Universidad de Las Palmas de Gran Canaria

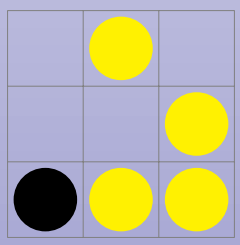
Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería

Contributions to Underwater Glider Path Planning

Contribuciones a la Planificación de Rutas para Planeadores Submarinos

by

Enrique Fernández Perdomo



A thesis submitted for the degree of
Doctor of Philosophy

January 26, 2014

Las Palmas de Gran Canaria, Spain

Under the supervision of

Antonio C. Domínguez Brito
Daniel Hernández Sosa ❖ Josep Isern González

Contributions to Underwater Glider Path Planning

Contribuciones a la Planificación de Rutas para Planeadores Submarinos

Universidad de Las Palmas de Gran Canaria

Instituto Universitario de Sistemas Inteligentes y
Aplicaciones Numéricas en Ingeniería

Enrique Fernández Perdomo

Las Palmas de Gran Canaria, Spain

A thesis submitted for the degree of
Doctor of Philosophy

Under the supervision of
Antonio C. Domínguez Brito
Daniel Hernández Sosa ❖ Josep Isern González

January 26, 2014

To my family

The fundamental cause of the trouble is that in the modern world
the stupid are cocksure while the intelligent are full of doubt.

The Triumph of Stupidity
Bertrand Russell

Contents

List of Figures

List of Tables

List of Algorithms

Abstract

Path or Motion Planning is a Robotics discipline that deals with the search of feasible or optimal paths. For most vehicles and environments, it is not a trivial problem and therefore we find a great diversity of algorithms to solve it, not only in Robotics and Artificial Intelligence, but also as part of the Optimization literature, with Numerical Methods and Bio-inspired Algorithms, like Genetic Algorithms and Ant Colony Optimization.

The particular case of variable cost scenarios is considerably difficult to address because the environment where the vehicle moves varies over time. The present thesis work studies this problem and proposes a number of practical solutions for Underwater Robotics applications. The thesis focuses on Autonomous Underwater Glider (AUG) Path Planning, considering its motion model and the effect of ocean currents. Gliders are a type of Autonomous Underwater Vehicle (AUV) that moves with a relatively slow nominal speed when compared with ocean currents. For this reason, they conform an interesting research field from the perspective of path planning. Ocean current maps provide the snapshot of this physical parameter at different time instants, which along with other physical parameters, are computed with Ocean Models. In those zones close to the coastline they exhibit a high temporal variability, so we have to perform path planning in a variable cost environment, where the cost function is also asymmetric.

This work attacks the problem using different approaches. Starting with classical path planning algorithms adapted to the problem, like A*, up to optimization techniques, Genetic Algorithms, Probabilistic Sampling methods, among others. The thesis also provides a number of proposals to solve this particular problem. For instance, some variants of A* combined with probabilistic sampling techniques and several proposals of optimization algorithms within the configuration space of the feasible bearings the vehicle can follow from each surfacing point, are described and analyzed. A thorough comparison of these techniques is covered by this document, with the intention of assessing which ones are more suitable for the given problem, as well as analyzing their computational cost. Finally, the work explores sampling strategies for movable ocean structures.

Resumen

La Planificación de Rutas o Caminos es un disciplina de Robótica que trata la búsqueda de caminos factibles u óptimos. Para la mayoría de vehículos y entornos, no es un problema trivial y por tanto nos encontramos con un gran diversidad de algoritmos para resolverlo, no sólo en Robótica e Inteligencia Artificial, sino también como parte de la literatura de Optimización, con Métodos Numéricos y Algoritmos Bio-inspirados, como Algoritmos Genéticos y el Algoritmo de la Colonia de Hormigas.

El caso particular de escenarios de costes variables es considerablemente difícil de abordar porque el entorno en el que se mueve el vehículo cambia con el tiempo. El presente trabajo de tesis estudia este problema y propone varias soluciones prácticas para aplicaciones de Robótica Submarina. La tesis está enfocada en la Planificación de Rutas para Planeadores Autónomos Submarinos —*Autonomous Underwater Gliders (AUGs)* en inglés—, considerando su modelo de movimiento y el efecto de las corrientes oceánicas. Los planeadores son un tipo de Vehículo Submarino Autónomo —*Autonomous Underwater Vehicle (AUV)* en inglés— que se desplaza con un velocidad nominal relativamente baja en comparación con las corrientes oceánicas. Por este motivo, constituyen un interesante campo de investigación desde la perspectiva de la planificación de caminos. Los mapas de corrientes oceánicas proporcionan una captura de estos parámetros físicos en diferentes instantes de tiempo, que juntos con otros, son computados por Modelos Oceánicos. En aquellas zonas cercanas a la costa exhiben una alta variabilidad temporal, de modo que tenemos que realizar la planificación de caminos en un entorno de costes variables, donde la función de coste es además asimétrica.

