4.11}$$

which correctly defines the multiplier evolution $\dot{\mu}_i$ as long as $\nu'(\mathbf{q}(t))\frac{\partial g_i}{\partial \mathbf{x}}^{\mathsf{T}}$ is not null. Because of the structure of the normalization function's jacobian matrix, $\nu'(\mathbf{q}(t))\frac{\partial g_i}{\partial \mathbf{x}}^{\mathsf{T}}$ is null if and only if $\mathbf{q}(t)$ and $\frac{\partial g_i}{\partial \mathbf{x}}^{\mathsf{T}}$ are aligned, which corresponds to a control vector pointing either fully towards or outwards of the obstacle. This is a degenerate case that can be neglected in a first approach, thus Eq. 4.11 provides a time evolution for μ_i.

So, the system evolution is entirely governed by the combination of the state evolution 4.9, the alternate costate evolution 4.6 and the multiplier evolution 4.11. This provides a way to keep track exactly of the costate vector, but in what follows we will proceed without tracking it, resorting directly to the use of the directional time-optimal control.

Continuity of the multipliers

In the PMP-G, the $(\mu_i)_{i\in\{1,\dots,n_{\text{obs}}\}}$ are continuous. This means that the trajectory hitting the obstacle in Fig. 4.2, is not an extremal of the problem, because $\mu_i(t_{\text{obs}}^-) = \mu_i(t_0) = 0$ and clearly a non-zero μ_i value is required at t_{obs}^+ to follow the border $\mu_i(t_{\text{obs}}^+) < 0$. So, the true extremals are only the trajectories entering obstacles with *a tangent ground velocity vector*.

But tangent trajectories are difficult to capture using the in-depth sampling algorithm. Indeed, in the finite collection of extremals shot in the algorithm, while there are usually several trajectories touching a given obstacle, it is not sure that some of them will be *tangent* to it, in the sense $\frac{\partial g_i}{\partial \mathbf{x}}\mathbf{v}_g \approx 0$. So, instead, we resort to using a tolerance a_{tol}, and decide to allow the entrance of the obstacle mode to all pseudo-tangent trajectories with $\left|\frac{\partial g_i}{\partial \mathbf{x}}\mathbf{v}_g\right| / \left(\left\|\frac{\partial g_i}{\partial \mathbf{x}}\right\| \left\|\mathbf{v}_g\right\|\right) < a_{\text{tol}}$. Em-

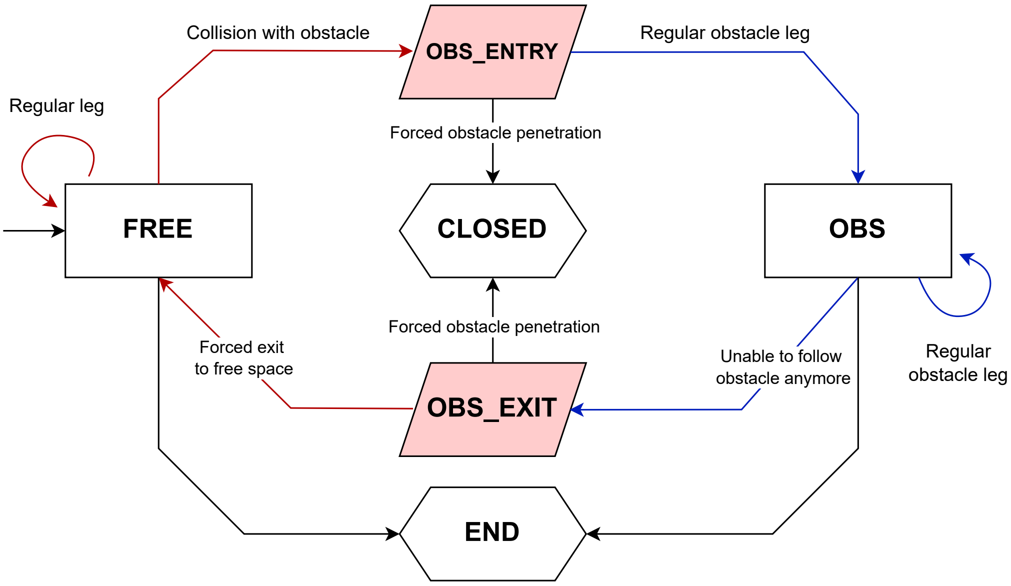


Figure 4.3: State machine for the integration of extremals in the presence of obstacles. The entry point is the 'FREE' state. Integration is performed using IVP 4.8 on red arrows and IVP 4.9 on blue arrows. Red parallelograms are transition states, meaning that the integration never stops in these cells, either resuming by switching integration mode or falling to the 'CLOSED' state. Hexagon states are terminal states.

pirically, the parameter a_{tol} was sometimes required to be as large as 0.7, allowing angles up to 45 deg, so that the algorithm does not trim out useful extremals.

State machine

We now have useful ingredients to build an algorithm able to deal with obstacles. We will integrate the extremals in time using a combination of free legs using IVP 4.8 and constrained legs using IVP 4.9. The general principle is summed up in Fig. 4.3.

4.2.2 Algorithm

The previous sections paved the way to adapt the in-depth interpolated sampling algorithm to the presence of obstacles. The new algorithm taking obstacles into account will be called the *in-depth interpolated obstacle-aware* algorithm. We do not provide a pseudo-code for it, as its principle is very close to the in-depth interpolated sampling Algorithm 4 but with technicality in the handling of the sub-cases that would make a pseudo-code not readable. Instead, we refer the reader to the Python implementation of the algorithm in the code of the DABRY module.

In the following sections, we describe the additional principles to build the in-depth interpolated obstacle-aware algorithm.

Interpolated resampling between FREE and OBS

If we are to apply the same principle as in Algorithm 4, we have to explain how to create an interpolated extremal between two neighbors with one in the free mode and the other in the obstacle mode. Let's suppose we are in Algorithm 4, line 22 and that $\mathbf{x}^-(t_\varepsilon)$ is in the free space while $\mathbf{x}^+(t_\varepsilon)$ lies on an obstacle, say $g_i(\mathbf{x}^+(t_\varepsilon)) = 0$. Then, we face the problem that we chose not to track the costate in obstacle mode for simplicity. We thus don't have a $\mathbf{q}^+(t_\varepsilon)$ to create $\mathbf{q}_{\text{interp}}$. But we have the control value $\mathbf{u}^+(t_\varepsilon)$ that constrains the direction of $\mathbf{q}^+(t_\varepsilon)$ thanks to Eq. 4.7. Still, we lack the norm of $\mathbf{q}^+(t_\varepsilon)$. So, we overcome the difficulty by defining an approximate alternate costate built from the direction of the control in obstacle mode and the norm of the costate in free mode $\widetilde{\mathbf{q}^+} := \|\mathbf{q}^-(t_\varepsilon)\| \frac{\mathbf{u}^+(t_\varepsilon)}{\|\mathbf{u}^+(t_\varepsilon)\|}$ and thus we take $\mathbf{q}_{\text{interp}} = \frac{1}{2}\left(\mathbf{q}^-(t_\varepsilon) + \widetilde{\mathbf{q}^+}\right)$.

Voluntary exits

In the PMP-G, extremals are allowed to exit the obstacle mode. If $\mathbf{y}(\cdot)$ satisfies the PMP-G up to t_{exit} and lies on obstacle i at t_{exit}, *i.e.* $g_i(\mathbf{x}(t_{\text{exit}})) = 0$, then denoting by $\mathbf{y}_{\text{FREE}}(\cdot)$ the solution of IVP 4.8 with initial condition $\mathbf{y}_{\text{FREE}}(t_{\text{exit}}) = \mathbf{y}(t_{\text{exit}})$ over $[t_{\text{exit}}, t_{\text{exit}} + \tau]$ (τ sufficiently small so that the trajectory lies in free space), the concatenation of $\mathbf{y}(\cdot)$ over $[t_0, t_{\text{exit}}]$ and $\mathbf{y}_{\text{FREE}}(\cdot)$ over $[t_{\text{exit}}, t_{\text{exit}} + \tau]$ still satisfies the PMP-G. But resuming the integration of IVP 4.9 with $\mathbf{y}_{\text{OBS}}(t_{\text{exit}}) = \mathbf{y}(t_{\text{exit}})$ integrated over $[t_{\text{exit}}, t_{\text{exit}} + \tau']$ and concatenating $\mathbf{y}(\cdot)$ and $\mathbf{y}_{\text{OBS}}(\cdot)$ leads to a trajectory also satisfying the PMP-G. This is depicted in Fig. 4.4. Both trajectories share the same ground velocity vector at t_{exit} but not the same curvature, due to the ODE being different.

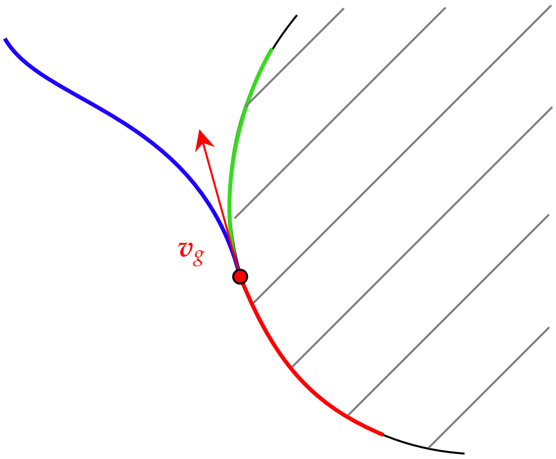


Figure 4.4: A captive leg (in red) that can be continued either by a free leg (in blue) or a captive leg (in green), depending if IVP 4.8 or IVP 4.9 is used.

In particular, in obstacle mode, $\dot{\mu}_i \neq 0$, but in free mode, $\dot{\mu}_i = 0$ which changes the dynamics of the alternate costate vector. We can see the change in $\dot{\mu}_i$ on obstacle exit in Fig. 4.1.

So, in the PMP-G, extremals are not only characterized by the initial costate $\mathbf{p}_0$, but also their sequence of exit durations. For instance, if we look for solutions of the problem with a maximum of 3 obstacle legs, then any extremal would be characterized by its initial costate $\mathbf{p}_0$ and a sequence (τ_1, τ_2, τ_3) of exit durations, with also the possibility $\tau_i = \infty$ if the extremal does not exit the obstacle.

This is a problem for the bisection principle: when two neighboring extremals have different initial costates $\mathbf{p}_0^-$ and $\mathbf{p}_0^+$ and exit durations τ_1^- and τ_1^+, how to set up the child trajectory? There is no reason that taking 'middles' leads to the creation of a 'median' extremal that increases the precision between parent extremals. Worse, there may be degeneration in the sense that multiple couples $(\mathbf{p}_0, \tau_1)$ entail extremals that are identical from the time they leave the first obstacle.

Fortunately, the interpolated resampling described in page 134 approximates exiting trajectories with good accuracy. So, in the algorithm, we do not consider voluntary exits and let the resampling procedure approximate these exiting trajectories.

Forced obstacle exits

Because of the flow field, an extremal can be forced out of an obstacle. In this case, the same problem as in page 134 arises: we have to provide an alternate costate vector initial value to initialize IVP 4.8. Its direction is opposite to the control vector, but its norm is lacking. In this case, we assume the extremal has a free neighbor, and use its norm to build the alternate costate in a similar fashion as in page 134. If it happens that the extremal only has neighbors in obstacle mode, then we use 1 as default norm. This may still be working fine in most cases when the costate is not evolving much in amplitude.

Neutering

We have to add a neutering screening in the algorithm to avoid bringing the bisection principle into dead ends. With $\left(\mathbf{p}_0^-,\ \mathbf{p}_0^+\right)$ a pair of neighboring costates, we denote by $\mathbf{y}^-(\cdot)$ the extremal associated to $\mathbf{p}_0^-$ and $\mathbf{y}^+(\cdot)$ the one associated to $\mathbf{p}_0^+$. We say that we *neuter* a pair of neighboring costates $\left(\mathbf{p}_0^-,\ \mathbf{p}_0^+\right)$ if we remove the latter from the list of costates considered for child creation (l'_{neigh} in Algorithm 4). Consequently, a neutered pair does not yield the resampling of a child extremal.

We neuter a pair of neighboring costates $\left(\mathbf{p}_0^-,\ \mathbf{p}_0^+\right)$ if the associated extremals enter obstacle mode with dissimilar orientation $\sigma^- \neq \sigma^+$. It has to be ensured because when a branching on the obstacle orientation occurs between neighboring extremals, their children would then alternatively have one or the other orientation, and thus very dissimilar trajectories that will never ensure the proximity condition Eq. 3.11. This leads to infinite resampling of trajectories without the CAVI going forward. Hence the need for neutering.

4.2.3 Experiments

In this section, we demonstrate the results of the in-depth interpolated obstacle-aware algorithm.

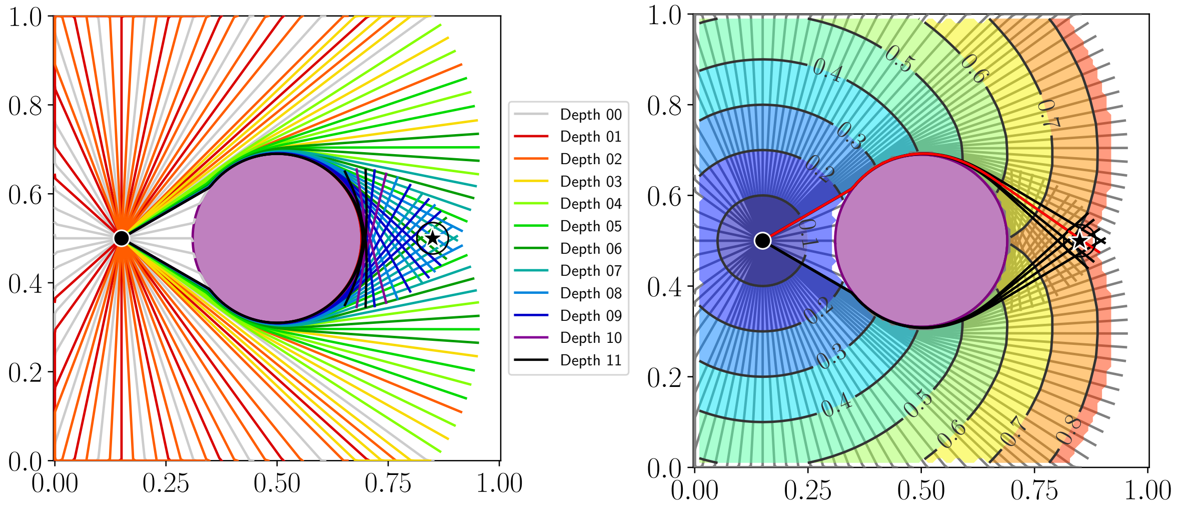


Figure 4.5: Disk obstacle. Left, the detail of the extremals shot, colored by depth in the in-depth interpolated obstacle-aware algorithm. It can be noticed that some grey, 0-depth trajectories reaching the obstacle are being closed, because their ground velocity vectors are not in the tangency tolerance. Right, the resulting value function as color contours, extremals reaching target in black and the solution extremal in red.

No flow

We first study the behavior of extremals with obstacles in the absence of flow field. We run the computation[2] in the presence of a disk obstacle (Fig. 4.5), two bars and a concave shape (Fig. 4.6). In each case, the extremals are able to find a consistent value function for the problems at stake. It must be noted that the tangency tolerance value shall be adjusted depending on the problems to avoid missing obstacle-captured trajectories.

With flow

Next, we see how the algorithm performs in the presence of a flow field. We run[3] the algorithm in the classical case of the linear flow field but with a barrier blocking the way. We also run again the two bars example but add adversarial flow field around the corners (Fig. 4.7) with

[2] The code can be found in the notebook `obstacle_simple.ipynb` at DOI 10.5281/zenodo.13939206.

[3] The code can be found in the notebook `obstacle_moving.ipynb` at DOI 10.5281/zenodo.13939206.

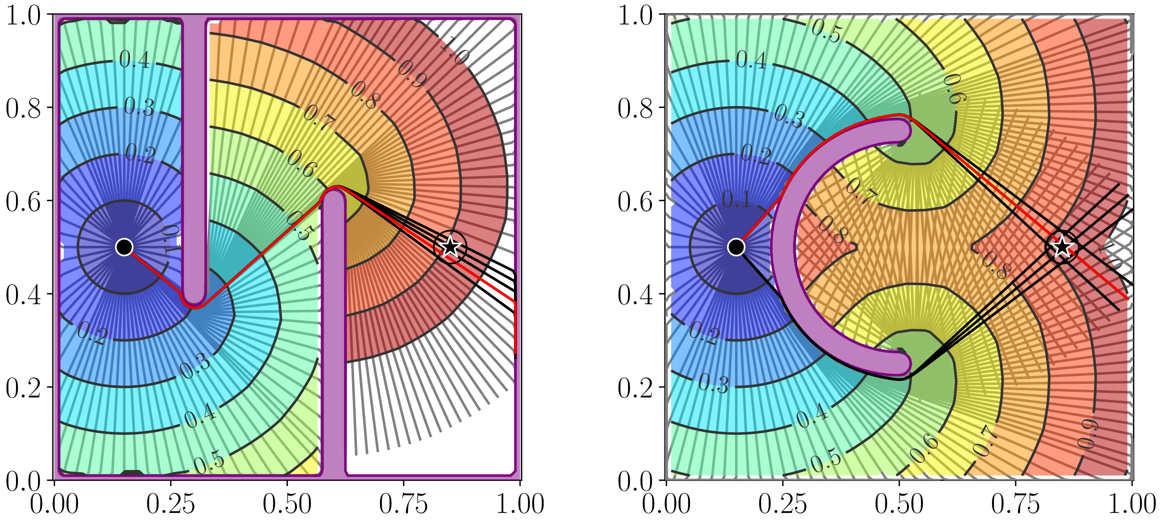


Figure 4.6: Two examples of shortest path computation using extremals in the absence of flow field with obstacles featuring concavities.

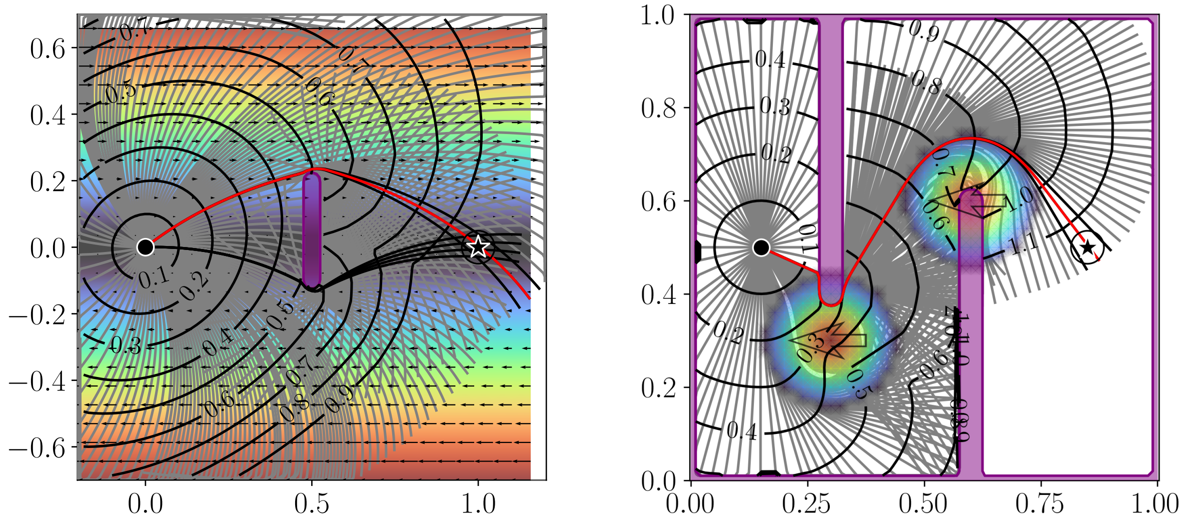


Figure 4.7: Two examples featuring a steady flow field and obstacles. Left, the classic linear flow but with a barrier in the middle. Right, the bars example with wind bubbles blocking the path around corners.

equation:

$$\mathbf{v}_f(\mathbf{x}) := \begin{pmatrix} -\max\left(r_1 - \|\mathbf{x} - \mathbf{x}_{c,1}\|, 0\right) \\ 0 \end{pmatrix} + \begin{pmatrix} -\max\left(r_2 - \|\mathbf{x} - \mathbf{x}_{c,2}\|, 0\right) \\ 0 \end{pmatrix}$$

with the centers $\mathbf{x}_{c,1} = (0.3\ 0.3)^\intercal$, $\mathbf{x}_{c,2} = (0.6\ 0.6)^\intercal$ and the radiuses $r_1 = r_2 = 0.15$.

Again, the algorithm shows consistent results. Computation times never exceed a minute per case on a laptop[4].

4.3 Moving obstacles

The preceding section presented the main modifications of the in-depth interpolated sampling algorithm to make it take still obstacles into account. The restriction to still obstacles was useful to give all the bases of the new algorithm without too much information.

In this section, we will explain the modifications to take time-varying obstacles into account. Obstacles are now described by time-space smooth functions $(B_i(t))_{1 \leq i \leq n_{\text{obs}}} := \{\mathbf{x} \mid g_i(t, \mathbf{x}) \geq 0\}$.

4.3.1 Directional offset time-optimal control

When obstacles are time-varying, staying on the border of the obstacle is not simply having a ground velocity colinear to the border, because of the obstacle's own velocity.

If $\mathbf{x}(t)$ lies on the border of obstacle i, then $g_i(t, \mathbf{x}(t)) = 0$, which we differentiate w.r.t. to get:

$$\left.\frac{\partial g_i}{\partial t}\right|_{t,\mathbf{x}(t)} + \left.\frac{\partial g_i}{\partial \mathbf{x}}\right|_{t,\mathbf{x}(t)} \big(\mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x}(t))\big) = 0$$

So, defining $\mathbf{d} = R\left(\frac{\pi}{2}\right) \frac{\frac{\partial g_i}{\partial \mathbf{x}}^\intercal}{\left\|\frac{\partial g_i}{\partial \mathbf{x}}^\intercal\right\|}$ and $\mathbf{n} = R\left(\frac{\pi}{2}\right) \mathbf{d}$, one has the equation:

[4] Intel® Core™ i5-10210U CPU @ 1.60GHz × 8

$$\mathbf{n}^\intercal \mathbf{u} = -\mathbf{n}^\intercal \mathbf{v}_f + \left\| \frac{\partial g_i}{\partial \mathbf{x}}^\intercal \right\|^{-1} \frac{\partial g_i}{\partial t}$$

which is reminiscent of Property 2.1. So, in a similar fashion as for Property 2.1, we define the *directional offset time-optimal control*:

Definition 14 (Directional offset time-optimal control)**.** *The directional offset time-optimal control for direction* **d** *and offset a is:*

$$\mathfrak{u}(\mathbf{d}, \mathbf{v}_f, a) := \begin{cases} \operatorname{argmax}_{\mathbf{u} \in \mathcal{U}} \mathbf{d}^\intercal \left(\mathbf{u} + \mathbf{v}_f(t, \mathbf{x}) \right) \\ \mathbf{n}^\intercal \left(\mathbf{u} + \mathbf{v}_f(t, \mathbf{x}) \right) = a \end{cases}$$

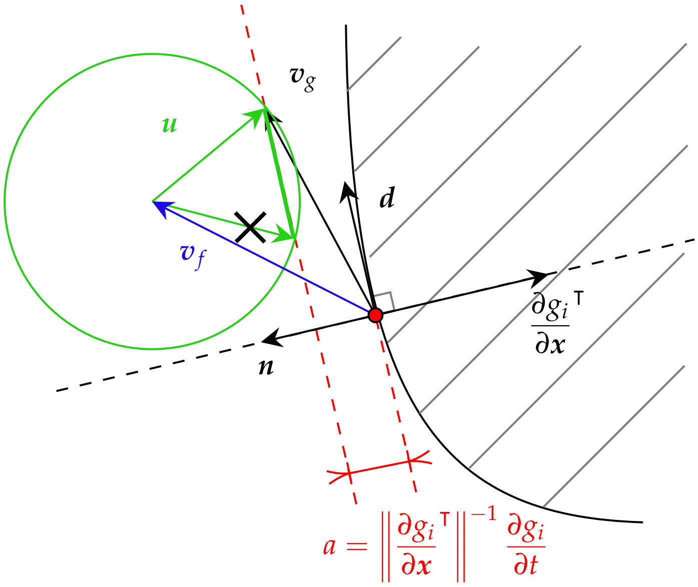
> **Remark**
> We use the same symbol as the directional time-optimal control because both can be discriminated against the other by the number of arguments. And we have the relation: $\mathfrak{u}(\mathbf{d}, \mathbf{v}_f) = \mathfrak{u}(\mathbf{d}, \mathbf{v}_f, 0)$.

The directional offset time-optimal control existence is found using the same ideas as in Property 2.1.

So, we see how to modify the algorithm for time-varying obstacles. The constrained legs will now be integrated using:

$$(\text{OBS-TV}) \quad \begin{cases} \dot{\mathbf{x}}(t) = \mathfrak{u} \left(R\left(\sigma \frac{\pi}{2}\right) \frac{\left. \frac{\partial g_i}{\partial \mathbf{x}} \right|_{t, \mathbf{x}(t)}^\intercal}{\left\| \left. \frac{\partial g_i}{\partial \mathbf{x}} \right|_{t, \mathbf{x}(t)}^\intercal \right\|}, \mathbf{v}_f(t, \mathbf{x}(t)), \left\| \left. \frac{\partial g_i}{\partial \mathbf{x}} \right|_{t, \mathbf{x}(t)}^\intercal \right\| \left. \frac{\partial g_i}{\partial t} \right|_{t, \mathbf{x}(t)} \right) + \mathbf{v}_f(t, \mathbf{x}(t)) \\ \mathbf{x}(t_{\text{obs}}) = \mathbf{x}_{\text{obs}} \end{cases} \tag{4.12}$$

The same rules hold as regards the interpolation of children, forced obstacle exits, neutering.



v_g
u
d
v_f
$\frac{\partial g_i}{\partial x}^\intercal$
n
$a = \left\lVert \frac{\partial g_i}{\partial x}^\intercal \right\rVert^{-1} \frac{\partial g_i}{\partial t}$

Figure 4.8: Directional offset time-optimal control to follow a moving obstacle. The green circle is $\{\mathbf{u} \mid \|\mathbf{u}\| = u_{\max}\}$. The green, thick segment are the values of the control respecting $\mathbf{n}^\intercal \left(\mathbf{u} + \mathbf{v}_f(t,\mathbf{x})\right) = a$, with a the obstacle 'local speed'. The directional offset time-optimal control is the green arrow that is not barred, achieving maximum ground speed in the direction $\mathbf{d}$. The resulting ground velocity is depicted in black. For this figure, $\frac{\partial g_i}{\partial t} > 0$, meaning the obstacle is extending in space.

4.3.2 Experiments: No flow case

We first test the obstacle-aware algorithm without flow field on moving obstacles. We test a moving disk (Fig. 4.9), a moving bar (Fig. 4.10) and a collection of moving objects (Fig. 4.11).

- For the moving disk case, the disk has radius 0.1. Its center starts at $(0.5\ 0.7)^\intercal$ and moves towards decreasing y with a speed of 0.4.
- For the moving bar case, the bar has width 0.05, length 0.4 and rounded corners. Its exact obstacle function is $g(t,\mathbf{x}) := \inf_{\mathbf{b}\in B(t)} \|\mathbf{x}-\mathbf{b}\| - 0.025$ with B the bar, *i.e.* $B = \{0.5\} \times ([0.5, 0.9] - v_b\, t)$ with speed $v_b = 0.4$. The bar is moving towards decreasing y.
- For the moving objects case or 'road crossing' example, three obstacle objects are moving. Each one of the three obstacles is a bar, with obstacle function $g_i(t,\mathbf{x}) := \inf_{\mathbf{b}\in B_i(t)} \|\mathbf{x}-\mathbf{b}\| - \frac{w_i}{2}$ for $i = 1,2,3$. $B_1(t) = \{0.5\} \times ([0.75, 0.85] - v_1\, t)$ with $v_1 = 0.5$, $B_2(t) = \{0.5\} \times ([0.45, 0.55] - v_2\, t)$ with $v_2 = 0.5$, and $B_3(t) = \{0.7\} \times ([0.3, 0.5] + v_3\, t)$

with $v_3 = 0.3$. The widths are all the same $w_1 = w_2 = w_3 = 0.05$. The total obstacle function is the minimum of all obstacle functions $g(t, \mathbf{x}) = \min_{i=1,2,3} g_i(t, \mathbf{x})$.

The algorithm manages to track extremals on the border of these moving obstacles. It also manages to resample interpolated extremals between free and captive extremals so that the reachability front is tracked by extremals. However, in all three cases, we see that the algorithm is not suited to resolve the value function in the wake of obstacles. For instance, Fig. 4.9, the value function is clearly wrong in the wake of the disk because there is a jump in the value function (for instance from the inter-level set between 0.3 and 0.4 and the one between 0.6 and 0.7 around point $(0.5 \;\; 0.6)$). This is because the first extremals passing in the zone are the one who performed a full loop around the obstacle, while in reality, the optimal commuting to $(0.5 \;\; 0.6)$ would be to follow an extremal reaching the obstacle, following the obstacle without momentum relative to it ('riding' the obstacle) and waiting to be 'dropped' at the right place. But this kind of behavior is not encompassed in the PMP-G as presented by Arutyunov and Karamzin (2020). Indeed, the maximum principle they derive is valid only if the endpoints (origin and destination) are *compatible* with the constraints (see Arutyunov and Karamzin 2020, Definition 2.4), which means in particular that the origin and destination lie out of obstacles. This explains why the extremals we sample are not capable of drawing the right value function in the wake of obstacles: if we were to place the destination point in a position that gets eclipsed by an obstacle at a time when it would be reachable without the obstacle, then the optimal trajectory to this destination is in general *not* an extremal in the sense of Arutyunov and Karamzin 2020, Definition 2.10 (the one we use).

The previous observation is not a problem as long as the destination stays out of obstacles, which for practical applications is a reasonable assumption: in a situation where the destination alternates between being forbidden and allowed, it seems that finding fastest routes to the destination can be less important than finding *safe* routes, so, in such a case, extremals are not the most relevant tool for the problem.

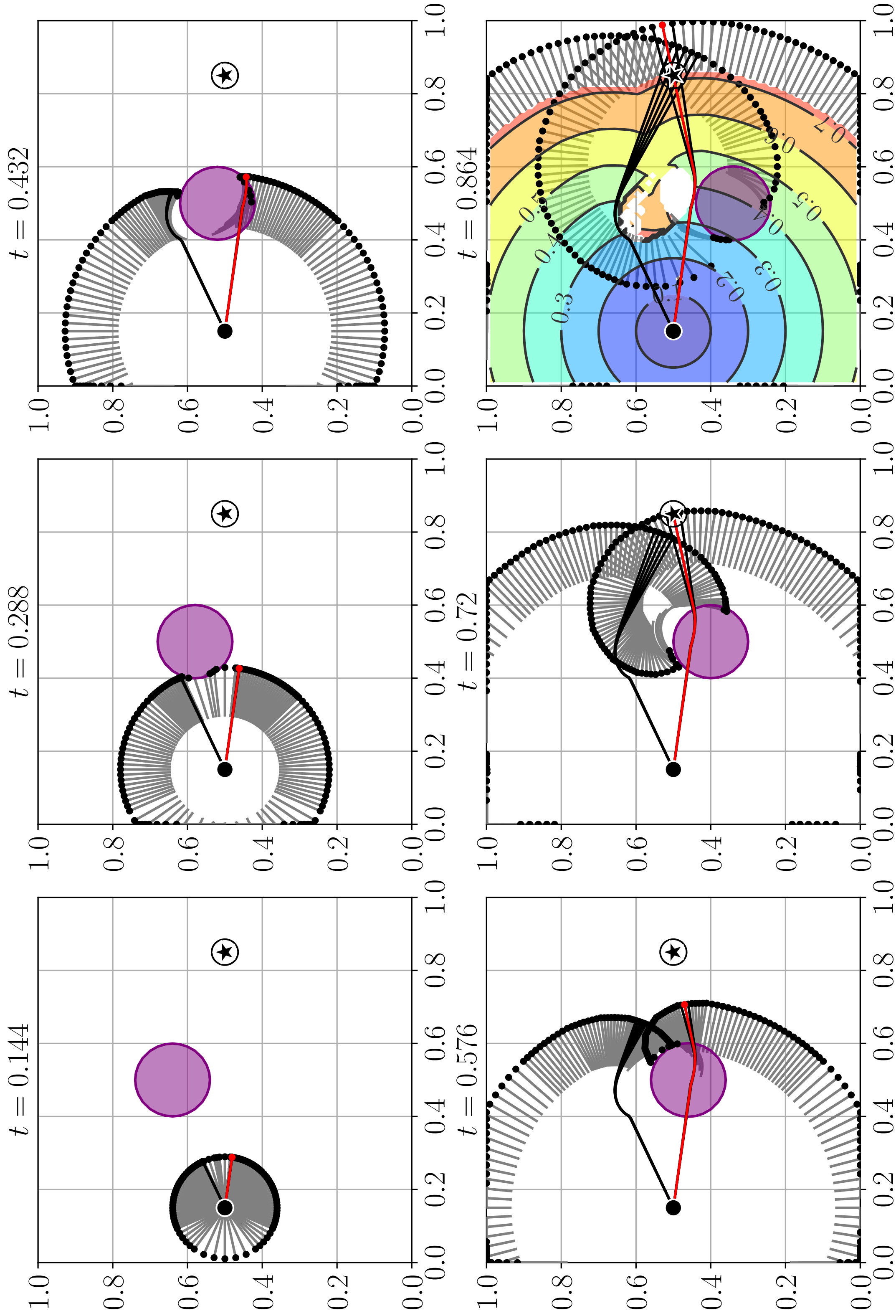


Figure 4.9: Moving disk obstacle. Extremals computed by the obstacle-aware in-depth interpolated sampling algorithm are depicted with their endpoints in black and their previous positions since the last frame as a grey line. The time-minimal trajectory is depicted in red. Other extremals reaching the target are depicted in black. On the last frame, the optimal cost map is displayed.

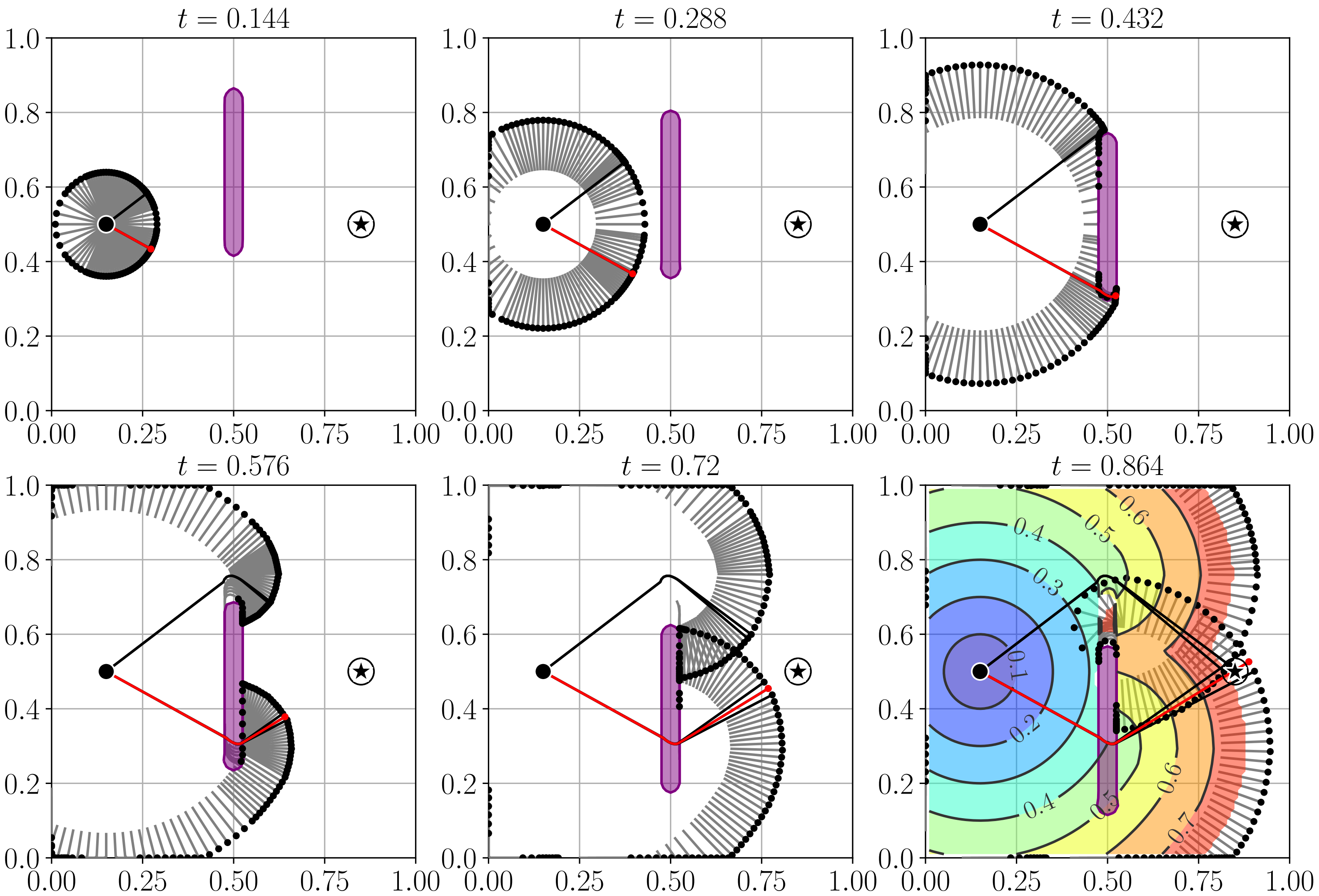


Figure 4.10: Moving bar obstacle. Extremals computed by the obstacle-aware in-depth interpolated sampling algorithm are depicted with their endpoints in black and their previous positions since the last frame as a grey line. The time-minimal trajectory is depicted in red. Other extremals reaching the target are depicted in black. On the last frame, the optimal cost map is displayed.

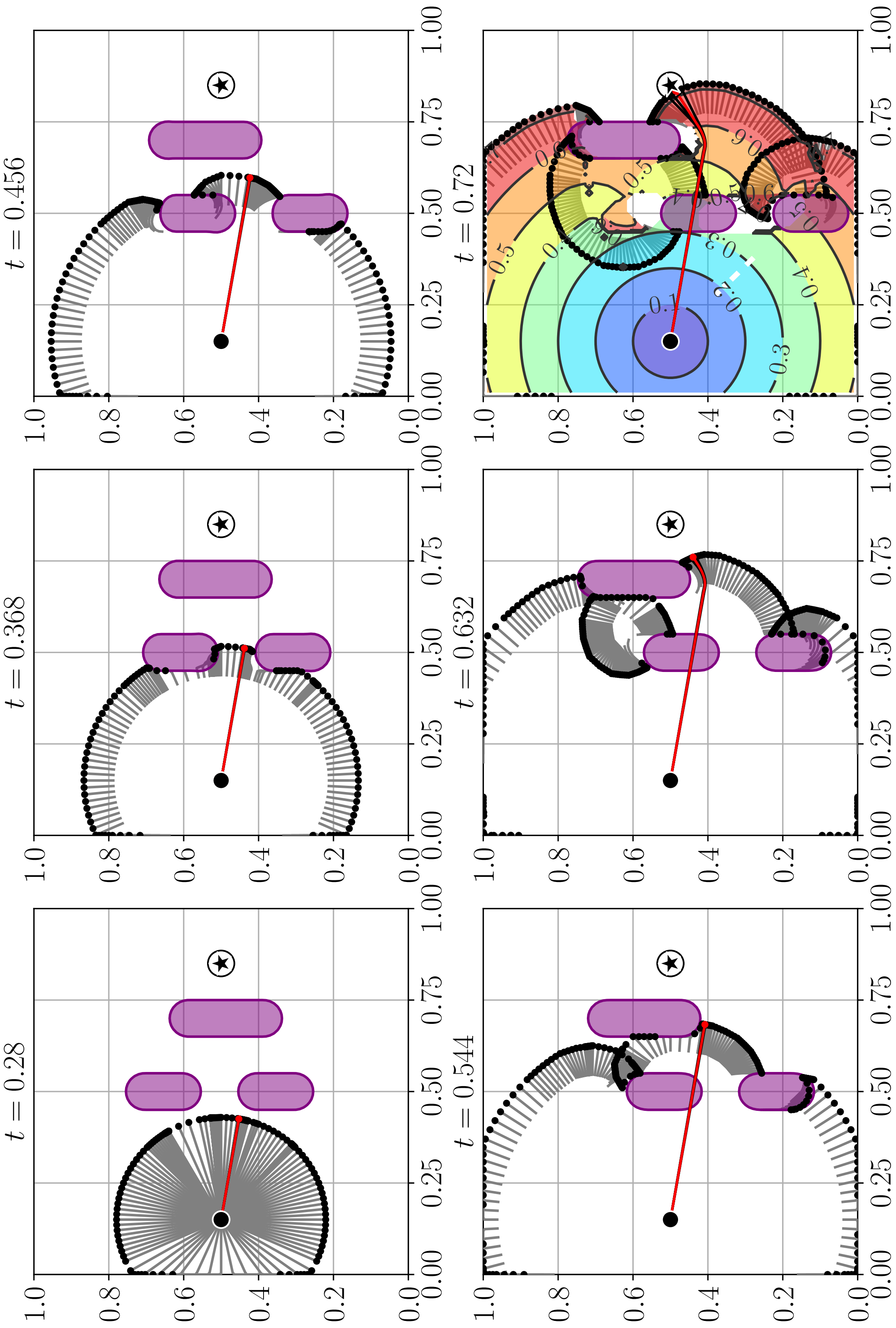


Figure 4.11: 'Road crossing' example. Extremals computed by the obstacle-aware in-depth interpolated sampling algorithm are depicted with their endpoints in black and their previous positions since the last frame as a grey line. The time-minimal trajectory is depicted in red. Other extremals reaching the target are depicted in black. On the last frame, we display the value function.

Fast obstacle

The case of a fast obstacle is also studied, 'fast' meaning the obstacle moves faster than the vehicle. Once again, a disk a radius 0.1 is considered, starting from position $(0.5\ 1.2)^\intercal$ and moving towards decreasing y with speed 1.4 greater than the vehicle speed which is 1. The results are depicted in Fig. 4.12. The optimal trajectory shows a sharp turn that corresponds approximately to the moment when the path to the target is clear from the moving disk. The sharp turn is a result of the resampling scheme that happens between a free extremal and another one captured on the border of the moving obstacle. On the last subfigure, the optimal cost map is displayed. Starting from the point where the turn happens (around $(0.4\ 0.6)^\intercal$) and going towards decreasing y, the optimal cost map is not resolved by the method on a band of width approximately 0.1. This illustrates once again the 'shadowing' effect of moving obstacles in the optimal cost map computed by extremals. The band where the optimal cost is not resolved indeed lies again in the wake of the obstacle. The information about the optimal cost in this band would have been brought by trajectories following the obstacle with non-maximal speed in the orthoradial direction in the obstacle frame, which are not part of the extremals set we consider, as explained for the previous examples. However, in the direction of the target, the propagation of extremals is consistent and the information about the optimal cost is resolved.

This example shows the ability of the method to deal with obstacles faster than the vehicle, in the limit of the shadowing effect of moving obstacles.

4.3.3 Experiments: Active flow case

Analytical flow field

We then test the algorithm with a moving obstacle immersed in the classical linear flow field (Fig. 4.13). Once again, the algorithm proves capable of computing the extremals. The obstacle moves sufficiently slowly so that the reachability front is well resolved behind the obstacle,

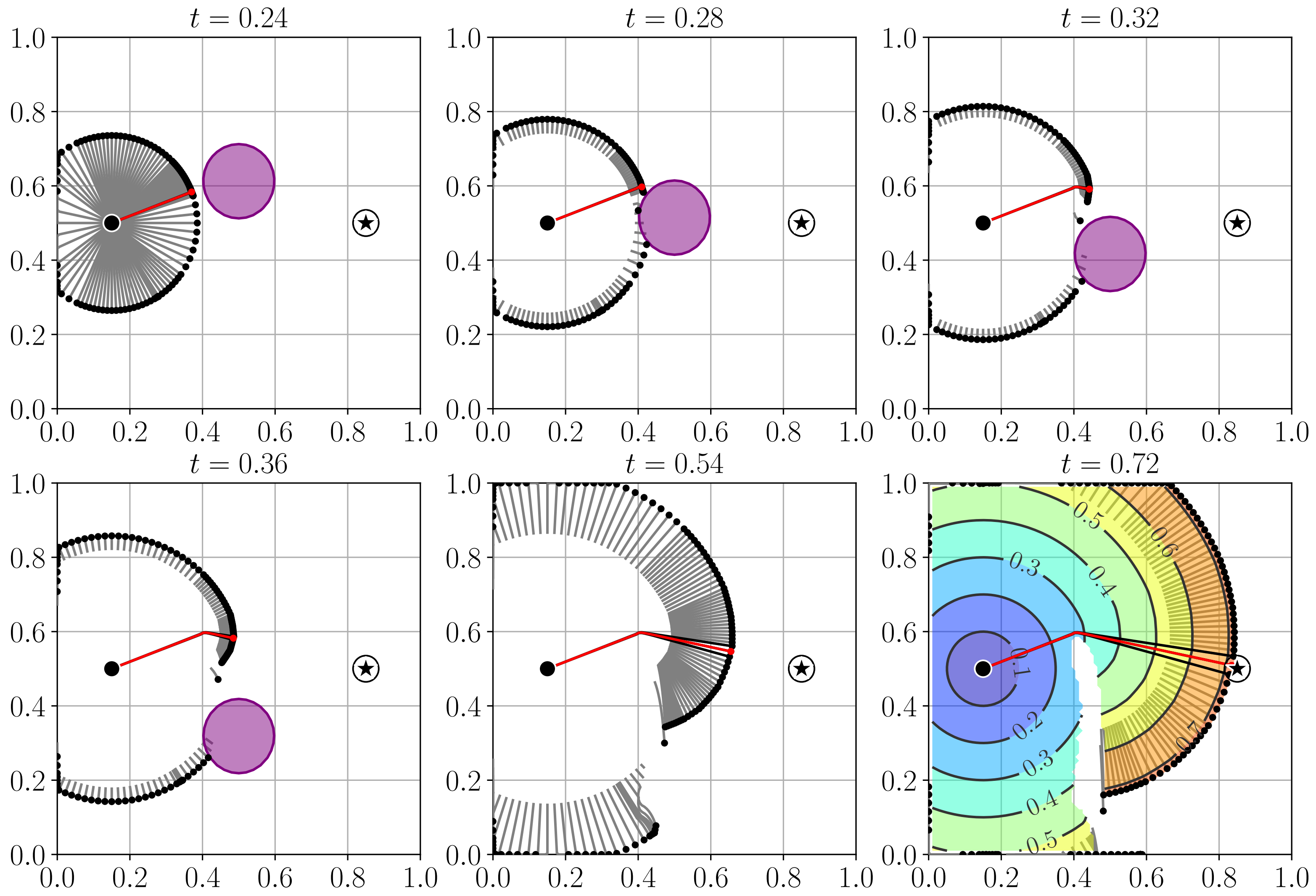


Figure 4.12: Fast disk obstacle. Extremals computed by the obstacle-aware in-depth interpolated sampling algorithm are depicted with their endpoints in black and their previous positions since the last frame as a grey line. The time-minimal trajectory is depicted in red. Other extremals reaching the target are depicted in black. On the last frame, the optimal cost map is displayed. The time difference is not the same between all subfigures. Between the first four subfigures, it is of 0.04 while for the last three it is of 0.18.

as opposed to the moving disk case. So, even though the PMP-G extremals are not made to resolve the value function in the wake of obstacles, in this case they manage to resolve it.