Este trabajo aborda el problema usando diferentes aproximaciones. Empezando por algoritmos de planificación de caminos clásicos adaptados al problema, como el A*, hasta técnicas de optimización, Algoritmos Genéticos, métodos de Muestreo Probabilístico, entre otros. La tesis también ofrece varias propuestas para resolver este problema particular. De hecho, varias variantes del A* combinadas con técnicas de muestreo probabilístico y varias propuestas de algoritmos de optimización en el espacio de configuración de los rumbos de aguja factibles que el vehículo puede seguir desde cada punto de emersión, son descritos y analizados. Este documento recoge una amplia comparativa de estas técnicas, con la intención de valorar cuáles son las más apropiadas para un problema dado, así como analizar su coste computacional. Finalmente, el trabajo explora estrategias de muestreo para estructuras oceánicas móviles.

Acknowledgements

This is probably the section most difficult to write of this document because, honestly, all the work done would not have been possible without the help and advice of many people. Colleagues or friends, who have been there since the beginning, or just at crucial stages throughout these last four years. Certainly, it is also difficult because the emotion dampens my eyes.

Surely, it is not easy to express on the paper these emotions and how thankful I feel now. Maybe, it is neither easy that those to whom I address these acknowledgements, recognize how sincere and touching it is writing these lines. Therefore, I have thought it would be a great idea to say thanks in their mother tongue —although it were just a sign of esteem. Because they always sound more emotive, when the words are said in the language we use with our close friends.

Empezaré por los miembros del grupo de investigación en el que no sólo he trabajado, sino convivido, hace ya algo más de cuatro años. Somos un grupo pequeño, y quizás por ello todo se magnifica. Lo bueno y lo malo, aunque lo cierto es que al final todo queda en tablas. Pero hoy quiero recordar lo bueno, que es mucho y quedará en mi memoria para siempre. Espero que también en la de los que me han ayudado a llegar hasta aquí.

Casi sin darme cuenta terminé la carrera con un proyecto sobre submarinos autónomos, y a renglón seguido empecé con el Máster del Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería. El SIANI, que es como también llamamos al laboratorio donde nos vemos a menudo. Aunque en realidad somos una división dentro del instituto, para mi el SIANI siempre ha sido Jorge, Daniel, Antonio Carlos y Josep. Sé que son muchos más, y les guardo especial aprecio: Modesto, Javier, Antonio Falcón, ... es imposible nombrarlos todos; incluso Juan Ignacio, que siempre se maravillaba con el material que tenía que inventariar en nuestro laboratorio: "dónde la pongo".

El 16 de octubre de 2009 fue una fecha clave. Jorge y yo asistimos a una charla sobre un vehículo que planeaba bajo el agua intentando realizar la proeza de cruzar autónomamente el Atlántico. Así conocimos a Antonio G. Ramos. Puro entusiasmo. Pronto empezamos a planificar rutas para ayudar a RU27 —el heroíco planeador o *glider*— a alcanzar la costa de Baiona. Aún guardo las notas que tomé en aquella charla. Gracias a Antonio, y a Pep y Alex, por poner las cosas tan fáciles para que nos pudiéramos poner en marcha desde el primer

momento. Igualmente, gracias a *Puertos del Estado*, que nos ofreció acceso a las salidas del Modelo Oceánico Regional del proyecto ESEOO, y a la Universidad de Rutgers, por el acceso a los datos de la misión trans-Atlántica de RU27.

Ahí nació la División de Robótica y Oceanografía Computacional y, aunque quizás menos importante, mi tema de tesis. Éramos un grupo de investigación con una larga trayectoria haciendo robótica, y por aquellas fechas ya habíamos empezado a hacer robótica submarina. Aquello fue un empujón en un momento propicio. Me habían concedido la beca de posgrado de la ULPGC, a la que agradezco que haya hecho posible que me dedique a investigar, dándome un sueldo y formación. Tenía tema de tesis y tiempo para llevarla a término.