Real wind field

We now test the algorithm with real-world, time-varying wind and moving obstacles. For this, we extract the 100m wind reanalysis from ERA5 for Sep. 28, 2021 and a three days time window. We also extract the total precipitation reanalysis. This date corresponds to a creation of a storm between Dakar and Natal, so this constitutes challenging weather for the Mermoz drone. Storms create high winds that can be attractive for a time-optimal path planner. But going into the eye of the storm may lead to the loss of the vehicle, and this is where moving obstacles are going to be useful to forbid this zone. In this case, we forbid the access to the eye of the storm by defining a circle of radius 427 km (0.2 in rescaled space) around the position of maximum precipitation as a moving obstacle (Fig. 4.14). As the time evolution of the maximum is erratic, in the sense that this maximum can jump to another location at the next time frame, we smooth the sequence of centers by averaging positions over a 10 hours window (one time step every hour, so smoothing over 10 time steps). We want to see if the algorithm can solve the constrained problem, and if so, how the new optimum compares with the no-obstacle situation.

We project the wind and rain fields to planar space using a Lambert projection with standard parallels -7 deg and 14 deg, showing a maximum relative error in the distances of 1.7 % at the top and bottom borders of the problem (meaning measuring a distance on the Lambert map leads to a measure that is 1.7 % more than the real distance). A discussion on the difference between working in projected space or in spherical coordinates for this case is proposed in Appendix D.

We scale distances and speeds so that space and time coordinates are of the order of 1, for good numerical conditioning. The start date is chosen to Sep 29th, 2021, 06:00Z for the storm to hinder the UAV on its path.

We show the results in Fig. 4.15. The computation takes around a minute to complete on

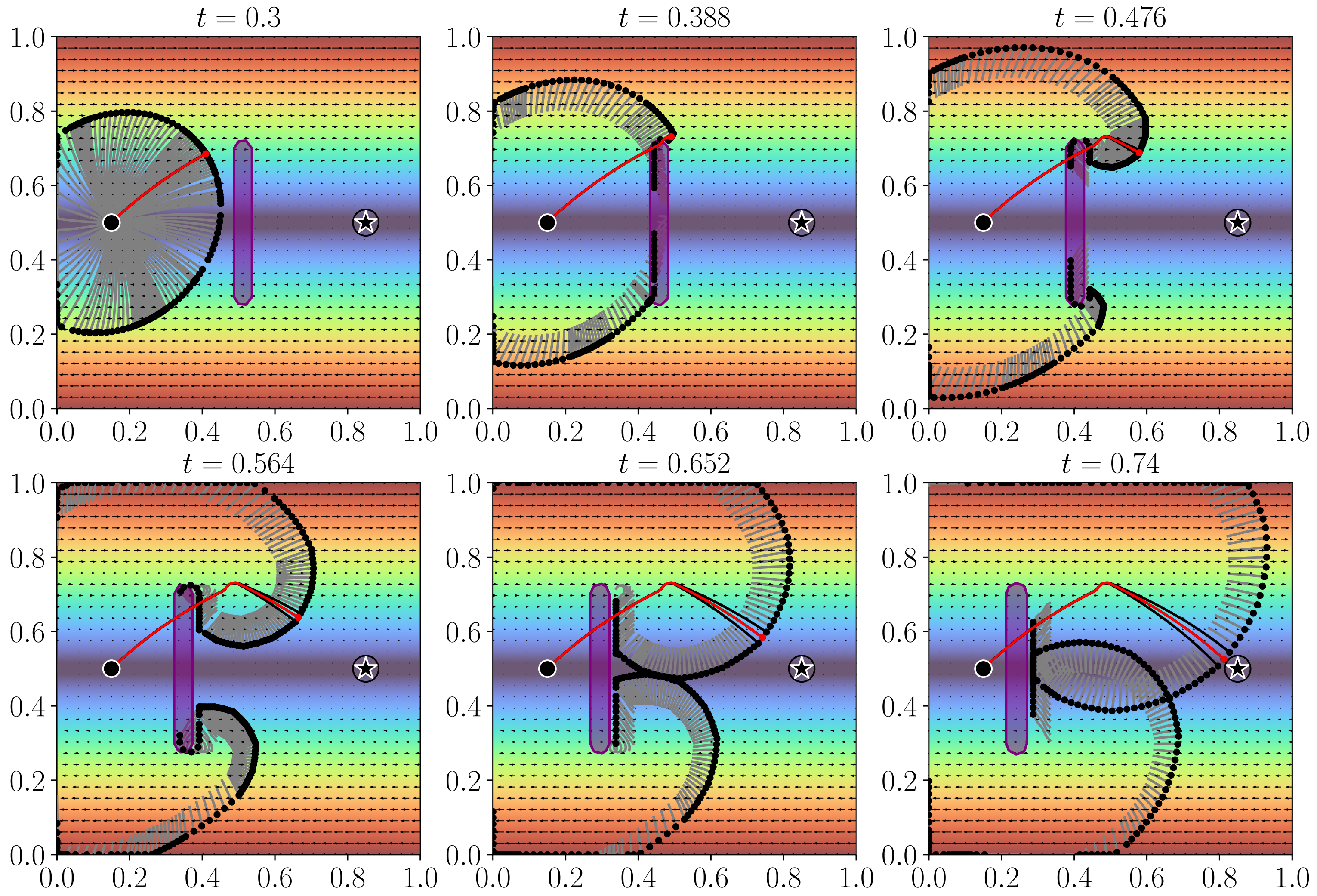
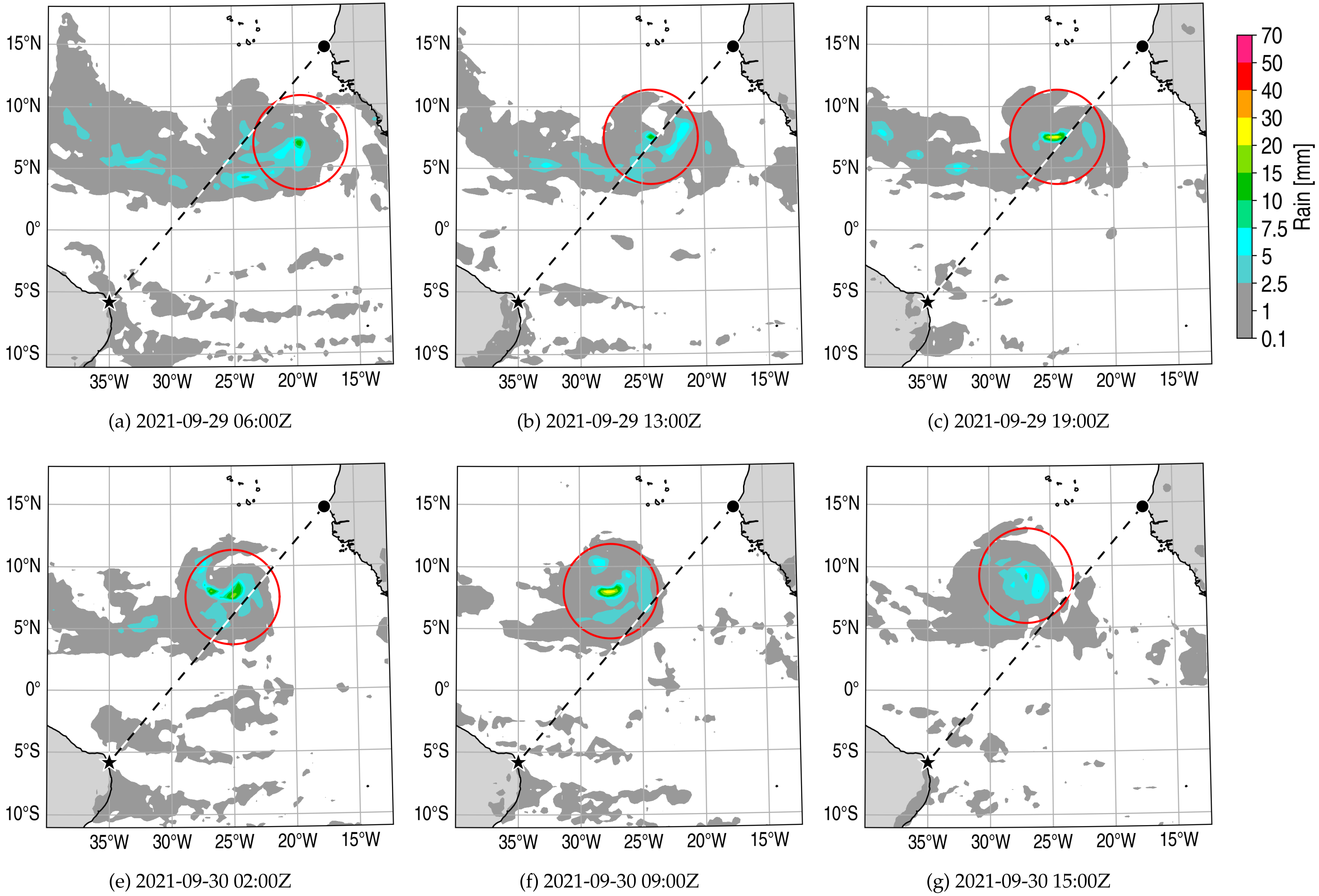


Figure 4.13: Moving bar in a linear flow field. Extremals computed by the obstacle-aware in-depth interpolated sampling algorithm are depicted with their endpoints in black and their previous positions since the last frame as a grey line. The time-minimal trajectory is depicted in red. Other extremals reaching the target are depicted in black. On the last frame, we display the value function.



(a) 2021-09-29 06:00Z

(b) 2021-09-29 13:00Z

(c) 2021-09-29 19:00Z

(e) 2021-09-30 02:00Z

(f) 2021-09-30 09:00Z

(g) 2021-09-30 15:00Z

Figure 4.14: ERA5 Reanalysis of the one hour accumulated rain field between Dakar and Natal. The maximum of precipitation is enclosed in a circle of radius 427 km.

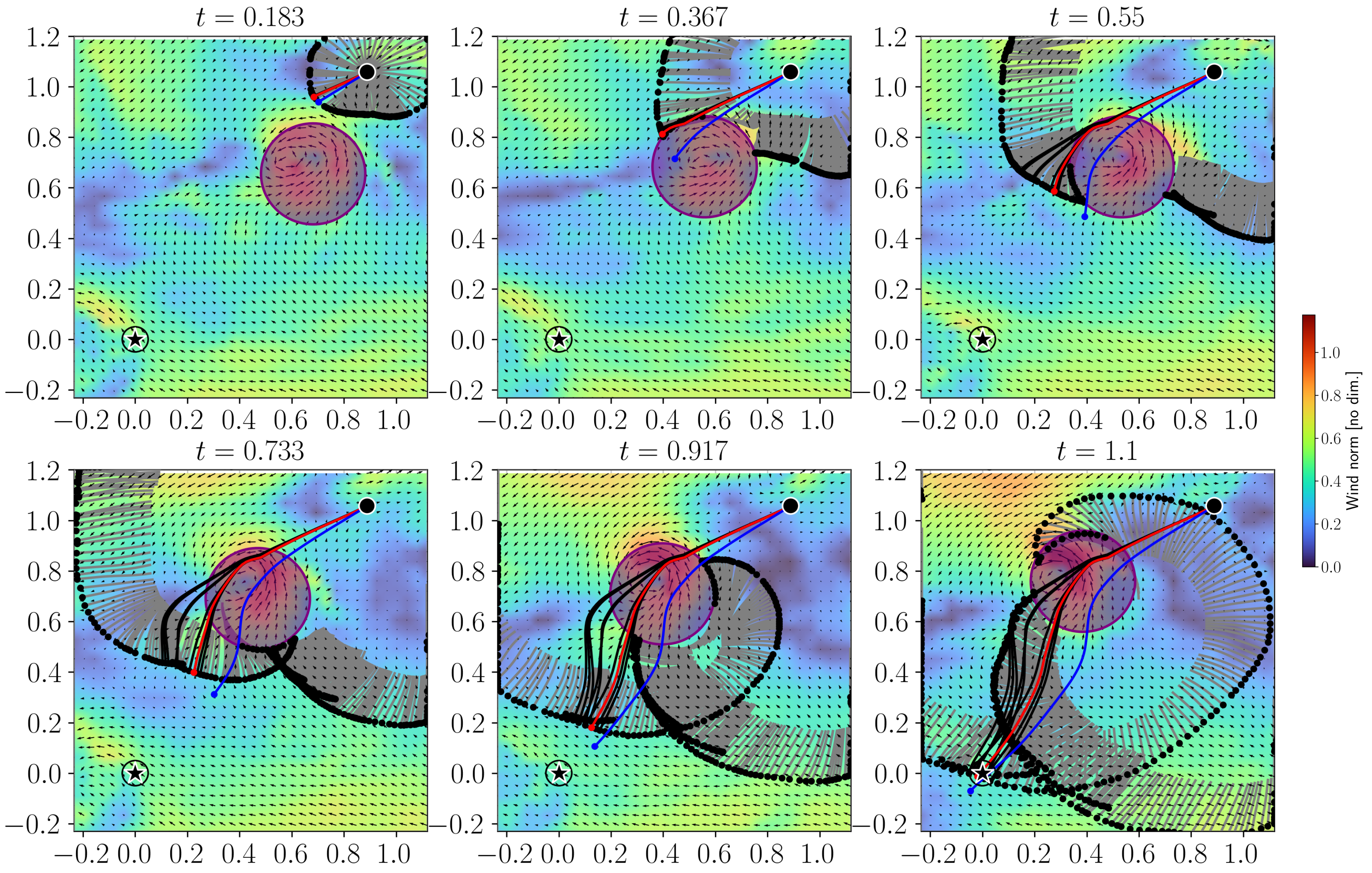


Figure 4.15: Storm forming between Dakar, Senegal (black dot, up right) and Natal, Brazil (star, down left). The optimal trajectory is depicted in red. Other extremals reaching the target are depicted in black. The optimal trajectory in the absence of obstacle is shown in blue.

a laptop[5]. The algorithm manages to find a solution in the constrained case. This solution makes use of the obstacle border. This example show the ability of the method to deal with real-world trajectory planning scenarios with constraints that can be formulated as moving forbidden zones.

> **Remark**
> This storm case was used in (Schnitzler et al., 2023a), but at the time the article was written, the DABRY module was not able to deal with moving obstacles so a fixed obstacle was used.

Conclusion of the chapter

We presented how to handle hard obstacles in time-optimal trajectory planning when using extremals. State constraints in Pontryagin's Maximum Principle are very challenging, still we gave some references trying to overcome the difficulty in trajectory planning problems. We explained a set of best-effort adaptations to add to the extremal-based time-optimal algorithm for it to work in the presence of obstacles. The resulting algorithm is capable of finding the time-optimal solutions of different analytical test problems and also on one real-world example as well. Still, the extremals we used are not appropriate if one is to compute the full value function of the navigation problem. Indeed, 'shadow zones', where no extremal propagate, appear around moving obstacles, where consequently no information on the optimal travel time is available. This is not a problem for time-optimal trajectory planning as long as the destination point lies out of moving obstacles.

With obstacles now included in the time-optimal trajectory planning procedure, one has then a tool for trajectory planning that tends to comprehensiveness. Yet, time may not be the only cost of interest in navigation problems. Indeed, focusing on the energy expense rather than on travel time changes the optimization procedure. It is studied in the following chapter.

[5] Intel® Core™ i5-10210U CPU @ 1.60GHz × 8

Chapter's main questions – Answers

- How to compute efficiently trajectories that minimize the travel time in the long-range navigation problem, when there are strictly forbidden zones in the environment?

  The exist adaptations of the necessary conditions of optimality for trajectories in the presence of obstacles, whether still or moving. From these conditions, extremals can be characterized, and an efficient method can be built by shooting extremals in a similar fashion as when obstacle are absent.

- Can we provide guarantees that a given trajectory is indeed a global optimum of the problem in this case?

  In a similar fashion as in the absence of obstacles, sampling extremals enables the building of the optimal cost map of a problem. However, extremals designed for navigation problems where the endpoints lie out of obstacles are consequently not able to propagate the information of the optimal travel time in the wake of obstacles. Different extremal trajectories, with free speed along obstacle borders (in obstacle frame) would be required to get this information.

- How do extremals behave for the time-optimal navigation problem in the presence of state constraints?

  In the presence of state constraints, extremals are composed of a succession of legs alternating between the free space and the borders of obstacles. Extremals designed for problems where the endpoints lie out of obstacles are using the maximum allowed speed at all times.

Chapter 5

Energy-time optimal navigation problem

Abstract

In this chapter, Zermelo's navigation problem in unsteady and strong flow fields is extended by adding the notion of energy. In the large scale point of view kinematic model, it is shown that the power function can be convexified. The power function can be made non-dimensional, with different properties whether the idling of the vehicle is costly or not. The properties of the non-dimensional two-terms drag polar power function for aircraft are thoroughly reviewed. Necessary conditions of optimality for trajectories are written using Pontryagin's Maximum Principle, characterizing the candidates to energy-time-optimality (extremals) for any kind of power function. Energy-time-optimal extremals are computed in the cases of a one-dimensional moving gust of wind, a bidimensional gyre flow and realistic wind fields over the Atlantic. In each of the previous cases, trade-offs between total travel time and total energy expense are studied. Tenths of percent reduction in energy consumption can be expected from using energy-time-optimal trajectories instead of time-optimal trajectories in favorable cases. Lastly, hints are given on how to adapt the previously built extremal-based algorithms to the energy-time-optimal case.

Résumé en français

Dans ce chapitre, le problème de navigation de Zermelo dans un champ de vent fort et instationnaire est étendu en ajoutant la notion d'énergie. Dans le modèle cinématique grande échelle, il est montré que la fonction de puissance peut être convexifiée. La fonction de puissance peut être adimensionnée, avec des propriétés différentes selon que le régime de vitesse grande échelle nul soit coûteux ou non. Les propriétés de la fonction de puissance du vol issue de la polaire en deux termes et adimensionnée sont examinées en détail. Les conditions nécessaires d'optimalité des trajectoires sont données à l'aide du principe du maximum de Pontryagin, caractérisant les candidats à l'optimalité en énergie et temps (extrémales) pour tout type de fonction de puissance. Les extrémales optimales en énergie et temps sont calculés dans les cas d'une rafale de vent mobile unidimensionnelle, d'un écoulement tourbillonnaire bidimensionnel et de champs de vent réalistes au-dessus de l'Atlantique. Dans chacun des cas précédents, les compromis entre le temps de déplacement total et la dépense énergétique totale sont étudiés. Dans les cas favorables, on peut s'attendre à une réduction de la consommation d'énergie de l'ordre de quelques dizaines de pourcents en utilisant des trajectoires optimales en énergie et temps plutôt que des trajectoires optimales en temps. Enfin, des indications sont données sur la manière d'adapter les précédents algorithmes basés sur les extrémales au cas de l'optimisation en énergie et temps.

Contents

In the previous chapters, we studied the problem of minimizing the travel time from an origin to a destination. We focused on travel time because it is the most simple metric for this kind of control problem, and it is of obvious interest for navigation problems. Nevertheless, we have seen that time-optimal navigation requires the vehicle to be operated at maximum SRF (Property 3.1). It is an energy-intensive way to operate the vehicle. Most navigation missions are not races and the most important goal is the fulfillment of the mission rather than sparing half an hour of travel time over 30 hours. Thus, studying the possibility of reducing and adapting the SRF to the flow field conditions is interesting to understand how much energy can be saved this way, and consequently how much margin can be gained on the expected final onboard energy at the end of the mission, concurring in reducing the risk of mission failure.

In this chapter, we will focus on adapting optimality conditions when the key metric is energy rather than travel time.

Chapter's main questions

- Is it beneficial for the vehicle to make its own speed vary throughout travel in terms of total energy expense?
- If so, what is the amount of energy that can be saved?
- How to compute numerically such energy-saving trajectories?
- How different are energy minimizing trajectories from time minimizing ones?

5.1 Literature review

The focus of this thesis is on long-range trajectory optimization. By nature, for long-range mission, autonomy and thus energy is a key metric. While time-optimal trajectory were the entry point in the literature for trajectory optimization, most authors also enlarge the problem by adding the energy dimension to it.

For underwater vehicles, Albarakati et al. (2020) used a control parameterization approach for time-energy trajectory optimization of AUVs in 3D underwater environment. The same type of methods was used by Bonnin (2015) to find albatross-like, energy-minimizing 3D trajectories for UAVs. Control parameterization was also used by González Arribas (2019) to perform time-energy optimization of aircraft trajectories. The approach proved efficient in all mentioned cases to find optimal trajectories, still control parameterization cannot guarantee the global optimality of the solutions found.

RRT methods were used by Rao and Williams (2009) to find AUV paths using less energy than constant SRF paths.

For 2D, time-energy optimal trajectory planning, Doshi et al. (2023) solve the PDE governing the vehicle's reachable set in an augmented space composed of the physical space and the energy dimension. They are able to build the Pareto front of time-energy pairs for a navigation problem, with guarantees of optimality. Though, Hamilton-Jacobi type methods scale badly with the dimension, and the 3D augmented state space increases much the computation effort.

For commercial aircraft, Franco and Rivas (2011) or Jafarimoghaddam and Soler (2023) use extremals to perform time-energy 2D trajectory optimization, on a more refined aircraft model than the kinematic model (Eq. 2.1). Extremal methods are also used by Dobrokhodov et al. (2020) for the long-range time-energy-optimal navigation of a liquid hydrogen powered UAV, which is very similar to the driving application case of this thesis (the Mermoz challenge). Commercial aircraft performance variation with airspeed and wind conditions were also investigated by Delgado and Prats (2013) from the air traffic control point of view. The authors showed reducing airspeed when approaching a congested airport can reduce delays and fuel, and that this is exacerbated in the presence of wind.

Without surprise, the diversity of methods illustrated in the literature review chapter is also found for the time-energy-optimal navigation problem. The following development still focuses on extremal methods for time-energy optimization. Thus, connections are made, when possible, with the aforementioned references belonging to the group of indirect methods.

5.2 Power expense

5.2.1 Kinematic model, large scale point of view

We want to study how energy is spent in the large-scale navigation of vehicles. We still want to keep the useful LSPOV that represents the vehicle with a kinematic model. We recall that the vehicle evolves in space with:

$$\dot{\mathbf{x}}(t) = \mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x}(t))$$

The control $\mathbf{u}$ is a 2D variable and its norm is called the Speed Relative to the Flow (SRF), which we note $v_r = \|\mathbf{u}\|$, while the direction vector is called the heading vector and we note $\mathbf{h} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$. The vehicle's energy is spent to counter the forces acting against the vehicle, resulting in a mechanical equilibrium that gives the vehicle an SRF of v_r. So, to maintain this equilibrium, the vehicle has to provide power to an amount that is governed by the SRF, noted $g_{\text{total}}(v_r)$. However, in what follows, we are going to focus on the pure mechanical power that is spent to counter the fluid's forces, which we note $g_{\text{SS}}(v_r)$. The difference with the previous one is that the total power expense (one can think fuel flow or current out of a battery) takes into account the different efficiencies involved in the power chain. By focusing on the mechanical power, we do not need a model of the full power chain, and we can provide a lower bound on the energy spent by the vehicle. The SS subscript means that it is the small scale power expense, as opposed to the power function seen by the LSPOV. The LSPOV indeed sees a different power expense than the small scale. We note $g(v_r)$ the LSPOV power function, without subscript as it is the one that is going to be used in the following sections. Because of the LSPOV abstraction, we have a relation between the small scale power function and the large scale power function.

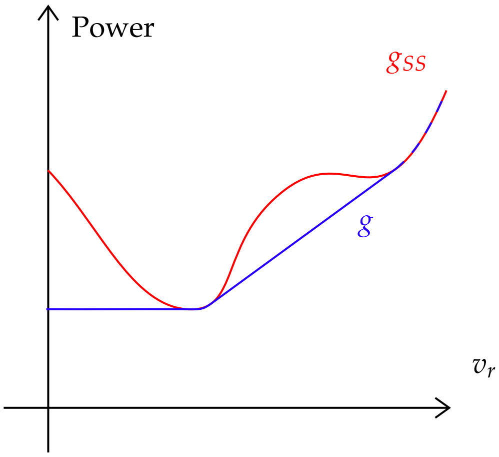


Figure 5.1: Example of a fictitious small scale power function (g_{SS}, red) and associated LSPOV power function (g, blue).

Property 5.1

g is the convex hull of the function

$$\tilde{g}(v_r) := \begin{cases} \min g_{\mathrm{SS}} & \text{if } v_r = 0 \\ g_{\mathrm{SS}}(v_r) & \text{else} \end{cases}$$

Proof. 1. Why the LSPOV power is the convex hull of the small scale power.

If $g_{\mathrm{SS},1} := g_{\mathrm{SS}}(v_{r,1})$ and $g_{\mathrm{SS},2} := g_{\mathrm{SS}}(v_{r,2})$ are possible power values, then alternating in the small scale between SRF $v_{r,1}$ and $v_{r,2}$ in a frequency that is high compared to the large scale time scale will give the vehicle an apparent SRF of $\frac{1}{2}(v_{r,1} + v_{r,2})$ and an apparent power expense of $\frac{1}{2}(g_{\mathrm{SS},1} + g_{\mathrm{SS},2})$ in the LSPOV. Thus,

- if $g_{\mathrm{SS}}\left(\frac{1}{2}(v_{r,1} + v_{r,2})\right) \leq \frac{1}{2}(g_1 + g_2)$, then flying at $v_r = \frac{1}{2}(v_{r,1} + v_{r,2})$ in the small scale is better than alternating between $v_{r,1}$ and $v_{r,2}$, and the resulting large scale power is $g(v_r) = g_{\mathrm{SS}}(v_r)$;
- else if $g\left(\frac{1}{2}(v_{r,1} + v_{r,2})\right) > \frac{1}{2}(g_1 + g_2)$ then alternating between $v_{r,1}$ and $v_{r,2}$ in the small scale is better than operating at fixed $\frac{1}{2}(v_{r,1} + v_{r,2})$ and the resulting large-scale power is

$\frac{1}{2}(g_1 + g_2)$.

In both cases, the LSPOV sees an SRF of $\frac{1}{2}(v_{r,1} + v_{r,2})$ and a power that is less than or equal to $\frac{1}{2}(g_1 + g_2)$. Thus, the LSPOV power is the convex hull of the small scale power function.

2. Why the point with zero SRF and minimal power can be added to the graph of the convex large scale power.

If we note $v_{r,\min}$ the SRF achieving minimum small scale power $v_{r,\min} = \arg\min g_{\mathrm{SS}}(v_r)$, then, as explained in Chap. 2, we can operate the vehicle at $v_{r,\min}$ in the small scale but artificially lengthen the small scale path so that the large scale resulting SRF is a lower value $v_r < v_{r,\min}$. In particular, we can fake the idling of the vehicle: the LSPOV sees $v_r = 0$ but the vehicle is doing circles in the small scale at $v_{r,\min}$. This justifies that the vehicle spends a power $g_{\mathrm{SS}}(v_{r,\min})$ at $v_r = 0$.

□

The convex hull relation from the previous property is sketched in Fig. 5.1. This property shows that in the LSPOV, only convex increasing power functions are consistent with the large scale abstraction.

5.2.2 Specific range

To cross the infinitesimal length $\mathrm{d}l$ at SRF v_r, the vehicle takes $\mathrm{d}t = \frac{\mathrm{d}l}{v_g}$ where v_g is the ground speed of the vehicle. But the infinitesimal energy spent is:

$$\mathrm{d}E = g(v_r)\mathrm{d}t = \frac{g(v_r)}{v_g}\mathrm{d}l$$

Thus, the *specific range* $R_s := \frac{\mathrm{d}l}{\mathrm{d}E}$, *i.e.* the distance that the vehicle can cross with one unit of energy is:

$$R_s = \frac{v_g}{g(v_r)}$$

We note that since $v_g = \left\|\mathbf{u} + \mathbf{v}_f\right\|$, the flow field appears in the formula and thus changes the local specific range depending on the ground velocity direction. The inverse of the specific range $\frac{1}{R_s}$ will be called the *energy intensity* in what follows.

5.2.3 Drag models

Aircraft drag

In steady level flight, an aircraft spends its propulsive energy countering the drag, which has the following expression:

$$D = \frac{1}{2}\rho\, S\, v_r^2\, C_D$$

where ρ is the air specific density, S the aircraft's wing surface and C_D is the drag coefficient. The corresponding power expense is:

$$P_D = \frac{1}{2}\rho\, S\, v_r^3\, C_D$$

Generally, the drag coefficient can depend on many parameters, but at a given altitude, in steady level flight with a known aircraft, it only depends on the lift coefficient C_L through a drag polar:

$$C_D = f(C_L)$$

where the lift coefficient is:

$$C_L = \frac{W}{\frac{1}{2}\rho\, S\, v_r^2} \tag{5.1}$$

with W the weight of the aircraft, assuming that lift equals weight (steady level flight). The drag polar is usually tabulated using wind tunnel testing.

So, noting $K_{C_L} = \frac{W}{\frac{1}{2}\rho\, S}$ and $K_P = \frac{1}{2}\,\rho\, S$, the power of flying at SRF v_r at the small scale for an aircraft (AC) is of the form:

$$g_{\mathrm{AC}}(v_r) = K_P\, v_r^3\, C_D \left(\frac{K_{C_L}}{v_r^2} \right)$$

Two-terms drag polar

Sometimes, an abstraction accurately approximates the drag polar, for instance the two-terms drag polar (Torenbeek, 2013, Sec. 4.4.1) originating from Prandtl's lifting line theory[1]:

$$C_D = C_{D,0} + K_L C_L^2$$

The power function corresponding to this Two-Terms (TT) drag polar is:

$$g_{\mathrm{TT}}(v_r) = K_{p,1}\, v_r^3 + K_{p,2}\, \frac{1}{v_r}$$

with $K_{p,1} = \frac{1}{2}\,\rho\, S\, C_{D,0}$ and $K_{p,2} = \frac{K_L\, W^2}{\frac{1}{2}\,\rho\, S}$.

The maximum range (R) SRF[2] for this power function is:

$$v_r^{(R)} = \sqrt[4]{\frac{K_{p2}}{K_{p1}}} = \sqrt{\frac{W}{\frac{1}{2}\rho S}}\sqrt[4]{\frac{K_L}{C_{D,0}}}$$

It is also the minimum drag airspeed.

The minimum power (P) SRF is different than the no-wind maximum range:

$$v_r^{(P)} = \sqrt[4]{\frac{K_{p2}}{3K_{p1}}} = \sqrt{\frac{W}{\frac{1}{2}\rho S}}\sqrt[4]{\frac{K_L}{3C_{D,0}}} = \frac{1}{\sqrt[4]{3}} v_r^{(R)}$$

It is also the maximum loitering airspeed (speed at which one can spend the longest time doing circles over a position).

[1] For even more accuracy, a non-symmetrical drag polar of form $C_D = C_{D,0} + K_L(C_L - C_{L,0})^2$ can also be plugged in the g_{AC} power function.

[2] In French: *Vitesse de finesse maximale*

Non-dimensioning For further study, it is useful to non-dimension this power function. With:

$$g(v_r) = \widetilde{K_{p,1}}\, v_r^3 + \widetilde{K_{p,2}}\, \frac{1}{v_r}$$

as generic two-terms drag polar power function, we want its maximum range SRF $v_r^{(R)}$ to be 1. This imposes:

$$\widetilde{K_{p,2}} = \widetilde{K_{p,1}}$$

Then, normalizing the power expense at the maximum range SRF $g(1) = 1$ gives:

$$2\,\widetilde{K_{p,2}} = 1 \implies \widetilde{K_{p,2}} = \frac{1}{2}$$

Thus, the Standard (non-dimensional) Two-Terms (STT) drag polar power function is:

$$g_{\text{STT}}(v_r) = \frac{1}{2}\left(v_r^3 + \frac{1}{v_r}\right) \tag{5.2}$$

This expression encompasses the power expense of a wide range of fixed-wing flying vehicles for which the two-terms drag polar is accurate. However, effects such as stall or transsonic drag are neglected in this model. The minimum of this function is reached for the loiter SRF which is $\frac{1}{\sqrt[4]{3}} \approx 0.760$. We note the minimum $g_{\min} = g_{\text{STT}}\left(\frac{1}{\sqrt[4]{3}}\right) \approx 0.877$. Thus, the Standard LSPOV Two-Terms (SLTT) drag polar power law is:

$$g_{\text{SLTT}}(v_r) = \begin{cases} \frac{1}{2}\left(v_r^3 + \frac{1}{v_r}\right) & \text{if } v_r > \frac{1}{\sqrt[4]{3}} \\ g_{\min} & \text{else} \end{cases} \tag{5.3}$$

We depict the graphs of g_{STT} and g_{SLTT} in Fig. 5.2.

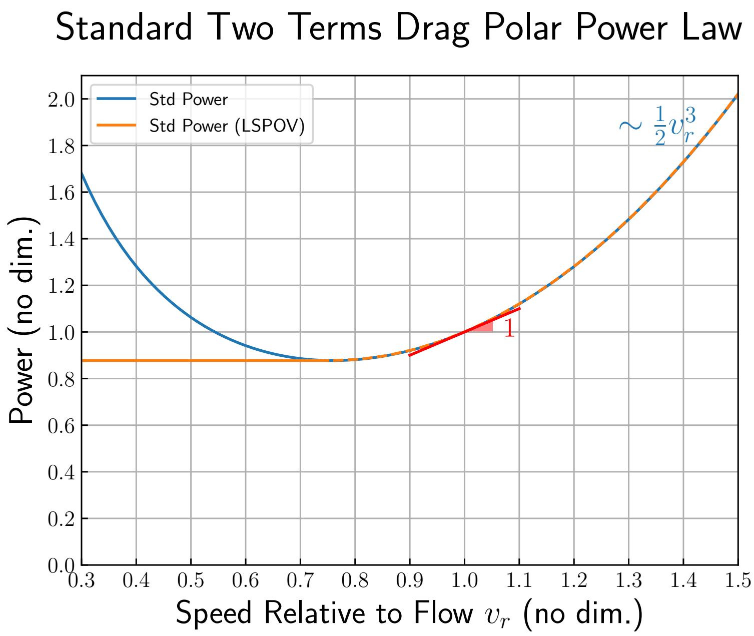


Figure 5.2: Standard Two-Terms Drag Polar Power function, with or without the LSPOV. Speed is in units of the maximum lift-to-drag ratio speed. Power is in units of the power corresponding to the maximum lift-to-drag ratio speed.

Submarine drag

For underwater vehicles, the drag writes the same as for aircraft:

$$D = \frac{1}{2} \rho \, S \, v_r^2 \, C_D$$

But in this case, there is no changing of lift coefficient with airspeed that changes the drag coefficient. Even for underwater gliders alternating up and down phases, overall, the drag coefficient can be considered constant. So, the resulting power is simply in the cube of the vehicle's speed:

$$g(v_r) = k \, v_r^3$$

However, this relation is only valid under the cavitation limit, *i.e.* for speeds under a given threshold. For speeds over the latter threshold, the vehicle goes so fast that water vaporizes in fixed or moving parts of the vehicle at the interface with the flow, making the drag diverge and

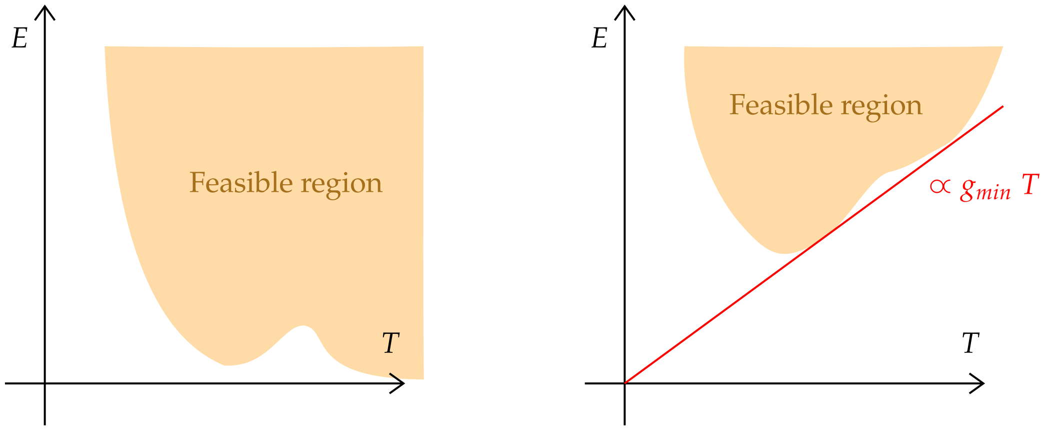


Figure 5.3: Time-energy graph possible shape for costly idling vehicles (left) or free idling vehicles (right).

possibly damaging the vehicle. Thus, the normal operation mode for underwater vehicles is to stay under this limit.

5.2.4 Costly idling vs. free idling vehicles

The large-scale models for airborne and underwater vehicles show a main difference in that the idling of the vehicle can be costly (*i.e.* $g(v_r = 0) > 0$, airborne case) or not ($g(v_r = 0) = 0$, underwater case). This difference will give different features to the time-energy diagram, *i.e.* the diagram of a navigation problem where we plot all the possible time-energy pairs from origin to destination. For instance, the time-energy diagram for costly idling vehicles will be at most tangent to the line $E = g_{\text{min}} \times T$, whereas the time-energy diagram for free idling vehicle will not have such a bound. This is depicted in Fig. 5.3.

Another fundamental difference is that for costly idling vehicles, there exists an optimal SRF achieving maximum range. For aircraft, it is the maximum lift-over-drag ratio SRF (or airspeed). It is the one achieving maximum specific range. For free idling vehicles, this optimum SRF does not exist because the power is not bounded by a positive constant. In other words, reducing the SRF always reduces the energy expense on a given travel, in the absence of flow field, for this kind of vehicles.

In the next section, we study the influence of the flow field on the optimal SRF for costly

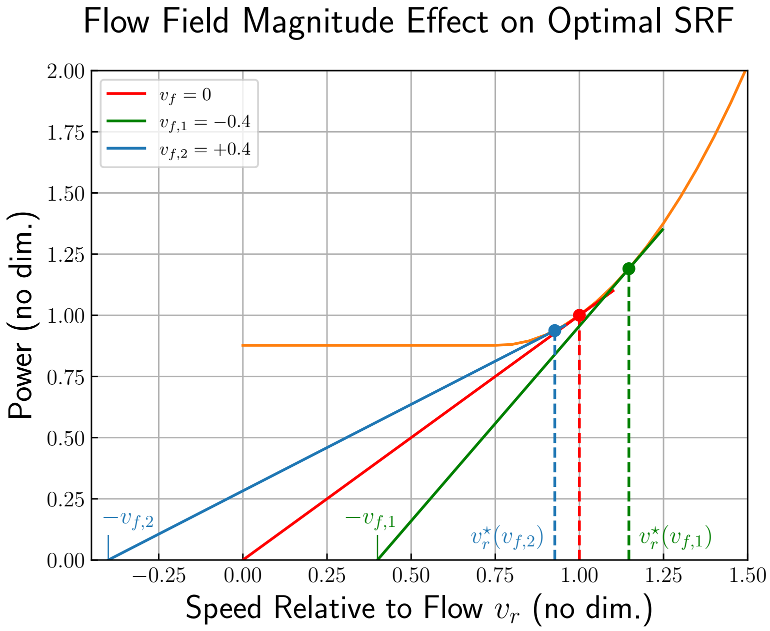


Figure 5.4: Standard LSPOV Two-Terms Drag Polar Power Law and different tangent lines for varying flow field magnitudes. Speed is in units of the maximum lift-to-drag ratio speed (without wind). Power is in units of the power corresponding to the maximum lift-to-drag ratio speed (without wind).

idling vehicles.

5.2.5 Flow field influence on the optimal SRF for costly idling vehicles

We now consider the effect of the flow magnitude on the best SRF to minimize the energy expense. For simplicity, we start by studying what happens when the flow field is colinear to the ground velocity. We note v_f the signed value of the flow field (positive is flow field pushing the vehicle, negative is flow field against the vehicle).

Locally, the best SRF is defined by achieving the maximum specific range:

$$R_s = \frac{v_r + v_f}{g_{\text{SLTT}}(v_r)}$$

This corresponds graphically to finding the maximum slope α such that $v_r \mapsto \alpha(v_r + v_f)$ is tangent to the graph of $v_r \mapsto g_{\text{SLTT}}(v_r)$. This is depicted in Fig. 5.4. The optimal SRF is thus in direct relation with the flow field value v_f, which relation we note $v_r^\star(v_f)$. The optimal SRF $v_r^\star$ is the minimizer of the energy intensity:

$$v_r \mapsto \frac{v_r^3 + \frac{1}{v_r}}{2(v_r + v_f)}$$

We solve numerically this minimization and plot the results in Fig. 5.5.

From applying NCO to function $v_r \mapsto \frac{v_r^3+\frac{1}{v_r}}{2(v_r+v_f)}$, the optimal SRF solves

$$\underbrace{v_r^5 + \frac{3}{2}\, v_f\, v_r^4 - v_r - \frac{v_f}{2}}_{\varphi(v_r, v_f)} = 0 \tag{5.4}$$

The slope $\frac{\mathrm{d}v_r^\star}{\mathrm{d}v_f}$ is found by differentiating the previous relation:

$$\begin{aligned} &\underbrace{\varphi(v_r^\star, v_f)}_{0} + \frac{\partial \varphi}{\partial v_r}\,\mathrm{d}v_r^\star + \frac{\partial \varphi}{\partial v_f}\,\mathrm{d}v_f = 0 \\ \Longrightarrow\ &\frac{\mathrm{d}v_r^\star}{\mathrm{d}v_f} = -\frac{\frac{\partial \varphi}{\partial v_f}}{\frac{\partial \varphi}{\partial v_r}} = \frac{1 - 3\,v_r^4}{10\,v_r^4 + 12\,v_f v_r^3 - 2} \end{aligned}$$

Hence the slope $\frac{\mathrm{d}v_r^\star}{\mathrm{d}v_f} = -\frac{1}{4}$ at $v_f = 0$ (thus $v_r^\star = 1$) appearing in Fig. 5.5.

Similarly, when $v_f \gg 1$, Eq. 5.4 looks like:

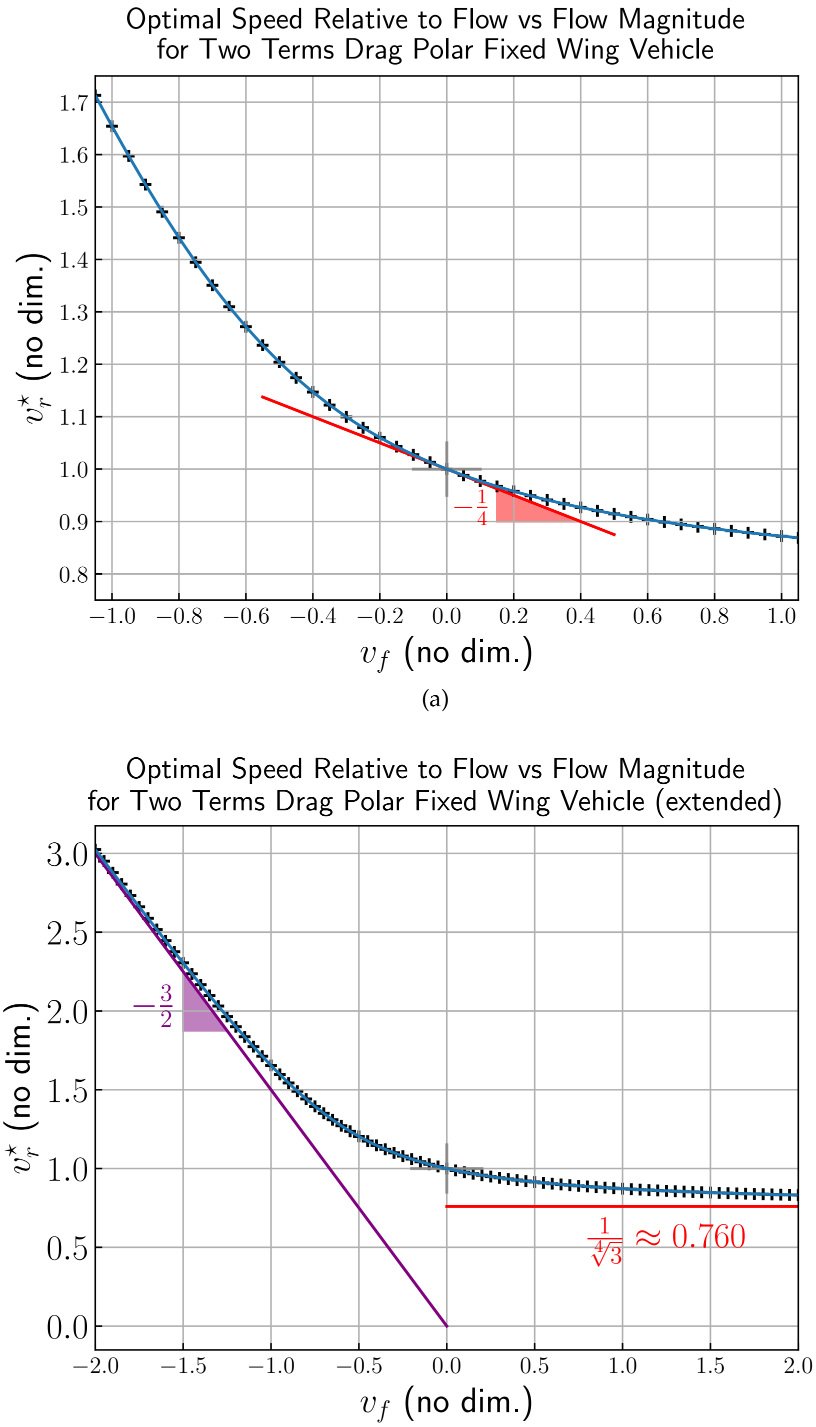
$$\frac{3}{2}\, v_f\, v_r^4 - \frac{v_f}{2} = 0$$

which gives $v_r = \frac{1}{\sqrt[4]{3}}$, the asymptotic value of $v_r^\star$ for a high positive flow magnitude (depicted in Fig. 5.5b).

Looking for the asymptotic behavior for high $|v_f|$, negative v_f, we write $v_r = \alpha\, v_f$ and inject in Eq. 5.4 to have:

$$\left(\frac{3}{2}\,\alpha^4 + \alpha^5\right)\, v_f^5 - \left(\alpha + \frac{1}{2}\right)\, v_f = 0$$

So, for $|v_f| \gg 1$, only the term in v_f^5 matters and we find $\alpha = -\frac{3}{2}$.



(a)

(b)

Figure 5.5: Standard Two-Terms Drag Polar optimum SRF for varying flow field magnitudes. (a) Variation of the optimal SRF with the flow field value. (b) The same relation but with a twice larger value range for the flow field. Positive flow field is pushing the vehicle forward, negative is acting against it.

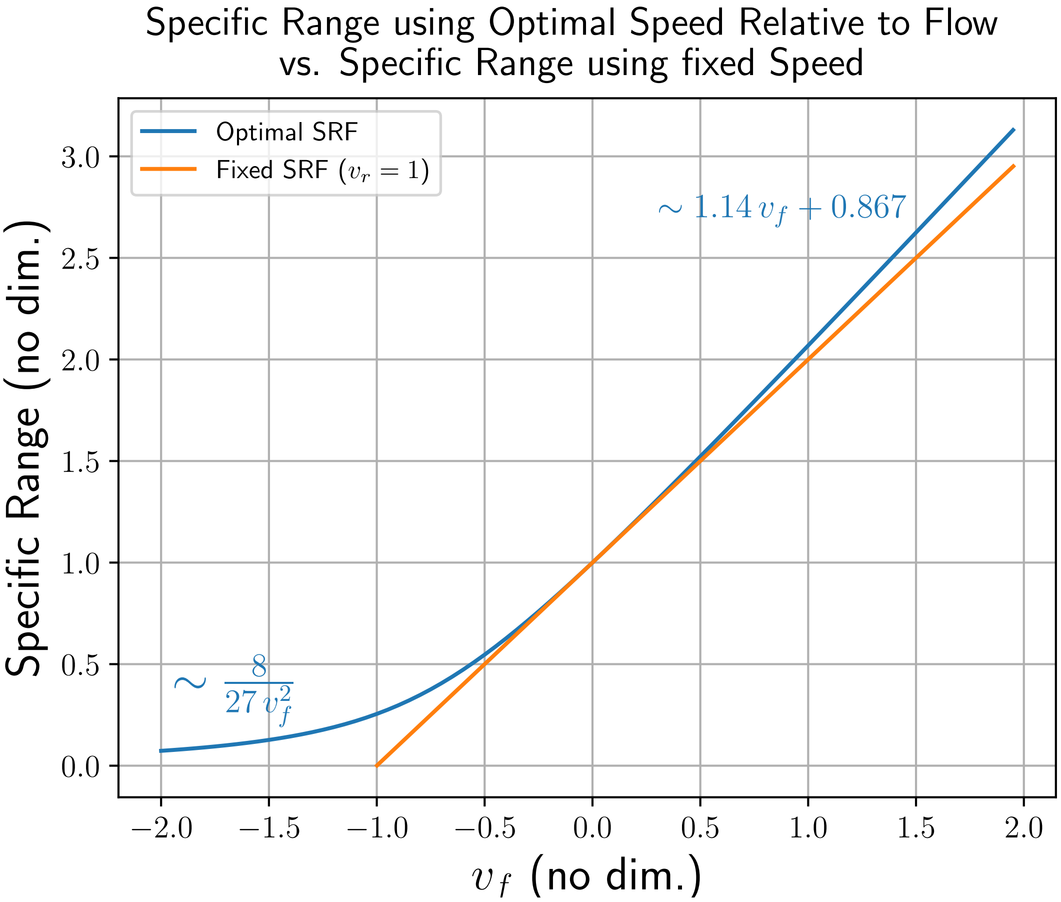


Figure 5.6: Specific range as a function of the longitudinal flow field. Flow field pushes the vehicle when its value is positive, and is acting against it when its value is negative. The graph is normalized so that the specific range is 1 for the no-flow case $v_f = 0$.

Effect on specific range Now that we analyzed the best SRF for varying flow magnitude, we are interested in knowing how using the optimal value affects the specific range. In Fig. 5.6, we plot the specific range as a function of the flow field magnitude, assuming the vehicle operates at the optimal SRF. We also compare it to the operation mode where the SRF does not vary and is fixed to the no flow field, maximum range value (1 in our case).

All in all, the specific range extension provided by adjusting the SRF is not significant when the flow field is small compared to the nominal vehicle speed (no wind maximum range SRF). In Table 5.1, we give the bounds on the flow field magnitude that guarantee a bound on the maximum loss of specific range entailed by operating at fixed SRF compared to adjusting the

Specific range variation	$v_{f,\min}$	$v_{f,\max}$
-0.5%	-0.171	0.244
-1%	-0.228	0.381
-2%	-0.300	0.635
-5%	-0.419	1.60

Table 5.1: Specific range maximum variation when operating at fixed SRF equal to the no-flow maximum range SRF instead of adjusting the SRF to the flow-optimal value.

speed to the optimal value. For instance, if in the considered navigation problem the flow field magnitude (absolute value) is less than 22.8% of the nominal vehicle speed when acting against the vehicle and less than 38.1% when pushing it, then it is guaranteed that the range extension that one would get by varying the SRF cannot exceed 1% of the achieved range. This really states that the flow magnitude should be strong compared to the nominal speed before seeing a benefit from the variation of SRF for saving energy.

In the weak flow regime, the range extension or diminution entailed by the flow field varies linearly with the flow magnitude (kinematic behavior). For strong pushing flow field ($v_f \gg 1$), the range extension increases linearly with the flow magnitude, at a rate of 0.14 units of specific range gained with one unit of flow magnitude, when using the flow-optimal SRF. For strong flow field acting against the vehicle, the fixed SRF strategy falls down when the flow value reaches -1, as at this point, the vehicle does not move anymore. The flow-optimal SRF adjusts to such flow fields, and even bigger magnitudes. When $v_f \ll -1$, $v_r^\star \approx -\frac{3}{2}\,v_f$, so the ground speed in this regime is $v_g \approx -\frac{1}{2}\,v_f$. The specific range never vanishes with this approach, but tends asymptotically to zero.

5.3 The one-dimensional case

Now that we clarified how power is spent for the considered classes of vehicles, we see how we can optimize energy resources in the presence of a flow field. We start by considering a 1D navigation problem.

5.3.1 Problem statement

We focus on the following problem

$$\text{(ENP-1D)}\begin{cases} \displaystyle\min_{v_r(\cdot),h(\cdot)} \int_{t=0}^{T} g(v_r(t))dt \\ \dot{x}(t) = v_r(t)\,h(t) + v_f(t,x(t)) \\ x(t_0) = x_0, \quad x(t_f) = x_f \end{cases} \tag{5.5}$$

where the 'heading' h is simply a scalar. For this system, we have the following Hamiltonian:

$$H(t,x,p,v_r,h,\lambda) = p\,(v_r\,h + v_f(t,x)) + \lambda g(v_r)$$

We look for normal extremals and thus assume $\lambda = 1$. First, the Hamiltonian is clearly minimized in h for $h = -\text{sign}(p)$. Then, we have:

$$\frac{\partial H}{\partial v_r} = p\,h + g'(v_r) = -|p| + g'(v_r)$$

and $\frac{\partial^2 H}{\partial v_r^2} = g''(v_r) \geq 0$ since g is convex. Thus, if $|p| \leq g'(u_{\max})$, $\frac{\partial H}{\partial v_r} = 0$ has one solution $(\widetilde{g}')^{-1}(|p|)$ which minimizes the Hamiltonian. Else, if $|p| > g'(u_{\max})$ then $\frac{\partial H}{\partial v_r} < 0$ and the minimum in the Hamiltonian is reached for the maximal possible SRF $v_r = u_{\max}$. Thus, the optimal SRF is:

$$v_r^\star = \min\left\{(\widetilde{g}')^{-1}(|p|);\, u_{\max}\right\}$$

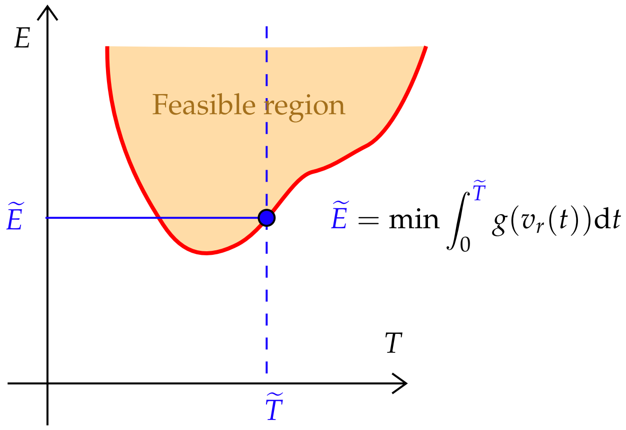

E
Feasible region
$\widetilde{E}$
$\widetilde{E} = \min \int_0^{\widetilde{T}} g(v_r(t))\mathrm{d}t$
T
$\widetilde{T}$

Figure 5.7: For a fixed $\widetilde{T}$, the minimum energy to reach the target $\widetilde{E}$ forms with $\widetilde{T}$ a point that lies on the bottom part of the time-energy graph of all possible time-energy pairs reaching the destination.

Trajectories candidate to optimality for this problem are solutions of the ODE in the augmented vector $\mathbf{y} = (x \;\; p)^\mathsf{T}$:

$$\begin{cases} \dot{\mathbf{y}}(t) = \begin{pmatrix} -v_r^\star\left(|p(t)|\right)\,|p(t)| + v_f(t, x(t)) \\ -\left.\dfrac{\partial v_f}{\partial x}\right|_{t,x(t)} p(t) \end{pmatrix} \\ \mathbf{y}(0) = \begin{pmatrix} x_0 \\ p_0 \end{pmatrix} \end{cases} \tag{5.6}$$

If the final time T is considered fixed, shooting extremals over the window $[0, T]$ will help find the energy-minimal trajectory arriving in time T at destination. In other words, extremals can be used to find the time-energy pairs (T, E) with E being the minimal amount of energy to reach the target in duration exactly T. This means that extremals can be used to find the bottom of the time-energy graph, as depicted in Fig. 5.7.

Free final time If the final time T is considered free, this means that we are looking for the absolute minimum of energy spent for the navigation problem, and we don't know for which travel time T this minimum will be found. In this case, as for the time-optimal case, there is a transversality condition at final time on the reduced Hamiltonian (Hamiltonian in which the optimal control values are plugged). The reduced Hamiltonian writes:

$$H_r(t, x, \widetilde{v_r}^\star) = -g'(\widetilde{v_r}^\star)\ \left(\widetilde{v_r}^\star + v_f(t, x)\right) + g(\widetilde{v_r}^\star)$$

The terminal condition $H_r(T, x(T), \widetilde{v_r}^\star(T)) = 0$ thus writes:

$$-g'(\widetilde{v_r}^\star(T))\ \left(\widetilde{v_r}^\star(T) + v_f(T, x(T))\right) + g(\widetilde{v_r}^\star(T)) = 0$$

We recognize the equation for an extremum of the energy intensity $\frac{g(v_r)}{v_r+v_f}$.

Steady flow field If the flow field is steady, the reduced Hamiltonian does not depend on time and the terminal condition is valid for all times. In other words, in a steady flow field, the minimum energy trajectory is the one using the maximum range SRF adapted to the flow field magnitude at all times.

Unsteady flow field With an unsteady flow field, the previous remark is not valid anymore as the reduced Hamiltonian now depends on time. It is generally false that the vehicle uses the flow-optimal maximum range SRF at all times in an unsteady flow field. The main reason is that, in the presence of time-varying flow field, it may be interesting to spend some more power at some time, being energetically inefficient, to reach a favorable zone that will enable a lower power expense, compensating the previous increase. This is shown in Section 5.3.2.

Consequence of the power function's convexity Let's suppose we have a vehicle reaching target in duration T with an SRF law $t \mapsto v_r(t)$. Its energy expense is:

$$E_{\text{varying}} = \int_{t=0}^{T} g(v_r(t))\mathrm{d}t$$

Now, let's suppose another vehicle operates at a fixed SRF equal to the averaged SRF $\widetilde{v_r} := \frac{1}{T}\int_{t=0}^{T} v_r(t)\mathrm{d}t$ and that this vehicle arrives at time T'. Its energy expense is thus $E_{\text{fixed}} :=$

$T'g(\widetilde{v_r})$. But the convexity of g gives:

$$g(\widetilde{v_r}) = g\left(\frac{1}{T}\int_{t=0}^{T} v_r(t)\mathrm{d}t\right) \leq \frac{1}{T}\int_{t=0}^{T} g(v_r(t))\mathrm{d}t = \frac{1}{T}E_{\text{varying}}$$

Thus,

$$E_{\text{fixed}} \leq \frac{T'}{T}E_{\text{varying}}$$

So, if the averaged-SRF vehicle ever reaches the destination earlier than the varying SRF vehicle $T' < T$, then for sure it would have spent less energy in total than the varying SRF trajectory. In particular, in the absence of flow field, both strategies arrive simultaneously but the varying SRF consumes more energy.

This short analysis shows that varying the SRF consumes more energy than using a fixed SRF in the absence of flow field. So, if a cut in energy expense shall exist for varying SRF trajectories, it can only be because the varying SRF allows the vehicle the harness the surrounding flow field, eventually reducing its total spent energy. This phenomenon is shown in the following section.

5.3.2 Case study: the moving gust

Description

To better understand the time-energy trade-off, we choose to study a simplified, idealized case. We consider the problem of navigating in 1D space from $x_0 = 0$ to $x_f = 1$, with a gust flow field defined by:

$$v_f(t,x) = \begin{cases} v_{\text{gust}} \text{ if } |x - c(t)| \leq w \\ 0 \text{ else} \end{cases}$$

and the center of the carpet $c(t) = c_0 + v_{\text{center}}\, t$. We call $l_0 = c_0 - w$ the spatial offset of the gust with $x = 0$ at $t = 0$. We set $c_0 = 0.25$, $w = 0.15$, $v_{\text{center}} = 1$, $v_{\text{carpet}} = 0.5$. This flow field is depicted in Fig. 5.8.

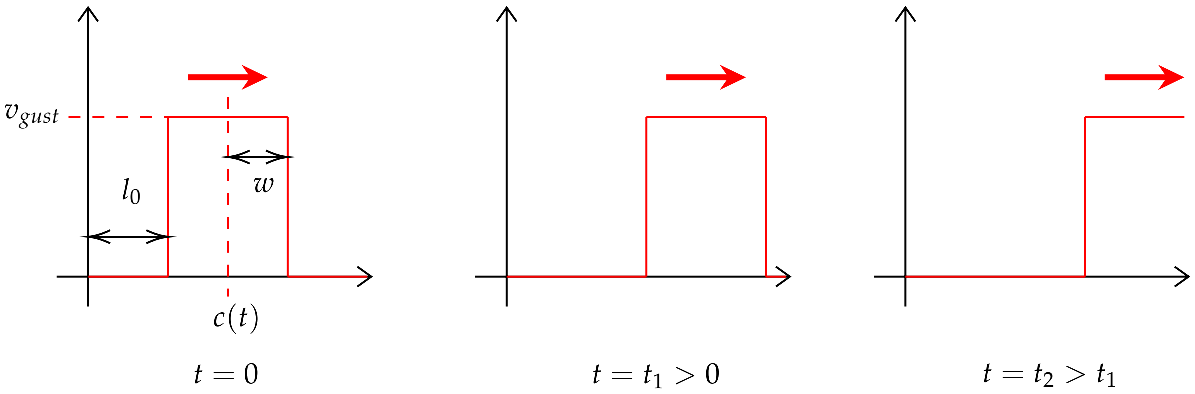


Figure 5.8: Moving gust of wind on a single axis. The gust is moving away from the origin with a flow field vector in the same direction.

From the equations of extremals we deduce that the control shall be constant in regions where the flow field is constant. Thus, we consider two types of trajectories:

1. varying SRF trajectories, that adopt an SRF $v_{r,1}$ in the beginning and switch to a different SRF $v_{r,2}$ when reaching the gust;

2. fixed SRF $v_{r,f}$ trajectories.

For varying SRF trajectories, we limit the study to trajectories effectively reaching the gust before the target, so:

$$v_{r,1} \geq \frac{v_{\text{center}}}{1 - l_0}$$

Similarly, inside the gust, we discard trajectories that are too slow to follow the gust. So we enforce:

$$v_{r,2} \geq v_{\text{center}} - v_{\text{gust}}$$

The time spent before reaching the gust is:

$$\Delta t_{\text{free}} = \frac{l_0}{v_{r,1} - v_{\text{center}}}$$

When reaching the gust, the position is $v_{r,1}\,\Delta t_{\text{free}}$. So, the duration left before reaching destination is

$$\Delta t_{\text{gust}} = \frac{1 - \frac{v_{r,1} l_0}{v_{r,1} - v_{\text{center}}}}{v_{r,2} + v_{\text{gust}}} = \frac{(1 - l_0) v_{r,1} - v_{\text{center}}}{(v_{r,1} - v_{\text{center}})(v_{r,2} + v_{\text{gust}})}$$

All in all, the total energy expense is $g(v_{r,1})\,\Delta t_{\text{free}} + g(v_{r,2})\,\Delta t_{\text{gust}}$, which we note:

$$E_{\text{varying}}(v_{r,1}, v_{r,2}) = g(v_{r,1})\,\frac{l_0}{v_{r,1} - v_{\text{center}}} + \Delta x_1(v_{r,1})\,\frac{g(v_{r,2})}{(v_{r,2} + v_{\text{gust}})}$$

where we stress the dependency on the variables $v_{r,1}$ and $v_{r,2}$ and where $\Delta x_1 = \frac{(1-l_0)v_{r,1} - v_{\text{center}}}{(v_{r,1} - v_{\text{center}})}$ is the remaining distance to cross after entering the gust.

In the previous expression, we notice the minimum of E_{varying} w.r.t. $v_{r,2}$ is independent of $v_{r,1}$: it results of the minimization of the term $\frac{g(v_{r,2})}{(v_{r,2}+v_{\text{gust}})}$, where we recognize the expression of the energy intensity. So, the minimization of the energy expense in the gust phase is simply setting the SRF to the less energy intensive value.

Similarly, there is an energy expense associated to fixed-SRF trajectories, for $v_{r,f} \geq \frac{v_{\text{center}}}{1-l_0}$:

$$E_{\text{fixed}}(v_{r,f}) = g(v_{r,f})\,\frac{v_{r,f} + l_0\,v_{\text{gust}} - v_{\text{center}}}{(v_{r,f} - v_{\text{center}})(v_{r,f} + v_{\text{gust}})}$$

In this development, we voluntarily kept the power function g undefined. We will now see how the two different energy expenses vary, depending on the chosen power model.

Quadratic polar

In this subsection, we choose a quadratic power model for the vehicle

$$g(v_r) = 1 + k\,v_r^2$$

For $k \approx 1$, this models a vehicle with costly idling and increasing power expense with v_r. If $k \to 0$, the model is independent of the SRF. If $k \to +\infty$, it is as if the idling was not costly (idling power negligible compared to the power for $v_r > 0$), up to the rescaling of the energy axis.

For this power model, we have an analytical expression for the flow-optimal SRF.

Property 5.2

For the quadratic polar, the less energy-intensive SRF is:

$$v_r^\star = \sqrt{\frac{1}{k} + v_{\text{gust}}^2} - v_{\text{gust}}$$

For numerical applications, we take $k = 1$. Energy-time pairs are displayed in Fig. 5.9 for the constant SRF law without the gust, the constant SRF law with the gust, and the varying SRF law to compare for total travel time and total energy.

We first see that the gust reduces the travel time and energy for fixed SRF laws compared to a no flow situation. The gust indeed pushes the vehicle towards its objective. Furthermore, some varying SRF laws achieve time-energy pairs that dominate constant SRF performances. In particular, the minimum amount of energy is ≈ 0.9, spent for $v_{r,1} \approx 1.6$ and $v_{r,2} \approx 0.8$. It is less than the reference minimum energy without flow field, which is 1, and also the constant SRF approach, which achieves an energy expense of at least 0.95.

Two-terms drag polar

We now change the power law for the standard LSPOV two-terms drag polar power law from Eq. 5.3. As in the previous case, energy-time pairs are computed for the constant SRF law without the gust, the constant SRF law with the gust, and the varying SRF law. Results are displayed in Fig. 5.10.

The same behavior as for the preceding case is observed, in that the travel time and energy are smaller for fixed SRF than in the no flow case, and some time-energy pairs for varying SRF trajectories dominate constant SRF performances. However, the time-energy domain gained with varying SRF trajectories is smaller than for the previous power law. The minimum amount

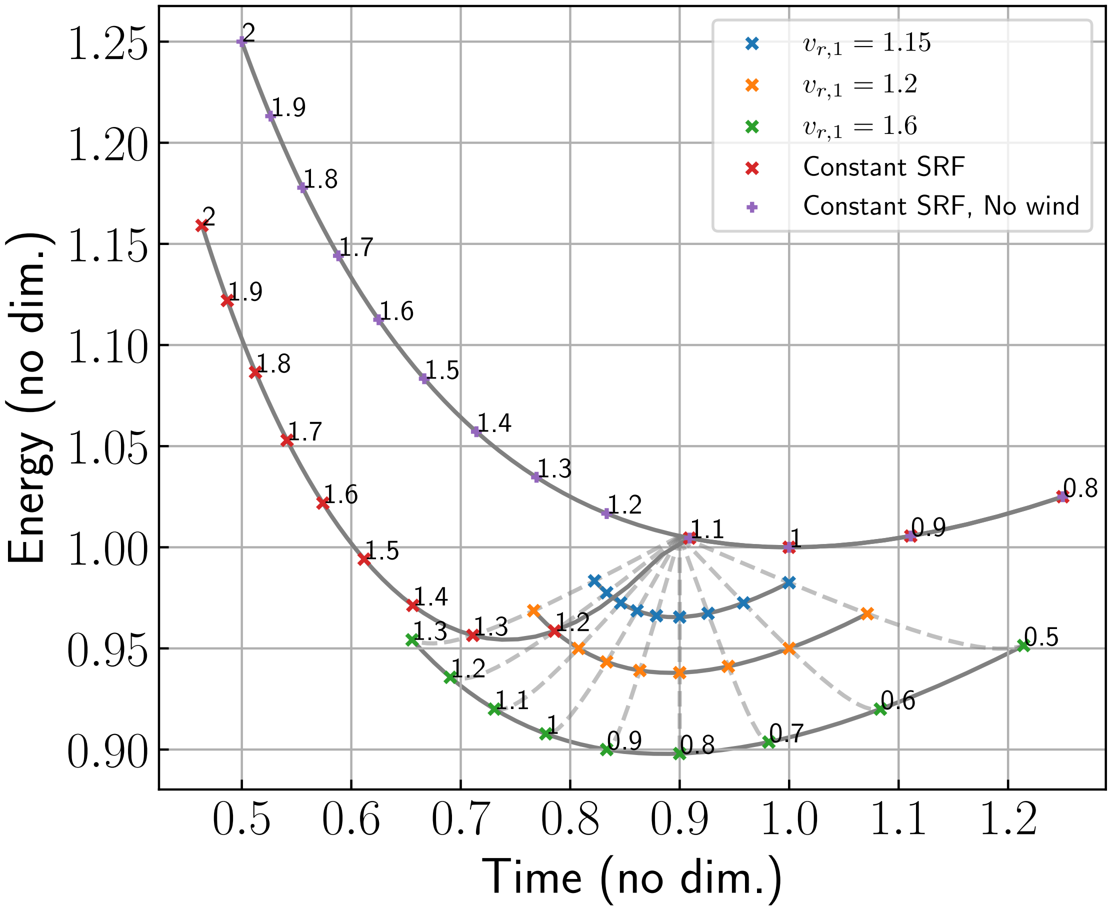


Figure 5.9: Energy-time trade-off for the moving gust case with a quadratic polar. The purple crosses are the reference time-energy points in the no-flow case, with the quadratic power model. Red crosses are the result of choosing a fixed SRF for the travel. Blue, yellow and green crosses are the result of choosing a varying SRF law with values $v_{r,1}$ and $v_{r,2}$. The $v_{r,1}$ value is given by the color, while the $v_{r,2}$ value is labelled on the green sequence, with dashed lines indicating same values for $v_{r,2}$.

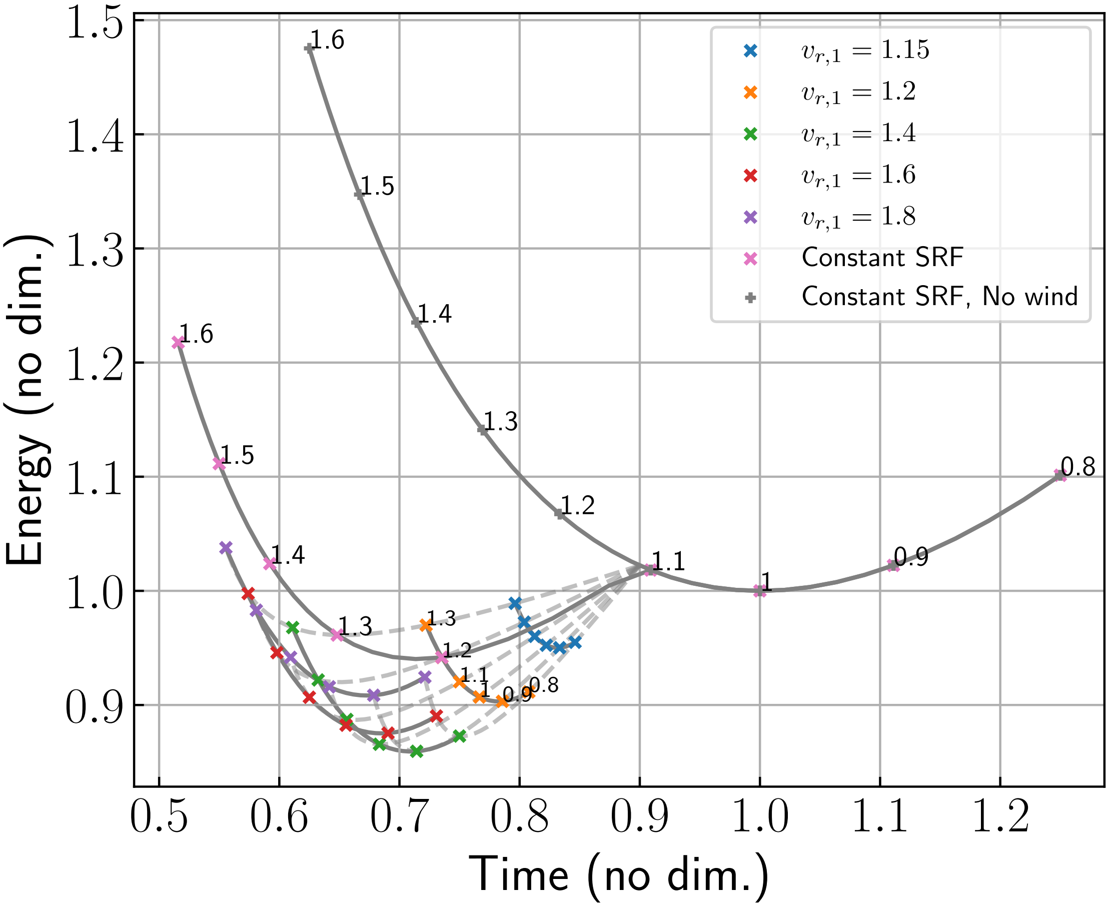


Figure 5.10: Energy-time trade-off for the moving gust case with the two-terms LSPOV drag polar power law. For a description of the sequences, refer to Fig. 5.9. For the varying SRF sequences, $v_{r,1}$ corresponds to the color of the sequence and $v_{r,2}$ is labelled on top of the yellow sequence, with dashed lines indicating constant $v_{r,2}$.

of energy is found to be ≈ 0.85, achieved for varying SRF trajectories. The reference minimum energy for the travel without flow field is still 1. The minimum energy spent by constant SRF trajectories is ≈ 0.95.

These experiments show that a cut in energy expense by an order of magnitude of 10% can be expected from the adoption of a varying SRF law.

5.4 The bidimensional case

5.4.1 Problem statement

We now open up the state space and consider a vehicle evolving in free 2D planar space. The total energy spent over duration T by this vehicle is:

$$E = \int_{t=0}^{T} g\left(v_r(t)\right) dt$$

In a similar fashion as for the 1D case, the energy-optimal navigation problem writes:

$$\text{(ENP)} \begin{cases} \displaystyle \min_{v_r(\cdot),\mathbf{h}(\cdot)} \int_{t=0}^{T} g(v_r(t))dt \\ \dot{\mathbf{x}}(t) = v_r(t)\,\mathbf{h}(t) + \mathbf{v}_f(t,\mathbf{x}(t)) \\ \mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(T) = \mathbf{x}_f \end{cases} \tag{5.7}$$

We note $\mathbf{h}(\theta) = (\cos\theta \;\; \sin\theta)^\intercal$ to parametrize the control with variables (v_r, θ). The Hamiltonian for this problem writes:

$$H(t,\mathbf{x},\mathbf{p},v_r,\theta,\lambda) = \mathbf{p}^\intercal(v_r\,\mathbf{h}(\theta) + \mathbf{v}_f(t,\mathbf{x})) + \lambda g(v_r)$$

We study normal extremals, so we can rescale the pair $(\mathbf{p}, \lambda)$ so that $\lambda = 1$. The variations of the Hamiltonian with the control values are:

$$\frac{\partial H}{\partial v_r} = \mathbf{p}^\mathsf{T}\mathbf{h}(\theta) + g'(v_r)$$
$$\frac{\partial H}{\partial \theta} = v_r\,\mathbf{p}^\mathsf{T} R\left(\frac{\pi}{2}\right)\mathbf{h}(\theta)$$

The Hessian matrix of the Hamiltonian w.r.t. the control variables is:

$$\nabla^2 H = \begin{pmatrix} \frac{\partial^2 H}{\partial v_r^2} & \frac{\partial^2 H}{\partial v_r \partial \theta} \\ \star & \frac{\partial^2 H}{\partial \theta^2} \end{pmatrix} = \begin{pmatrix} g''(v_r) & \mathbf{p}^\mathsf{T} R\left(\frac{\pi}{2}\right)\mathbf{h}(\theta) \\ \star & -\mathbf{p}^\mathsf{T}\mathbf{h}(\theta) \end{pmatrix}$$

where we hid the redundant symmetric coefficient with a star. So, if (v_r, θ) minimizes the Hamiltonian, we have:

$$\frac{\partial H}{\partial \theta} = 0 \Leftrightarrow v_r\,\mathbf{p}^\mathsf{T} R\left(\frac{\pi}{2}\right)\mathbf{h}(\theta) = 0$$

which means:

$$\mathbf{h}^\star = \varepsilon \frac{\mathbf{p}}{\|\mathbf{p}\|}$$

with $\varepsilon = \pm 1$. Now, for such a value of $\mathbf{h}$, the Hessian is:

$$\begin{pmatrix} g''(v_r) & 0 \\ 0 & -\varepsilon\,\|\mathbf{p}\| \end{pmatrix}$$

But (v_r, θ) is assumed to minimize the Hamiltonian so this matrix has to be non-negative. The coefficient $g''(v_r)$ is positive because the LSPOV power function is convex. Thus $\varepsilon = -1$ and the optimal heading is still the same as in the time-optimal case:

$$\mathbf{h} = -\frac{\mathbf{p}}{\|\mathbf{p}\|}$$

Then, we use the second NCO:

$$\frac{\partial H}{\partial v_r} = 0 \Leftrightarrow \mathbf{p}^\intercal \left(-\frac{\mathbf{p}}{\|\mathbf{p}\|}\right) + g'(v_r) = 0 \Leftrightarrow g'(v_r) = \|\mathbf{p}\| \tag{5.8}$$

Here, the LSPOV is very useful since the function g is convex, thus g' is increasing. Moreover due to the definition of g, the minimum tangent slope is to be found at $v_r = 0$ and has value $g'(0)$. For the considered models, this value is zero $g'(0) = 0$, and the limit in $+\infty$ is $+\infty$, $\lim_{v_r \to +\infty} g'(v_r) = +\infty$. The monotonicity and these boundary conditions makes Equation 5.8 have a unique solution. So, the optimal SRF is:

$$v_r^\star = (g')^{-1}(\|\mathbf{p}\|) \tag{5.9}$$

If there is a maximum SRF $u_{\max}$, and for some $\mathbf{p}(t)$ we have $(g')^{-1}(\|\mathbf{p}(t)\|) > u_{\max}$, then since:

$$\frac{\partial H}{\partial v_r} = g'(v_r) - \|\mathbf{p}\|$$

we have $\forall v_r \leq u_{\max}$, $\frac{\partial H}{\partial v_r} \leq 0$. So, $v_r = u_{\max}$ still minimizes the Hamiltonian. So, in the case of a bounded SRF, one needs to modify Eq. 5.9 in the following manner:

$$v_r^\star = \min\left\{(g')^{-1}(\|\mathbf{p}\|),\ u_{\max}\right\} \tag{5.10}$$

Transversality condition As in the 1D case, when the final time T is not fixed, which means we look for the absolute minimum of energy for the navigation problem, there is a terminal condition on the reduced Hamiltonian $H_r(t, \mathbf{x}, \mathbf{p}) = H(t, \mathbf{x}, \mathbf{p}, v_r^\star, \mathbf{h}^\star)$:

$$H_r(T, \mathbf{x}(T), \mathbf{p}(T)) = 0$$

Which gives:

$$g(v_r^\star(T)) - g'(v_r^\star(T))\left(v_r^\star(T) + \mathbf{h}^\star(T)^\intercal \mathbf{v}_f(T, \mathbf{x}_f)\right) = 0$$

When $\mathbf{v}_f(T, \mathbf{x}_f) = 0$, it reduces to $g(v_r^\star(T)) - g'(v_r^\star(T))v_r^\star = 0$, which is the equation of the no-flow maximum range SRF. When $\mathbf{v}_f(T, \mathbf{x}_f) \neq 0$, we could have expected that, in a similar way than in the 1D case, the previous equation would be the equation for the flow-optimal SRF, but it is not the case in general.

As always in the PMP, this transversality condition is an additional necessary condition for optimality, and in this case, necessary condition to have an absolute minimum in energy. So, if one discards the condition, one is only left with more extremals but is still guaranteed to find the solution of the problem within the set of extremals.

5.4.2 Optimal SRF expressions

Powers of the SRF For power expenses of the form $g(v_r) = k\,v_r^n$, $n \geq 2$, Eq. 5.9 gives an analytical link between the costate and the optimal SRF:

$$v_r^\star = \left(\frac{\|\mathbf{p}\|}{k\,n}\right)^{\frac{1}{n-1}}$$

> **Remark**
>
> This formula can be found in Doshi et al. (2023).

Two-terms drag polar In the two-terms drag polar case, Eq. 5.9 gives an analytical link between the costate and the optimal SRF

Property 5.3

For the standard two-terms drag polar power profile 5.2, the optimal SRF is:

$$v_r^\star = \frac{1}{\sqrt{3}}\sqrt{\|\mathbf{p}\| + \sqrt{\|\mathbf{p}\|^2 + 3}}$$

Proof. We have:

$$g'(v_r) = \frac{1}{2}\left(3\,v_r^2 - \frac{1}{v_r^2}\right)$$

So $g'(v_r) = \|\mathbf{p}\|$ can be rewritten to:

$$v_r^4 - \frac{2\,\|\mathbf{p}\|}{3}\,v_r^2 - \frac{1}{3} = 0$$

which is a degree two polynomial in v_r^2. There is thus two solutions for v_r^2 but one of the two is negative and cannot be a square. We conclude by taking the root of this solution for $v_r^\star$, as SRFs are positive quantities. □

> **Remark**
>
> It is the same formula as (Dobrokhodov et al., 2020, Eq. 21).

5.4.3 Case studies

We now study particular cases for energy-time trade-offs. The focus is on determining when the SRF variation entails a clear benefit in the time-energy performance. Furthermore, the focus is mostly on airborne vehicles, with two cases out of three using the standard two-terms drag polar power function.

Linear flow field, quadratic power

We begin case studies with the particular example of the standard linear flow field navigation problem, with a quadratic power expense. This case is interesting because it is one of the very rare cases for which an analytical formula can be found for the varying-SRF energy-time Pareto front. For this case, the navigation settings are:

$$\mathbf{x}_0 = (0,\ 0)$$
$$\mathbf{x}_f = (1,\ 0)$$
$$\mathbf{v}_f(\mathbf{x}) = \begin{pmatrix} 0 & g \\ 0 & 0 \end{pmatrix} \mathbf{x}$$

with power:

$$g(\|\mathbf{u}\|) = k\,\|\mathbf{u}\|^2$$

The derivation of the following formulae is kept in Appendix C. The energy-time Pareto front for SRF-varying trajectories is:

$$E_{\text{vary}} = \frac{k}{t_f\left(1 + \frac{g^2\,t_f^2}{12}\right)}$$

Unfortunately, the energy-time curve for fixed-SRF trajectories does not have an analytical expression (to the best of the author's knowledge). Thus, we solve numerically

$$\frac{v_r}{g}\left(\frac{\sin\theta_0}{\cos^2\theta_0} + \operatorname{arctanh}(\sin\theta_0)\right) = 1 \tag{5.11}$$

for the initial angle θ_0 and get the resulting optimal travel time:

$$t_f = \frac{2\,\tan\theta_0}{g} \tag{5.12}$$

The corresponding energy expense is $E_{\text{fixed}} = k\,t_f\,v_r^2$.

For different values of the flow field gradient g, we plot the energy-time graphs in Figure 5.11. We take $k = 1$ as it is a multiplicative scaler for both E_{vary} and E_{fixed}.

First of all, in all situations and for fixed SRF strategies as well as varying SRF strategies, allowing more travel time reduces the total energy expense. Varying SRF trajectories achieve better time-energy pairs, but the difference with constant SRF time-energy pairs is not signif-

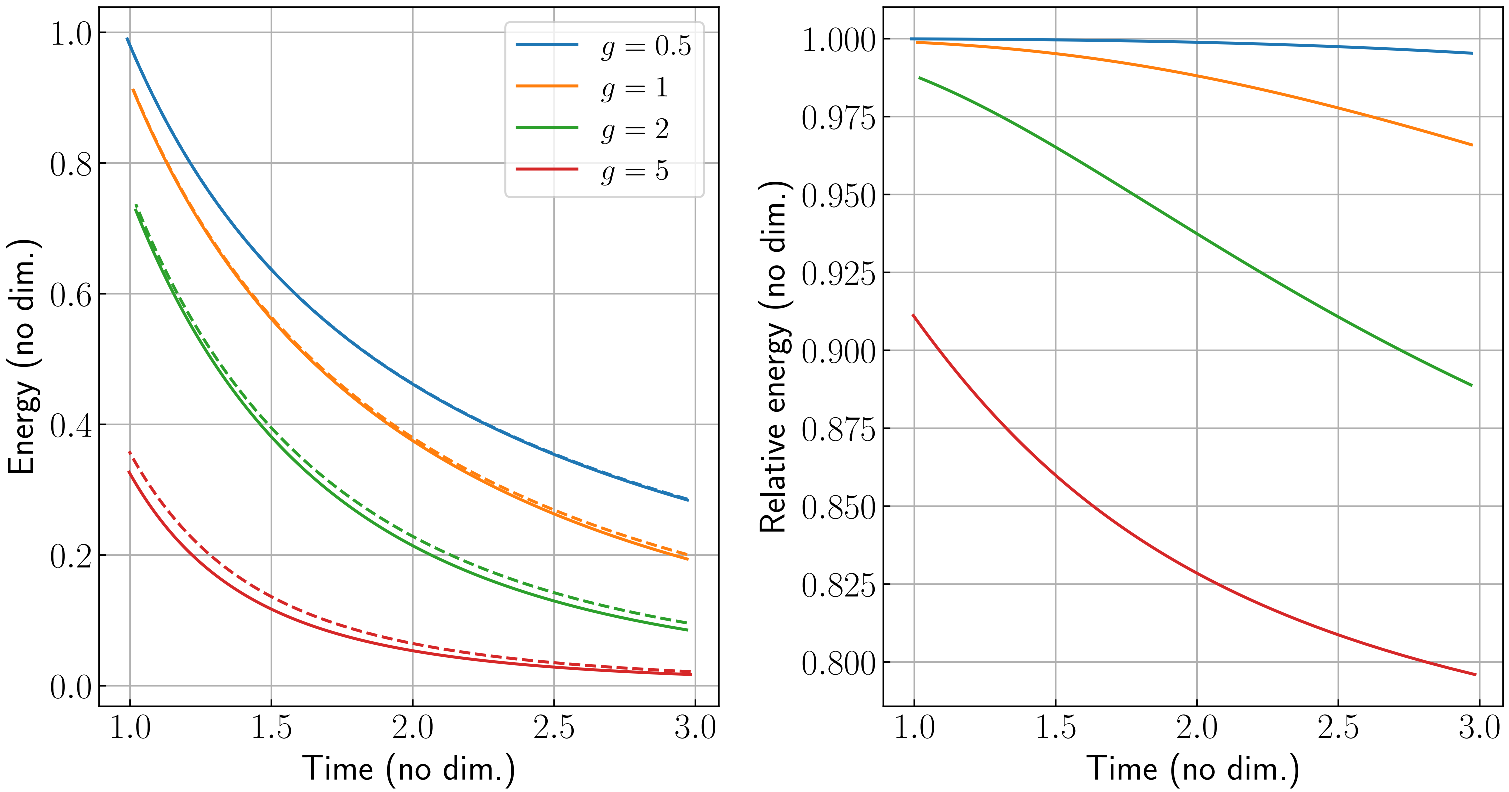


Figure 5.11: Energy-time Pareto front for the linear flow field, quadratic power case. Left, plain lines are the Pareto front for the varying-SRF trajectories. Dashed lines show the energy-time curve for fixed-SRF trajectories when the SRF value changes. Right, the relative energy expense of using the varying-SRF approach compared to using the fixed-SRF approach for the same arrival time.

icant. The relative difference is only significant in very high flow field gradient ($g = 5$, red curves) and large travel times. For travel times close to 1 and a flow field gradient close to 1 as well, the performances are very close to one another.

The observed behavior was expected. Indeed, Fig. 5.6 shows that the difference in energy savings in 'tailwind' conditions is weak between constant and varying SRF. Furthermore, this simple analytical flow field does not show a complex spatial structure, neither does it show time variations, limiting the available features for the varying SRF approach to save energy.

Gyre flow, airborne-like power function

We now move on to a more spatially complex flow field with the gyre flow. We go gradually in the flow field features by choosing a steady flow field in this second case. We use the same navigation settings as in Sec. 9. In a similar way as for time-optimal extremals, we can plug the necessary conditions for optimality into the dynamics to get an augmented system on $\mathbf{y} := (\mathbf{x}\ \mathbf{p})^\intercal$:

$$\begin{cases} \dot{\mathbf{y}}(t) = \begin{pmatrix} -v_r^\star\left(\|\mathbf{p}(t)\|\right) \frac{\mathbf{p}(t)}{\|\mathbf{p}(t)\|} + \mathbf{v}_f(t, \mathbf{x}(t)) \\ -\left.\dfrac{\partial \mathbf{v}_f}{\partial \mathbf{x}}\right|_{t,\mathbf{x}(t)}^\intercal \mathbf{p}(t) \end{pmatrix} \\ \mathbf{y}(0) = \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{p}_0 \end{pmatrix} \end{cases} \tag{5.13}$$

Because of the term $v_r^\star\left(\|\mathbf{p}(t)\|\right)$, this system is *not* invariant in the costate norm anymore. Thus, to be exhaustive, the sampling of extremals cannot be reduced to $\mathbf{p}_0 \in \mathbb{S}^1$ anymore. For this example, we empirically adjust a lattice of initial costate vectors of the form:

$$\mathbf{p}_0^{(i,j)} = r_i \begin{pmatrix} \cos\theta_j \\ \sin\theta_j \end{pmatrix}$$

with $(r_i)_{i\in\{0,1,\dots,n-1\}}$ and $(\theta_j)_{j\in\{0,1,\dots,m-1\}}$ being uniform samplings of given intervals. In the numerical application, $r_0 = 0.1$, $r_{n-1} = 10$, $n = 20$, $\theta_0 = 0.6$, $\theta_{m-1} = 1.7$, $m = 400$.

We shoot extremals over the time interval $[0, 2]$ with $n_t = 150$ time discretization points. We verify empirically that the trajectories shot end up around the target, and that the resolution in the extremal front is fine enough around the target. We select candidate solutions as trajectories close enough to the target, *i.e.* trajectories for which there exist some t_f for which $\|\mathbf{x}(t_f) - \mathbf{x}_f\| < r_f$, with $r_f = 0.1$. We also use the time-optimal algorithm with different fixed SRF values to get the energy-time curve for fixed SRF trajectories. The energy-time curves and the SRF laws are depicted in Fig. 5.12, and the resulting trajectories in Fig. 5.13.