En este sentido también teníamos recursos a través de varios proyectos de investigación. En concreto, este trabajo está parcialmente financiado por los proyectos PI2007/039 y PI2010-0062 del Gobierno de Canarias (Consejería de Educación, Cultura y Deportes) con fondos FEDER, el proyecto ProID20100062 de la Agencia Canaria de Investigación, Innovación y Sociedad de la Información (ACIISI) con fondos FEDER, y el proyecto TIN2008-06068 del Ministerio de Ciencia e Investigación. Así como por la Secretaría de Estado de Universidades e Investigación (Ministerio de Ciencia e Innovación) y FEDER con el contrato UNLP08-3E-010.

En definitiva, nos pusimos a trabajar todos. Doy mil gracias a Josep y Daniel por lo mucho que me han ayudado para sacar adelante los algoritmos de planificación que desarrollamos. También doy gracias a Antonio Carlos, por sus precisos consejos, y por su infinita ayuda con la tesis y mis primeras clases como docente. Jorge se merece un premio. Sus consejos trascienden lo académico. Sus conocimientos son enciclopédicos. Sus charlas, a menudo largas —especialmente con un servidor—, ayudaban a *este burro a sacar la cabeza del surco*. Gracias, Jorge.

Pronto pudimos publicar y tuve la suerte de poder ir a varios congresos. Guardo un especial recuerdo del primero y de sus asistentes: el WAF'09, en Cáceres. Los siguientes fueron más lejos, pero de todos traía alguna delicia exótica: bombones de higo, galletas de mantequilla, y una mermelada asquerosa de Bangkok.

My second conference was at Sydney: the OCEANS'10. More than twelve hours away. There I met Mike Eichhörn, who chaired my session. We ended up chatting after the talks, and some months later we wrote a paper together. Danke schön. I also met a very proactive researcher and friendly colleague: Ryan Smith. It was great to find him in my first international conference. He was very interesting in our work, and we are still in contact. Indeed, we met again one year later at ICRA'11 in Shanghai, and spend some days —and nights— catching up. Thanks, Ryan. I hope we will fly some gliders together in the future!

I also went twice to a Summer School in Bremen, at Jacobs University. There I met Andreas Birk and his teammates: Max, Ravi, Narunas, ..., and specially Heiko. I spent two great weeks with them and the classmates. I learned a lot those days, and also had time to go on bike around the beautiful Schönebeck and Vegesack. I am still in contact with some of them. Danke schön.

Tot i que el més freqüent és fer l'estada fora d'Espanya i durant tres mesos, la meva va durar sis mesos i va ser a Girona. No me'n penedeixo en absolut, vaig sentir-me com a casa i vaig fer bons contactes i grans amistats. És per això que vull donar les gràcies a tothom.

A l'Arnau, per donar-me conversa i entretenir-me en els mesos en els que va ser el company de la taula del costat. A en David, que amb la seva admirable claredat d'idees em va explicar els detalls dels filtres de Kalman. A en Narcís, per ser tan proper i faciliatar-me les coses a la feina. A en Carles, que em va ajudar a posar-me en marxa amb el manipulador Staubli. A en Lluís, sempre tan ressolutiu en donar-me la solució de qualsevol tasca de mecanitzat o electrònica. A en Sharad, perquè aquells dinars mentre parlàvem de la feina i de la vida em van donar la perspectiva necessària per veure les coses en la seva justa mesura.

En especial a la Tali, única noia del grup i la companya que tenia més energia de tots nosaltres —Besalú és el més bonic que vaig veure a Girona! I l'Aggelos, el mariner del grup, per emprar sempre les paraules adequades i parlar poc però fer molt. És un gran exemple de com fer les coses correctament. És una persona seriosa però divertida. Aquest parell em van deixar tancat al CIRS més d'una vegada però els ho perdono!. Una nit la vàrem fer grossa!. No s'apagava l'alarma i van haver de venir els guardes de seguretat del campus —crec però, que aquest cop no vàren ser ells— ens hi vàrem estar fins a les onze de la nit.

Per descomptat a l'Emili. El vaig veure poc perquè ell estava escrivint la seva tesi, però vaig poder anar a la defensa. Això no obstant, temps enrere haviem coincidit a Shanghai i em va ensenyar el CIRS en una visita un any abans de la meva estada. En aquella ocasió em va convèncer definitivament per a fer l'estada amb ells —i per anar amb ell a un busseig improvisat prop de les Illes Medes! A en Marc, que cada cop que parlava amb ell em transmetia la calma de fer les coses poc a poc i amb bona lletra. I a en Pere i en Nuno, els meus supervisors.Els seus consells em van ajudar a avançar en la tasca que vaig desenvolupar amb ells. També a en Martí i les seves noies, per aquells dinars tan bons a "sis i mig" —de vegades "vuit i mig"— al Parc Cafeteria. És una llàstima que no tinguin servei a domicili a Canàries! Guardo també un especial record de la Joseta, en Xevi, en Miki, en Rafa i tants d'altres que van fer que la meva estada fos més amena i enriquidora.