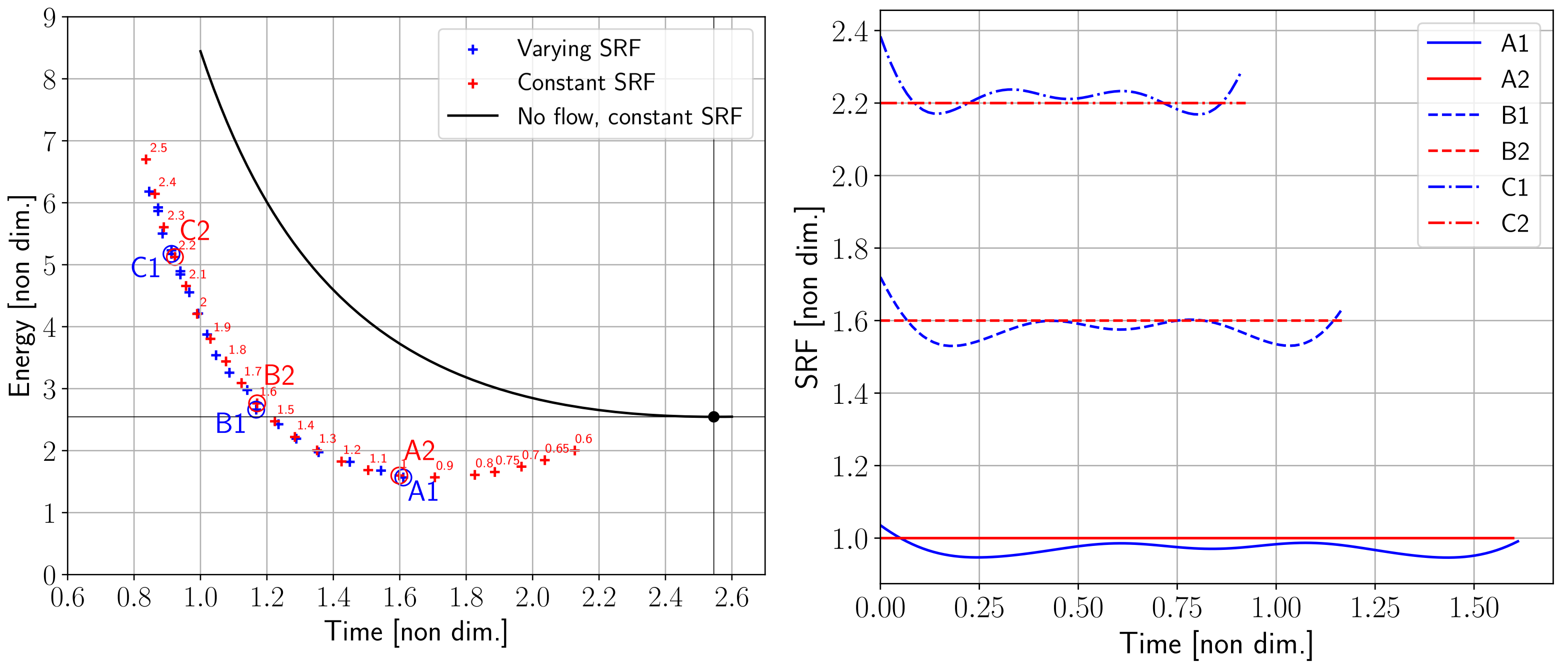
Overall, the gyre flow field enables savings as regards time and energy for constant SRF trajectories as well as varying SRF trajectories. However, no improvement is found by varying the SRF. On the energy-time diagram, constant SRF and varying SRF end up describing two curves that are not distinguishable from one another. Moreover, varying SRF laws do not vary much in SRF, and are thus close to the value of the constant SRF achieving the same time-energy performance. Lastly, trajectories are very similar in the state space.

These observations were expected, because optimal trajectories harness the 'tailwind' regime where the flow field pushes the vehicle forward at all times. As shown in Fig. 5.6, the benefit from using the optimal SRF in the tailwind regime for a two-terms drag polar is very weak and explains why varying SRF trajectories do not gain performance on the energy-time graph.

This canonical trajectory planning example shows that vehicles with two-terms drag polar power expense do not show easily a benefit from adopting a varying SRF, especially if the case features a tailwind route to the target.

Atlantic crossing by an air drone

We complete the case study with a last example, where we consider a realistic, time-varying flow field. We use the same problem settings as the Dakar to Natal crossing from Sec. 4.3.3, but



(a)

(b)

Figure 5.12: (a) Energy-time graph for the gyre problem. Varying SRF performances are noted in blue, constant SRF performances in red. For constant SRF trajectories, the SRF value is annotated above data points. The black curve indicates the energy-time expense in the absence of flow field. The black dot corresponds to the time-energy pair of minimal energy in the no-flow case, which is a reference for all values. (b) For three pairs of varying-SRF/constant-SRF points, the SRF laws are plotted.

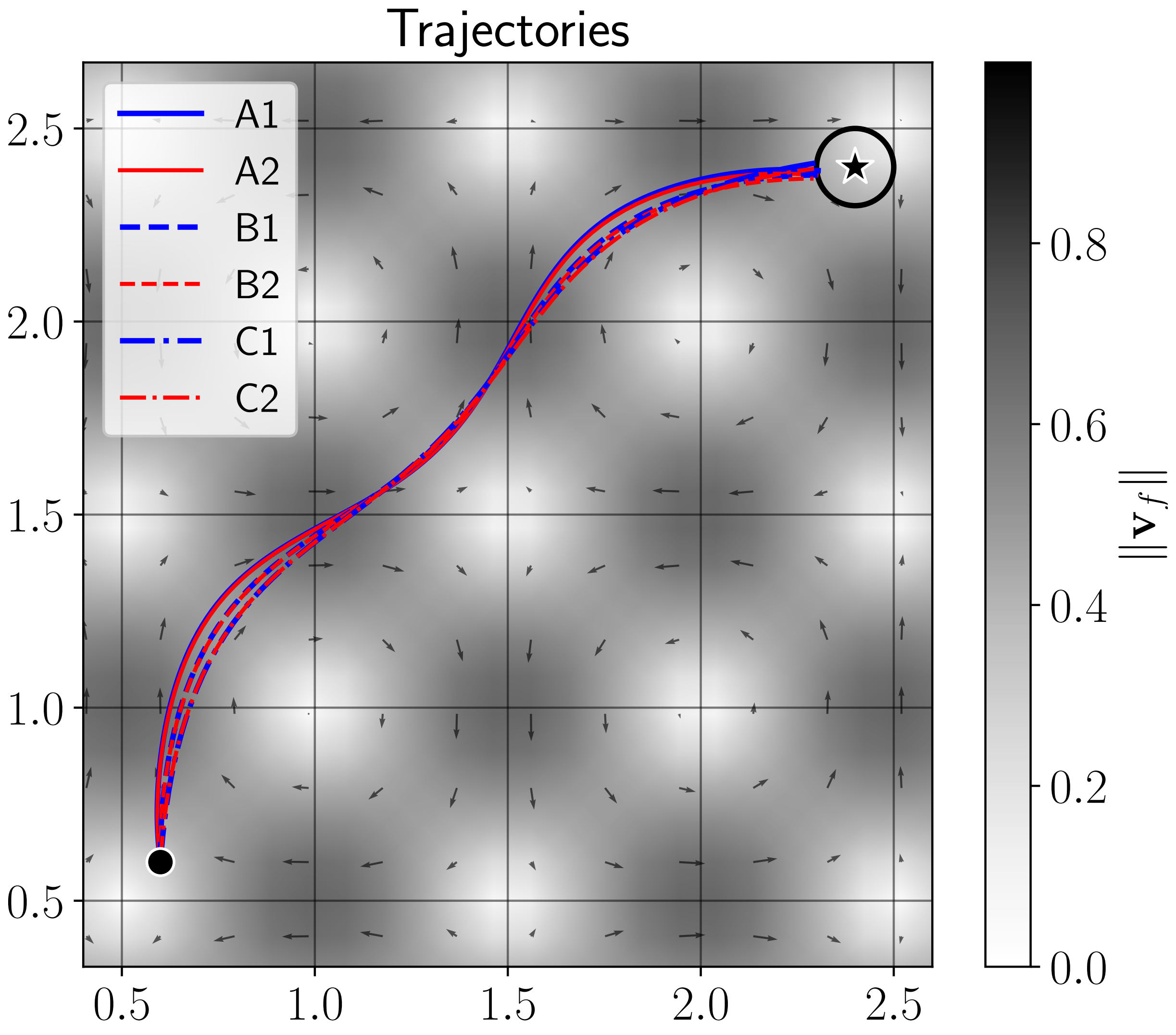


Figure 5.13: Fixed and varying SRF trajectories in the physical space. Varying SRF performances are noted in blue, constant SRF performances in red.

	Energy (no dim.)		
	Constant SRF	Varying SRF	Difference
Group A	1.64	1.51	-8.0%
Group B	2.01	1.77	-12%
Group C	2.63	2.34	-11%

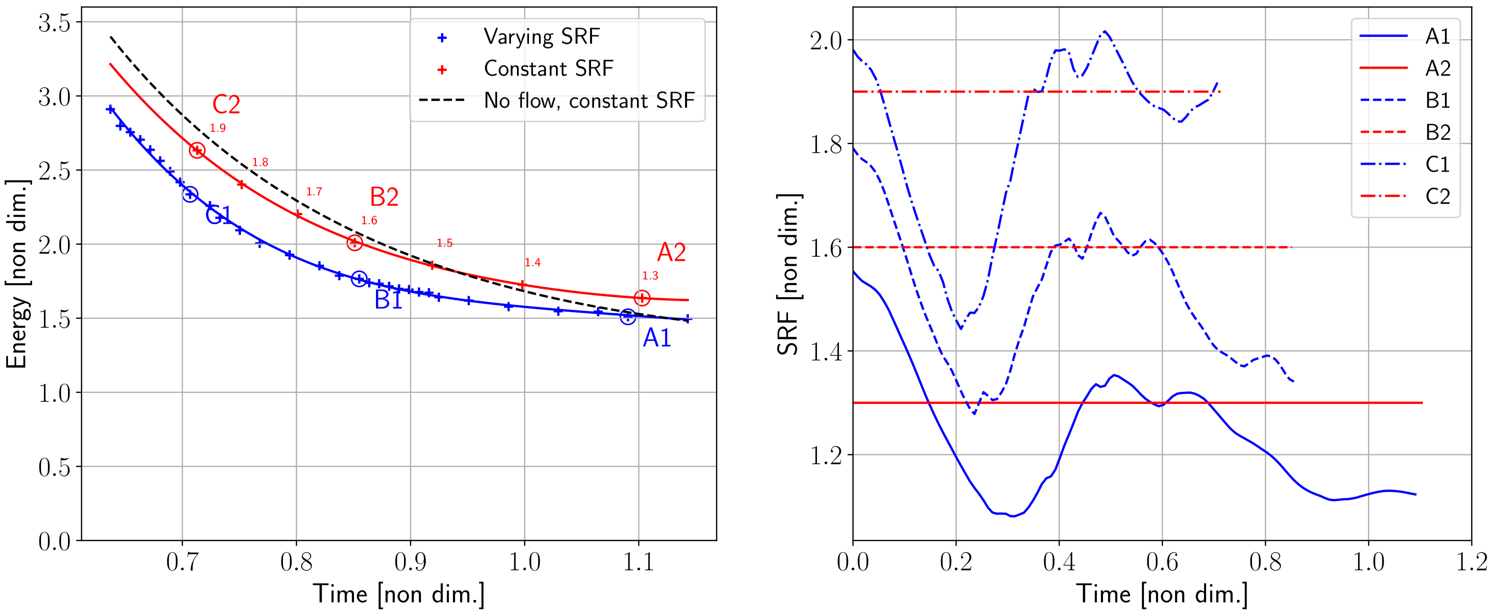
Table 5.2: Comparison of energy expense for the pairs A1/A2, B1/B2 and C1/C2 that have very close travel duration. The difference is relative to the constant SRF value.

without obstacles. The goal here is to use energy-optimal extremals to build the Pareto front of time-energy pairs for varying SRF trajectories, and to compare it to fixed SRF trajectories, to see if a benefit can be gained on real-world airborne examples when applying a varying SRF law.

For this case, we have $r_0 = 3$, $r_{n-1} = 7$, $n = 30$, $\theta_0 = -2.56$, $\theta_{m-1} = -2.26$, $m = 151$. As presented in Sec. 4.3.3, the problem is rescaled for good numerical conditioning. In particular, speeds are divided by 16.46 m/s. In the rescaled space, we use the standard two-terms drag polar power function that has a maximum range airspeed of 1, so it means implicitly that the airborne vehicle in the real world has a maximum range airspeed of 16.46 m/s. The energy-time curves and the SRF laws are depicted in Fig. 5.14, and the resulting trajectories in Fig. 5.15.

In this case, we are able to see a benefit from using a varying SRF law compared to a constant SRF law. As for the previous example, we isolate three pairs of varying-SRF/constant-SRF trajectories and study their differences. Table 5.2 compares the energy expense for these pairs (that have very similar travel duration). The overall benefit from a varying SRF law is of the order of a 10% reduction for this case. It is bigger at high SRF than low SRF. On the SRF graph (Fig. 5.14), we see that the main variation in SRF occurs in regions of high flow field. The strategy is always to reduce the speed when passing through high tailwinds and augmenting it back when the wind weakens.

There is also a remarkable feature: the heading law and the speed law are almost decoupled. For instance, the pair showing the biggest difference in their ground paths is A1/A2. For this



(a)

(b)

Figure 5.14: (a) Energy-time graph for the Dakar Natal, Sep. 29th 2021 case. Varying SRF performances are noted in blue, constant SRF performances in red. An order 4 polynomial regression curve is drawn for both series in respectively blue and red. For constant SRF trajectories, the SRF value is annotated above data points. The dotted black curve indicates the energy-time expense in the absence of flow field. (b) For three pairs of varying-SRF/constant-SRF points, the SRF laws are plotted. The underlying wind field is the time-varying wind field frozen at its first data frame, *i.e.* 2021-09-29 06:00Z

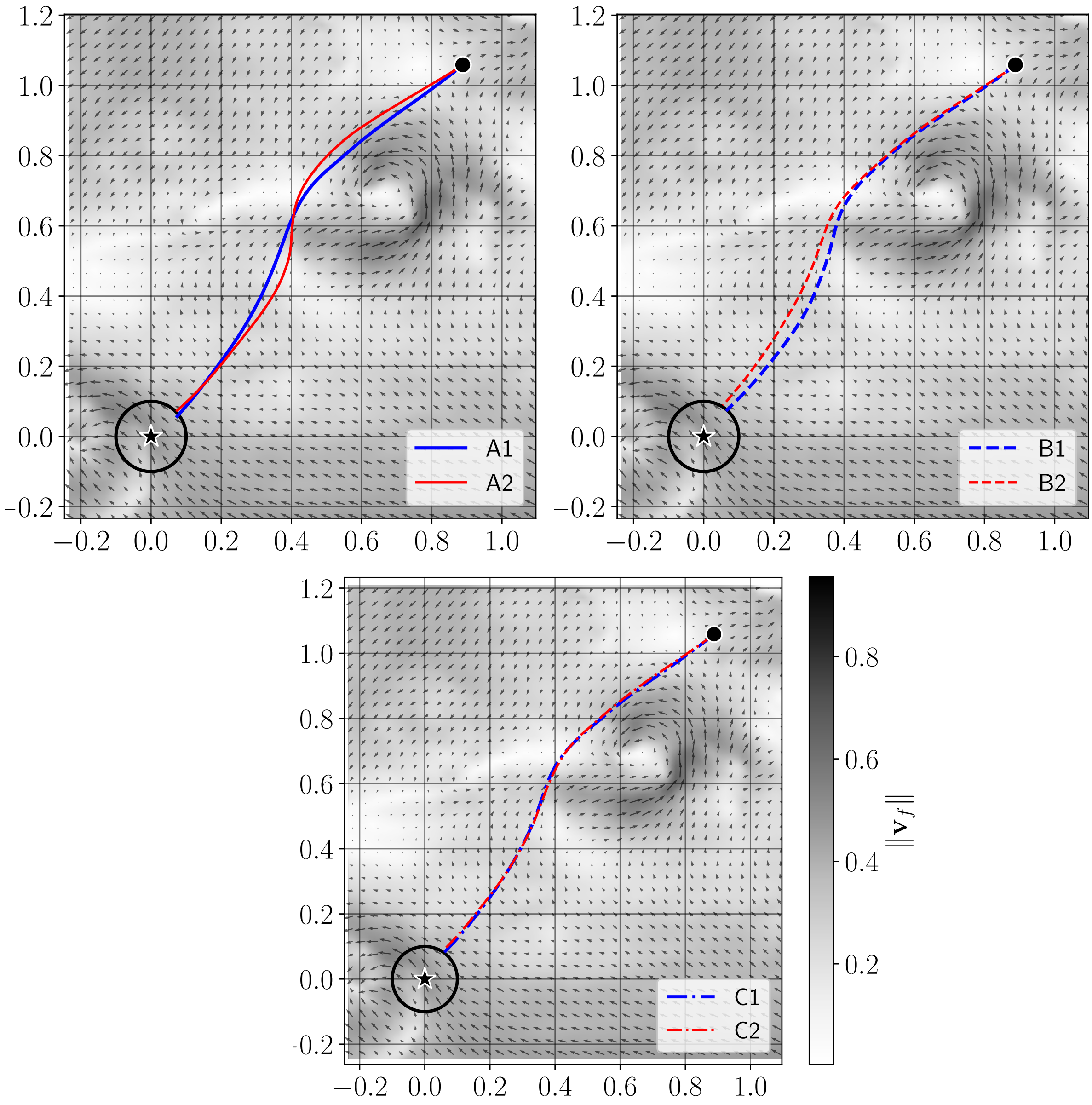


Figure 5.15: Fixed and varying SRF trajectories on the gyre case. Fixed SRF trajectories are depicted in red and varying SRF in blue. The underlying wind field is the time-varying wind field frozen at its first data frame, *i.e.* 2021-09-29 06:00Z

pair, the maximum distance between ground paths is approximately of 4% their total length. But in comparison, the airspeed diagram shows variations up to approximately 20%, so the difference is much more significant in airspeed than on the ground path. This is even more true for B1/B2 and C1/C2 as their ground paths are closer and they still show high variations in the airspeed.

The previous remark tends to hint at a simplified optimization procedure: compute constant-SRF (time-optimal) trajectories for various SRF values and build the corresponding time-energy diagram; let an operator select the satisfying time-energy trade-off; draw the corresponding path and optimize for airspeed along this path. This approach is computationally less intensive than looking for energy-optimal trajectories and can lead to a suboptimal choice that can be very close to the true optimum, because of the heading/SRF decoupling phenomenon.

5.5 Idea of algorithm

We finish this chapter by giving hints of how to build an algorithm based on extremals, in a similar fashion as in Chapter 3, that can find the energy-time Pareto front of a navigation problem.

Initial sampling As noticed in Sec. 5.4.3, the extremals now depend on the norm of the initial costate. So, the initial sampling of costate vectors must be bidimensional. Because of the different roles played by the heading and the SRF, we propose to do this sampling on an annulus-like grid of type:

$$\mathbf{p}_0^{(i,j)} = r_i \begin{pmatrix} \cos\theta_j \\ \sin\theta_j \end{pmatrix}$$

with $(r_i)_{i\in\{0,1,\dots,n-1\}}$ being a uniform sampling of some interval $[r_{\min}, r_{\max}]$ bounding the costate norm and $(\theta_j)_{j\in\{0,1,\dots,m-1\}}$ is a uniform sampling of $[0, 2\pi]$. Such a sampling is depicted is Fig. 5.16.

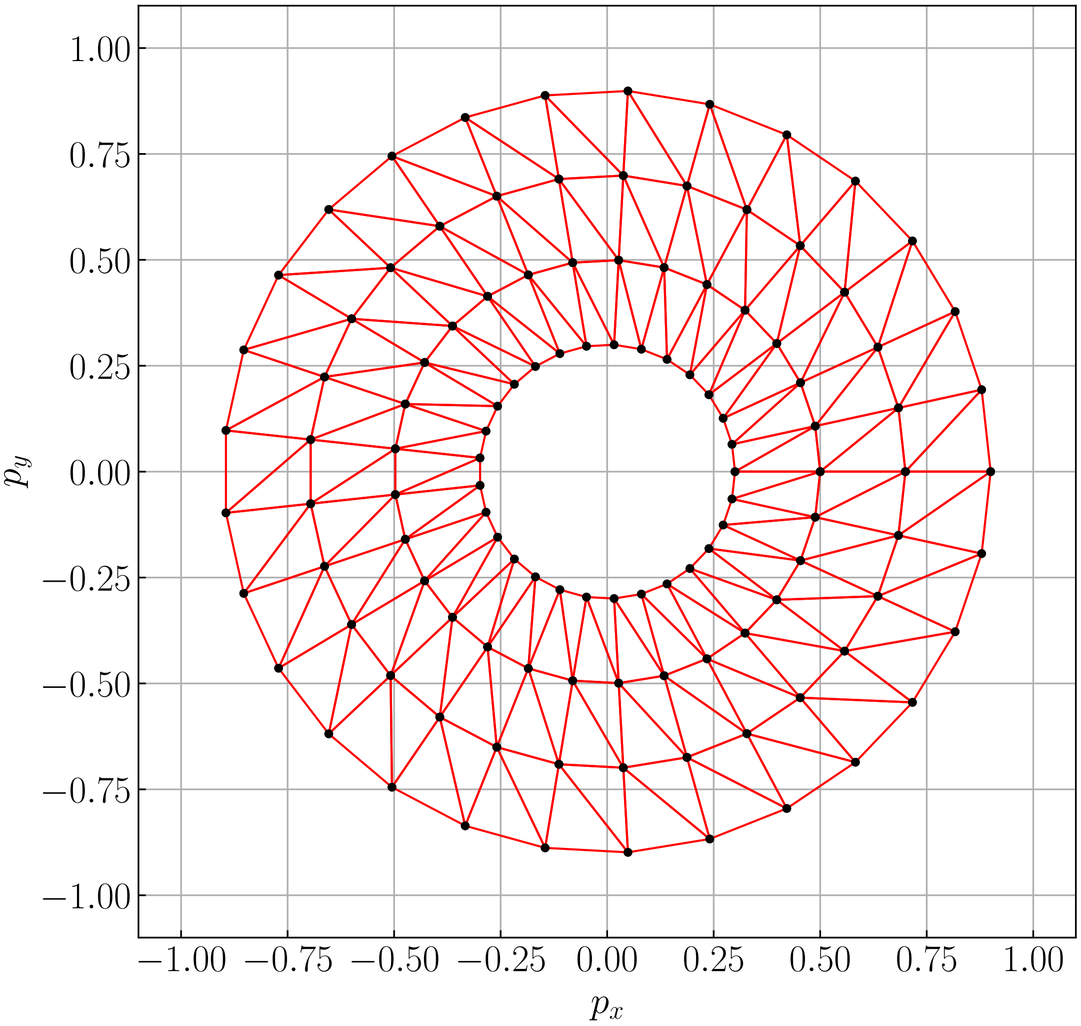


Figure 5.16: Example of costate space sampling to initialize extremals. Here, four different values for the costate norm are set, corresponding to the four crowns, and a discretization for angles is also chosen. Neighboring relations are set on a the lattice of triangles depicted here.

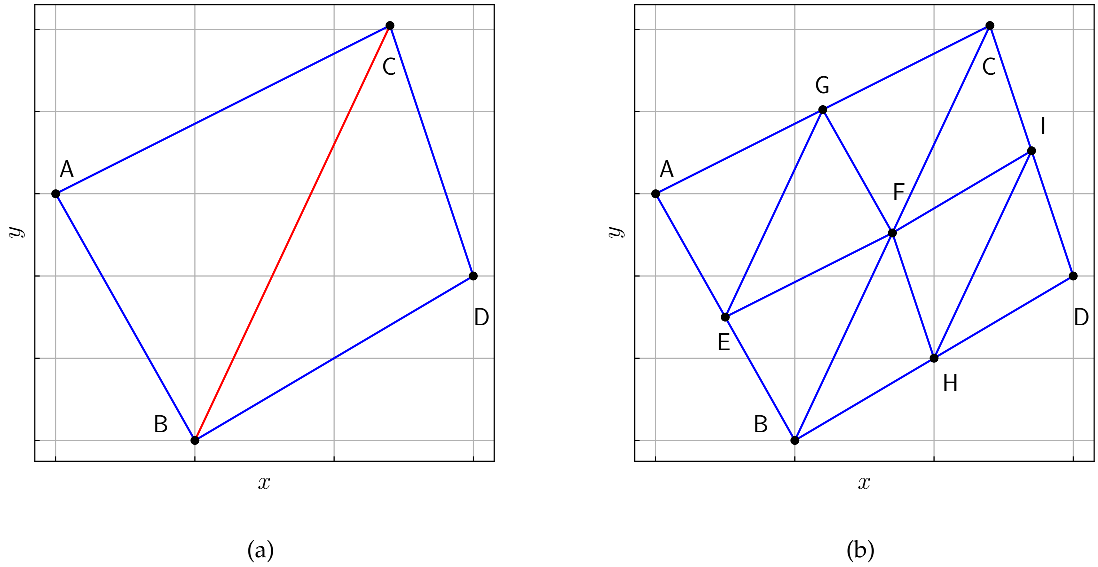


Figure 5.17: 'Embedded triangles' resampling procedure. (a) At some time, trajectories B and C are too far from one another. (b) Resampling is performed, adding middle points to all edges, and connecting edges consequently.

Resampling Once a discretization is chosen, trajectories can be integrated forward in time. As for the time-optimal case, neighboring trajectory will diverge at some point, and the information about the optimal cost between them will be inaccurate if not lost. So once again, resampling of extremals will be useful to keep the neighboring distances under a given threshold. Because the collection of extremals changed its geometry from a loop to an annulus when moving to energy-optimality, the resampling scheme has to be thought in order to preserve the structure of the extremal lattice. We propose to resample using an 'embedded triangles' scheme, as depicted in Fig. 5.17. When two extremals get too far from one another, e.g. B and C in Fig. 5.17, we take the middle point, F and initialize it with mean costate value (as in Algorithm 4) for further integration in time, and connect it to neighbors B and C. But to preserve a structure of extremals that is auto-similar, we also add the corresponding neighbors E, G, H and I to form new triangles of relation EFG and FHI. To preserve computation effort, the new extremals that did not result from the violation of the distance threshold (E, G, H, I) can be build from interpolating their parent neighbors. For instance, the state and costate values for E can be build from the mean of A and B values instead of starting a costly integration of E. But this shall stop when the distance threshold between A and B is violated, and a proper integration of extremal E must then start.

Conclusion of the chapter

In this chapter, the consequences of adding the vehicle's Speed Relative to the Flow as a variable on the total energy expense were studied. In the large-scale abstraction model, airborne and underwater vehicles show a similar type of relation between spent power and SRF: the power increases with the SRF and is convex. Still, airborne vehicles have a cost for idling while underwater vehicles don't. Without flow field, varying the SRF can only increase the total energy expense on a given travel. However, in the presence of a flow field, varying the SRF can be very beneficial, as demonstrated on a one-dimensional gust example, where higher SRF values were used to quickly reach the gust and lower values used when inside the gust, the latter pushing in the direction of motion, with overall reduction in total travel time compared to a fixed SRF approach. In a realistic airborne setting, it was found that savings of the order of ten percents could be expected when using a varying SRF law compared to a fixed SRF law. While for almost every example considered, the energy-time Pareto front was approximated numerically, in the case of a linear flow field and a quadratic power law, an analytical expression for the energy-time Pareto front was found, providing easy insight for a parametric study of its variations. Finally, hints are left to build a comprehensive extremal sampling algorithm for energy-time trajectory optimization in a similar fashion as the time-optimal algorithms from Chapter 3.

This chapter broadened the trajectory optimization problem to the case when several costs are to be minimized: namely, the energy expense and the travel time. But there may also be other costs which shall be minimized in a trajectory planning problem. In particular, if there is a diffuse hazard in the environment that one wants to avoid, then it is of particular interest to find trade-off trajectories between travel time and exposure to hazard. This is under study in the following chapter.

Chapter's main questions – Answers

- Is it beneficial for the vehicle to make its own speed vary throughout travel in terms of total energy expense?

  When there is no flow, since the power function is convex, no, it increases the energy expense. In the presence of a flow field, depending on the flow field magnitude, energy can be saved by increasing speed to reach favorable flow field zones and reducing speed when in the zone.

- If so, what is the amount of energy that can be saved?

  When the flow is weak, only a few percents of energy can be saved by varying the vehicle's speed. In realistic airborne settings, orders of tenths of percent reduction in the total energy expense can be reached when adapting the speed.

- How to compute numerically such energy-saving trajectories?

  All the methods applied to find time-optimal trajectories can generalize to the case of energy minimizing with varying speed. In particular, there are still differential equations to characterize the trajectories minimizing energy for fixed arrival time (extremals), that enables the use of shooting to find energy minimizing trajectories.

- How different are energy minimizing trajectories from time minimizing ones?

  In realistic settings, the shape of energy minimizing trajectories does not change much from time minimizing ones. However, the vehicle's speed can show high deviations from its averaged value to minimize the energy expense.

Chapter 6

Hazard-time optimal navigation problem

Abstract

In this chapter, Zermelo's navigation problem in unsteady and strong flow fields is extended to the presence of time-varying hazard in the environment. A level-set based method is proposed to compute hazard-time-optimal trajectories achieving trade-offs between travel time and exposure to hazard. The method is used on two case studies: the first one is the crossing of the Atlantic Ocean by an air drone avoiding stormy conditions and the second one is a collaboration mission between an air drone and a sea drone where each of the vehicle has to avoid hazard in the environment while completing its navigation mission. The case studies demonstrate that significant reduction in the exposure to hazard can be achieved when diverting from time-optimal trajectories to hazard-time-optimal trajectories.

Résumé en français

Dans ce chapitre, le problème de navigation de Zermelo dans un écoulement fort et instationnaire est adapté à la présence d'un danger instationnaire diffus dans l'environnement. Une méthode basée sur les ensembles de niveau est proposée pour calculer des trajectoires réalisant un compris entre temps de trajet et exposition au danger. La méthode est utilisée sur deux cas d'application: le premier étudie la traversée de l'Atlantique par un drone aérien en évitant les conditions orageuses et le second présente une mission collaborative entre un drone aérien et un drone sous-marin où chacun doit éviter une forme de danger tout en atteignant son but. Ces études de cas montrent qu'une réduction significative de l'exposition au danger est possible avec un temps de trajet modérément plus élevé en suivant les trajectoires danger-temps optimales à la place des trajectoires temps-optimales.

Contents

In the introduction, we briefly depicted the Atlantic crossing by Jean Mermoz on May 12$^{\text{th}}$. One important challenge that the crew faced was the crossing of the ITCZ, where large thunderstorms form, creating very unfavorable conditions for flight. The previous chapters presented methods to optimize routes for vehicles in flow fields, providing a way to find quickest or minimum energy routes, while avoiding forbidden zones. One may thus include thunderstorm areas as obstacles in the navigation problem. But looking at the data (see e.g. Fig. 6.2), these areas are very large, and the risk within, diffuse. Thus, the rigid-obstacle approach would be overconservative for such a problem. Instead, one can model thunderstorms as a space-time hazard field with high values where the thunderstorm is active and lower values else. This way, an accumulated exposure to hazard can be computed along trajectories, so that one can ponder trajectories on their travel duration and their total exposure to hazard, to decide what route is the best. In this chapter, we thus study hazard-time optimal trajectories, *i.e.* trajectories achieving a trade-off between exposure to a space-time hazard field and travel time to destination.

Chapter's main questions

- How can we compute trajectories achieving trade-offs between travel time and exposure to hazard?
- Is it possible to give guarantees that a trajectory is indeed the best possible trade-off?
- How much exposure to hazard can be avoided when adapting paths on real examples?
- How do trajectories change (shape, travel time) when we allow for longer travel time to avoid hazard?

6.1 Background

6.1.1 General background

Hazard-time optimal path planning has been addressed in the past, both for UAVs and AUVs, with different sources of hazard.

In Miller et al. (2011), UAV hazard-time-optimal path planning is solved using extremals (solving a Boundary Value Problem (BVP)) in the context of a UAV flying over regions of threat either in 2D or in 3D. However, no wind field is considered in this approach. Weather hazard for aircraft is tackled in Zhang et al. (2014), where the authors implement a path planning principle based on D* which accounts for the hazard of thunderstorms by computing an empirical risk based on weather ensemble predictions[1]. Paths achieving a trade-off between travel time and exposure to hazard are computed, and the authors demonstrate the possibility to reduce exposure to hazard when allowing for longer travel times. Similarly, Rudnick-Cohen et al.

[1] Weather ensemble forecast will be addressed specifically in Chapter 7.

(2016) use graph-based-like methods to minimize travel time and crash risk for UAVs. Still, the flow field advection is not incorporated in the previous approaches. Pereira et al. (2013) combine ocean current advection and hazard avoidance in the problem of routing AUVs in the ocean while avoiding maritime traffic. The authors formulate the problem as an Markov Decision Process (MDP) and solve it using policy iteration to provide risk-minimizing paths in expectation. This work is a little out of scope of this chapter because it relies on a stochastic model of the vehicle, which will only be addressed in Chapter 7.

6.1.2 Level-set methods for path planning

The work conducted in this chapter was the result of a collaboration with the MIT-MSEAS research group (the author took part in a 3-months visit at MIT). MIT-MSEAS is specialized in level-set methods for routing underwater vehicles in the ocean, as testify the numerous level-set method references given in the literature review that belong to this group. Thus, the work on hazard-time optimal trajectory planning presented in this chapter is not based on extremals, but based on the numerical resolution of the PDE governing the level-set function describing the vehicle's reachable set (Lolla et al. (2014), Bhabra et al. (2020)).

6.2 Methodology

In the 'Front propagation methods' section from the literature review, we already explained the principle of defining a function $\phi(t, \mathbf{x})$ such that $\phi(t, \mathbf{x}) < 0$ if $\mathbf{x}$ is reachable in time $t' < t$, $\phi(t, \mathbf{x}) = 0$ if $\mathbf{x}$ is reachable in time exactly t and $\phi(t, \mathbf{x}) > 0$ if $\mathbf{x}$ is not reachable in time less than t, and we gave the PDE satisfied by this function (see Lolla et al. (2014)). We also recalled that, because reachable fronts can feature shocks, *i.e.* non-differentiabilities, the correct framework for solutions of this type of PDE is the framework of viscosity solutions, which satisfy the PDE in a weak sense and allow non-differentiabilities.

In this chapter, we extend the level-set function to an augmented state space composed of the physical space variable $\mathbf{x}$ and the accumulated hazard variable η. We will do our development keeping a 2D planar model for the physical space, but it should be noted that the development adapts straighforwardly to 3D physical space. The only main difference is that PDE based methods scale badly with dimension, and using a 3D physical space can lengthen computation times significantly. The work presented in what follows is adapted from Schnitzler et al. (2024).

6.2.1 Problem statement

The problem at stake is that of computing the paths of vehicles that minimize both travel time in a dynamic flow environment $\mathbf{v}_f(t, \mathbf{x})$ and accumulated exposure to a dynamic hazard field[2] $\mathfrak{h}(t, \mathbf{x})$. Indeed we want to predict the reachable sets in the spatial and cumulative hazard dimensions, and all Pareto-optimal paths for the corresponding two costs, travel time and cumulative hazards.

The space-time instantaneous hazard field $\mathfrak{h}(t, \mathbf{x})$ is a positive field representing a hazard for the operation of the vehicle that is diffuse in the sense that what matters is the accumulation of exposure to hazard more than the instantaneous value itself. This models phenomenons that can deteriorate the vehicle with some probability at all time instants, with independence between the instants. The overall expected safety state for the vehicle at the end of the time window is thus really a matter of the integral of all risk taken. Hence, the modelling of hazard through the $\mathfrak{h}$ field.

Examples of hazard fields $\mathfrak{h}$ can be:

- a function of the precipitation (rain) field $\mathfrak{p}(t, \mathbf{x})$, *i.e.* $\mathfrak{h} = \mathfrak{h}(t, \mathbf{x}, \mathfrak{p})$;
- a function of hazardous ocean currents or wind fields $\mathbf{v}_f(t, \mathbf{x})$, *i.e.* $\mathfrak{h} = \mathfrak{h}(t, \mathbf{x}, \mathbf{v}_f)$;
- a function of variables such as expected vessel traffic density field $\rho_v(t, \mathbf{x})$, *i.e.* $\mathfrak{h} =$

[2] Note the notations are different from Schnitzler et al. (2024) to match the conventions of this thesis. For instance, in this thesis, the time variable always appears first in space-time functions.

$\mathfrak{h}(t, \mathbf{x}, \rho_v)$ (can be of interest for underwater vehicles but also for airborne vehicles flying at low altitude such as the Mermoz drone);

- …

In the $(\mathbf{x}, \eta)$ space, the vehicle evolution is now composed of the physical evolution and the accumulated hazard evolution:

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x}) \\ \dfrac{\mathrm{d}\eta}{\mathrm{d}t} = \mathfrak{h}(t, \mathbf{x}) \end{cases} \tag{6.1}$$

The Hazard-time-optimal Navigation Problem (HNP) at stake can be summarized as the following optimal control problem:

$$(\text{HNP}) \begin{cases} \min\limits_{\mathbf{u}(\cdot)} \, [t_f, \eta(t_f)] \\ \text{Eqs. 6.1} \\ \mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f \\ \eta(t_0) = 0 \end{cases} \tag{6.2}$$

6.2.2 Resolution

To solve the HNP, we compute the augmented reachable set for the vehicle, *i.e.*, the set of all values $(\mathbf{x}, \eta)$ that are reachable at some time t. We represent the reachable set by a scalar function $\phi(t, \mathbf{x}, \eta)$ whose subzero level set is the reachable set at time t, in a similar manner as for time-optimality but this time including the hazard dimension with the η variable. It can be proven (see the derivation in Bhabra (2021)) that the level set function ϕ is governed by the following exact Hamilton-Jacobi-Bellman PDE:

$$\frac{\partial \phi}{\partial t} + u_{\max} \left\| \frac{\partial \phi}{\partial \mathbf{x}} \right\| + \frac{\partial \phi}{\partial \mathbf{x}} \mathbf{v}_f(t, \mathbf{x}) + \mathfrak{h}(t, \mathbf{x}) \frac{\partial \phi}{\partial \eta} = 0 \tag{6.3}$$

For each of the applications presented in the following section, we scale variables and inputs (domain boundaries, $\mathbf{v}_f(t,\mathbf{x})$, $\mathfrak{h}(t,\mathbf{x})$) such that the governing equations are non-dimensional and the problem is numerically well conditioned, *e.g.*, the coordinate variables x, y, η are of an order of magnitude of 1. To compute the values of $\phi(t,\mathbf{x},\eta)$, we initialize ϕ to ϕ_0, the signed distance to a ball centered at the origin $(\mathbf{x}_0, 0)$ with radius ρ_{init}:

$$\phi_0(\mathbf{x},\eta) := \left\| \begin{pmatrix} \mathbf{x} \\ \eta \end{pmatrix} - \begin{pmatrix} \mathbf{x}_0 \\ 0 \end{pmatrix} \right\| - \rho_{\text{init}}$$

In theory, the perfect initialization would be obtained for $\rho_{\text{init}} = 0$, but in practice, this would lead to no grid point being part of the subzero level set, so the numerical scheme could not compute the evolution of the latter. We thus set ρ_{init} to be twice the grid spacing.

To compute the evolution of the subzero level set, we use a second-order Essentially Non-Oscillatory (ENO) scheme in space and a second-order scheme in time (described for instance in Doshi et al. (2023)).

Once $\phi(t,\mathbf{x},\eta)$ is computed, we obtain the Pareto-optimal ways to reach any destination point $\mathbf{x}_f$ in physical space and cumulative hazard level η. If $\mathbf{x}_f$ is reachable, we have the minimum duration $\tau^\star$ to reach $\mathbf{x}_f$ as:

$$\tau^\star := \min\left\{\tau \mid \exists \eta \in [\eta_{\min}, \eta_{\max}], \quad \phi(\tau, \mathbf{x}_f, \eta) = 0\right\} .$$

It is the first time the zero level set of ϕ reaches the segment $\left\{(\mathbf{x}_f, \eta) \mid \eta \in [\eta_{\min}, \eta_{\max}]\right\}$ in the augmented state space. Then, for larger durations $\tau \geq \tau^\star$, we can extract the minimum possible amount of hazard to reach the destination point in the exact given duration, *i.e.* :

$$\eta^\star(\tau) = \min\left\{\eta \mid \phi(\tau, \mathbf{x}_f, \eta) = 0\right\} .$$

The collection of points:

$$\{(\tau, \eta^\star(\tau)) \mid \tau \geq \tau^\star\}$$

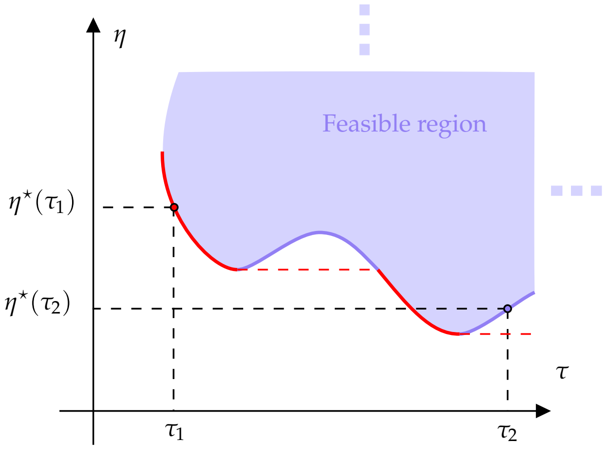


Figure 6.1: Hazard-time solution domain: feasible region and Pareto front. The horizontal axis is the total travel time and the vertical axis is the total accumulated hazard. The shaded region corresponds to feasible solutions (τ, η). The thicker lower curve contains the pairs $(\tau, \eta^\star(\tau))$ corresponding to hazard-only minimizing trajectories, for each time τ. The red portion of the curve encompasses the hazard-time optimal or so-called Pareto-optimal solutions.

is then the hazard-minimal curve to reach $\mathbf{x}_f$. A pair of travel time value and total accumulated hazard value (τ, η) is said to be Pareto-dominated if there exist a trajectory reaching the target in time τ' and hazard η', with $\tau' < \tau$ or $\eta' < \eta$. If a pair is not Pareto-dominated by any other pair, it is Pareto-optimal. These kind of pairs are particularly interesting in multi-objective optimization as they provide the best possible trade-offs between the objective values. Pareto-optimal pairs $(\tau, \eta^\star(\tau))$ can be deduced from the hazard-minimal curve by removing Pareto-dominated points. An example of such a curve is given in Fig. 6.1. In this figure, the red dot $(\tau_1, \eta^\star(\tau_1))$ corresponds to a Pareto-optimal hazard-time pair while the purple dot $(\tau_2, \eta^\star(\tau_2))$ is Pareto-dominated, because there are trajectories arriving earlier with the same amount of accumulated hazard.

When a designated pair $(\tau, \eta^\star(\tau))$ is chosen, one may then ask how to compute a trajectory that links the starting point in the augmented state space $(\mathbf{x}_0, 0)$ to the destination point with this performance. To solve for this, we perform *backtracking* of trajectories using the gradient of

ϕ. Using Eqs. 6.1, the backtracking ODEs in the augmented space are:

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} & = & -v_{\max}\,\mathbf{h}^{\star}(t,\mathbf{x},\eta) - \mathbf{v}_f(t,\mathbf{x}) \\ \dfrac{\mathrm{d}\eta}{\mathrm{d}t} & = & -\mathfrak{h}(t,\mathbf{x}) \end{cases} \tag{6.4}$$

where $\mathbf{h}^{\star}(t,\mathbf{x},\eta) := \left\| \left.\dfrac{\partial\phi}{\partial\mathbf{x}}\right|_{t,\mathbf{x},\eta} \right\|^{-1} \left.\dfrac{\partial\phi}{\partial\mathbf{x}}\right|_{t,\mathbf{x},\eta}$. These ODEs are initialized at time τ at position $(\mathbf{x}_f, \eta^{\star}(\tau))$. After performing the integration, one obtains a trajectory $(\mathbf{x}(\cdot), \eta(\cdot))$ from $(\mathbf{x}_0, 0)$ at $t = 0$ to $(\mathbf{x}_f, \eta^{\star}(\tau))$ at $t = \tau$.

6.3 Case studies

We propose a first application case where we consider airborne drones and hazard-time optimal long-distance missions with rain avoidance. It focuses on the conditions that the Mermoz drone would face over the Atlantic with storms to avoid.

In a second application case, we study hazard-time optimal collaborative mission between air and sea drones. An air drone must transport an ocean vehicle optimally to a target location exploiting winds and avoiding storms, and the ocean vehicle must subsequently complete its mission leveraging currents and avoiding vessel traffic hazards. It is motivated by missions off the US East Coast, around the New England Seamounts region (MSEAS NESMA Ex., 2024).

6.3.1 Rain-avoidance–time optimal planning for air drones

We consider the crossing of the Atlantic between Dakar, Senegal, and Natal, Brazil, by a UAV with a cruising speed of $23\,\mathrm{m\,s^{-1}}$, 100 m above sea level. In this application, we predict optimal trajectories for such an airborne drone providing all possible trade-offs between travel time by exploiting instantaneous winds and hazards by limiting thunderstorm exposure. Most precisely for the latter, we select the rain field as a proxy for thunderstorms. Since thunderstorms commonly imply heavy rain, this choice is conservative in avoiding rain, possibly avoiding zones with only some rain but no thunderstorms. Another reason why we work with rain data rather than thunderstorm data is because the former is available in practically all weather products. For instance, rain is available in both ERA5 reanalysis data and ECMWF forecasts, two open-access data sources. This is not the case for parameters such as the number of lightning per surface unit area or the cloud coverage, which may be found in reanalyses but not necessarily in forecasts.

We thus take $\mathfrak{h}(t, \mathbf{x}, \mathfrak{p}(t, \mathbf{x})) := a\, \mathfrak{p}(t, \mathbf{x})$, where $\mathfrak{p}(t, \mathbf{x})$ is the ECMWF 3h-accumulated rain forecast and a is a dimensional scaling parameter that relates the hazards to the drone as a function of the rain. Of course, different parameterization for instantaneous hazards can be selected such as higher-order polynomials or other functions that increase hazards nonlinearly as the rain increases and are representatives of the risks to the air drone. For our specific application, we select a linear function of the 3h-accumulated rain and the ECMWF rain forecast issued on 2024-04-25 00:00Z. For the environmental flows $\mathbf{v}_f(t, \mathbf{x})$, we employ the corresponding ECMWF wind forecast. Snapshots of these wind and rain fields are given in Fig. 6.2, clearly indicating the prevailing northeasterly trade winds and strong rain storms around the equator.

The air drone takes off on 2024-04-27, at 15:00Z. It flies to the destination in the fastest possible time leveraging winds while avoiding high accumulated exposure to rain. Example key questions include: Will rain avoidance lead to significantly different paths from the fastest ones, both in travel time and shape of the path? What are all of the hazard-time optimal paths so that the drone operators can select the paths most appropriate to their level of risk and desired arrival time? What is the cumulative hazard that corresponds to the fastest time path? Is there an arrival time that avoids the forecast rain? Our hazard-time optimal analysis provides clear answers to all such questions.

Solving Eq. 6.3, we obtain the hazard-time reachable set and reachability front. We can then compute all rain-travel-time Pareto-optimal paths solving the backtracking Eqs. 6.4. They minimize the accumulated rain and travel time and include the overall minimum travel time path, i.e., the path that minimizes travel time by optimally exploiting the wind field while ignoring

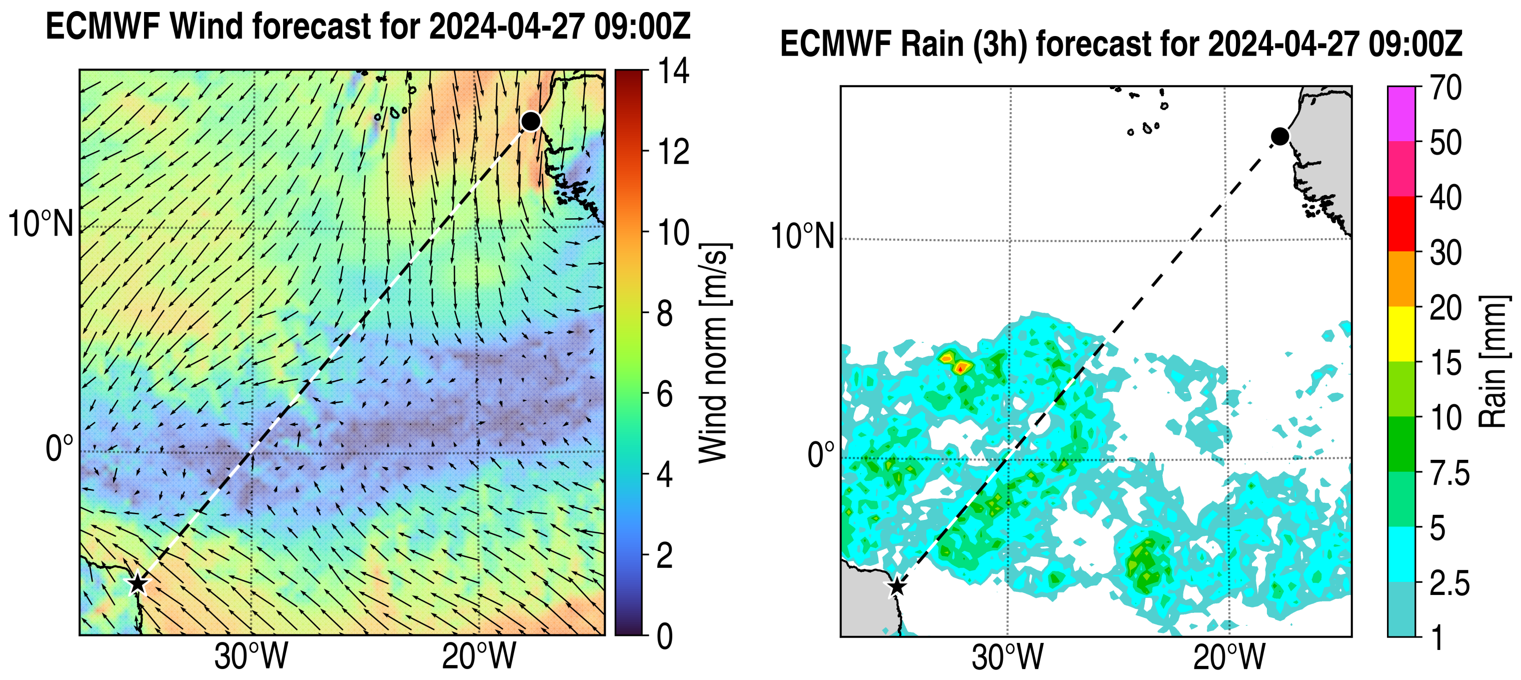
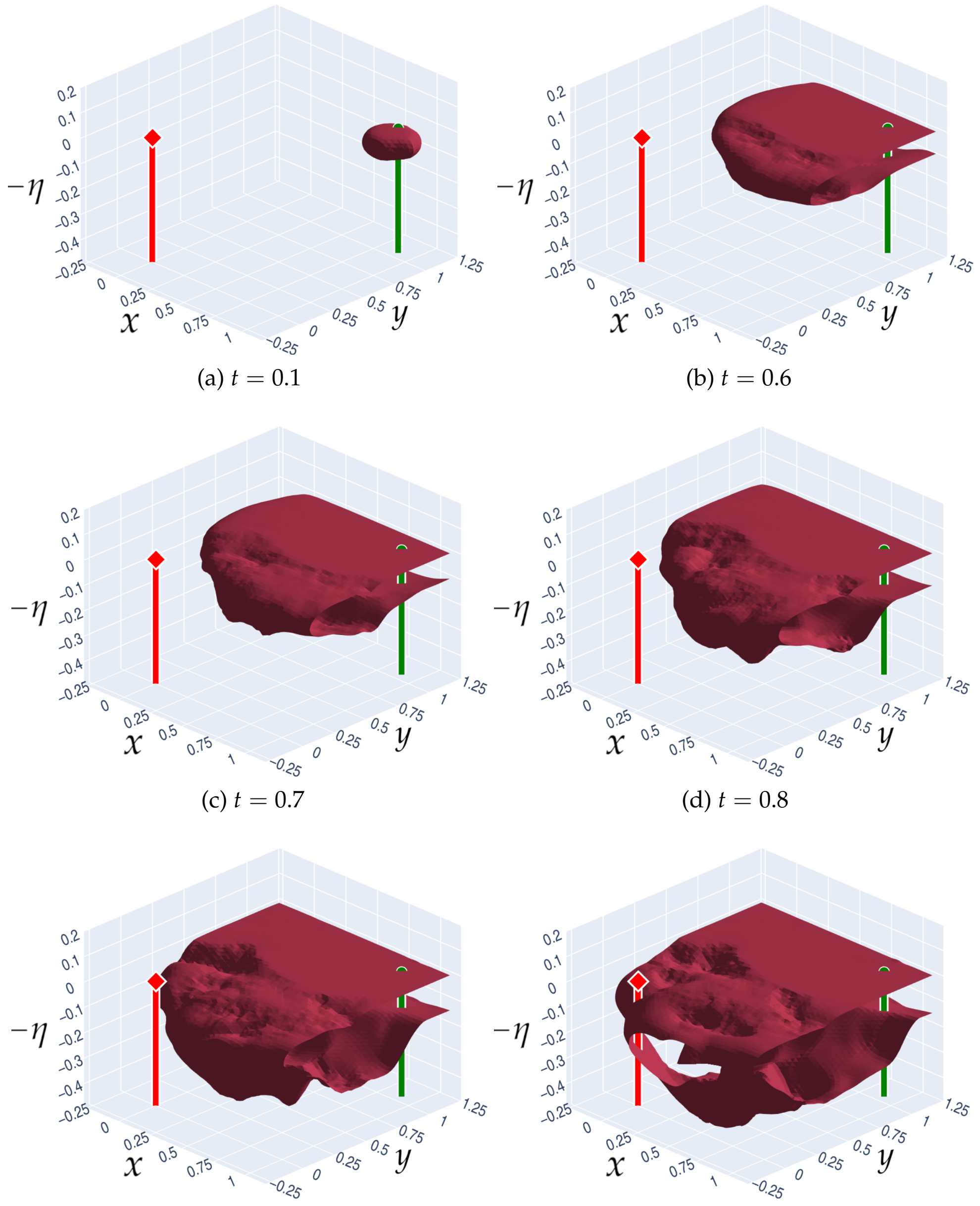

ECMWF Wind forecast for 2024-04-27 09:00Z
10°N
0°
30°W
20°W
Wind norm [m/s]
14
12
10
8
6
4
2
0
ECMWF Rain (3h) forecast for 2024-04-27 09:00Z
10°N
0°
30°W
20°W
Rain [mm]
70
50
40
30
20
15
10
7.5
5
2.5
1

Figure 6.2: Instantaneous wind (100 m) and 3h-accumulated rain forecast snapshots from ECMWF. The path planning start point is depicted as a black circle, the endpoint as a black star, and between them the shortest-distance great circle is drawn.



(a) $t = 0.1$

(b) $t = 0.6$

(c) $t = 0.7$

(d) $t = 0.8$

(e) $t = 0.9$

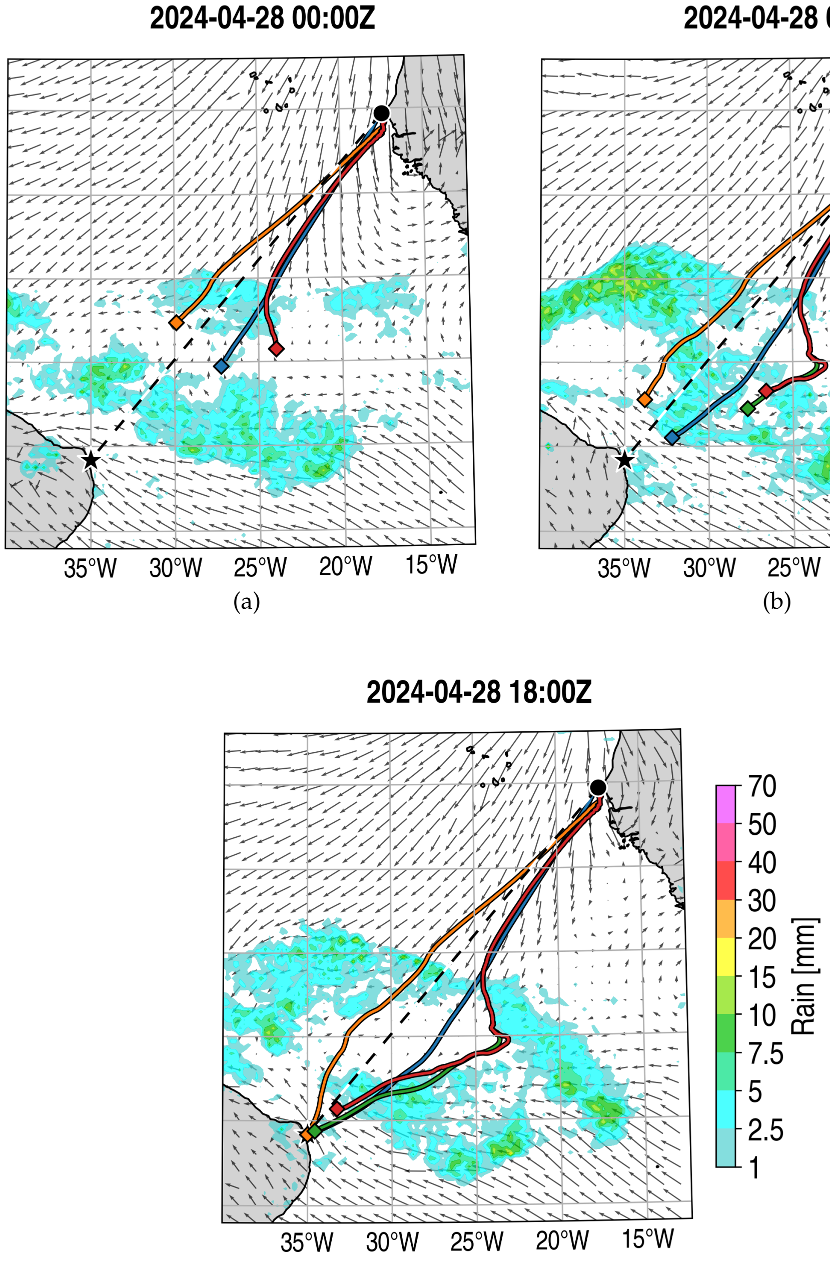
(f) $t = 1$

Figure 6.3: Hazard-time optimal air drone crossing the Atlantic Ocean: Reachable set evolution in the augmented state space (x, y, η). The hazard dimension is flipped (the vertical axis is indeed a $-\eta$ axis) to more directly visualize the hazard-minimizing part of the reachable set; in these graphs, the minimum hazards are thus at the top of the 3D reachable set. The start location is depicted as a green dot with a green line in the direction of increasing hazards. Similarly, the final location is depicted as a red diamond with a red line in the same direction. All times t, space (x, y), and cumulative hazard η are non-dimensional.

rain. In Fig. 6.3, we display the evolution of the reachability front, which is the zero-level set of the value function ϕ in the cumulative hazard η and physical space (x, y). In Fig. 6.3a, the reachable set started propagating from its initial position. There is no rain around the starting point so the shape of the reachable set is governed by time-optimal trajectories in the winds. The width of the reachable set on this frame in the hazard direction remains twice ρ_{init}. At $t = 0.6$ (Fig. 6.3b), the reachable set has reached the rain zone and hazards started to accumulate: the front there changed appearance from flat to rough, reflecting the rain patterns. As stronger hazardous rain is encountered, the reachable set goes down (Fig. 6.3c). In Figs. 6.3d–e, the part of the front that is the closest to the red line corresponds to trajectories close to the time-optimal ones. This part of the front accumulates much hazard and is thus much lower than the original flat portion of the front. Finally, in Fig. 6.3f, we observe how slower but less hazardous trajectories are found within portions of the front intersecting the destination red line with lower accumulated hazard.

In Fig. 6.4, we show three snapshots of hazard-time optimal paths for four optimal travel times and cumulative hazards, the first path (blue) being the strictly time-optimal path (ignoring rain). We find that, while the pure time-optimal path rushes through the rain, other trajectories computed for longer travel time but lower exposure to hazard manage to avoid heavy rains. We also observe that winds are quite steady, but the rain field evolves quickly. The time-optimal trajectory in blue crosses a zone of high precipitation (Fig. 6.4b). The orange trajectory encounters less rain by taking a route north of the great circle between Dakar and Natal. The green and red trajectories deviate even more from the great circle path: while the rain exposure is not significant enough, they both first follow the strictly time-optimal path, but then leave it (Fig. 6.4a) and meander along an evolving, quiet zone (Fig. 6.4b) with less rain before reaching the final destination (Fig. 6.4c).

In Fig. 6.5, we show Pareto-optimal travel times and cumulative hazards. The values are given in Table 6.1. We find that a moderate increase in travel time can result in a significant decrease in exposure to hazards. For example, simply changing from the strictly time-optimal,



(a)

(b)

(c)

Figure 6.4: Hazard-time optimal air drone crossing the Atlantic: (a, b, c) Snapshots at three different times of four hazard-time Pareto-optimal trajectories overlaid on the corresponding rain and wind fields. The paths are colored according to their arrival time and cumulative rain hazard level, as shown in Fig. 6.5.

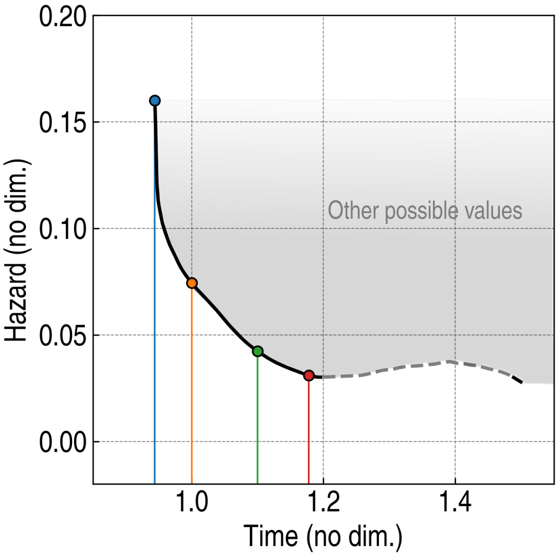


Figure 6.5: Hazard-time optimal air drone crossing the Atlantic: Minimum total cumulative rain hazard for various travel times for the Dakar-Natal crossing. Solid curves are Pareto optimal values. The four colored dots on the Pareto-front correspond to paths shown with their color in Fig. 6.4.

blue trajectory to the orange one gives a reduction in rain hazards of 54%, while increasing the travel time by only 5.9%. The table shows the operational benefit of our joint hazard-time trajectory optimization by providing the operator with a variety of optimal paths with different performance. The operator or the drone itself can select its preferred optimal path depending on the criticality of being on time compared to being exposed to hazards.

In Fig. 6.5, we also show non-Pareto-optimal portions on the hazard-time graph (dashed line). This confirms that increasing the travel time is not a sufficient condition for lowering exposure to rain. For instance, if heavy rain is barring the road, any trajectory passing through to ensure a given travel time will inevitably have a high accumulated hazard when reaching the destination.

6.3.2 Hazard-time optimal air-sea collaboration

This second mission consists of two parts. First, a transport air drone departs from Boston, MA on 2024-02-17, 10:00 UTC, and drops a sea drone in the region of the New England seamounts

Travel duration τ		Total hazard $\eta^{\star}(\tau)$	
0.944	-	0.160	-
1.000	+5.9%	0.0743	-54%
1.100	+17%	0.0423	-74%
1.178	+25%	0.0310	-81%

Table 6.1: Hazard-time optimal air drone crossing the Atlantic: Travel times and cumulative rain hazards for the four backtracked trajectories of the Dakar-Natal air mission. Both travel time and hazard are non-dimensionalized variables which have no units.

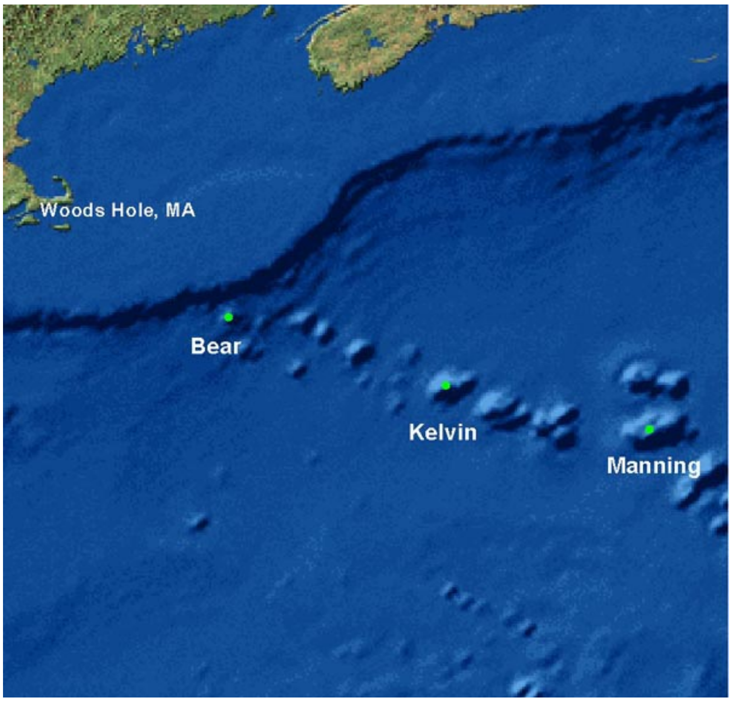


Figure 6.6: A map showing the position of the New England seamounts. (Source: NOAA)

(see Fig. 6.6) at point [63°W, 38.5°N] in fastest time, taking advantage of favorable winds and avoiding unfavorable ones. However, it faces forecast stormy conditions with high wind bursts and hazardous rain that require a trade-off between exposure to hazards and travel time. Second, once the sea drone has been dropped, its mission is to proceed to the north of the Atlantis II seamount in fastest time in accord with dynamic ocean currents. However, it needs to limit its exposure to hazards from expected vessel traffic, hence a second ocean hazard-time optimal reachability analysis is completed.

For this collaborative air-sea mission, example key questions include: Can the transport air drone reach its final location soon enough for the sea drone while avoiding regions with too strong storms? What are all the rain-hazard, vessel-traffic-hazard, and time optimal choices for the air and sea drone operators? What are the optimal sea drone paths that collect the desired ocean data in fastest time while minimizing vessel-traffic hazards? Our collaborative hazard-time optimal analysis can answer all such questions.

Rain-hazard and time optimal transport air drone

For our application, for the transport by the air drone, we employ data from the ERA5 atmospheric reanalysis from the European Center for Medium-range Weather Forecasts (ECMWF), accessible through the Copernicus climate Data Store (CDS). For the environmental flows $\mathbf{v}_f(t, \mathbf{x})$, we extract the hourly 100 m wind field, and for the rain hazard, the 1h-accumulated rain field $\mathfrak{p}(t, \mathbf{x})$. Once again, we define the instantaneous hazard from the rain field as $\mathfrak{h}(t, \mathbf{x}, \mathfrak{p}(t, \mathbf{x})) := a\, \mathfrak{p}(t, \mathbf{x})$ where the dimensional scaling parameter a ensures that the resulting $\mathfrak{h}$ represents the instantaneous hazard due to rain. We depict the wind and rain conditions at 2024-02-17 18:00 UTC in Fig. 6.7. There is a storm with counter clock-wise winds passing through the region, creating favorable conditions for trajectories that bend southward. The maximum of precipitation passes through the shortest-distance great circle between the start and destination locations, also encouraging trajectories to avoid this shortest-distance path to the destination. The dynamic behavior can also be seen in Fig. 6.8.

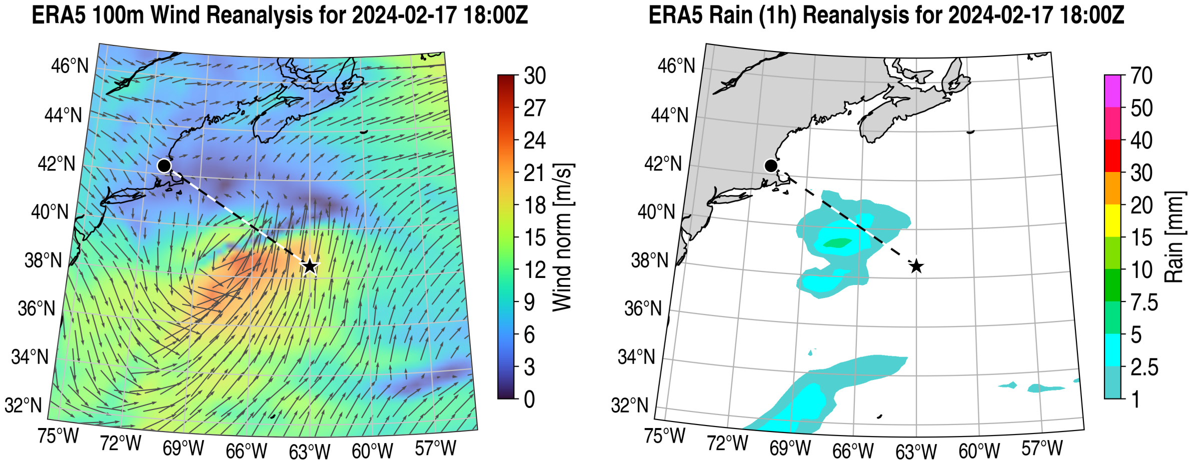
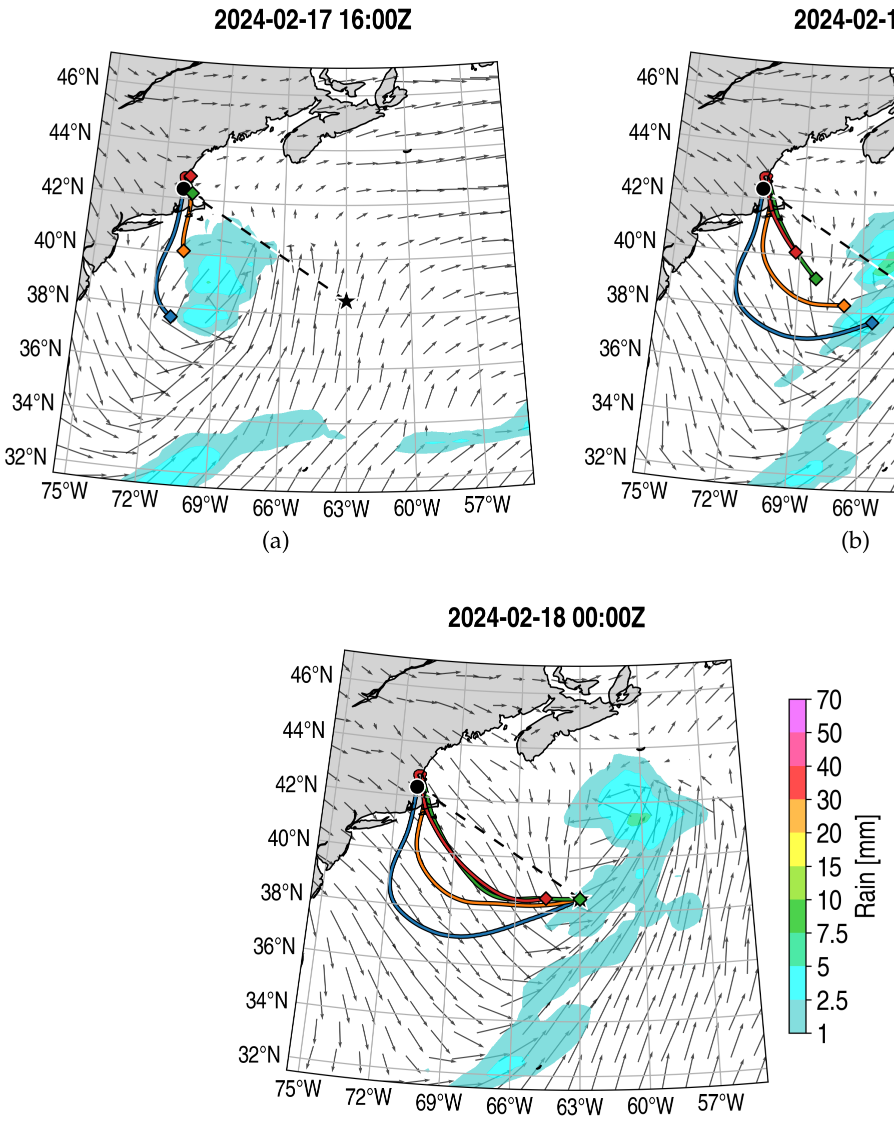


Figure 6.7: Instantaneous wind (100 m) and rain snapshots from the ERA5 reanalysis. The path planning start point is depicted as a black circle, the endpoint as a black star, and between them the shortest-distance great circle is drawn.

We again solve Eq. 6.3 to obtain the hazard-time reachability front and the Pareto front, and then the backtracking Eqs. 6.4 to highlight Pareto-optimal paths to the destination. Four of these Pareto-optimal trajectories are depicted in Fig. 6.8. The corresponding Pareto front is shown in Fig. 6.9.

As shown in Figs. 6.8-6.9, the presence of a storm featuring high winds accelerating the vehicle but also much rain entails a clear variation in the amount of total hazard depending on the desired travel time. From the monotonic Pareto front (Fig. 6.9), trajectories going the fastest are the most exposed to the rain. Other trajectories accepting longer travel times can follow the wake of the storm, thus avoiding most of the rain. What is noticeable is that with an order of magnitude of 23 m/s for the speed of the air drone, optimal trajectories are bending significantly south, taking the western side of the storm and differing much from the great circle joining the start to the destination. For example, the fastest blue trajectory is much longer than this shortest distance. Nonetheless, it encounters rain in the back of the storm (Fig. 6.8b-c). Other Pareto-optimal trajectories that don't take full advantage of the strong winds can drastically reduce their total exposure to this rain hazard by remaining in the dryer side of the storm while still arriving only a bit later. For example, the orange and green trajectories



(a)

(b)

(c)

Figure 6.8: Hazard-time optimal air-sea collaboration. Air drone reaching the New England Seamounts in hazard-time optimal fashion, transporting the sea drone: (a, b, c) Snapshots of four hazard-time Pareto-optimal trajectories, overlaid on rain and wind fields. The paths are colored according to their arrival time and cumulative rain-hazard level, as shown in Fig. 6.9.

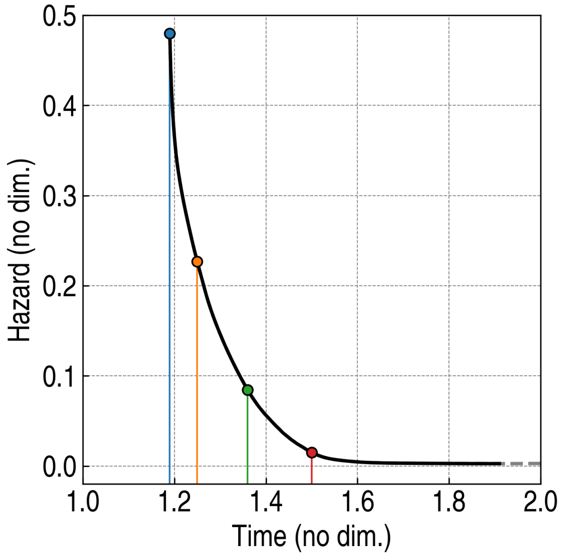


Figure 6.9: Hazard-time optimal air-sea collaboration: Minimum total cumulative rain-hazard for various travel times for the air drone transporting a sea drone to the New England Seamounts. Solid curves are Pareto optimal values. The four colored dots on the Pareto-front correspond to paths shown with their color in Fig. 6.8.

(Fig. 6.8) reduce the cumulative rain hazard by about 100 to 500 percent while only increasing travel time by 10 to 20 percent, see Fig. 6.9.

Vessel-traffic-hazard and time optimal sea drone

Once the air drone reaches its destination at approximately 2024-02-18 01:00 UTC, it drops a sea drone in the vicinity of the New England Seamounts. This sea drone then travels along the ocean surface at a nominal speed of 3 m/s to location [64°W, 40.5°N], north of the Atlantis II seamount in fastest time to collect data and rendezvous with a research vessel in the area. During this journey, it faces a trade-off between travel time and exposure to expected hazards due to interfering vessel traffic.

For the ocean environmental flows $\mathbf{v}_f(t, \mathbf{x})$, we utilize the ocean current hindcasts from the MIT-MSEAS primitive-equation ocean model (Haley and Lermusiaux, 2010; Haley et al., 2015; Lermusiaux et al., 2024). For the surface vessel traffic hazard, we employ the historical traffic density data from the Global Maritime Traffic Density Service (GMTDS) in terms of hours of vessel traffic per square kilometer. We thus define the instantaneous hazard from this expected vessel traffic density field $\rho_v(t, \mathbf{x})$ as the linear function $\mathfrak{h}(t, \mathbf{x}, \rho_v(t, \mathbf{x})) := b\, \rho_v(t, \mathbf{x})$ where the dimensional scaling parameter b ensures that the resulting $\mathfrak{h}$ represents the instantaneous hazard due to vessel traffic. As noted for the rain hazards (see Sec. 6.3.1), different parameterization for instantaneous vessel hazards can be selected such as higher-order polynomials or other functions that increase hazards nonlinearly as vessel traffic density increases and are representatives of the risks to the chosen ocean vehicle. In general, ρ_v can be data that varies with the time of the day or a fully dynamic forecast $\rho_v(t, \mathbf{x})$. In our example, we utilized the historical time-averaged vessel traffic GMTDS data so ρ_v is a spatial field steady in time.

We solve Eq. 6.3 to obtain the hazard-time reachability front and all Pareto-optimal solutions. We then solve the backtracking Eqs. 6.4 to highlight Pareto-optimal paths the destination. Three of these Pareto-optimal trajectories are depicted in Fig. 6.10. The corresponding traffic-hazard and time Pareto front is shown in Fig. 6.11.

As shown in Fig. 6.10, the sea drone is operating in an area with several historical transit lanes, visible in yellow and white in the figure background. A fast Pareto-optimal path (shown in blue) takes advantage of the forecast currents but goes across some high-density and wide-double vessel traffic lanes hence has a high cumulative hazard, as shown in Fig. 6.11. Two other Pareto-optimal paths with lower cumulative hazard (shown in red and green) minimize hazard-time by first crossing the southern transit lane at the area of lowest historical traffic density at approximately [63°W, 39.5°N]. Both of the lower hazard paths then move to the west. By doing so, the ocean vehicle is only required to cross one additional transit lane (instead of a wide double-lane to the east), and avoids the large intersection that occurs near [63.5°W, 40°N].

Conclusion of the chapter

In this chapter, hazard was introduced in the environment as a diffuse quantity to avoid. It leaves more freedom to design trajectories avoiding possibly dangerous zones than the ban-

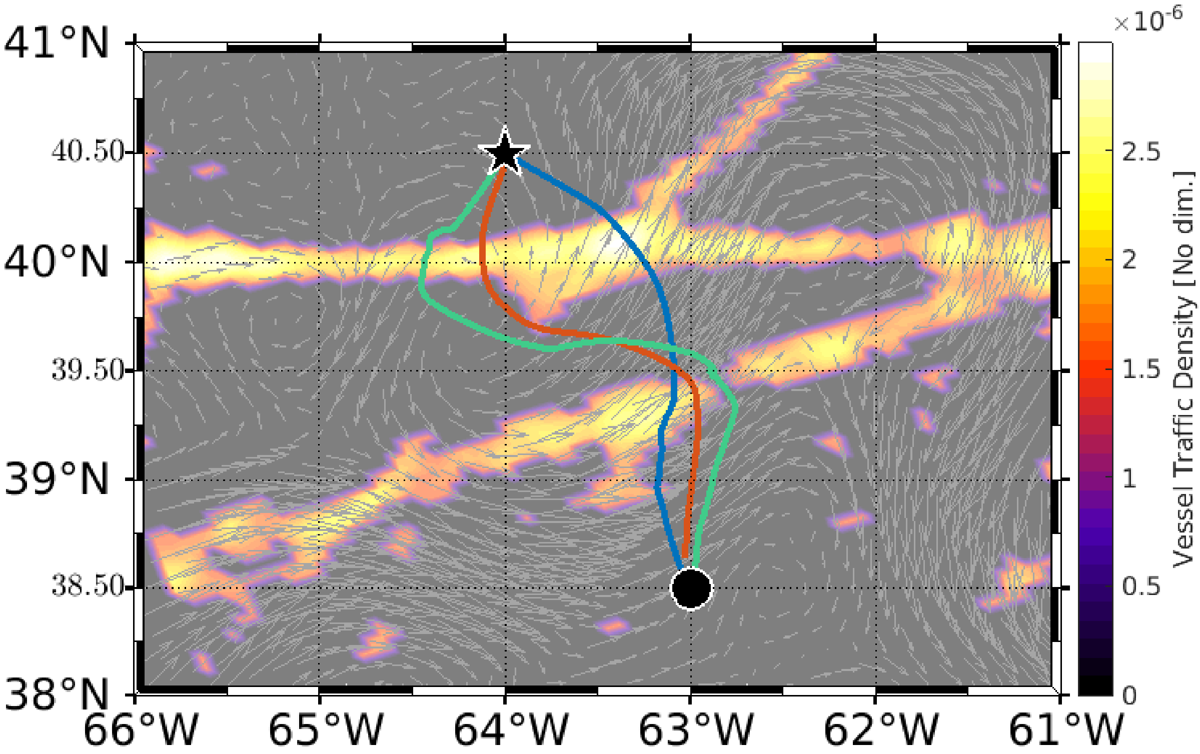


Figure 6.10: Hazard-time optimal air-sea collaboration: Pareto-optimal trajectories for the sea drone, overlaid on the non-dimensional traffic density field and MIT-MSEAS forecast ocean current vectors. The three paths are colored according to their arrival time and cumulative traffic-hazard level, as shown in Fig. 6.11. From Schnitzler et al. (2024).

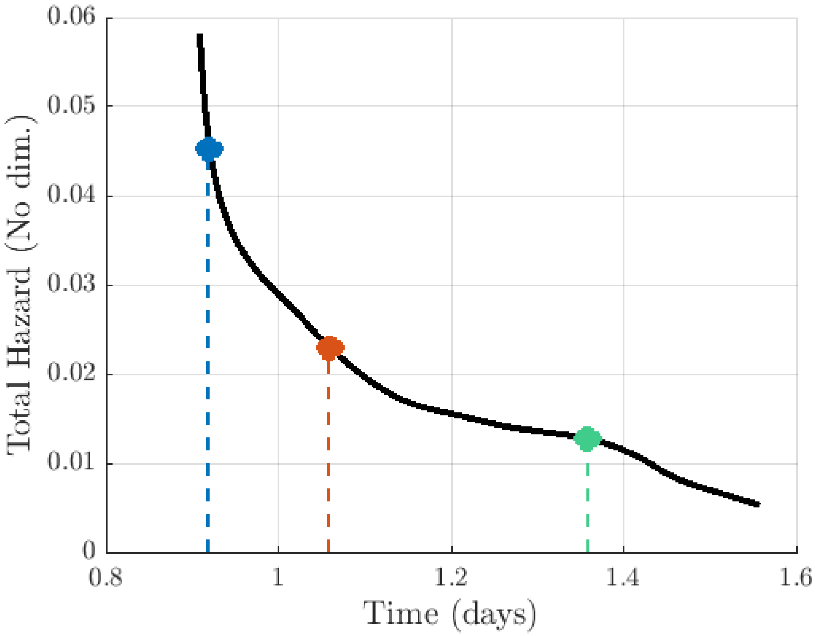


Figure 6.11: Hazard-time optimal air-sea collaboration: Minimum total cumulative traffic-hazard for various travel times for the sea drone. Solid curves are Pareto optimal values (Pareto front). The three colored dots on the Pareto-front correspond to paths shown with their color in Fig. 6.10. From Schnitzler et al. (2024).

ning of areas with hard obstacles. It is shown that hazard-time-optimal trajectory planning can be performed by using already existing tools for collection-time-optimal trajectory planning. A Partial Differential Equation on a level-set function describing the reachable space for the vehicle was solved to find all hazard-time optimal trajectories. The problem's dimension is increased by one when adding the hazard dimension, which is challenging for Hamilton-Jacobi-like methods solving PDEs. However, the level set method showed satisfying performance for strategic planning with computation times never exceeding half an hour for each considered application case. Hazard-avoiding paths were shown to differ much from time-optimal paths in the physical space but sometimes with little difference in the total travel time and significant reduction in the exposure to hazard, which proves that hazard-time-optimal trajectory planning is relevant in applied contexts. Hazard-time-optimal path planning also ensured more safety for the success of collaborative air-sea missions than pure time-optimal path planning, showing that storms and vessel traffic could be avoided in the paths of an air drone and a sea drone, with moderate, hence likely acceptable, increase in time at rendez-vous points and highly reduced exposure to hazardous phenomenons.

In this chapter, risks in the navigation were accounted for through the use of a deterministic, time-varying hazard field, assessed on environmental data. But risks can also come from the uncertainty in the capacity to predict the state of the environment at future times. In the following chapter, the focus will be on the risk coming from an uncertain weather prediction.

Chapter's main questions – Answers

- How can we compute trajectories achieving trade-offs between travel time and exposure to hazard?

  By adding the hazard dimension to the state space, the problem of finding trajectories achieving trade-offs between travel time and exposure to hazard boils down to computing the reachable set for the vehicle in the augmented state space. The evolution of the reachable set is governed by a partial differential equation that can be solved numerically. From the knowledge of the reachable set evolution in time, one can find optimal trajectories by the backtracking of the gradient of the level set function describing the reachable set.

- Is it possible to give guarantees that a trajectory is indeed the best possible trade-off?

  If one is able to compute the exact reachable set for the vehicle in the aforementioned augmented state space, as is the case when solving the governing partial differential equation, then one knows for what times and hazard accumulation values the destination is reachable or unreachable, and thus one can check the Pareto-optimality of a given trajectory, or in other words if a pair of travel time and accumulated hazard is a best trade-off or not.

- How much exposure to hazard can be avoided when adapting paths on real examples?

  Depending on the case, a division by 3 or 4 in the total hazard was demonstrated in practical cases. In special cases with large null hazard zones it is even possible to avoid all the hazard with longer travel times.

- How do trajectories change (shape, travel time) when we allow for longer travel time to avoid hazard?

  The hazard-time-optimal trajectories' shape can differ much from the time-optimal trajectory, harnessing the areas of minimal hazard. On the considered real cases, from the minimal travel time up to 50% more, the total hazard decreases significantly, but for longer travel time, the decrease is weaker.

Chapter 7

Optimal navigation in the presence of uncertainty

Abstract

In this chapter, Zermelo's navigation problem in unsteady and strong flow fields is extended by adding uncertainty in the problem. Uncertainty models are distinguished whether they put uncertainty in the infinitesimal motion or only in the flow field data. In the case of ensemble prediction for the flow field on a real world example, as many time-optimal trajectories as weather scenarios are computed, and the travel times resulting from following these trajectories in each of the different scenarios are computed. This analysis shows that there is little variation in the expected travel time among paths, but that some paths can reduce the variance in travel time. In a second time, a trajectory model based on ground paths and an average-time-optimal path planning problem are introduced. For the latter, necessary conditions of optimality for paths are given using Pontryagin's Maximum Principle the characterize trajectories candidate to optimality (extremals). The ability of path-based extremals to find the minimum average-time path is demonstrated on an analytical case of a moving vortex with unknown direction of motion. There is however no significant gap in the optimum average time found compared to the averaged travel time on other paths computed using the regular time-optimal approach on the considered example.

Résumé en français

Dans ce chapitre, le problème de navigation de Zermelo dans un écoulement fort et instationnaire est étendu en ajoutant de l'incertitude au problème. Les modèles d'incertitude sont différenciés selon qu'ils introduisent de l'incertitude dans le mouvement infinitésimal ou seulement dans les données de l'écoulement. Dans un cas réel de prédiction d'ensemble pour l'écoulement, autant de trajectoires optimales en temps que de scénarios météorologiques sont calculées, et les temps de parcours résultant du suivi de ces trajectoires dans chacun des différents scénarios sont calculés. Cette analyse montre qu'il y a peu de variation dans le temps de parcours attendu entre les trajectoires, mais que certaines trajectoires peuvent réduire la variance en temps de parcours. Dans un deuxième temps, un modèle de trajectoire basé sur les chemins au sol et un problème de planification de trajectoire optimale en temps moyen sont introduits. Pour ce dernier, les conditions nécessaires d'optimalité des trajectoires sont données en utilisant le principe du maximum de Pontryagin pour caractériser les trajectoires candidates à l'optimalité (extrémales). La capacité des extrémales basées sur les chemins à trouver la trajectoire minimale en temps moyen est démontrée sur un cas analytique d'un tourbillon en mouvement avec une direction de mouvement inconnue. Il n'y a cependant pas d'écart significatif entre le temps moyen optimal trouvé et les temps de parcours moyennés sur les autres chemins optimaux en temps sur l'exemple considéré.

Contents

Throughout all the previous chapters, maybe the reader found fault with the high confidence placed in the problem data (flow field values, obstacle description, hazard fields, . . .). The present chapter will try to open up the discussion to the cases for which the problem data is not precisely known. The most canonical uncertainty source comes from the weather prediction, since atmospheric or oceanic systems are chaotic and thus increasingly difficult to predict the further we go in the future. But perturbations can also occur on the small scale: positioning errors, unpredicted sub-grid phenomena in the flow field, actuator malfunction. . . In any case, taking into account these sources of uncertainty in the planning process is beneficial for the decision-making, provided one has a good model of uncertainties.

Chapter's main questions

- When the problem contains sources of uncertainty, is it possible to plan for an optimal operation of the vehicle that prevents unexpected situations in the navigation?
- Can the benefit of choosing safer or riskier trajectories be quantified?

7.1 Uncertainty models

The literature shows two main ways to deal with uncertainty. The first one is the framework of stochastic processes, where there is random noise acting at the local scale perturbing the vehicle's trajectory. The second one is based on weather scenarios, that is, when weather prediction centers generate not a single prediction but a collection of predictions.

7.1.1 Stochastic processes

The formalism of stochastic processes is the most general framework to model quantities that evolve in time and that are subject to random perturbations. The goal of this subsection is to describe how stochastic processes can be used to model uncertainty, without digging too much into the details. In particular, we keep the mathematical definition minimal, and refer to textbooks for further details.

Particular attention has been put on *Markov processes*. These processes do not depend on the full history of their values, *i.e.* if one knows the history of values H_t up to time t for the

Markov process $\mathbf{x}(t)$, then the conditional probability density $p(\mathbf{x}(s) \mid H_t)$ of a future value at time $s > t$, only depends on the last value of the history $p(\mathbf{x}(s) \mid H_t) = p(\mathbf{x}(s) \mid \mathbf{x}(t))$. The previous property make Markov process convenient for calculus.

Forgetting the maths, from the operational perspective, one can be interested in modelling uncertainty through the addition of noise in the dynamics

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x}) + \boldsymbol{\varepsilon}(t) \tag{7.1}$$

with $\boldsymbol{\varepsilon}$ a 2D vector of white noise. It is as if the ground velocity is perturbed at all time instants by an error $\boldsymbol{\varepsilon}(t)$ that shows no special structure of auto-correlation in time. The problem is that the latter is not a well posed mathematical formulation since introducing a discontinuous function (the white noise) in an ODE is not rigorous. A rigorous way to describe an evolution in time influenced by noise can be done through the use of Stochastic Differential Equations (SDEs), in particular Itô's SDEs. The non-rigorous Equation 7.1 would be turned into the SDE

$$\mathrm{d}\mathbf{x} = \big(\mathbf{u}(t) + \mathbf{v}_f(t, \mathbf{x})\big)\,\mathrm{d}t + \boldsymbol{\Sigma}\,\mathrm{d}\mathbf{W} \tag{7.2}$$

with $\mathbf{W}$ a 2D Brownian motion and $\boldsymbol{\Sigma}$ a 2×2 coefficient matrix. This SDE models the infinitesimal behavior of the vehicle influenced by perturbations through the $\boldsymbol{\Sigma}\,\mathrm{d}\mathbf{W}$ term. To build trajectories out of this SDE, one uses *Itô's integral*, which is an extension of Riemannian integration to stochastic processes. The Itô integral of Eq. 7.2 is a stochastic process. It is even a Markov process. Realizations of this process are possible trajectories for the vehicle under the influence of random perturbations. For a thorough introduction to stochastic processes and Itô integrals, the reader is referred to Särkkä and Solin (2019).

Trying to minimize a cost, for instance the expected travel time, with a stochastic model of motion is the interest of *stochastic optimal control* (see *e.g.* Fleming and Rishel (1975)).

The independence of noise in time is a satisfying way to model positioning errors, unexpected deviations of the flow's true value from the forecast, or any perturbation in general.

In Parkinson et al. (2020), a model based on SDEs is used for human navigation in mountainous environments, and a corresponding HJB equation is formulated for the optimal navigation in the presence of uncertainty, with the cost being an expected deviation from target. The authors solve the HJB equation to find an optimal control law and then simulate trajectories using this law to see the variability of possible paths, and how the magnitude in the noise influences the trajectories. The formalism of viscosity solutions still applies to solve stochastic optimal control problems with an HJB PDE (Gozzi et al., 2000). The authors demonstrate the efficiency of a Hamilton-Jacobi approach in a stochastic context to solve for optimal trajectories with an averaged cost.

The discrete analog to diffusion processes is the MDP, where the environment is discretized into a finite collection of states for an agent to explore, and at all times the latter has a finite collection of actions to pick what action to do. Usually, it is not possible to compute explicitly the optimal policy (*i.e.* the optimal mapping from state to action) because it requires too much space or computation effort, and an approximate approach such as Reinforcement Learning (RL) shall be used. It is the approach used in Biferale et al. (2019) to solve Zermelo's navigation problem in very chaotic flow field environments. The goal is to build a control law (or policy, in the RL terminology) with optimal performances in the presence of perturbations (here the chaos in the flow field), and RL provides an approximate yet successful way to do so. But modern approaches based on MDPs can now solve for the optimal policy without resorting to approximate approaches. In Chowdhury et al. (2022) the AUV energy-time, stochastic routing problem is turned into a MDP and solved explicitly, making use of GPU computation to cut the computation time and allow for the exact computation of the policy function. An MDP was also used by Hentzen et al. (2018) for aircraft navigation in the presence of thunderstorms. The motion of the aircraft is modelled though an MDP and the thunderstorm evolution is described probabilistically.

7.1.2 Scenario-based uncertainty

While the framework of stochastic processes is well established, it is somewhat insufficient to capture the nature of uncertainty in long-range trajectory planning problems. A major uncertainty source is in the flow field prediction. Because systems such as the ocean or the atmosphere are chaotic, predicting their state becomes increasingly difficult when the time horizon for the prediction increases. But some weather conditions show more dispersion than others, and in some favorable cases, the system can be less chaotic than normal. To get a better grasp of how dispersive the atmosphere or ocean conditions are, *ensemble predictions* were developed (see Lewis (2005); Leutbecher and Palmer (2007)). The principle of ensemble forecasting is to extend the main prediction with a collection of other predictions, produced from perturbed initial conditions or parameters, or both. When looking at the values of these ensemble members at a particular date, one can then see if the members 'agree', which means their values are close, or 'disagree', if the value are dispersed. An important fact is that each member results from the resolution of physics equations, so each member is (in principle) physically consistent. This motivates an approach that deals with each weather scenario independently, instead of building local, point-wise statistics from the ensemble and applying the formalism of stochastic processes over it.

In González-Arribas et al. (2018), the authors optimize aircraft routes on metrics that are aggregated from weather ensemble predictions. Using collocation methods, they find paths minimizing a compound metric featuring travel time, fuel burn and difference between maximum travel time and minimum travel time for robustness. They thus demonstrate the usefulness of weather ensemble predictions for practical flight path planning. However, collocation methods cannot prove the optimum found are the global minimizers of the considered problem.

In Subramani and Lermusiaux (2019), the authors compute as many time-optimal trajectories as the number of flow field realizations $\mathbf{v}_f(t, \mathbf{x}, \omega_i)$, $i = 0, 1, \dots, N_{\text{ens}}$ (realizations can be thought as ensemble members). The important question is then how to merge the information from the optimal trajectory to decide what control law, what trajectory or what path to follow to get optimal performances in the uncertain flow field. In particular, one needs a way to assess the performance of optimum trajectory $\mathbf{x}^\star(t, \omega_i)$ or control $\mathbf{u}^\star(t, \omega_i)$ that was optimized in scenario $\mathbf{v}_f(t, \mathbf{x}, \omega_i)$ in an unexpected weather scenario $\mathbf{v}_f(t, \mathbf{x}, \omega_j)$. The authors provide three ways of doing so:

1. computing the Fréchet distance between pairs of optimal trajectories $\mathbf{x}^\star(t, \omega_i)$ and $\mathbf{x}^\star(t, \omega_j)$;

2. applying control law $\mathbf{u}^\star(t, \omega_i)$ in flow scenario $\mathbf{v}_f(t, \mathbf{x}, \omega_j)$ and noting the resulting distance from target at the end of the time window (time upper bound being the optimal travel time in the j-th scenario);

3. using the optimal trajectory $\mathbf{x}^\star(t, \omega_i)$ as a path to follow in the j-th scenario, and computing the resulting delay, with baseline the optimal travel time in the j-th scenario.

When choosing one of these cross-scenario performances, one can then assess which of the optimal strategy leads to the best statistics of performance among all scenarios.

In Wang et al. (2016), an ensemble of flow realizations is generated by perturbing an analytic model of gyre flow field. In each scenario, the authors compute the optimal travel time using extremals. They first shoot a collection of backward-integrated extremals and use the closest to origin to set a BVP solver. Once the optimal trajectories are computed, they use the same metric as Subramani and Lermusiaux's third error metric (the difference in travel time from following paths). They can then study the statistics of performance of the different optimal paths when crossing scenarios.

In what follows, we will use our time-optimal extremal-based algorithm on a real case of UAV path planning with an ensemble weather prediction. We will study the variability in the performances of the different optimal paths to see what can be expected from real world airborne navigation cases.

7.2 Case study: multi-scenario trajectory planning in real wind data

7.2.1 Definition and methods

We set up a navigation problem with real data over the Northern Atlantic. We define the origin at St John's, Canada and the destination at Dublin, Ireland. We extract an ECMWF ensemble prediction issued on 2023-08-25 06:00Z for the following 90 hours. The ECMWF ensemble is composed of 51 predictions: a reference prediction (the zeroth member) which is unperturbed and 50 perturbed members. For the ensemble at stake, we depict two particular members that show a high difference in Fig. 7.1. On this figure, we can see how uncertainty builds by looking at the difference of wind fields between ensemble member 16 and member 49. At the beginning of the time window, overall, the difference is weak. After 30 hours, a ridge appears in the difference field, with high magnitude, corresponding to the uncertainty in predicting a large stream of wind appearing in each individual member. After 63 hours, the difference field starts to have features with an order of magnitude comparable to the wind fields themselves. In this environment, the vehicle has an SRF of 22.8 m/s, which gives a no-wind travel time from origin to destination to 40.1 h.

We set the trajectory planning start date at 2023-08-26 12:00Z, as if the current date is 2023-08-25 06:00Z and the vehicle shall take off in 30 hours. Thus, the planning problem is immersed from the start in an environment that is uncertain. We first want to compute the time-optimal trajectory in each weather scenario. We then want to evaluate the travel time from following a given trajectory as waypoint orders in a different weather scenario.

Path-following trajectory computation Suppose we have a trajectory already in ground path form $\lambda \in [0, L] \mapsto \mathbf{x}_i(\lambda)$ that was computed in flow field $\mathbf{v}_{f,i}$ and that we want to follow in a different flow field $\mathbf{v}_{f,j}$. We first define a mapping $\mathbf{d}_i(\mathbf{x})$ that maps any position to the nearest direction of the path to follow, *i.e.* $\mathbf{d}_i(\mathbf{x}) := \left.\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}\lambda}\right|_{\lambda_{\text{proj}}}$ with $\lambda_{\text{proj}} := \arg\min_{\lambda \in [0,L]} \|\mathbf{x}(\lambda) - \mathbf{x}\|$. We

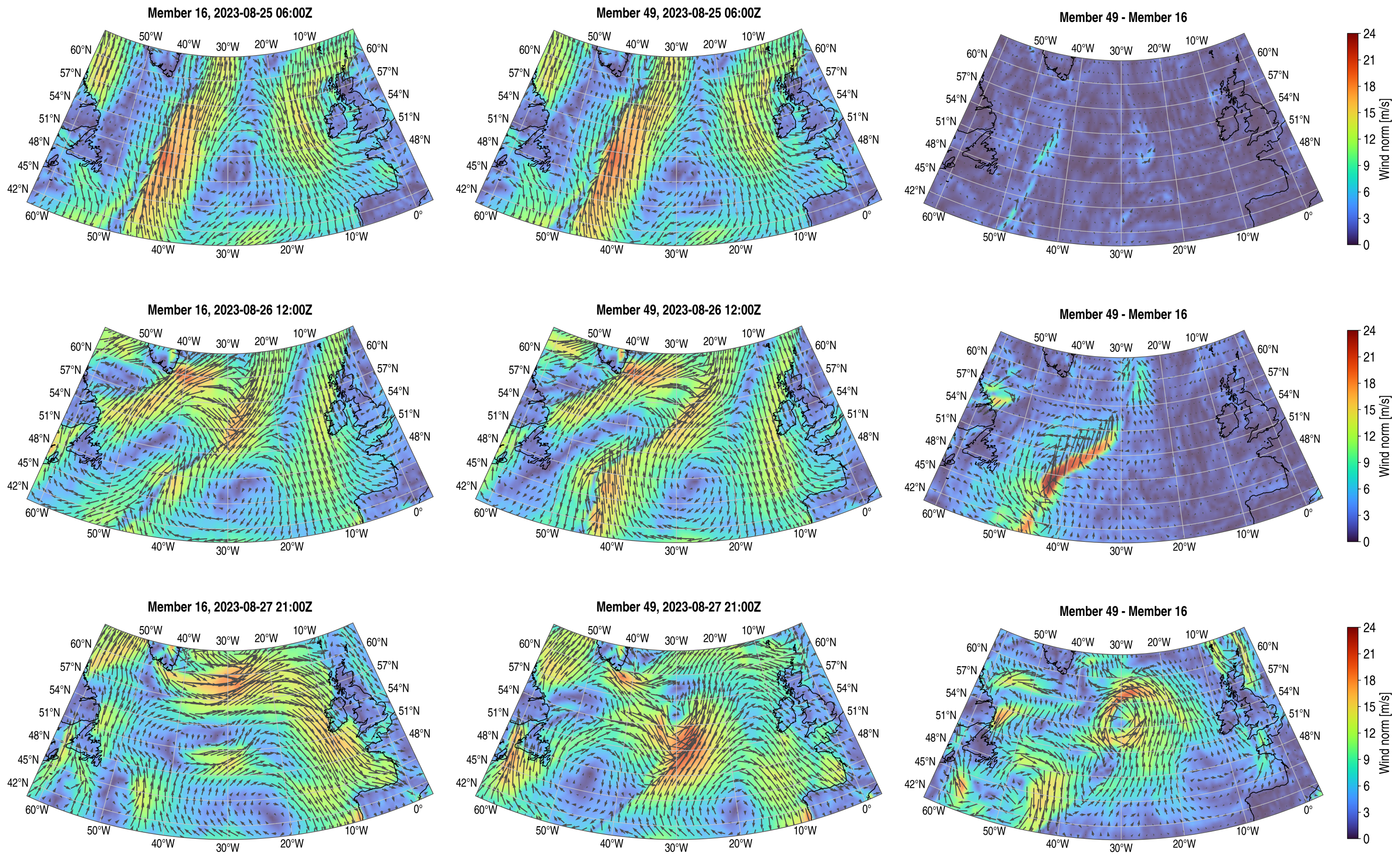

Member 16, 2023-08-25 06:00Z
Member 49, 2023-08-25 06:00Z
Member 49 - Member 16
Member 16, 2023-08-26 12:00Z
Member 49, 2023-08-26 12:00Z
Member 49 - Member 16
Member 16, 2023-08-27 21:00Z
Member 49, 2023-08-27 21:00Z
Member 49 - Member 16
60°N
57°N
54°N
51°N
48°N
45°N
42°N
60°W
50°W
40°W
30°W
20°W
10°W
0°
Wind norm [m/s]
24
21
18
15
12
9
6
3
0

Figure 7.1: ECMWF ensemble forecast issued at 2023-08-25 06:00Z. The data is the wind field at 1000hPa (just above sea level). The rows show data for different time stamps: the upper row is for 2023-08-25 06:00Z, the middle row for 2023-08-26 12:00Z (path planning start time) and the lower row is for 2023-08-27 21:00Z (path planning end time). The rightmost column shows the 16th ensemble member, the middle row the 49th and the leftmost row the difference between the 49th member and the 16th member.

then use the directional time-optimal control $\mathfrak{u}$ to define an ODE

$$\frac{\mathrm{d}\mathbf{y}_j}{\mathrm{d}t} = \mathfrak{u}(\mathbf{d}_i(\mathbf{y}_j(t)), \mathbf{v}_{f,j}(t, \mathbf{y}_j(t))) \tag{7.3}$$

If we set $\mathbf{y}_j(0) = \mathbf{x}_0$ the origin of the navigation problem, then if $\mathbf{x}_i(\lambda)$ is followable in flow field $\mathbf{v}_{f,j}$, the trajectory resulting from the integration of Eq. 7.3 follows exactly the ground path $\mathbf{x}_i(\lambda)$ and arrives at destination with some arrival time (can be greater or lower than the travel time of $\mathbf{x}_i(\lambda)$ since the flow field changes). The reason why a global mapping for the direction of the path was required is that in the integration process, numerical errors always entail some small difference in position between the path and the trajectory that is supposed to follow it. The direction vector mapping is thus defined everywhere in space so that it can be evaluated at any point. But of course, this mapping only makes sense in the vicinity of the considered path.

To implement $\mathbf{d}_i$ in practice, we discretize the reference ground path in $(\mathbf{x}_i^{(k)})_{k\in\{0,1,\dots,N-1\}}$ with $\mathbf{x}_i^{(k)} := \mathbf{x}_i(\frac{k}{N-1}L)$, and $k_{\text{proj}} := \arg\min_{k\in\{0,1,\dots,N-2\}} \left\|\mathbf{x} - \mathbf{x}_i^{(k)}\right\|$, and take for the direction:

$$\mathbf{d}_i(\mathbf{x}) = \frac{\mathbf{x}_i^{(k_{\text{proj}}+1)} - \mathbf{x}_i^{(k_{\text{proj}})}}{\left\|\mathbf{x}_i^{(k_{\text{proj}}+1)} - \mathbf{x}_i^{(k_{\text{proj}})}\right\|} \tag{7.4}$$

7.2.2 Results

We run the time-optimal in-depth interpolated sampling algorithm from Chapter 3 in each individual weather scenario to get the time-optimal trajectory in each case. The resulting trajectories are depicted in Fig. 7.2. We then evaluate the cross scenario performances. We plot the per-ensemble member statistics in Fig. 7.3 and the per-path statistics in Fig. 7.4.

The overall mean travel duration across all scenarios and all paths is of 29.9 hours, 10.2 h less than the no-wind travel time, so that the wind is beneficial to the travel. The associated standard deviation is of 0.783 h which is approximately 47 min. In Fig. 7.3, we see that the flow field realization highly influences the travel duration. The best performance is achieved in ensemble member 10. Overall, the dispersion around the mean is of the same order of magnitude

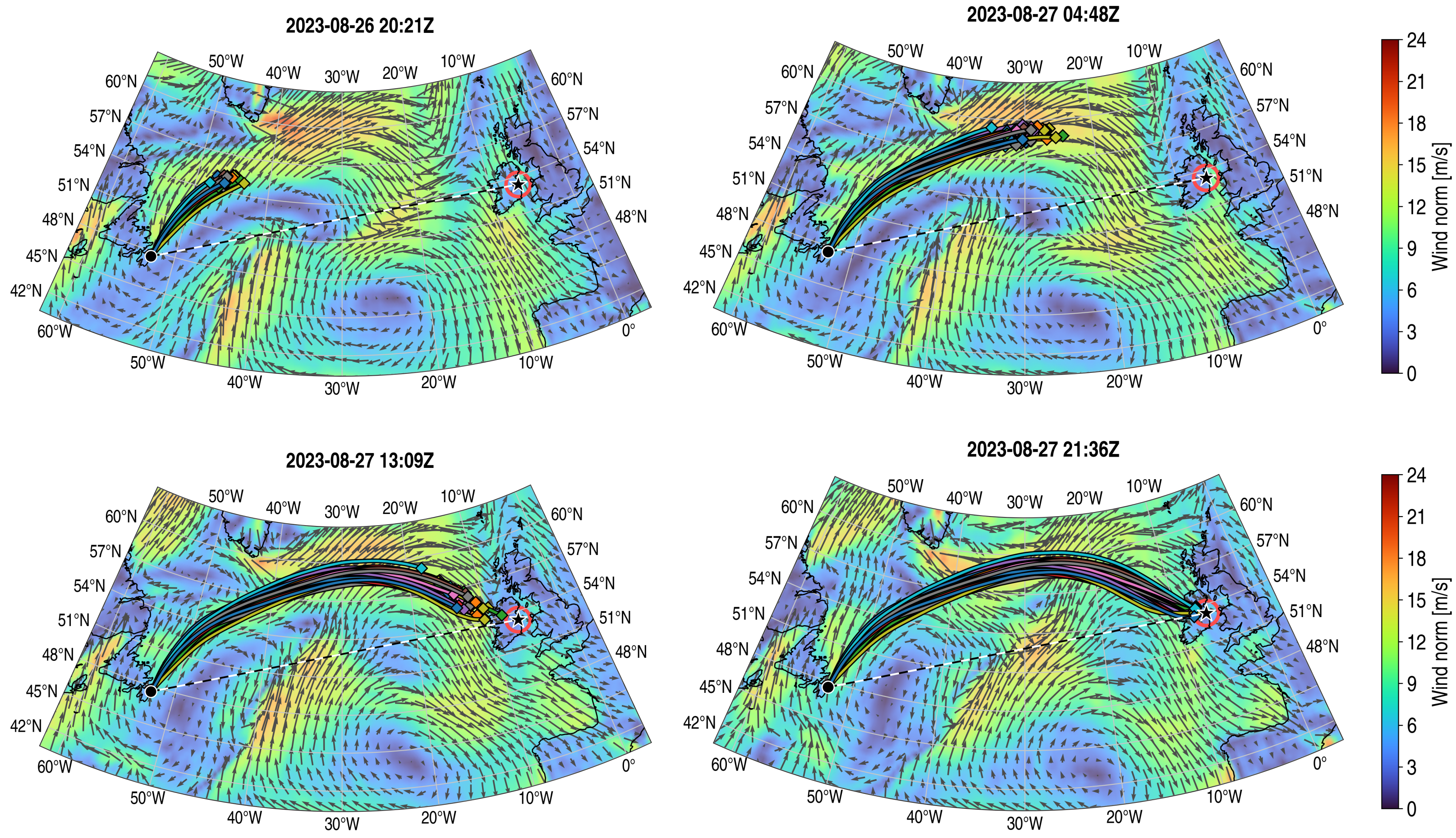


Figure 7.2: Time-optimal trajectories computed for each weather ensemble member. The underlying flow field is the zeroth member *i.e.* the reference prediction. The path arriving first (green) is obtained in the 10th ensemble member, while the path arriving last (sky blue) is obtained in the 49th ensemble member

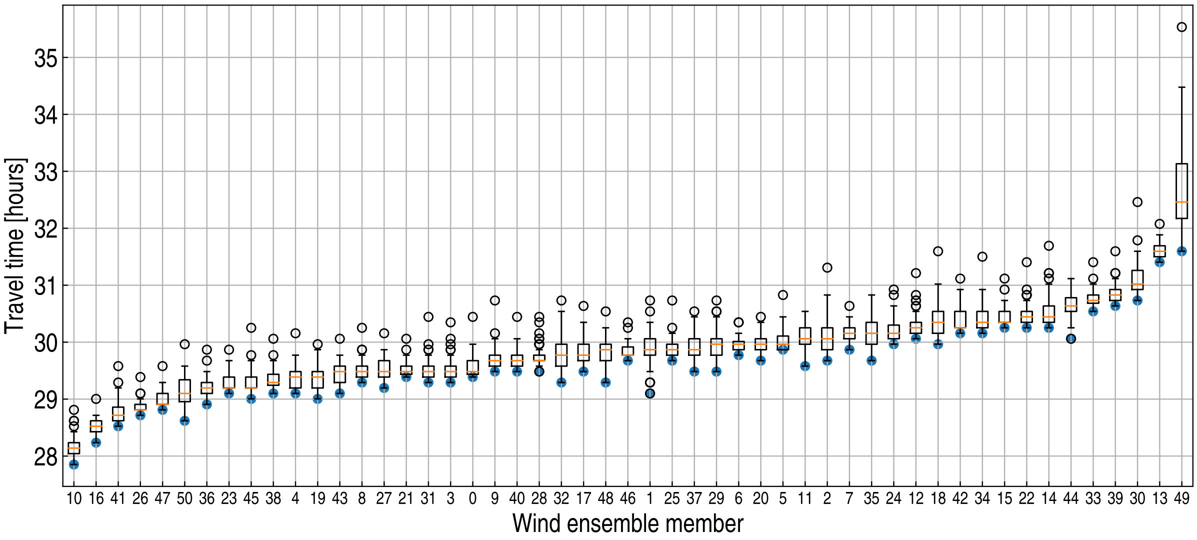


Figure 7.3: Travel time in each wind ensemble member for the North Atlantic case. Members are sorted by increasing mean travel time across all paths. The blue dots are the optimal travel times in each scenario.

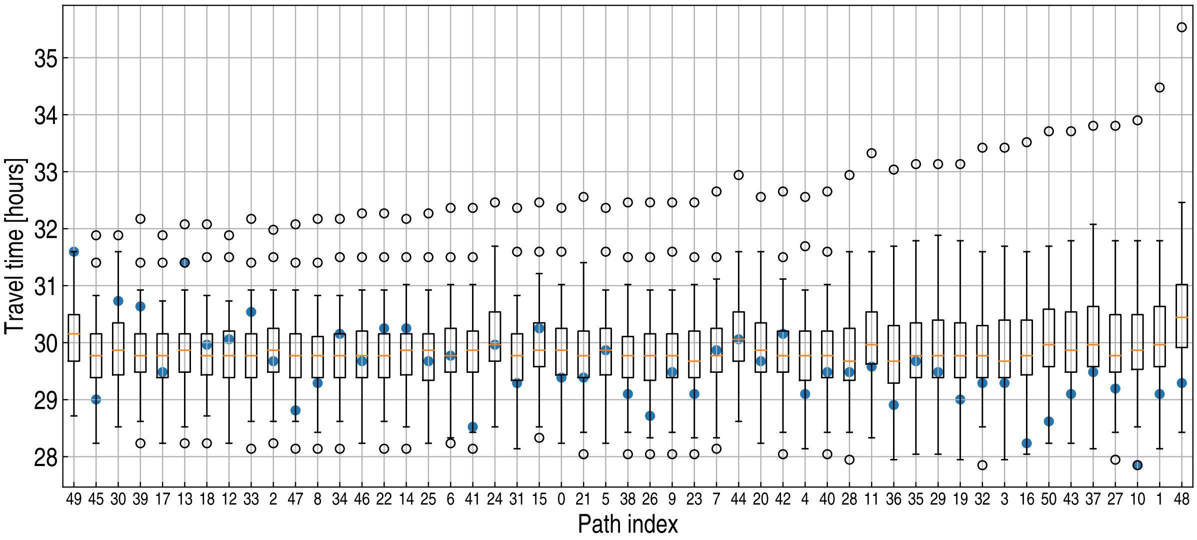


Figure 7.4: Travel time for each path in ensemble prediction for the North Atlantic case. Paths are sorted by increasing standard deviation of travel time in the different wind scenarios. The path minimal travel duration in its nominal scenario is depicted as a blue dot.

for all scenarios, with the exception of the 49th one, which shows a higher spread. It is also the worst scenario for travel time. Snapshots of this 49th ensemble member are given in Fig. 7.1.

In Fig. 7.4, the path-wise performances show almost similar performances across the paths, with a sensibly constant mean performance and varying dispersion. Path 49 shows the least standard deviation from its mean performance, with a worst-case travel time of 31.6 h. In comparison, path 48 can see its travel time going up to 35.5 h in an unfavorable case.

Reverse travel

To get other results with this approach, the origin and destination are reversed, keeping all other parameters identical. For simplicity, only the first ten weather members are used: the control member at index 0 and nine others. In the reverse direction, the wind is generally facing the vehicle, increasing the travel time and making the trajectory problem more complex. The same methodology as previously is applied. The results are depicted in Fig. 7.5. The travel times are displayed in Fig. 7.6.

Trajectory number 3 has the best nominal travel time (*i.e.* travel time in the scenario for which it was optimized) with 42.9 hours of travel, and trajectory number 4 has the worst one with 48.0 hours of travel. As in the previous case, some paths show more variance in travel time than others. But what catches attention is that path number 1 has a significantly different shape than all the others, as can be seen in Fig. 7.5. However, this path has a nominal travel time comparable to the others, with 44.7 hours. One can think that this trajectory is an outlier, in the sense that it is one of the possible optimums of the navigation problem in flow field number 1, so that another optimal trajectory exists with a similar travel time but a shape similar to all the other paths. But looking at Figure 7.3, there is at least 2 hours of difference with any other path in the flow field scenario number 1, excluding the preceding possibility.

Besides, path number 1 has the highest standard deviation among all others, as can be seen in Figure 7.4 (box plots are sorted according to standard deviation). This illustrates the primary importance of considering multiple scenarios in the presence of uncertainty. Indeed,

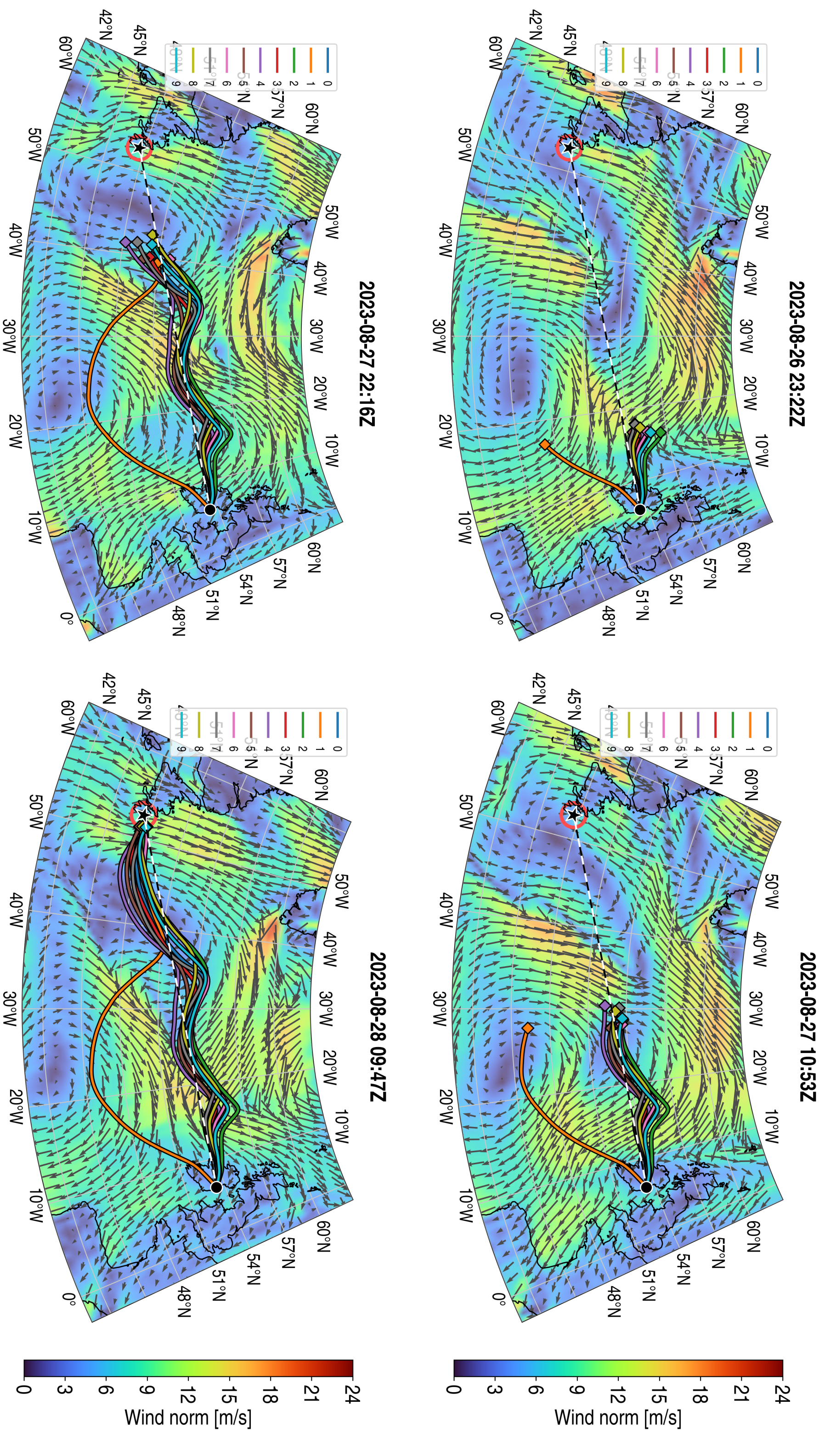
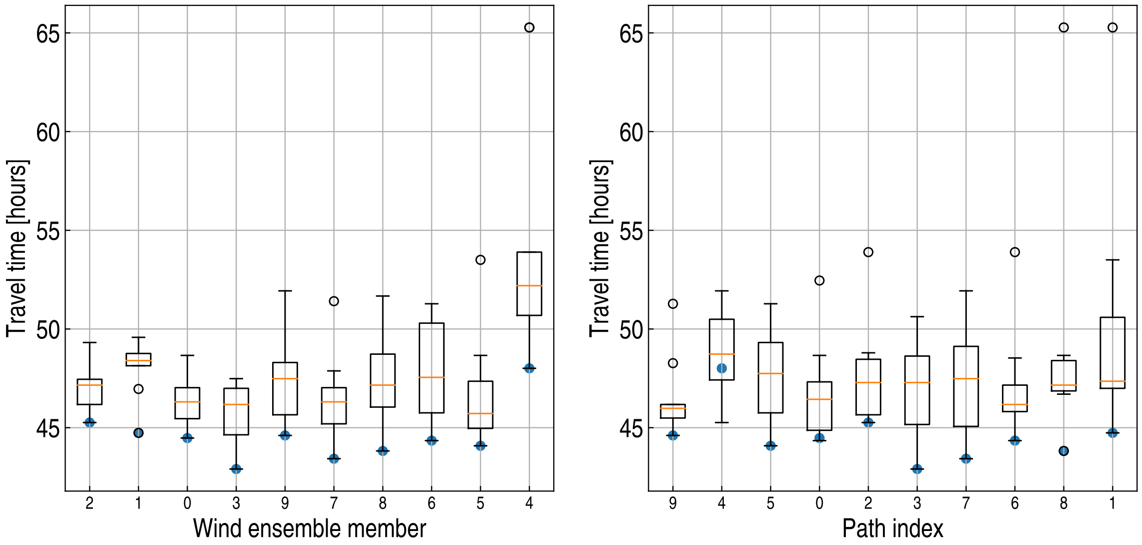


Figure 7.5: Time-optimal trajectories computed for each weather ensemble member in the reverse North Atlantic case. The underlying flow field is the zeroth member *i.e.* the reference prediction.



(a) (b)

Figure 7.6: Reverse travel in North Atlantic ensemble wind prediction. (a) Travel times in each wind ensemble member. Members are sorted by increasing mean travel time across all paths. (b) Travel times for each path in the different wind scenarios. Paths are sorted by standard deviation. For both graphs, the blue dots are the optimal travel times in the nominal scenarios.

if flow field number 1 would have been the control run (the zeroth member), and an operator was to optimize its trajectory in the standard prediction without consideration for perturbed scenarios, the latter would have computed a very uncertain path, in the sense that its variance in travel time would have been very large with a travel time spanning from around 45 hours in the best scenario to 65 hours in the worst, while there exists other paths with reduced variance.

Conclusions of the case study This case study shows that for UAVs such as the Mermoz drone, the uncertainty in the weather prediction can change the travel time by a non-negligible amount, indeed hours on a travel time of around 30 hours. No path was shown to outperform the others as regards the expected travel time, however some show reduced variance compared to the others. Overall, the ground paths show little deviation from one another.

7.3 Path-based average-time-optimal trajectory planning

In the previous section the main performance metric was the variation in travel time over a given path when the weather scenario changes. We built the performances of the different paths based on this criteria and could then select the best path for the problem. The selected path has the best performances, but only among the explored trajectories. It may happen that there exists a path with better performances and which was not considered by the previous approach. This is because we never optimize on the statistical performance, but on the deterministic performance in each scenario.

In this section, we propose an approach that optimizes directly on the aggregate performance across scenarios, indeed the average travel time. To do so, we choose to change the point of view from trajectories to ground paths (see Chapter 2, Sec. 2.4 for an introduction to ground paths). In Davis et al. (2009), the authors also derive a framework for trajectory optimization based on paths, with similar equations as the ones presented in what follows.

We note $\mathbf{x}(\lambda)$ a ground path in which we choose to parametrize through the chord length λ. In what follows, we note ϕ the angle of the ground path in the ground frame. It must not be confused with the heading of the vehicle, which could be different because of the presence of a flow field. The evolution over the path is:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\lambda} = \begin{pmatrix} \cos\phi(\lambda) \\ \sin\phi(\lambda) \end{pmatrix} \tag{7.5}$$

where we stress that ϕ is a function of λ. The derivative vector is of norm 1 since the path is parametrized in chord length. There is no notion of velocity in the environment for now, as we only exposed the differential equation satisfied by the path $\mathbf{x}(\lambda) = (x(\lambda)\ \ y(\lambda))^\intercal$. To introduce the time evolution, we define the *slowness* of the vehicle along the path. At position $\mathbf{x}$, imposing a ground path angle of ϕ to the vehicle makes it find the appropriate heading to match the direction of motion ($\mathbf{d} := (\cos\phi\ \ \sin\phi)^\intercal$, the vehicle chooses $\mathbf{u} = \mathrm{u}(\mathbf{d}, \mathbf{v}_f)$, the directional time-optimal control defined in Chapter 2), and the resulting ground speed v_g along the path is:

$$v_g = v_f^{\|} + \sqrt{v_r^2 - \left(v_f^{\perp}\right)^2}$$

with $v_f^{\|} = \mathbf{v}_f(t,\mathbf{x}) \cdot \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix}$ and $v_f^{\perp} = \mathbf{v}_f(t,\mathbf{x}) \cdot \begin{pmatrix} -\sin\phi \\ \cos\phi \end{pmatrix}$ and v_r the SRF of the vehicle. We thus define the slowness as the inverse of the ground speed:

$$s(t,\mathbf{x},\phi) = \frac{1}{v_f^{\|} + \sqrt{v_r^2 - \left(v_f^{\perp}\right)^2}} \tag{7.6}$$

which is a function of time t, position $\mathbf{x}$ and direction of ground path ϕ. In the strong flow case, some directions of motion are not possible for the vehicle, in which case we set $s(t,\mathbf{x},\phi) = +\infty$.

The time variable is now also parametrized in chord length: $t(\lambda_1)$ is the time it takes for the vehicle to follow the path from $\lambda = 0$ to $\lambda = \lambda_1$. The derivative of the time variable with chord length is precisely the slowness of the vehicle:

$$\frac{\mathrm{d}t}{\mathrm{d}\lambda} = s(t(\lambda),\mathbf{x}(\lambda),\phi(\lambda)) \tag{7.7}$$

So, with the current formulation, the *Path-based Time-optimal Navigation Problem* (P-TNP) writes:

$$\text{(P-TNP)} \begin{cases} \min\limits_{\phi(\cdot)} \; t(L) \\ \text{Eqs. } 7.5, 7.7 \\ \mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x}(L) = \mathbf{x}_f \end{cases} \tag{7.8}$$

with control variable being the law of angle for the ground path $\phi(\cdot)$. This time, the cost appears as a terminal cost and not an integral one. In this problem, we did not introduce the variability of the flow field yet. We first derive the optimality conditions in the deterministic case, before going through the uncertain case.

7.3.1 Deterministic path-based extremals

We once again use the PMP to derive necessary conditions for optimality for the path-based extremals. We introduce an adjoint state $\mathbf{p} = \begin{pmatrix} p_x & p_y & p_t \end{pmatrix}^\intercal$ with as many components as state variables for the problem, *i.e.* three: two for the physical state and one for the time variable. Its evolution in time is given by:

$$\begin{aligned} \frac{\mathrm{d}p_x}{\mathrm{d}\lambda} &= -\frac{\partial s}{\partial x}\, p_t \\ \frac{\mathrm{d}p_y}{\mathrm{d}\lambda} &= -\frac{\partial s}{\partial y}\, p_t \\ \frac{\mathrm{d}p_t}{\mathrm{d}\lambda} &= -\frac{\partial s}{\partial t}\, p_t \end{aligned} \tag{7.9}$$

The Hamiltonian for this system is:

$$H = p_x \cos\phi + p_y \sin\phi + s(t, \mathbf{x}, \phi)$$

The latter expression is already normalized and excludes abnormal extremals, as, once again, there are not fundamental for a first approach based on extremals. The Hamiltonian always has a minimum in ϕ because:

1. in the controllable case ($\left\|\mathbf{v}_f\right\| < v_r$) it is continuous in ϕ and ϕ takes values in the compact set $[0, 2\pi]$;

2. in the non-controllable case, the slowness takes finite values over an interval $]\phi_{\min}, \phi_{\max}[$, with $\lim_{\phi\to\phi_{\min}} = +\infty$ and $\lim_{\phi\to\phi_{\max}} = +\infty$, thus there is a minimum for the Hamiltonian over $]\phi_{\min}, \phi_{\max}[$.

We first note that the Hamiltonian is autonomous, in the sense that it has no direct dependence on the λ parameter, *i.e.* $\frac{\partial H}{\partial \lambda} = 0$. Moreover, the final chord length L is not fixed so there is a terminal condition on the Hamiltonian $H(t(L), \mathbf{x}(L), \phi(L)) = 0$. Because the Hamiltonian is autonomous, the transversality indeed holds for all $\lambda \in [0, L]$ and we have:

$$\forall \lambda \in [0, L], \quad p_x(\lambda)\cos\left(\phi(\lambda)\right) + p_y(\lambda)\sin(\phi(\lambda)) + s(t(\lambda), \mathbf{x}(\lambda), \phi(\lambda)) = 0 \tag{7.10}$$

The Hamiltonian is smooth in ϕ so we have the NCO $\frac{\partial H}{\partial \phi} = 0$ for the the minimizer $\phi^\star$, which writes:

$$-p_x \sin\phi^\star + p_y \cos\phi^\star + p_t \left.\frac{\partial s}{\partial \phi}\right|_{t,\mathbf{x},\phi^\star} = 0$$

The previous equation holds for all $\lambda \in [0, L]$, thus we can differentiate w.r.t. λ to get:

$$\begin{aligned}-\frac{\mathrm{d}p_x}{\mathrm{d}\lambda}\sin\phi^\star - p_x\cos(\phi^\star)\frac{\mathrm{d}\phi^\star}{\mathrm{d}\lambda} + \frac{\mathrm{d}p_y}{\mathrm{d}\lambda}\cos\phi^\star - p_y\sin(\phi^\star)\frac{\mathrm{d}\phi^\star}{\mathrm{d}\lambda} + &\\ -\frac{\partial s}{\partial t}\frac{\partial s}{\partial \phi}\,p_t + &\\ \left[s\,\frac{\partial^2 s}{\partial\phi\partial t} + \frac{\partial^2 s}{\partial\phi\partial x}\,\cos\phi^\star + \frac{\partial^2 s}{\partial\phi\partial y}\,\sin\phi^\star + \frac{\partial^2 s}{\partial\phi^2}\,\frac{\mathrm{d}\phi^\star}{\mathrm{d}\lambda}\right] p_t = 0&\end{aligned}$$

Now, using Eqs. 7.9, 7.10 simplifies the previous equation to:

$$\left[s\frac{\mathrm{d}\phi^\star}{\mathrm{d}\lambda} + s\,\frac{\partial^2 s}{\partial\phi\partial t} + \left(\frac{\partial^2 s}{\partial\phi\partial x} - \frac{\partial s}{\partial y}\right)\cos\phi^\star + \left(\frac{\partial^2 s}{\partial\phi\partial y} + \frac{\partial s}{\partial x}\right)\sin\phi^\star + \frac{\partial^2 s}{\partial\phi^2}\,\frac{\mathrm{d}\phi^\star}{\mathrm{d}\lambda} - \frac{\partial s}{\partial t}\,\frac{\partial s}{\partial \phi}\right] p_t = 0$$

Since p_t cannot cancel out (non null initial condition and given its ODE), we have an equation for $\frac{\mathrm{d}\phi^\star}{\mathrm{d}\lambda}$:

$$\boxed{\frac{\mathrm{d}\phi^\star}{\mathrm{d}\lambda} = -\frac{s\,\frac{\partial^2 s}{\partial\phi\partial t} + \left(\frac{\partial^2 s}{\partial\phi\partial x} - \frac{\partial s}{\partial y}\right)\cos\phi^\star + \left(\frac{\partial^2 s}{\partial\phi\partial y} + \frac{\partial s}{\partial x}\right)\sin\phi^\star - \frac{\partial s}{\partial t}\,\frac{\partial s}{\partial \phi}}{s + \frac{\partial^2 s}{\partial\phi^2}}} \tag{7.11}$$

With this ODE, one can shoot extremal paths as done in Chapter 3 to look for the optimal path to target. An important fact is that this equation is independent from the costate values, which is reminiscent of the invariance to costate positive scaling for the PMP applied to the TNP (Property 3.2).

7.3.2 Average-time-optimal path-based extremals

If the flow field is uncertain and described by an ensemble of possible realizations $(\mathbf{v}_{f,i})_{i\in\{0,1,\dots,N_{\text{ens}}-1\}}$, the advantage of considering ground paths is that paths will be the same in each scenario, only the travel time will vary. We note $t_i(\lambda)$ the variable accounting for the travel time along the path $\mathbf{x}(\cdot)$ up to chord length λ. We note $s_i(t_i, \mathbf{x}, \phi)$ the slowness of moving in direction ϕ when at position $\mathbf{x}$ and time t_i in flow field scenario i. At this point we also note the caveat that in some cases, a path may be unfollowable in a strong flow scenario, in which case $t_i = +\infty$ from some λ.

The system evolution is now:

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}\lambda} &= \cos\phi(\lambda) \\ \frac{\mathrm{d}y}{\mathrm{d}\lambda} &= \sin\phi(\lambda) \\ \frac{\mathrm{d}t_i}{\mathrm{d}\lambda} &= s_i(t_i(\lambda), \mathbf{x}(\lambda), \phi(\lambda)) \qquad i = 0, 1, \ldots, N_{\text{ens}} - 1 \end{aligned} \tag{7.12}$$

In this uncertain environment, we look for the path that will have the best expected travel time. We thus want to solve the following Average Time-optimal Navigation Problem (ATNP):

$$(\text{ATNP}) := \begin{cases} \min\limits_{\phi(\cdot)} \sum\limits_{i\in\{0,1,\ldots,N_{\text{ens}}-1\}} a_i\, t_i(L) \\ \qquad \text{Eqs. 7.12} \\ \mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x}(L) = \mathbf{x}_f \end{cases} \tag{7.13}$$

where $(a_i)_{i\in\{0,1,\ldots,N_{\text{ens}}-1\}}$ is an arbitrary weighting of scenarios, *i.e.* $\forall i,\ \ a_i > 0$, $\sum a_i = 1$. The regular average travel time is obtained for the weighting $a_i = \frac{1}{N_{\text{ens}}}$, but we want to allow general weightings of scenarios with this formulation. For instance, the zeroth member plays a particular role and its importance can be amplified compared to the perturbed members.

Resolution We once again introduce an adjoint state $\mathbf{p} = (p_x\ p_y\ p_0\ \ldots\ p_{N_{\text{ens}}-1})^\intercal$ with $N_{\text{ens}} + 2$ components corresponding to the state variables: 2 physical and N_{ens} time variables. Its evolution in time is given by:

$$\begin{aligned} \frac{\mathrm{d}p_x}{\mathrm{d}\lambda} &= -\sum_{i=0}^{N_{\text{ens}}-1} \frac{\partial s_i}{\partial x}\, p_i \\ \frac{\mathrm{d}p_y}{\mathrm{d}\lambda} &= -\sum_{i=0}^{N_{\text{ens}}-1} \frac{\partial s_i}{\partial y}\, p_i \\ \frac{\mathrm{d}p_i}{\mathrm{d}\lambda} &= -\frac{\partial s_i}{\partial t_i}\, p_i \qquad\qquad i = 0, 1, \dots, N_{\text{ens}} - 1 \end{aligned} \tag{7.14}$$

The Hamiltonian writes:

$$H = p_x \cos\phi + p_y \sin\phi + \sum_{i=0}^{N_{\text{ens}}-1} p_i\, s_i(t_i, \mathbf{x}, \phi)$$

For the same reasons than the previous case, this Hamiltonian admits a minimum for some $\phi^\star$ either somewhere in $[0, 2\pi[$ if the vehicle is controllable in each flow field or somewhere on an interval $]\phi_{\min}, \phi_{\max}[$ which is the biggest possible for all the s_i to take finite values. So once again, we differentiate the Hamiltonian w.r.t. ϕ and use that $\frac{\partial H}{\partial \phi}(\phi^\star) = 0$ to write:

$$-\,p_x \sin\phi^\star + p_y \cos\phi^\star + \sum_{i=0}^{N_{\text{ens}}-1} p_i \left.\frac{\partial s_i}{\partial \phi}\right|_{t_i, \mathbf{x}, \phi^\star} = 0 \tag{7.15}$$

The Hamiltonian is once again autonomous (does not depend explicitly on λ), and satisfies a transversality condition on the final chord length L, so similarly to the previous section, we have:

$$\forall \lambda \in [0, L], \quad p_x(\lambda) \cos\left(\phi(\lambda)\right) + p_y(\lambda) \sin(\phi(\lambda)) + \sum_{i=0}^{N_{\text{ens}}-1} p_i(\lambda)\, s_i(t_i(\lambda), \mathbf{x}(\lambda), \phi(\lambda)) = 0 \tag{7.16}$$

We now differentiate Eq. 7.15 w.r.t. λ and get:

$$-\frac{\mathrm{d}p_x}{\mathrm{d}\lambda}\sin\phi^\star - p_x\frac{\mathrm{d}\phi^\star}{\mathrm{d}\lambda}\cos\phi^\star + \frac{\mathrm{d}p_y}{\mathrm{d}\lambda}\cos\phi^\star - p_y\frac{\mathrm{d}\phi^\star}{\mathrm{d}\lambda}\sin\phi^\star +$$
$$-\sum_i \frac{\partial s_i}{\partial t}\frac{\partial s_i}{\partial \phi}p_i +$$
$$\sum_i \left[\frac{\partial^2 s_i}{\partial\phi\partial t}s_i + \frac{\partial^2 s_i}{\partial\phi\partial x}\cos\phi^\star + \frac{\partial^2 s_i}{\partial\phi\partial y}\sin\phi^\star + \frac{\partial^2 s_i}{\partial\phi^2}\frac{\mathrm{d}\phi^\star}{\mathrm{d}\lambda}\right]p_i = 0$$

which, because of Eqs. 7.14, 7.16, the previous equation ends up writing:

$$\sum_i p_i \left[\left(s_i + \frac{\partial^2 s_i}{\partial\phi^2}\right)\frac{\mathrm{d}\phi^\star}{\mathrm{d}\lambda} + \frac{\partial^2 s_i}{\partial\phi\partial t}s_i + \left(\frac{\partial^2 s_i}{\partial\phi\partial x} - \frac{\partial s_i}{\partial y}\right)\cos\phi^\star + \left(\frac{\partial^2 s_i}{\partial\phi\partial y} + \frac{\partial s_i}{\partial x}\right)\sin\phi^\star - \frac{\partial s_i}{\partial t}\frac{\partial s_i}{\partial\phi}\right] = 0$$

which finally writes:

$$\boxed{\frac{\mathrm{d}\phi^\star}{\mathrm{d}\lambda} = -\frac{\sum_i p_i\left[s_i\frac{\partial^2 s_i}{\partial\phi\partial t} + \left(\frac{\partial^2 s_i}{\partial\phi\partial x} - \frac{\partial s_i}{\partial y}\right)\cos\phi^\star + \left(\frac{\partial^2 s_i}{\partial\phi\partial y} + \frac{\partial s_i}{\partial x}\right)\sin\phi^\star - \frac{\partial s_i}{\partial t}\frac{\partial s_i}{\partial\phi}\right]}{\sum_i p_i\left(s_i + \frac{\partial^2 s_i}{\partial\phi^2}\right)}} \tag{7.17}$$

In this equation, the values of the costate do not cancel out anymore as in the previous case. Thus, the transversality condition on the costate at final time cannot be discarded anymore. In this case, we have a terminal cost function:

$$\psi(x, y, t_0, t_1, \dots, t_{N_{\text{ens}}-1}) = \sum_{i\in\{0,1,\dots,N_{\text{ens}}-1\}} a_i\, t_i$$

The target set in the $\boldsymbol{\xi} := (x, y, t_0, t_1, \dots, t_{N_{\text{ens}}-1})$ space is $\{x_f\}\times\{y_f\}\times\underbrace{\mathbb{R}\times\cdots\times\mathbb{R}}_{N_{\text{ens}}}$. The tangent space to this target set is thus $\mathcal{T} = \{0\}\times\{0\}\times\underbrace{\mathbb{R}\times\cdots\times\mathbb{R}}_{N_{\text{ens}}}$. The transversality condition for the costate writes[1]:

$$\mathbf{p}(L) - \left.\frac{\partial\psi}{\partial\boldsymbol{\xi}}\right|_{x(L),y(L),t_0(L),\dots,t_{N_{\text{ens}}}(L)} \perp \mathcal{T}$$

[1] For general extremals, there is a factor in front of the $\frac{\partial\psi}{\partial\xi}$ term. But we neglected abnormal extremals in this approach, so this coefficient can be taken to 1.

with $\frac{\partial \psi}{\partial \xi} = (0\ 0\ a_0\ \dots\ a_{N_{\text{ens}}-1})^\intercal$. Thus, there is no condition on p_x and p_y at final chord length, but there are on the $(p_i)_{i\in\{0,1,\dots,N_{\text{ens}}-1\}}$, which write:

$$\forall i \in \{0, 1, \dots, N_{\text{ens}} - 1\}, \quad p_i(L) = a_i \tag{7.18}$$

To get average-time-optimal trajectories, one has thus to solve a boundary value problem by integrating ODEs on ϕ and $(p_i)_{i\in\{0,1,\dots,N_{\text{ens}}-1\}}$ (Eqs. 7.17, 7.14) and finding the appropriate initial conditions such that, for some L, $\mathbf{x}(L) = \mathbf{x}_f$ and $\forall i \in \{0, 1, \dots, N_{\text{ens}} - 1\}$, $p_i(L) = a_i$.

The previous development led to a differential characterization of the paths which are candidate to average time optimality. By shooting such paths, it is expected that minimum average travel time trajectories can be found in a given problem. The following section will demonstrate this on a real world example.

In this section, the focus was put on the minimization of average travel time, because the linearity of the average operation made it easy to integrate this new cost in optimization problems. Average travel time served as a first interesting metric to study the behavior of the newly formulated path-based extremals. In the presence of uncertainty, however, the only priority is not necessarily to minimize the expected travel time, and variance minimization or robustness to worst scenarios can be other criteria for which to optimize trajectories.

7.4 Case study: average-time-optimal path on an analytical example

7.4.1 Problem statement

We want to demonstrate that the extremals resulting from the previous resolution can be used to find average-time-optimal trajectories in flow fields. We set up an analytical case with a moving Rankine vortex:

$$\mathbf{v}_{f,i}(t,\mathbf{x}) = A \times R\left(\frac{\pi}{2}\right) \frac{\mathbf{x}-\mathbf{c}_i(t)}{\|\mathbf{x}-\mathbf{c}_i(t)\|} \times \begin{cases} \frac{\|\mathbf{x}-\mathbf{c}_i(t)\|}{r_0} & \|\mathbf{x}-\mathbf{c}_i(t)\| \leq r_0 \\ \frac{r_0}{\|\mathbf{x}-\mathbf{c}_i(t)\|} & \text{else} \end{cases} \tag{7.19}$$

where A is an amplitude parameter and r_0 a radius parameter. $\mathbf{c}_i(t)$ is the center of the vortex that is defined as:

$$\mathbf{c}_i(t) = \mathbf{c}_0 + \mathbf{d}_i\, t$$

The direction of motion for the vortex $\mathbf{d}_i = (\cos\alpha_i \;\; \sin\alpha_i)^\intercal$ is the parameter that changes between the different 'weather' scenarios. It can model for instance a prediction of a storm for which the direction of motion is uncertain.

For the numerical application, the values are $A = 0.9$, $r_0 = 0.2$, $\mathbf{c}_0 = (0.5 \;\; -0.2)^\intercal$. The angle takes three different values $\alpha_0 = 1.1$, $\alpha_1 = 1.6$ and $\alpha_2 = 1.9$ (in radians). These values are chosen in a non-uniform way to break symmetries, in particular so that the second value is not the average of the two others.

7.4.2 Methodology

We compute the time-optimal trajectories for each flow field scenario using the in-depth interpolated sampling algorithm. We discretize extremals from the algorithm on a uniform sampling of the chord length dimension on $[0, 1.4]$ with 500 points. The tolerance radius around the target in the physical space is of 0.05. We also build the mean flow field:

$$\overline{\mathbf{v}_f} = \frac{1}{3}\left(\mathbf{v}_{f,0} + \mathbf{v}_{f,1} + \mathbf{v}_{f,2}\right) \tag{7.20}$$

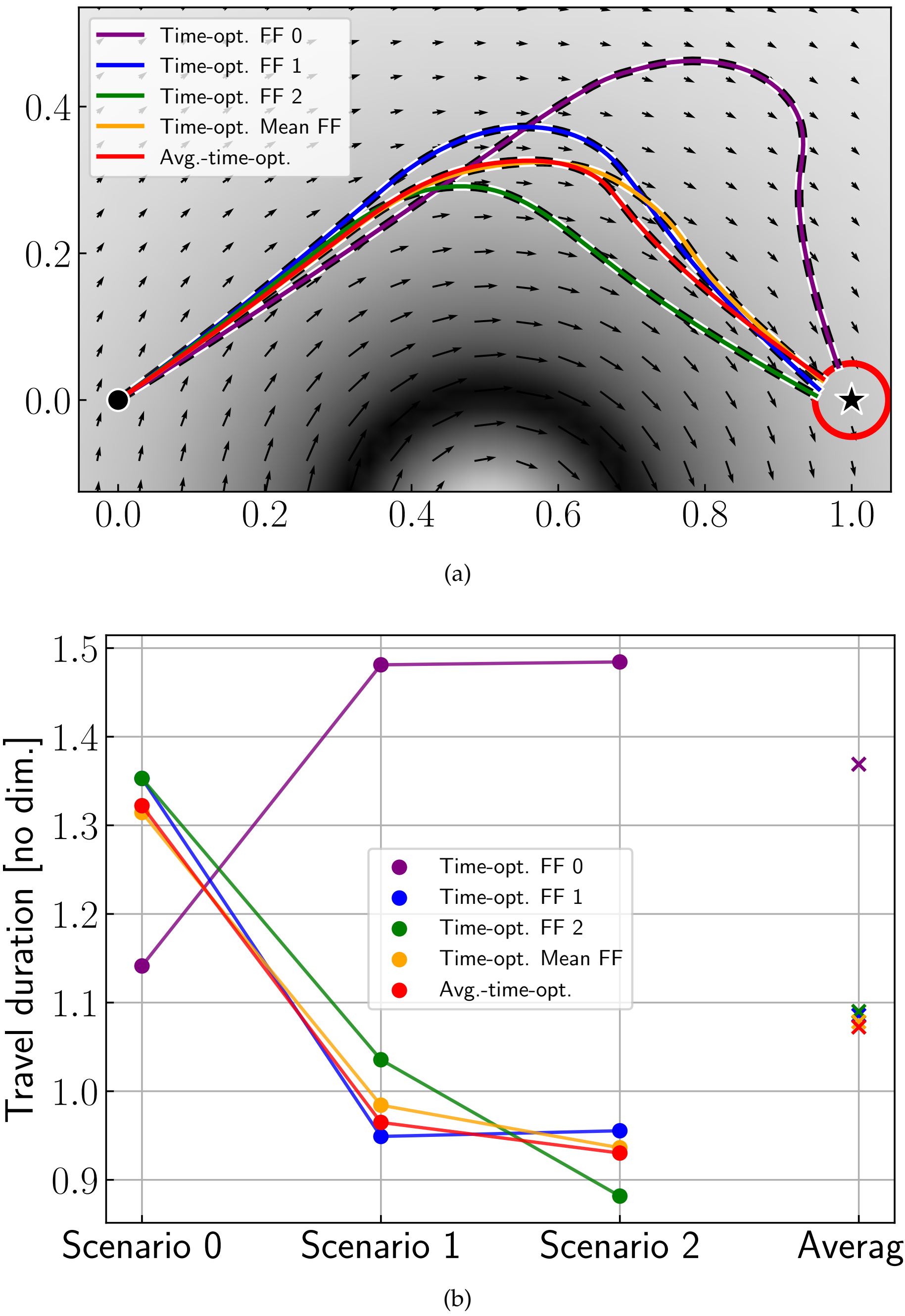
and compute the time-optimal trajectory in this flow field. In general, this mean flow field has no physical meaning. Still, the time-optimal trajectory in this mean flow field is an experimentally good baseline path when the weather scenarios are not too dispersed.

Path	Duration in FF 0	Duration in FF 1	Duration in FF2	Average
Time-opt. FF 0	1.141	1.481	1.484	1.369
Time-opt. FF 1	1.353	0.9490	0.9554	1.086
Time-opt. FF 2	1.353	1.036	0.8816	1.090
Time-opt. Mean FF	1.314	0.9842	0.9361	1.078
Avg.-time-opt.	1.322	0.9648	0.9302	1.072

Table 7.1: Travel duration for the different paths.

To compute the average-time-optimal path for this problem, we solve a boundary value problem as described in Sec. 7.3.2. We set up $\mathbf{x}(0) = \mathbf{x}_0$ and use $\phi(0)$ and $p_i(0)$, $i = 0, 1, \ldots, N_{\text{ens}} - 1$ as parameters. We then shoot the extremal parameterized with this initial condition by integrating Eqs. 7.17, 7.14 forward in time. We define an error metric $e(\mathbf{x}(\cdot), \mathbf{p}(\cdot)) = \sup_{\lambda \in [0, \lambda_{\max}]} \|\mathbf{x}(\lambda) - \mathbf{x}_f\| + \|\mathbf{p}(\lambda) - \mathbf{a}\|$ with $\mathbf{a} = (a_0 \; a_1 \; \ldots \; a_{N_{\text{ens}}-1})^\intercal$. This error metric adds up the error done on the distance to target in the physical space and the error in the costate variable for the final chord length transversality condition. To find the parameters $\phi(0)$ and $p_i(0)$, $i = 0, 1, \ldots, N_{\text{ens}} - 1$ that entail the average-time-optimal path, we minimize the error metric over the latter parameters. We use the global optimization, gradient free solver `scipy.optimize.direct` to perform the minimization, with bounds equal to $[0.45, 0.8]$ for $\phi(0)$ (inferred from the time-optimal trajectories) and $[0, 1]$ for $p_i(0)$, $i = 0, 1, \ldots, N_{\text{ens}} - 1$. We set a target maximum error of 0.01 for the error function.

Once the computation of all trajectories is completed, we turn each trajectory into a ground path by discarding the time variable. We then use a path-following control law as described in Sec. 7.2.1 to build trajectories following ground paths in different scenarios of flow field. Each path is evaluated in the three scenarios of flow field for the three possible direction of motion for the vortex.



(a)

(b)

Figure 7.7: Varying performance paths for the multi-scenarios vortex case. The underlying flow field is the flow field at $t = 0$, which is the same for all three scenarios. FF stands for Flow Field. The start point is depicted as a black dot and the destination as a black star. The circle around the target shows the tolerance radius for considering the target reached.

7.4.3 Results and discussion

The travel times computed by crossing ground paths and flow field scenarios are given in Table 7.1 and are also depicted in Fig. 7.7b. The five ground paths (time-optimal paths in flow field scenarios 0, 1, 2, time-optimal path in the mean flow field and average-time-optimal path) are depicted in Fig. 7.7a, and the position of the vehicle over the different paths is depicted for the three scenarios of flow field at different timestamps in Fig. 7.8.

While all the paths are similar over around one third of their length, the last two third can be very different from one path to another. In particular, the time-optimal path in flow field scenario 0 is widely bent towards increasing x and increasing y to benefit from the vortex advection. This path has the best performance in its nominal scenario with a travel duration of 1.141 that is 13.2% less than the best among the other paths (Time-opt. Mean FF), but in other scenarios the performance is highly deteriorated with a travel time of 1.481 in scenario 1 and 1.484 in scenario 2, respectively 43.0% and 55.3% more than the worst path among the others (Time-opt. FF 2 in FF 1 and Time-opt. FF 1 in FF 2). This shows the importance of considering at least a bulk of trajectories in the presence of multiple scenarios for the flow field, as one particular time-optimal trajectory in one scenario can have bad performances in other scenarios, even in the controllable regime ($\left\|\mathbf{v}_f\right\| \leq v_r$).

The time-optimal paths in flow field scenario 1 and 2 show similar and good performances, with an average performance close to the optimal average. The average-time-optimal path indeed has the lowest average travel time among all paths, but with a very narrow margin: it is only 0.56% less than the average travel time on the mean flow field time-optimal path.

For better insight into the behavior of the average-time-optimal path, the costate components are plotted for this path in Fig. 7.9. The components show cusps and steep variations in their values, with significant variation from their initialization values. At the optimal chord length $\lambda^\star$, the distance to the destination point in the physical space is less than the defined threshold of 0.01.

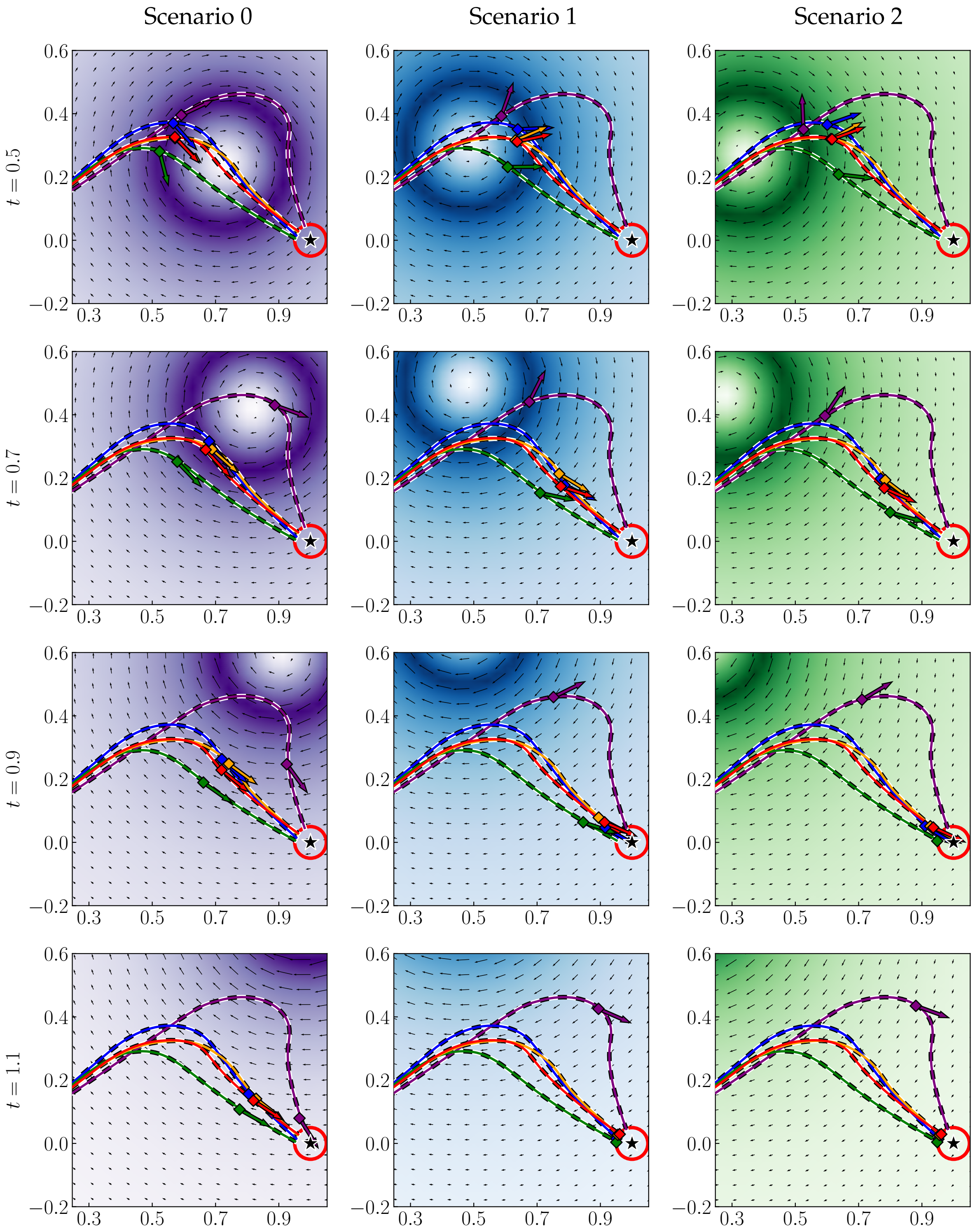


Figure 7.8: Positions of the vehicle while following the different paths in the three flow field scenarios. The left column corresponds to flow field scenario 0, the middle one to the scenario 1 and the right one to the scenario 2. The color of the paths is explained in Fig. 7.7a. For all position marks, the vehicle's heading vector is also depicted.

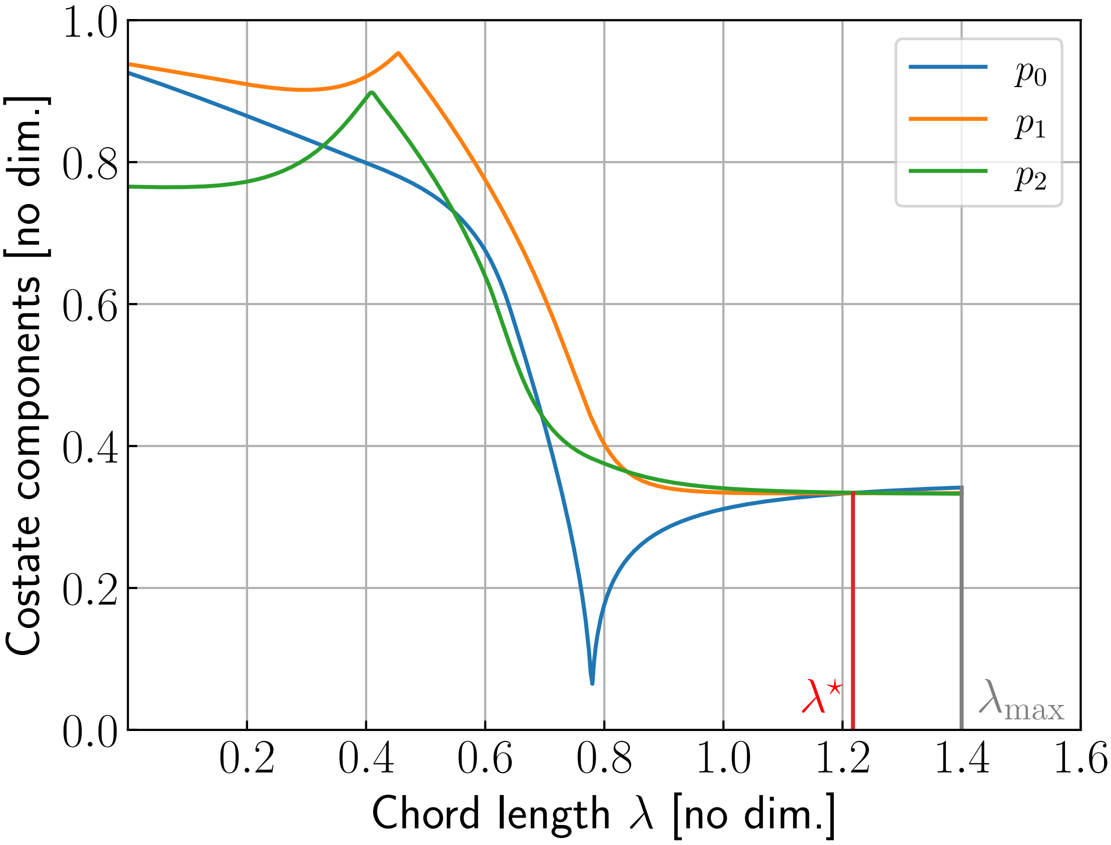


Figure 7.9: Costate components associated to the average-time-optimal path. $\lambda_{\max}$ is the maximum chord length over which extremals are computed. $\lambda^{\star}$ is the chord length at which the solution extremal minimizes the error metric.

Conclusions of the chapter

In this chapter, the addition of uncertainty in the trajectory planning problem was studied. The focus was put on an approach based on weather ensemble predictions rather than a stochastic behavior for the vehicle. The former indeed provides a satisfying way to consistently deal with weather uncertainty while the latter models the different random errors in positioning, control or measurements. For long-range trajectory planning, the weather uncertainty seems to be the main challenge to tackle, hence the choice of uncertainty model. The approach from the underwater vehicle routing literature consisting in computing one time-optimal trajectory per weather scenario and evaluating each trajectory performance in other scenarios was applied in an realistic airborne case. It shows that the average travel time is rather the same over all the different computed paths. Still, the variance of travel time was shown to be reduced for some paths, showing an opportunity to reduce the risk of unexpected high lengthening of travel time that can harm a given mission.

In a second part, the point of view was shifted to ground paths and it was shown that paths minimizing the average travel time over multiple weather scenarios can be characterized by differential equations by using Pontryagin's Maximum Principle. To the best of the author's knowledge, this approach is not seen in the literature to date. The approach was proved to work on an analytical example, but the minimal average travel time was not significantly different from the averaged travel time over classical per scenario time-minimizing paths. Further work would be needed to investigate how much one can reduce the expected travel time by applying the latter approach compared to the per scenario optimization. The present work was preliminary and meant to open up the study of uncertainty mitigation in navigation problems using ground paths and optimality conditions. The metric of average travel time was an entry point in uncertainty mitigation and it would also be interesting to try to find optimality conditions for paths minimizing the variance of travel time or maximizing the robustness to worst weather scenarios.

Chapter's main questions – Answers

- When the problem contains sources of uncertainty, is it possible to plan for an optimal operation of the vehicle that prevents unexpected situations in the navigation?

  When the uncertainty comes from the prediction of the flow field, ensemble forecast provides a set a physically consistent possible scenarios. Optimal path planning applied in each scenario provides a set of routes with different cross-scenario performances, for an operator to select its preferred one (best average travel time, reduced variance of travel time, . . .).

  When the uncertainty comes from random perturbations in the vehicle positioning, control or measurements, stochastic optimal control provides tools to optimize performances in a probabilistic sense.

- Can the benefit of choosing safer or riskier trajectories be quantified?

  Using weather ensemble forecast, the performance of any path reaching destination can be assessed in the different weather scenarios (travel time, energy, . . .). Depending on the mean and variance of performance, one can assess the expected benefit from choosing a path and weight it against its variability, that constitutes a risk for the mission if high deviations are possible.

Conclusion of the thesis

The success of long-range, high endurance missions for autonomous vehicles is conditioned by careful strategic trajectory planning that accounts for the different features of the environment, its threats and opportunities, its variability and its intrinsic uncertainty. A good route is a route that reaches the destination harnessing favorable environment conditions, avoiding hazard or forbidden areas and keeping a safety margin of autonomy at destination for unexpected outcomes. This work is based on the problem of crossing the Atlantic Ocean between Dakar, Senegal and Natal, Brazil with a hydrogen-powered air drone. For this problem, the fulfillment of the mission is of primary importance. This means that every feature concurring in increasing the probability of mission success shall be leveraged. Following trade winds to reduce travel time, adapting the speed to the wind conditions to save energy, avoiding the stormy conditions of the inter-tropical convergence zone as well as areas of possible interference with other vehicles and reducing the possible variability in travel time are all examples of features concurring in the mission success. These qualitative features are unknowingly describing an optimization problem and calling for the use of optimal control to design the best trajectory according to all the latter criteria.

Consequently, this work focused on optimization techniques to compute trajectories for vehicles evolving in complex unsteady and uncertain flow fields. The kind of long-range navigation problem at stake, when considered from a large scale point of view, is the same for aerial vehicle in atmospheric winds than for underwater vehicles in ocean currents. To solve this problem, a large literature spans on several different families of computer methods with different properties. One main distinction exists between direct and indirect methods depending if optimization happens after discretization or prior to discretizing. The second group of methods comes with the advantage of being able to prove the global optimality of the solutions, but are usually computationally more intensive than their direct counterparts. For strategic trajectory planning where computation is done offline with high computing power and where computation times up to an hour are acceptable compared to the planning time window that lasts tenths of hours or more, indirect methods are attractive as their computing intensity is not a problem, while the benefit of proving the optimality of trajectories is important for an operator to trust optimization results.

From the theoretical and numerical perspective, the work presented tried to derive efficient methods from the application of Pontryagin's Maximum Principle to the different problems at stake. The PMP is a powerful tool in optimal control that, unlike its Hamilton-Jacobi-Bellman counterpart, does not suffer *a priori* from the curse of dimension. Building algorithms on top of extremals proved very efficient in many application cases. By building value functions for the navigation problem from the set of extremals, guarantees for optimality were obtained. However, extremals start to behave badly the more strong, spatially and temporally complex the flow field becomes, in the sense that sub-optimal extremals start to outnumber extremals properly lying on the reachability front (and thus candidate to global optimality). The curse of dimensionality seems to be turned into the curse of the flow's intrinsic complexity. Heuristic trimming can be used to stop the spread of useless extremals, such as target-based trimming or trimming based on the shape of the extremal set. Overall, using extremals to solve such optimization problems requires great care in the sampling procedure, but has a good potential for fast resolution with optimality guarantees and scalabilty to higher dimensional problems.

It is known that dealing with obstacles (constraints on the state vector) is challenging for extremals. The necessary conditions of optimality reach one step higher in complexity to describe extremals compatible with state constraints. Still, using algorithmic reasoning to integrate the additional necessary conditions of optimality, a routine to find time-optimal trajectories in the presence of obstacle was successfully designed. It was proven able to compute time-optimal trajectories in a set of analytical and real world examples featuring obstacles. Still, because of the kind of optimality conditions used in this work, the obstacle-aware extremals lost the capacity to propagate the optimal cost everywhere in space. Indeed, in the wake of moving obstacles, the optimal cost is not resolved. This limit is not a caveat for the navigation problems at stake if the destination always lies out of obstacles.

The necessary conditions of optimality also proved useful when allowing the vehicle speed to vary to characterize the less energy intensive operation of the vehicle when harnessing the flow field. Because the power expense is in general a convex increasing function, there is no energetic benefit from adopting a varying speed in the absence of flow field. However, it is demonstrated that adjusting the vehicle speed to the flow field local values can much reduce the overall energy expense, by spending more energy in unfavorable flow conditions to reach favorable ones, where the power can be reduced. Practically, in the real world case of the Dakar to Natal crossing by an air drone in the presence of a tropical storm, it is shown that an order of 10% reduction in total energy expense is possible compared to fixed speed trajectories, with equal travel time. In this chapter, the computation based on extremals was done empirically, but hints were left for how to build a general algorithm to find energy-time-optimal trajectories based on extremals in a similar fashion as what was developed for time-optimal trajectories.

A step aside at MIT gave the opportunity to investigate how to compute trajectories achieving trade-offs between travel time and exposure to hazard using level-set methods. On realistic problems, it demonstrated the existence of paths with limited increase in travel time but highly reduced exposure to hazard. It is shown that the shape of hazard-time-optimal paths can change significantly from time-optimal trajectories to harness zones of lowered exposure to hazard in a given budget of additional travel time. The Hamilton-Jacobi methods used, in the tridimensional state space of planar physical space plus hazard space still performed well for strategic trajectory planning, with computation times always under half an hour on a modern laptop. The method provided a way to smoothly avoid unfavorable navigation conditions, such as the storms of the inter-tropical convergence zone for the Mermoz drone, without resorting to using hard obstacles.

Deterministic optimization methods for trajectories can also provide insight when the navigation problem data is uncertain. Atmospheric or ocean prediction centers provide multiple runs of their physical models with perturbed initial conditions or parameters called ensemble predictions, which gives an operator a collection of possible weather scenarios that help characterize the uncertainty in the predictions. By optimizing trajectories in each scenario and assessing the resulting trajectories in every other scenario, it is possible to measure the expected performance and associated dispersion when following a given path. This procedure, coming from the literature, while not being designed to minimize the expected travel time or the variance in travel time, still provides paths with satisfying performance (optimal paths per scenario), as it is demonstrated in an airborne case. To optimize the operation of the vehicle on a scenario-aggregated metric, path-based necessary conditions of optimality were formulated to minimize the scenario-averaged travel time. While not being the most critical metric in an uncertain setting (compared to variance in the travel or robustness to worst cases), averaged travel time served as an entry door into the path-based extremal formulation. The approach proved able to find the average-time-optimal trajectory in an analytical example, still without significant reduction in average travel time compared to paths computed using a regular time-optimal, per scenario method.

Overall, extremals as well as Hamilton-Jacobi methods proved to be key tools to design paths that optimize the vehicle's performances while satisfying navigation constraints. They need careful setting to work correctly and can demand computational power, but they are able to provide accurate results and guarantees when solving navigation problems in complex, time-varying and uncertain flow fields.

Perspectives

Throughout the whole thesis, and again the previous conclusions, hints were given at the possible extension of this work. They are summarized in what follows.

- **Energy-time-optimal planning: Algorithm.** There is a need to implement an energy-time-optimal extremal-based algorithm as hinted at the end Chapter 5. Because of the curse of dimension, an extremal-based solver can start to outperform a level-set based solver when the dimension increases, which is the case for the energy-time optimal navigation problem. Target-based trimming might be a key feature to avoid sampling too many extremals, that is, the elimination of extremals that are too far from the target to reach it before a maximal allowed time or a maximum level of energy.

- **Probabilistically optimal vehicle guidance.** There is a research gap in the literature of approaches able to optimize for a cost that aggregates the performance of the vehicle in an ensemble of flow field predictions. The approach described in chapter 7 of computing optimal trajectories in each scenario and evaluating their performance in other scenarios is a best-effort, already very efficient way to take uncertainty into account. But it is not designed to optimize for probabilistic performance, as for instance the expected travel time, the variance of travel time or the worst case travel time. In chapter 7, a path-based model was proposed to minimize the expected travel time on an object that stays the same in each scenario: the ground path. Only one test case was presented, and it did not show the existence of a path with significantly better average travel time than time-optimal paths computed in each scenario and used as baseline. So, more work is needed to study if there exist cases in which a clear gain in expected travel time can be found on particular paths that are not captured by per scenario time-optimal paths. Also, paths may also not be the right angle at which to look at the problem. The probabilistic optimal operation of the vehicle might also be better described as a feedback control, for instance. In each case, more work is needed to assess the possible gain of going further than the optimize and evaluate approach.

- **Energy to absorb uncertainty.** The combination of uncertainty with allowing the SRF to vary could lead to insightful analyses of trajectories in weather ensemble prediction, because the delay analysis performed by trying to follow a given trajectory in a weather scenario can be replaced by an analysis where the SRF is allowed to vary, with additional energy spent. This would remove cases when some paths are not followable in some weather scenarios, and would quantify uncertainty as an additional amount of energy to spend to follow a path in unfavorable conditions. This analysis would be helpful to size energy storage to the typical amount of uncertainty for the navigation missions planned.

- **Albatross-like energy optimization.** There is an interesting research topic lying in the combination of energy-time optimal trajectory planning at the large scale and dynamic soaring at the small scale for UAVs. Dynamic soaring consists in harnessing the gradients of wind in the atmospheric boundary layer near the ocean surface to lower the energy expense of flight, in a same manner as albatrosses do (see Bonnin (2015)). Dynamic soaring overall energy reduction can be abstracted at the large scale depending on macro-parameters (minimal allowed clearance above surface, angle to the wind, intensity of the wind, ...) and input in the energy-time problem to compute energy-time trajectory harnessing dynamic soaring. In particular, Bonnin (2015) showed highly significant reduction in the energy expense can be expected in the best conditions for dynamic soaring, and in particular at a path angle close to 45° with the wind field. The question would be to understand what optimal trajectories look like and how much energy can be saved in practice with the addition of this effect.

Appendices

Appendix A

Flow field mathematical model

In Chapter 2, hypothesis (H2) is not a regular hypothesis for control systems. Let's take a general control system

$$\dot{\mathbf{x}} = f(t, \mathbf{x}, \mathbf{u}) \tag{A.1}$$

with $t \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^d$ the state variable and $\mathbf{u} \in \mathbb{R}^m$ the control variable. A general textbook result is that under the following assumptions:

- f continuous in t
- f Lipschitz in $\mathbf{x}$
- f uniformly continuous in $\mathbf{u}$

with a bounded control law $\mathbf{u}(\cdot) \in L^\infty$, the following Cauchy problem admits a unique, globally defined solution:

$$\left\{ \begin{array}{rcl} \dot{\mathbf{x}}(t) & = & f(t, \mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{x}(t_0) & = & \mathbf{x}_0 \end{array} \right. \tag{A.2}$$

This ensures that for any chosen input, the system's behavior is unambiguously defined by its start point and dynamics.

If the system is defined in a bounded subspace $\mathcal{D}$ as it is the case in most studied navigation problem, then the previous theorem ensures there exist a unique solution to (A.2) which may be defined over $[t_0, +\infty[$ or $[t_0, t_{\max}]$, the second case happening when the trajectory reaches the border $\partial\mathcal{D}$ at some time $t_{\max}$ with outward velocity.

Why it still works for piece-wise Lipschitz continuous functions

If f is only assumed piece-wise continuous in $\mathbf{x}$, then there is a finite collection of pairwise disjoint open sets $(\Omega_j)_{j\in J}$ such that $\mathcal{D} = \cup_{j\in J}\overline{\Omega_j}$ and for all $t \in \mathbb{R}$ and all $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{y} \mapsto f(t, \mathbf{x}, \mathbf{u})$ is Lipschitz continuous on Ω_j. Let's call f_j the restriction of f over Ω_j. The following property will ensure that f_j can be extended to a Lipschitz continuous function over $\overline{\Omega}_j$.

Property A.1: Lipschitz continuous extension

Let Ω be an open set in $\mathbb{R}^d$ and φ a continuous function over Ω. Then there exists $\overline{\varphi}$ defined on $\overline{\Omega}$ such that

1. $\overline{\varphi}$ Lipschitz continuous on $\overline{\Omega}$
2. $\forall \mathbf{x} \in \Omega$, $\varphi(\mathbf{x}) = \overline{\varphi}(\mathbf{x})$
3. $\forall \overline{\mathbf{x}} \in \partial\Omega$, $\lim_{\mathbf{x}\to\overline{\mathbf{x}}} \varphi(\mathbf{x}) = \overline{\varphi}(\mathbf{x})$

Proof. To build $\overline{\varphi}$, we first set it to φ on Ω. It thus satisfies property (2).

Then, to define $\overline{\varphi}$ on $\overline{\mathbf{x}} \in \partial\Omega$, we use the fact that $\partial\Omega = \overline{\Omega}\backslash\mathring{\Omega}$ so in particular $\overline{\mathbf{x}}$ is in $\overline{\Omega}$ so there exist a sequence $(\mathbf{x}_n)_{n\in\mathbb{N}} \in \Omega^{\mathbb{N}}$ such that $\lim_{n\to\infty} \mathbf{x}_n = \overline{\mathbf{x}}$. Then $(\varphi(\mathbf{x}_n))_{n\in\mathbb{N}}$ is a Cauchy sequence since $||\varphi(\mathbf{x}_n) - \varphi(\mathbf{x}_m)|| \leq L||\mathbf{x}_n - \mathbf{x}_m||$. Thus, since $\mathbb{R}^d$ is complete, $(\varphi(\mathbf{x}_n))_{n\in\mathbb{N}}$ converges to a limit. This limit is independent of the generative sequence $(\mathbf{x}_n)_{n\in\mathbb{N}}$ as long as it converges to $\overline{\mathbf{x}}$. Indeed if it is the case for $(\mathbf{x}_n)_{n\in\mathbb{N}}$ and $(\mathbf{y}_n)_{n\in\mathbb{N}}$, then $||\varphi(\mathbf{x}_n) - \varphi(\mathbf{y}_n)|| \leq L||\mathbf{x}_n - \mathbf{y}_n||$ which shows that $(\varphi(\mathbf{x}_n))_{n\in\mathbb{N}}$ and $(\varphi(\mathbf{y}_n))_{n\in\mathbb{N}}$ have the same limit. Hence we note the common limit $\psi(\overline{\mathbf{x}})$. We set $\overline{\varphi} = \psi$ on $\partial\Omega$. This way, $\overline{\varphi}$ satisfies property (3).

The only work left is to prove that $\overline{\varphi}$ is Lipschitz continuous on $\overline{\Omega}$. Let $\mathbf{x}, \mathbf{y} \in \overline{\Omega}$. Pick two sequences $(\mathbf{x}_n)_{n\in\mathbb{N}}, (\mathbf{y}_n)_{n\in\mathbb{N}} \in \Omega^{\mathbb{N}}$ with $\lim_{n\to\infty} \mathbf{x}_n = \mathbf{x}$ and $\lim_{n\to\infty} \mathbf{y}_n = \mathbf{y}$. Then

$$||\overline{\varphi}(\mathbf{x}) - \overline{\varphi}(\mathbf{y})|| \leq ||\overline{\varphi}(\mathbf{x}) - \overline{\varphi}(\mathbf{x}_n)|| + ||\overline{\varphi}(\mathbf{x}_n) - \overline{\varphi}(\mathbf{y}_n)|| + ||\overline{\varphi}(\mathbf{y}_n) - \overline{\varphi}(\mathbf{y})|| \tag{A}$$

But $||\overline{\varphi}(\mathbf{x}_n) - \overline{\varphi}(\mathbf{y}_n)|| = ||\varphi(\mathbf{x}_n) - \varphi(\mathbf{y}_n)|| \leq L||\mathbf{x}_n - \mathbf{y}_n||$. And $\lim_{n\to\infty} ||\overline{\varphi}(\mathbf{x}) - \overline{\varphi}(\mathbf{x}_n)|| = \lim_{n\to\infty} ||\overline{\varphi}(\mathbf{y}_n) - \overline{\varphi}(\mathbf{y})|| = 0$. Thus passing to the limit in (A) leads to

$$||\overline{\varphi}(\mathbf{x}) - \overline{\varphi}(\mathbf{y})|| \leq L||\mathbf{x} - \mathbf{y}||$$

which finishes to prove that $\overline{\varphi}$ is Lispchitz continuous over $\overline{\Omega}$.

□

> **Remark**
>
> The Lipschitz continuous extension defined by the previous property is unique.

Noting $\overline{f_j}$ the Lipschitz continuous extension of f_j over $\overline{\Omega}_j$, we thus apply the Cauchy-Lipschitz theorem to $\overline{f_j}$ and get the existence and uniqueness of a trajectory $\mathbf{x}_j$ which may be defined either on $[t_0, +\infty[$ or $[t_0, t_{\max}]$.

Property A.2: Cauchy-Lipschitz for piece-wise Lipschitz continuous functions

Let $\mathcal{D}$ be a bounded set in $\mathbb{R}^d$ and

$$\begin{array}{rcl} f : \mathbb{R} \times \mathcal{D} \times \mathbb{R}^m & \rightarrow & \mathbb{R}^d \\ t, \mathbf{x}, \mathbf{u} & \mapsto & f(t, \mathbf{x}, \mathbf{u}) \end{array}$$

satisfy:

- f continuous in t
- f piece-wise Lipschitz in $\mathbf{x}$
- f uniformly continuous in $\mathbf{u}$

Then there exist a unique maximal solution $\mathbf{x}(\cdot)$ to Cauchy problem (A.2) which is either defined on $[t_0, +\infty[$ or $[t_0, t_{\max}]$ with $\mathbf{x}(t_{\max}) \in \partial\mathcal{D}$.

Proof. Let $(\Omega_j)_{j \in J}$ be the open sets such that $\mathcal{D} = \cup_{j \in J}\overline{\Omega_j}$ and for all $t \in \mathbb{R}$ and all $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{y} \mapsto f(t, \mathbf{x}, \mathbf{u})$ is Lipschitz continuous on Ω_j. Then define $\overline{f_j}$ the Lipschitz continuous extension of f_j over $\overline{\Omega}_j$.

Since $\mathcal{D} = \cup_{j \in J}\overline{\Omega_j}$, $\mathbf{x}_0$ belongs to some $\overline{\Omega_{j_0}}$ with $j_0 \in J$. Thus, problem (A.2) has a unique solution $\mathbf{y}_1(\cdot)$ either defined on $[t_0, +\infty[$ or $[t_0, t_1]$ with $\mathbf{y}_1(t_1) \in \partial\Omega_{j_0}$. In the first case, the result is proven, otherwise if solution stopped at $\mathbf{y}_1(t_1) \in \partial\overline{\Omega_{j_0}}$ and could not be further lengthened, then $f(t_1, \mathbf{y}_1(t_1), \mathbf{u}(t_1))$ is at outward direction for $\overline{\Omega_{j_0}}$: so it is either an outward direction for $\overline{\mathcal{D}}$ at $\mathbf{y}_1(t_1)$ or an inward direction for some $\overline{\Omega_{j_1}}$. In the first case, $\mathbf{y}_1(t_1)$ cannot be in $\mathring{\mathcal{D}}$ so $\mathbf{y}_1(t_1) \in \partial\mathcal{D}$ which proves the result over the maximal window $[t_0, t_1]$. In the second case, repeat the reasoning by initializing the Cauchy problem (A.2) at time t_1 and start position $\mathbf{x}_1 := \mathbf{y}_1(t_1)$ and using the Lipschitz continuous extension $\overline{f_{j_1}}$. End up by concatenating an either finite sequence of trajectory pieces $(\mathbf{y}_1, ..., \mathbf{y}_n)$ over $[t_0, t_1], ..., [t_{n-1}, t_n]$ with $\mathbf{y}_n(t_n) \in \partial\mathcal{D}$ or an infinite number of trajectory pieces defining a unique solution over $[t_0, +\infty[$. □

Appendix B

Zermelo's problem in a steady, linear flow field

B.1 General case

An analytical setting which is of primary importance to study time-optimal navigation is the case in which the flow field is steady and linear

$$\mathbf{v}_f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$$

For this appendix we assume the problem has been scaled so that $u_{\max} = 1$ and thus $\mathbf{u} = \mathbf{h}$ and also $t_0 = 0$.

In this case, the extremal system writes

$$\begin{cases} \begin{pmatrix} \dot{\mathbf{x}}(t) \\ \\ \dot{\mathbf{p}}(t) \end{pmatrix} = \begin{pmatrix} A\mathbf{x}(t) + \mathbf{b} + \mathbf{h}(t) \\ \\ -A^\intercal \mathbf{p}(t) \end{pmatrix} \\ \mathbf{h} = -\frac{\mathbf{p}}{\|\mathbf{p}\|} \\ \begin{pmatrix} \mathbf{x}(0) \\ \\ \mathbf{p}(0) \end{pmatrix} = \begin{pmatrix} \mathbf{x}_0 \\ \\ \mathbf{p}_0 \end{pmatrix} \end{cases}$$

Thus

$$\mathbf{p}(t) = e^{-tA^\intercal} \mathbf{p}_0$$

and

$$\mathbf{h}(t) = -\frac{e^{-tA^\intercal} \mathbf{p}_0}{\|e^{-tA^\intercal} \mathbf{p}_0\|}$$

And finally, using Duhamel's formula on the first ODE

$$\boxed{\mathbf{x}(t) = e^{tA} \left(\mathbf{x}_0 + \int_{s=0}^{t} e^{-sA} \left(\mathbf{b} - \frac{e^{-sA^\intercal} \mathbf{p}_0}{\|e^{-sA^\intercal} \mathbf{p}_0\|} \right) ds \right)} \tag{B.1}$$

The integral term prevents the formula from being purely analytical. Still, it may be evaluated independently by numerical methods and serve as reference optimal trajectory for a path-planner to compare to.

B.2 Planar case

In what follows, $\mathbf{x} \in \mathbb{R}^2$

B.2.1 Diagonalizable matrix

When A is diagonalizable and $b = 0$, we assume the problem is stated in the matrix's eigenbasis, or equivalently we assume

$$A = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

Using the heading angle formulation 3.6 of extremal trajectories, we have the heading ODE

$$\dot{\theta} = (\lambda_2 - \lambda_1)\cos\theta\sin\theta = \frac{\lambda_2 - \lambda_1}{2}\sin 2\theta$$

We are thus looking for solutions of Cauchy problem

$$\begin{cases} \dot{\theta} = \frac{\lambda_2-\lambda_1}{2}\sin 2\theta \\ \theta(0) = \theta_0 \end{cases} \tag{A}$$

There exist an analytical solution to (A), which we state in the following property

Property B.1

The solution to (A) writes

$$\begin{aligned} \theta_0 \not\equiv \tfrac{\pi}{2}[\pi] : \quad & \theta(t) = \arctan\left(\tan\theta_0\, e^{(\lambda_2-\lambda_1)t}\right) + \theta_\infty \\ \theta_0 \equiv \tfrac{\pi}{2}[\pi] : \quad & \theta(t) = \theta_0 \end{aligned}$$

with $\theta_\infty = n_0\pi$ and $n_0 = \lfloor \frac{1}{\pi}\left(\theta_0 + \frac{\pi}{2}\right) \rfloor$.

Proof. Because of periodicity, if $\theta(\cdot)$ is solution to (A), then $\theta(\cdot) + \pi$ is still solution of (A) with initial condition $\theta_0 + \pi$. So we only need to study solutions for $\theta_0 \in]-\frac{\pi}{2}, \frac{\pi}{2}]$. We call $k = \lambda_2 - \lambda_1$

For $\theta_0 = 0$ (resp. $\frac{\pi}{2}$), the solution is constant equal to 0 (resp. $\frac{\pi}{2}$).

Else if $\theta_0 \in]0, \frac{\pi}{2}[$, then we first write the ODE from (A) as :

$$\frac{2\dot{\theta}}{\sin 2\theta} = k \tag{B.2}$$

With $u = 2\theta$, the left term is of the form $\frac{\mathrm{d}}{\mathrm{d}t} f(u) = \dot{u} f'(u)$ with $f \,:\, x \mapsto \ln|\tan\frac{u}{2}|$.

For a given $\theta_0 \in]0, \frac{\pi}{2}[$, the Cauchy-Lipschitz theorem ensures that there exists a timescale $[0, t_*[$ over which the solution ϑ_{θ_0} exists. On this interval the solution is differentiable thus continuous so we take t_* sufficiently small so that $\forall t \in [0, t_*[$, $\vartheta_{\theta_0}(t) \in]0, \frac{\pi}{2}[$. So over this timescale, $u \in]0, \pi[$ over which f is well defined. We integrate (B.2) from 0 to $t \in [0, t_*[$ and we get:

$$\ln\left|\frac{\tan\vartheta_{\theta_0}(t)}{\tan\theta_0}\right| = kt$$

The absolute value falls since both tangents have the same sign and we get:

$$\vartheta_{\theta_0}(t) = \arctan\left(\tan\theta_0\, e^{kt}\right)$$

We notice that we can extend with $\mathcal{C}^1$ continuity this expression to $t \in [0, +\infty[$.

For $\theta_0 \in]-\frac{\pi}{2}, 0[$ the steps are the same and we get the same expression as before for $\vartheta_{\theta_0}(t)$ over $[0, +\infty[$.

We then conclude of a general expression by adding the appropriate shifting constant θ_∞.

□

Then, since $\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{h}$, we look for the solution as $\mathbf{x}(t) = e^{tA}\mathbf{y}(t)$ and we get the ODE

$$\dot{\mathbf{y}}(t) = e^{-tA}\mathbf{h}(t)$$

and

$$\mathbf{h}(t) = \begin{pmatrix} (-1)^{n_0} \frac{1}{\sqrt{1+\tan^2\theta_0\, e^{2(\lambda_2-\lambda_1)t}}} \\ (-1)^{n_0} \frac{\tan\theta_0\, e^{(\lambda_2-\lambda_1)t}}{\sqrt{1+\tan^2\theta_0\, e^{2(\lambda_2-\lambda_1)t}}} \end{pmatrix}$$

so

$$\begin{cases} y_1(t) &= (-1)^{n_0} \frac{e^{-\lambda_1 t}}{\sqrt{1+\tan^2\theta_0\, e^{2(\lambda_2-\lambda_1)t}}} \\ y_2(t) &= \tan\theta_0 (-1)^{n_0} \frac{e^{-\lambda_1 t}}{\sqrt{1+\tan^2\theta_0\, e^{2(\lambda_2-\lambda_1)t}}} \end{cases}$$

Thus the general solution is

$$\mathbf{x}(t) = e^{tA} \left(\mathbf{x}_0 + F(t) \begin{pmatrix} 1 \\ \tan\theta_0 \end{pmatrix} \right)$$

with

$$F(t) = (-1)^{n_0} \int_{s=0}^{t} \frac{e^{-\lambda_1 s}}{\sqrt{1+\tan^2\theta_0\, e^{2(\lambda_2-\lambda_1)s}}} ds$$

which has a closed-form expression that can be found using a formal calculus software

$$F(t) = -\frac{1}{\lambda_1} e^{-\lambda_1 t} \sqrt{1+\tan\theta_0 e^{2(\lambda_2-\lambda_1)t}}\, {}_2F_1\left(1, \frac{\lambda_1}{2(\lambda_1-\lambda_2)} + \frac{1}{2}; \frac{\lambda_1}{2(\lambda_1-\lambda_2)} + 1; -\tan\theta_0 e^{2(\lambda_2-\lambda_1)t}\right)$$

where ${}_2F_1$ is the hypergeometric function.

B.2.2 Gradient flow field

When $A = \begin{pmatrix} 0 & a \\ 0 & 0 \end{pmatrix}$ and $b = 0$, formula B.1 is explicit. It is given in Girardet 2014, Appendix F. This is a special case of the non-diagonalizable case when matrix A has only one eigenvalue equal to 0, then it can write as the previous form in the appropriate basis.

B.2.3 Complex eigenvalues

When A has complex conjugate eigenvalues, it has a simple expression in the appropriate basis.

Property B.2

Let $A \in \mathcal{M}_2(\mathbb{R})$ a matrix with complex (conjugate) eigenvalues $u + iv$ and $u - iv$. Then A is similar to $\begin{pmatrix} u & -v \\ v & u \end{pmatrix}$

Proof. Let $(\mathbf{e}_1, \mathbf{e}_2)$ the eigenvectors of A associated to $(u + iv, u - iv)$. Write

$$\begin{aligned} A\mathbf{e}_1 &= (u + iv)\mathbf{e}_1 \\ A\mathbf{e}_2 &= (u - iv)\mathbf{e}_2 \end{aligned}$$

Then make $L_1 \hookleftarrow L_1 + L_2$ and $L_2 \hookleftarrow i(L_1 - L_2)$

$$\begin{aligned} A(\mathbf{e}_1 + \mathbf{e}_2) &= u(\mathbf{e}_1 + \mathbf{e}_2) + iv(\mathbf{e}_1 - \mathbf{e}_2) \\ A(i(\mathbf{e}_1 - \mathbf{e}_2)) &= -v(\mathbf{e}_1 + \mathbf{e}_2) + iu(\mathbf{e}_1 - \mathbf{e}_2) \end{aligned}$$

Then define the new basis

$$\begin{aligned} \epsilon_1 &:= \mathbf{e}_1 + \mathbf{e}_2 \\ \epsilon_2 &:= i(\mathbf{e}_1 - \mathbf{e}_2) \end{aligned}$$

and notice that the new matrix in this basis is the required one. □

This special form of matrix is studied in Techy (2011).

Appendix C

Linear flow field, quadratic power

The goal of the appendix is to derive the analytical expression of the energy-time Pareto front for the navigation problem in the standard linear flow field with a quadratic power function.

The navigation settings are

$$\mathbf{x}_0 = (0,\ 0)$$
$$\mathbf{x}_f = (1,\ 0)$$
$$\mathbf{v}_f(\mathbf{x}) = \begin{pmatrix} 0 & g \\ 0 & 0 \end{pmatrix} \mathbf{x}$$

C.1 Varying SRF energy-time Pareto front

We use the energy-optimal extremals from Sec. 5.4.

The costate evolution is

$$\dot{\mathbf{p}} = \begin{pmatrix} 0 & 0 \\ -g & 0 \end{pmatrix} \mathbf{p}$$

We consider a quadratic power expense

$$g(\|\mathbf{u}\|) = k \|\mathbf{u}\|^2$$

So

$$(g')^{-1}(a) = \frac{1}{2k}a$$

The optimal control is thus

$$\mathbf{u}^\star = \frac{\|\mathbf{p}\|}{2k}\left(-\frac{\mathbf{p}}{\|\mathbf{p}\|}\right) = -\frac{\mathbf{p}}{2k}$$

Thus, the state evolution is

$$\dot{\mathbf{x}} = -\frac{\mathbf{p}}{2k} + \begin{pmatrix} 0 & g \\ 0 & 0 \end{pmatrix} \mathbf{x}$$

So the evolution of the augmented state is very simple in this case

$$\dot{\mathbf{z}} = \underbrace{\begin{pmatrix} 0 & g & -\frac{1}{2k} & 0 \\ 0 & 0 & 0 & -\frac{1}{2k} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -g & 0 \end{pmatrix}}_{M} \mathbf{z}$$

We have

$$M^2 = \begin{pmatrix} 0 & 0 & 0 & -\frac{g}{2k} \\ 0 & 0 & \frac{g}{2k} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad M^3 = \begin{pmatrix} 0 & 0 & \frac{g^2}{2k} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad M^4 = 0$$

Thus, the augmented costate is

$$\mathbf{z}(t) = e^{tM}\mathbf{z}_0 = \left(I_4 + tM + \frac{t^2M^2}{2} + \frac{t^3M^3}{6}\right)\mathbf{z}_0 = \begin{pmatrix} 1 & gt & -\frac{1}{2k}t + \frac{g^2}{12k}t^3 & -\frac{g}{4k}t^2 \\ 0 & 1 & \frac{g}{4k}t^2 & -\frac{1}{2k}t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -gt & 1 \end{pmatrix}\mathbf{z}_0$$

At $t = t_f$, the equation on $y(t_f) = 0$ gives

$$\frac{g}{4k}\, t_f^2\, p_{x,0} - \frac{1}{2k}\, t_f\, p_{y,0} = 0$$

which simplifies to

$$t_f = \frac{2\, p_{y,0}}{g\, p_{x,0}}$$

Then the equation $x(t_f) = 1$ gives

$$\left(-\frac{1}{2k}\left(\frac{2\, p_{y,0}}{g\, p_{x,0}}\right) + \frac{g^2}{12k}\left(\frac{8\, p_{y,0}^3}{g^3\, p_{x,0}^3}\right)\right) p_{x,0} - \frac{g}{4k}\left(\frac{4\, p_{y,0}^2}{g^2\, p_{x,0}^2}\right) p_{y,0} = 1$$

Which gives

$$-\, p_{y,0} - \frac{1}{3}\frac{p_{y,0}^3}{p_{x,0}^2} = k\, g \tag{C.1}$$

Then, we notice that the instantaneous power is

$$g\left(\frac{\|\mathbf{p}\|}{2k}\right) = \frac{\|\mathbf{p}\|^2}{4k} = \frac{1}{4k}\left(p_{x,0}^2 + \left(p_{y,0} - g\, p_{x,0}\, t\right)^2\right)$$

So the total energy expense is

$$\begin{aligned}
E &= \frac{1}{4k}\int_0^{t_f}\left(p_{x,0}^2 + \left(p_{y,0} - g\,p_{x,0}\,t\right)^2\right)\mathrm{d}t \\
&= \frac{p_{x,0}^2}{4k}t_f - \frac{1}{12k\,p_{x,0}\,g}\left[\left(p_{y,0} - g\,p_{x,0}\,t\right)^3\right]_0^{t_f} \\
&= \frac{p_{x,0}\,p_{y,0}}{2g\,k} + \frac{p_{y,0}^3}{6p_{x,0}\,g\,k} \\
&= -\frac{p_{x,0}}{2g\,k}\underbrace{\left(-p_{y,0} - \frac{1}{3}\frac{p_{y,0}^3}{p_{x,0}^2}\right)}_{k\,g}
\end{aligned}$$

Eventually

$$E = -\frac{1}{2}p_{x,0}$$

So if we use the initial vehicle heading θ_0 with the relation

$$\mathbf{p}_0 = -\left\|\mathbf{p}_0\right\|\left(\cos(\theta_0)\ \ \sin(\theta_0)\right)^\intercal$$

combining C.1 and C.1 gives

$$\tan\theta_0\left(1 + \frac{1}{3}\tan^2\theta_0\right) = \frac{g\,k}{2E}$$

that we can solve in θ_0 to find the initial heading of the trajectory satisfying the boundary conditions and reaching target with energy expense E.

Then, the costate norm is deduced using a reformulation of C.1

$$\left\|\mathbf{p}_0\right\| = \frac{2E}{\cos\theta_0}$$

With the explicit initial costate, Eq. C.1 rewrites

$$t_f = \frac{2\tan\theta_0}{g} \Leftrightarrow \tan\theta_0 = \frac{g\,t_f}{2}$$

Thus, plugging this expression in Eq. C.1 gives the relation between optimal energy expense and the corresponding travel duration

$$E = \frac{k}{t_f\left(1 + \frac{g^2\, t_f^2}{12}\right)}$$

C.2 Fixed SRF energy-time curve

Fixing the SRF and finding energy-optimal trajectories is equivalent to searching time-optimal trajectories. The time-optimal trajectories of the standard linear flow field navigation problem have expression (see Example 3.1.3)

$$\boldsymbol{\xi}_{\theta_0}(\theta) := \begin{pmatrix} \frac{1}{2}\frac{v_r}{g}\left(\frac{-1}{\cos\theta_0}\left(\tan\theta_0 - \tan\theta\right) + \tan\theta\left(\frac{1}{\cos\theta_0} - \frac{1}{\cos\theta}\right) - \ln\frac{\tan\theta_0 + \frac{1}{\cos\theta_0}}{\tan\theta + \frac{1}{\cos\theta}}\right) \\ \frac{v_r}{a}\left(\frac{1}{\cos\theta} - \frac{1}{\cos\theta_0}\right) \end{pmatrix} \tag{C.2}$$

For time-optimal extremals in these settings, the costate is

$$\mathbf{p} = \begin{pmatrix} p_{x,0} \\ p_{y,0} - p_{x,0}\, g\, t \end{pmatrix}$$

and $\tan\theta = \frac{p_y}{p_x}$. At the final position

$$\boldsymbol{\xi}_{\theta_0}(\theta_f) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{C.3}$$

and the lower equation gives $\theta_f = -\theta_0$. But we have

$$\begin{aligned} \tan\theta_f &= \tfrac{p_{y,0} - p_{x,0}\, g\, t_f}{p_{x,0}} \\ \tan\theta_0 &= \tfrac{p_{y,0}}{p_{x,0}} \end{aligned}$$

and $\tan\theta_f = -\tan\theta_0$, thus

$$t_f = \frac{2\ \tan\theta_0}{g} \tag{C.4}$$

Finally the upper equation in Eq. C.3 gives

$$\frac{v_r}{g}\left(\frac{\sin\theta_0}{\cos^2\theta_0} + \operatorname{arctanh}(\sin\theta_0)\right) = 1 \tag{C.5}$$

So to build the energy-time curve, one has to solve numerically Eq. C.5 in θ_0 and plug back the value in C.4 to find t_f. The corresponding energy expense if $t_f\, g(v_r) = k\, t_f\, v_r^2$.

Appendix D

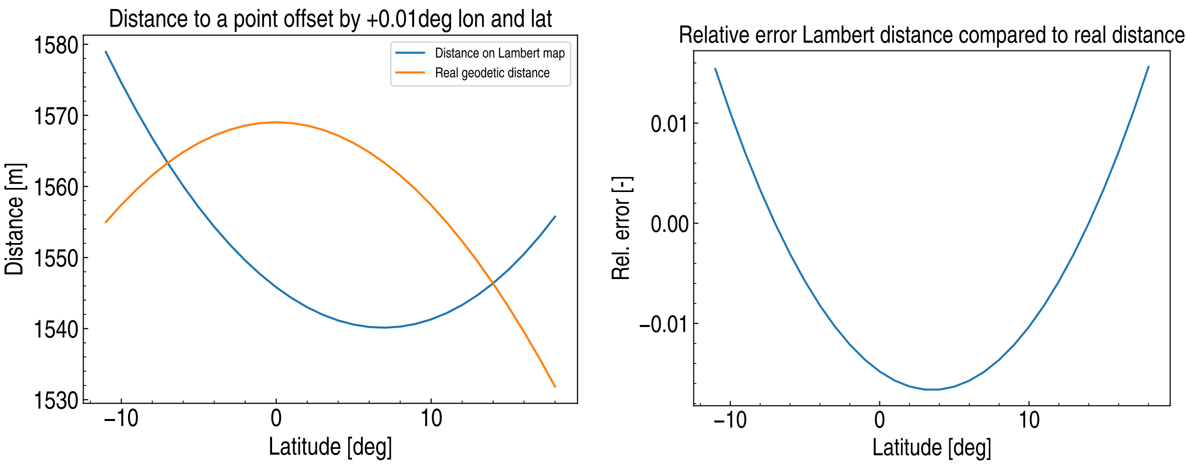
Comparison of optimizing in Lambert projection or in spherical coordinates

In this appendix, time-optimal trajectories computed in a real world case using a projection of Earth (thus approximations) are compared to the ones obtained using an exact spherical model.

The spherical model for time-optimal path planning was introduced in Chapter 2, Sec. 2.6, and the corresponding time-optimal problem was given in Chapter 3, Sec. 3.2. The in-depth interpolated resampling algorithm presented in Chapter 3 was adapted in the DABRY module in spherical coordinates.

In this appendix, the Dakar to Natal crossing by an air drone is considered. For the projected problem, the problem data is the same as presented in Chapter 4, Sec. 4.3.3 but without obstacles. For the spherical problem, the problem data is also the same but is not projected.

The selected projection is a conformal Lambert projection with two standard parallels. This projection preserves angles locally, and has the advantage to balance distance errors between slightly shrunk distances between the two paralllels compared to real distances and increased distances out of the two parallels. For the considered case, the standard parallels are chosen manually to keep the distortion as low as possible everywhere. The values are $-7°$ and $14°$. To measure distance distortion locally, pairs of points are considered, with the second point



(a) (b)

Figure D.1: Difference in the distance measurement on Lambert projection compared to the actual distance on the globe. (a) Absolute distance for a collection of points with varying latitude. (b) Relative error of the Lambert projection compared to the actual distance on the globe.

chosen close to the first one, indeed at 0.01° North and East of the first point. The Lambert projection distance distortion measured this way only depends on the latitude and not the longitude. Thus, a collection of first points are chosen at the middle longitude −26.2° and sampled uniformly from the lowest latitude at −11° to the highest latitude at 18°. The distance the the corresponding offset second points are computed and the result is shown in Fig. D.1 with the absolute distance and relative distance. The relative error in measuring distances on the Lambert projection is shown to be always under 1.7% in absolute value compared to the real distance on the globe, with underestimated distances between the two standard parallels and overestimated distances out of the standard parallels.

Now, the maximal local error in distances is known. But its impact on the computation of optimal trajectories still needs to be assessed. Time-optimal trajectories as well as optimal cost maps are computed on both the Lambert projected space and on the spherical space. They are depicted in Fig. D.2. In the case, time is scaled by the no-wind travel time from origin to destination at the nominal speed of 23 m/s rounded to hours, that is 36h. The arrival time of the trajectory computed on the Lambert projection is 0.997 in rescaled time, while the trajectory

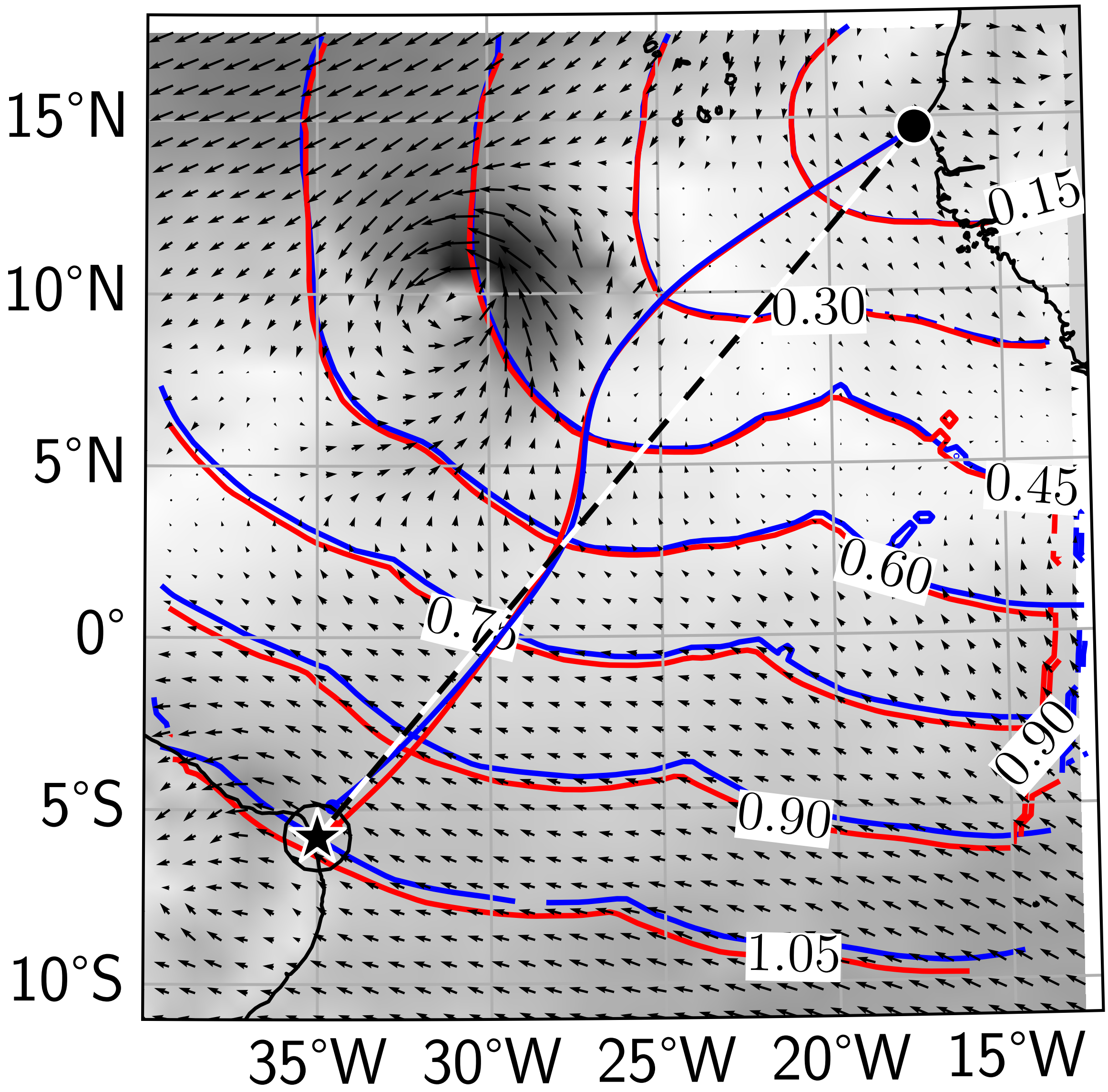


Figure D.2: Comparison of time-optimal path planning results obtained in a Lambert projected space (red) or directly in spherical coordinates (blue). Time-optimal trajectories are depicted, starting from the black dot and reaching the black star with a destination threshold depicted as a circle. Optimal travel time level sets are depicted and labelled in normalized time. The wind field at time 1.01 is depicted in the background.

computed in spherical coordinates reaches the target in time 1.011. Both time-optimal trajectories have a very similar shape. The difference in travel time can also be seen in the level sets of the optimal travel time mapping: the level set computed on the Lambert projection move faster in the direction of the target than the ones computed in spherical space. There are some numerical inaccuracies in the level sets on the right border of the problem.

The shorter travel time in the Lambert projection can be explained by the fact that the time-optimal trajectory in the Lambert projection evolves almost exclusively between the standard parallels, where distances are underestimated compared to their real values. Thus, trajectories go faster in the Lambert projection in this region that would be the case on the globe, hence the reduced travel time. This is corroborated by the shape of the 0.60 optimal travel time level set in Fig. D.2: in the inter-standard-parallel region, the Lambert level set lies in front of the spherical level set, but over 14°, it is the contrary, precisely in the region over the upper standard parallel.

Overall, the difference in shape, travel time and optimal cost map is shown to be of the order of some percent on the considered example, which is a 3000 km navigation mission by an air drone flying at 23 m/s in a wind field of maximum magnitude equal to 23.11 m/s. This serves as a baseline to understand the difference between planning on projection or planning on the globe. While planning directly on the globe does not suffer from distance distortion, planning in projected space comes with the great advantage that all optimization techniques developed in euclidean 2D space directly apply to it. The choice of working on the globe or in projected space is thus a cost-benefit trade-off between the complexity to integrate sphericity in methods compared to the benefit of being precise possibly lower than the percent on travel time, trajectory shape, etc. Yet, in some cases of very long range navigation problems, spherical coordinates are unavoidable. Also, computation times were longer with the spherical model compared to the planar model. This may be explained by the cosine term appearing in the kinematic model of the vehicle in spherical coordinates that is not present in the planar model and which can lengthen by some percents the integration of all trajectories.

Bibliography

Albarakati, S., Lima, R. M., Theußl, T., Hoteit, I., and Knio, O. M. (2020). Optimal 3D time-energy trajectory planning for AUVs using ocean general circulation models. *Ocean Engineering*, 218:108057.

Arutyunov, A. and Karamzin, D. (2020). A Survey on Regularity Conditions for State-Constrained Optimal Control Problems and the Non-degenerate Maximum Principle. *Journal of Optimization Theory and Applications*, 184(3):697–723.

Bakolas, E. and Tsiotras, P. (2010a). Time-optimal synthesis for the Zermelo-Markov-Dubins problem: The constant wind case. In *Proceedings of the 2010 American Control Conference*, pages 6163–6168, Baltimore, MD. IEEE.

Bakolas, E. and Tsiotras, P. (2010b). The Zermelo–Voronoi diagram: A dynamic partition problem. *Automatica*, 46(12):2059–2067.

Betts, J. T. and Cramer, E. J. (1995). Application of Direct Transcription to Commercial Aircraft Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, 18(1):151–159.

Bhabra, M. S. (2021). *Harvest-time optimal path planning in dynamic flows*. PhD Thesis, Massachusetts Institute of Technology.

Bhabra, M. S., Doshi, M., Koenig, B. C., Haley, Jr., P. J., Mirabito, C., Lermusiaux, P. F. J., Goudey, C. A., Curcio, J., Manganelli, D., and Goudey, H. (2020). Optimal harvesting with autonomous tow vessels for offshore macroalgae farming. In *OCEANS 2020 IEEE/MTS*, pages 1–10. IEEE.

Biferale, L., Bonaccorso, F., Buzzicotti, M., Clark Di Leoni, P., and Gustavsson, K. (2019). Zermelo's problem: optimal point-to-point navigation in 2D turbulent flows using reinforcement learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(10):103138. Publisher: AIP Publishing LLC.

Bijlsma, S. J. (1975). *On minimal-time ship routing*. Thèse, Mechanical, Maritime and Materials Engineering, Delft University of Technology, Staatsdrukkerij Den Haag, The Netherlands.

Bijlsma, S. J. (2009). Optimal Aircraft Routing in General Wind Fields. *Journal of Guidance, Control, and Dynamics*, 32(3):1025–1029.

Boltyanskiy, V., Gamkrelidze, R. V., Mishchenko, Y., and Pontryagin, L. (1962). Mathematical theory of optimal processes.

Bonnard, B., Cots, O., and Wembe, B. (2021). A Zermelo navigation problem with a vortex singularity. *ESAIM: Control, Optimisation and Calculus of Variations*, 27:S10.

Bonnin, V. (2015). *From Albatrosses to Long Range UAV Flight by Dynamic Soaring*. PhD thesis.

Borndörfer, R., Danecker, F., and Weiser, M. (2020). A Discrete-Continuous Algorithm for Free Flight Planning. *Algorithms*, 14(1):4.

Bryson, A. and Ho, Y. (1975). *Applied Optimal Control: Optimization, Estimation and Control*. Hemisphere publishing corporation, washington dc edition.

Bulirsch, R., Montrone, F., and Pesch, H. J. (1991a). Abort landing in the presence of windshear as a minimax optimal control problem, part 1: Necessary conditions. *Journal of Optimization Theory and Applications*, 70(1):1–23.

Bulirsch, R., Montrone, F., and Pesch, H. J. (1991b). Abort landing in the presence of windshear as a minimax optimal control problem, part 2: Multiple shooting and homotopy. *Journal of Optimization Theory and Applications*, 70(2):223–254.

Chakrabarty, A. and Langelaan, J. (2013). UAV flight path planning in time varying complex wind-fields. In *2013 American Control Conference*, pages 2568–2574, Washington, DC. IEEE.

Chertovskih, R., Karamzin, D., Khalil, N., and Lobo Pereira, F. (2020). Regular path-constrained time-optimal control problems in three-dimensional flow fields. *European Journal of Control*, 56:98–106.

Chertovskih, R., Karamzin, D., Khalil, N. T., and Pereira, F. L. (2021). An Indirect Method for Regular State-Constrained Optimal Control Problems in Flow Fields. *IEEE Transactions on Automatic Control*, 66(2):787–793.

Chowdhury, R., Navsalkar, A., and Subramani, D. (2022). GPU-Accelerated Multi-Objective Optimal Planning in Stochastic Dynamic Environments. *Journal of Marine Science and Engineering*, 10(4):533.

Crandall, M. G. and Lions, P.-L. (1983). Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American mathematical society*, 277(1):1–42.

Davis, R. E., Leonard, N. E., and Fratantoni, D. M. (2009). Routing strategies for underwater gliders. *Deep Sea Research Part II: Topical Studies in Oceanography*, 56(3-5):173–187.

Delgado, L. and Prats, X. (2013). Effect of Wind on Operating-Cost-Based Cruise Speed Reduction for Delay Absorption. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):918–927.

Dobrokhodov, V. N., Walton, C., Kaminer, I. I., and Jones, K. D. (2020). Energy-Optimal Guidance of Hybrid Ultra-Long Endurance UAV. *IFAC-PapersOnLine*, 53(2):15639–15646.

Doshi, M., Bhabra, M., Wiggert, M., Tomlin, C. J., and Lermusiaux, P. F. J. (2022). Hamilton-Jacobi Multi-Time Reachability. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 2443–2450, Cancun, Mexico. IEEE.

Doshi, M. M., Bhabra, M. S., and Lermusiaux, P. F. (2023). Energy–time optimal path planning in dynamic flows: Theory and schemes. *Computer Methods in Applied Mechanics and Engineering*, 405:115865.

Edelsbrunner, H. and Mücke, E. P. (1994). Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72.

Eichhorn, M. (2015). Optimal routing strategies for autonomous underwater vehicles in time-varying environment. *Robotics and Autonomous Systems*, 67:33–43.

Fleming, W. and Rishel, R. (1975). *Deterministic and Stochastic Optimal Control*. Springer.

Franco, A. and Rivas, D. (2011). Minimum-Cost Cruise at Constant Altitude of Commercial Aircraft Including Wind Effects. *Journal of Guidance, Control, and Dynamics*, 34(4):1253–1260.

Garau, B., Alvarez, A., and Oliver, G. (2005). Path Planning of Autonomous Underwater Vehicles in Current Fields with Complex Spatial Variability: an A* Approach. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 194–198, Barcelona, Spain. IEEE.

Gavrilovic, N., Moschetta, J.-M., and Barascud, Q. (2023). Development of a Hydrogen-powered UAV System for Crossing the Atlantic Ocean. In *AIAA SCITECH 2023 Forum*, National Harbor, MD & Online. American Institute of Aeronautics and Astronautics.

Gavrilovic, N., Vincekovic, D., and Moschetta, J. (2019). A long range fuel cell/soaring uav system for crossing the atlantic ocean. In *11th international micro air vehicle competition and conference, madrid, Spain. IMAV*, page 121e31.

Girardet, B. (2014). *Trafic aérien : détermination optimale et globale des trajectoires d'avion en présence de vent*. PhD Thesis.

Girardet, B., Lapasset, L., Delahaye, D., and Rabut, C. (2014). Wind-optimal path planning: Application to aircraft trajectories. In *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 1403–1408. IEEE.

González Arribas, D. (2019). *Robust Aircraft Trajectory Optimization under Meteorological Uncertainty*. Ph.D. Dissertation, Universidad Carlos III de Madrid, Madrid, Spain.

González-Arribas, D., Soler, M., and Sanjurjo-Rivo, M. (2018). Robust Aircraft Trajectory Planning Under Wind Uncertainty Using Optimal Control. *Journal of Guidance, Control, and Dynamics*, 41(3):673–688.

Gozzi, F., Rouy, E., and Swiech, A. (2000). Second Order Hamilton–Jacobi Equations in Hilbert Spaces and Stochastic Boundary Control. *SIAM Journal on Control and Optimization*, 38(2):400–430. _eprint: https://doi.org/10.1137/S0363012997324909.

Guitart, A., Delahaye, D., and Feron, E. (2022). An Accelerated Dual Fast Marching Tree Applied to Emergency Geometric Trajectory Generation. *Aerospace*, 9(4):180.

Haley, Jr., P. J., Agarwal, A., and Lermusiaux, P. F. J. (2015). Optimizing velocities and transports for complex coastal regions and archipelagos. *Ocean Modelling*, 89:1–28.

Haley, Jr., P. J. and Lermusiaux, P. F. J. (2010). Multiscale two-way embedding schemes for free-surface primitive equations in the "Multidisciplinary Simulation, Estimation and Assimilation System". *Ocean Dynamics*, 60(6):1497–1537.

Hargraves, C. and Paris, S. (1987). Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics*, 10(4):338–342.

Hentzen, D., Kamgarpour, M., Soler, M., and González-Arribas, D. (2018). On maximizing safety in stochastic aircraft trajectory planning with uncertain thunderstorm development. *Aerospace Science and Technology*, 79:543–553.

Hestenes, M. R. (1966). *Calculus of variations and optimal control theory*. Wiley.

Jafarimoghaddam, A. and Soler, M. (2023). Time-Fuel-Optimal Navigation of a Commercial Aircraft in Cruise With Heading and Throttle Controls Using Pontryagin's Maximum Principle. *IEEE Control Systems Letters*, 7:2970–2975.

Janson, L., Schmerling, E., Clark, A., and Pavone, M. (2015). Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *The International journal of robotics research*, 34(7):883–921. Publisher: SAGE Publications Sage UK: London, England.

Jardin, M. and Bryson, A. (2010). Methods for Computing Minimum-Time Paths in Strong Winds. In *AIAA Guidance, Navigation, and Control Conference*, Toronto, Ontario, Canada. American Institute of Aeronautics and Astronautics.

Jardin, M. R. and Bryson, A. E. (2001). Neighboring Optimal Aircraft Guidance in Winds. *Journal of Guidance, Control, and Dynamics*, 24(4):710–715.

Kularatne, D., Bhattacharya, S., and Hsieh, M. A. (2016). Time and Energy Optimal Path Planning in General Flows.

Legrand, K., Puechmorel, S., Delahaye, D., and Zhu, Y. (2018). Robust aircraft optimal trajectory in the presence of wind. *IEEE Aerospace and Electronic Systems Magazine*, 33(11):30–38.

Lermusiaux, P. F. J., Haley, Jr., P. J., Mirabito, C., Mule, E. M., et al. (2024). Real-time ocean probabilistic forecasts, reachability analysis, and adaptive sampling in the Gulf of Mexico. In *OCEANS 2024 IEEE/MTS Halifax*, Halifax. IEEE. In press.

Leutbecher, M. and Palmer, T. (2007). Ensemble forecasting. Publisher: [object Object].

Lewis, J. M. (2005). Roots of Ensemble Forecasting. *Monthly Weather Review*, 133(7):1865–1885.

Li, G., Hildre, H. P., and Zhang, H. (2020). Toward Time-Optimal Trajectory Planning for Autonomous Ship Maneuvering in Close-Range Encounters. *IEEE Journal of Oceanic Engineering*, 45(4):1219–1234.

Lolla, S. V. T. (2016). *Path planning and adaptive sampling in the coastal ocean*. Thèse, Massachusetts Institute of Technology, Department of Mechanical Engineering.

Lolla, T., Lermusiaux, P. F. J., Ueckermann, M. P., and Haley, P. J. (2014). Time-optimal path planning in dynamic flows using level set equations: theory and schemes. *Ocean Dynamics*, 64(10):1373–1397.

Mahoney, J., Bargteil, D., Kingsbury, M., Mitchell, K., and Solomon, T. (2012). Invariant barriers to reactive front propagation in fluid flows. *EPL (Europhysics Letters)*, 98(4):44005.

Mannarini, G., Pinardi, N., Coppini, G., Oddo, P., and Iafrati, A. (2016). VISIR-I: small vessels – least-time nautical routes using wave forecasts. *Geoscientific Model Development*, 9(4):1597–1625.

Mannarini, G., Subramani, D. N., Lermusiaux, P. F. J., and Pinardi, N. (2020). Graph-Search and Differential Equations for Time-Optimal Vessel Route Planning in Dynamic Ocean Waves. *IEEE Transactions on Intelligent Transportation Systems*, 21(8):3581–3593.

Marchidan, A. and Bakolas, E. (2016). Numerical Techniques for Minimum-Time Routing on Sphere with Realistic Winds. *Journal of Guidance, Control, and Dynamics*, 39(1):188–193.

McGee, T., Spry, S., and Hedrick, K. (2005). Optimal path planning in a constant wind with a bounded turning rate. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, California. American Institute of Aeronautics and Astronautics.

Megson, P. W., Najarian, M. L., Lilienthal, K. E., and Solomon, T. H. (2015). Pinning of reaction fronts by burning invariant manifolds in extended flows. *Physics of Fluids*, 27(2):023601.

Miller, B., Stepanyan, K., Miller, A., and Andreev, M. (2011). 3D path planning in a threat environment. In *IEEE Conference on Decision and Control and European Control Conference*, pages 6864–6869, Orlando, FL, USA. IEEE.

Mirebeau, J.-M. (2018). Fast-marching methods for curvature penalized shortest paths. *Journal of Mathematical Imaging and Vision*, 60:784–815. Publisher: Springer.

MSEAS NESMA Ex. (2024). New England Seamounts Experiment Acoustics (NESMA) 2024: New England Seamount Chain – July, 2024.

Nguyen, N. (2006). Singular Arc Time-Optimal Climb Trajectory of Aircraft in a Two-Dimensional Wind Field. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado. American Institute of Aeronautics and Astronautics.

Oettershagen, P., Achermann, F., Müller, B., Schneider, D., and Siegwart, R. (2017). Towards Fully Environment-Aware UAVs: Real-Time Path Planning with Online 3D Wind Field Prediction in Complex Terrain. *arXiv:1712.03608 [cs]*. arXiv: 1712.03608.

Parkinson, C., Arnold, D., Bertozzi, A., and Osher, S. (2020). A Model for Optimal Human Navigation with Stochastic Effects. *SIAM Journal on Applied Mathematics*, 80(4):1862–1881.

Pereira, A. A., Binney, J., Hollinger, G. A., and Sukhatme, G. S. (2013). Risk-aware Path Planning for Autonomous Underwater Vehicles using Predictive Ocean Models. *Journal of Field Robotics*, 30(5):741–762.

Petres, C., Pailhas, Y., Patron, P., Petillot, Y., Evans, J., and Lane, D. (2007). Path Planning for Autonomous Underwater Vehicles. *IEEE Transactions on Robotics*, 23(2):331–341.

Rao, D. and Williams, S. B. (2009). Large-scale path planning for Underwater Gliders in ocean currents. *Australasian Conference on Robotics and Automation*.

Rhoads, B. (2013). *Efficient Guidance of Underpowered Vehicles in Time-Varying Flow Fields*. PhD thesis, University of California Santa Barbara.

Rhoads, B., Mezic, I., and Poje, A. (2010). Minimum time feedback control of autonomous underwater vehicles. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5828–5834, Atlanta, GA, USA. IEEE.

Rhoads, B., Mezić, I., and Poje, A. C. (2013). Minimum time heading control of underpowered vehicles in time-varying ocean currents. *Ocean Engineering*, 66:12–31.

Rudnick-Cohen, E., Herrmann, J. W., and Azarm, S. (2016). Risk-Based Path Planning Optimization Methods for Unmanned Aerial Vehicles Over Inhabited Areas1. *Journal of Computing and Information Science in Engineering*, 16(2):021004.

Schneider, D. (2016). Path planning for fixed-wing unmanned aerial vehicles. Master's thesis, ETH Zürich.

Schnitzler, B., Drouin, A., Moschetta, J.-M., and Delahaye, D. (2023a). General extremal field method for time-optimal trajectory planning in flow fields. *IEEE Control Systems Letters*.

Schnitzler, B., Drouin, A., Moschetta, J.-M., and Delahaye, D. (2023b). General extremal field method for time-optimal trajectory planning in flow fields. In *IEEE Control and Decision Conference*, Singapore.

Schnitzler, B., Haley, Jr., P. J., Mirabito, C., Mule, E. M., Moschetta, J.-M., Delahaye, D., Drouin, A., and Lermusiaux, P. F. J. (2024). Hazard-Time Optimal Path Planning for Collaborative Air and Sea Drones. In *OCEANS 2024 IEEE/MTS Halifax*, Halifax. IEEE.

Sethian, J. A. (1996). A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595. Publisher: National Acad Sciences.

Sethian, J. A. and Vladimirsky, A. (2003). Ordered upwind methods for static Hamilton–Jacobi equations: Theory and algorithms. *SIAM Journal on Numerical Analysis*, 41(1):325–363. Publisher: SIAM.

Sharp, N. and Ross, S. D. (2015). Set-based optimal control in 3D time-varying velocity fields using alpha shapes. *Preprint*.

Soulignac, M. (2011). Feasible and Optimal Path Planning in Strong Current Fields. *IEEE Transactions on Robotics*, 27(1):89–98.

Sridhar, B., Ng, H. K., and Chen, N. Y. (2011). Aircraft Trajectory Optimization and Contrails Avoidance in the Presence of Winds. *Journal of Guidance, Control, and Dynamics*, 34(5):1577–1584.

Subramani, D. N. and Lermusiaux, P. F. (2016). Energy-optimal path planning by stochastic dynamically orthogonal level-set optimization. *Ocean Modelling*, 100:57–77.

Subramani, D. N. and Lermusiaux, P. F. (2019). Risk-optimal path planning in stochastic dynamic environments. *Computer Methods in Applied Mechanics and Engineering*, 353:391–415.

Särkkä, S. and Solin, A. (2019). *Applied stochastic differential equations*, volume 10. Cambridge University Press.

Techy, L. (2011). Optimal navigation in planar time-varying flow: Zermelo's problem revisited. *Intelligent Service Robotics*, 4(4):271–283.

Techy, L. and Woolsey, C. A. (2009). Minimum-Time Path Planning for Unmanned Aerial Vehicles in Steady Uniform Winds. *Journal of Guidance, Control, and Dynamics*, 32(6):1736–1746.

Torenbeek, E. (2013). *Advanced aircraft design: conceptual design, analysis and optimization of subsonic civil airplanes*. John Wiley & Sons.

Trélat, E. (2005). *Contrôle optimal: théorie & applications*, volume 36. Vuibert Paris.

Vinter, R. B. (2010). *Optimal control*, volume 2. Springer.

Wang, T., Le Maître, O. P., Hoteit, I., and Knio, O. M. (2016). Path planning in uncertain flow fields using ensemble method. *Ocean Dynamics*, 66(10):1231–1251.

Wang, T., Lima, R. M., Giraldi, L., and Knio, O. M. (2019). Trajectory planning for autonomous underwater vehicles in the presence of obstacles and a nonlinear flow field using mixed integer nonlinear programming. *Computers & Operations Research*, 101:55–75.

Zermelo, E. (1930). Über die Navigation in der Luft als Problem der Variationsrechnung. *Jahresbericht der deutschen Mathematiker–Vereinigung, Angelegenheiten*, 39:44–48.

Zhang, B., Tang, L., and Roemer, M. (2014). Probabilistic Weather Forecasting Analysis for Unmanned Aerial Vehicle Path Planning. *Journal of Guidance, Control, and Dynamics*, 37(1):309–312.

The illustration on page 1 is a stereographic projection of Earth with the fastest trajectory for an air drone flying at sea level at 23 m/s between Dakar, Senegal and Natal, Brazil starting at time 2021-09-29 06:00Z depicted in red. Isochrons, *i.e.* contours with the same travel time from the starting point are depicted in black and white. The underlying wind field is the ERA5 reanalysis wind field at sea level (1000hPa) at time 2021-09-29 06:00Z. A stream plot of this wind field is also depicted. The optimal trajectory and the isochrons are computed using the open-source DABRY Python module.