Nuno merece uma menção especial. Uma belíssima pessoa para com todos. Fez-me sentir como um convidado de honra. Tanto que acabei por tomar a cidade como minha graças à ele. Esteve sempre presente para ajudar-me com as dúvidas no trabalho, e para animar-me a desfrutar dos dias de folga. Muito obrigado, Nuno.

A més a més, en Nuno em va presentar les seves amistats i em va ajudar a conèixer Girona i la seva gent. Així vaig conèixer la Núria, a qui dono les gràcies per portar-se com una bona

amiga i ensenyar-me els indcrets més bonics de Girona i la Costa Brava, però sobretot per la seva amistat. Gràcies a tots els components del grup "Ens va la marxa, marxa": Cesar, Toni, Elsa, Imma, ... espero no deixar-me ningú! I en especial a l'Eva, que va veure coses en mí que jo no era capaç de veure. Van fer molt agradable el meu pas per Girona.

La estancia en Girona la compaginé con la participación en el SAUC-E'12, como un miembro más del equipo AVORA. Fue todo un honor, chicos. Gracias por hacerme un hueco y dejarme disfrutar de aquella semana en La Spezia. Esos días han renovado mi ambición. Anil, Aaron, Luis, David, Federico, y Daniel, gracias. Mucho ánimo a los participantes de este año.

También quiero dar gracias a mis amistades, muchos amigos de carrera. Con ellos viví una de las mejores etapas de mi vida. Gracias por sus ánimos estos últimos meses y por ayudarme a distraerme algún que otro fin de semana. En especial, gracias a Rodrigo y Eliezer, compañeros de prácticas y risas. También guardo un especial recuerdo de Orlando, Jorge, Samuel, Gastón, y muchos más. Gracias, porque lo que aprendí con ustedes me ha llevado hasta aquí.

Finalmente, mi familia. Gracias a Enrique senior, y a mi madre. Mi padre, por sus sabios consejos. La voz de la experiencia. Y por hacer de chófer tantas veces, pero sobre todo, por no pedir nada a cambio. Gracias mamá, por hablar conmigo, escucharme, y por el sancocho. También gracias a mi hermana Yolanda, con la que a veces puedo hablar de estas cosas tan técnicas que hago en el trabajo. Gracias a los tres por liberarme de tantas tareas estos últimos meses que llevo pegado al teclado. Y finalmente gracias a mis vecinos por ayudarme a mantenerme despierto estas últimas noches antes de acabar la tesis. Mientras yo hacía la memoria de la tesis, ... Gracias. Thank you. Danke. Gràcies. Obrigado. ✍

Notation

$g(n_i)$ Accumulated cost from n_{start} to n_i node.

α Angle of Attack (AoA).

AR Aspect Ratio $\mathit{AR} = \frac{b^2}{S}$.

ψ_g Glider bearing angle.

$|\mathcal{B}|$ Number of bearings (cardinality of $\mathcal{B}$).

$\mathcal{B}$ Bearings set that defines the path $\mathcal{P}$.

∇V_{bp} Buoyancy change due to the buoyancy engine.

$\mathcal{O}(\cdot)$ Complexity in big-$\mathcal{O}$ notation.

ε Compressibility of the hull.

$c(n_i, n_{i+1})$ Cost of going from node n_i to n_{i+1}.

C_D Drag coefficient.

$\mathbf{x}_{n-1}$ Ending location.

$f(n_i)$ f-cost for node n_i.

γ Glide angle $\gamma = \theta + \alpha$.

U Glider velocity through water along the glide path.

ψ_e Glider heading (or course) angle.

$\hat{\psi}_e$ Heading error.

$h(n_i)$ Heuristic value for node n_i.

$C_{D_{1,h}}$ Induced drag for the hull.

$C_{D_{1,w}}$ Induced drag for the wings.

C_L Lift coefficient.

a_h Lift slope coefficient for the hull.

a_w Lift slope coefficient for the wings.

m_g Glider mass.

$\|\cdot\|$ Euclidean norm, usually used as a distance.

$C_{D_{0,h}}$ Parasite drag for the hull.

$C_{D_{0,w}}$ Parasite drag for the wings.

$\mathcal{P}$ Path.

$\mathcal{P}_d$ Desired path, for the path following problem.

θ Pitch angle in the BODY frame.

T_0 Reference temperature for α_T.

ϕ Roll angle in the BODY frame.

e Span efficiency parameter $e \approx 0.8$.

SR Speed ratio.

$\mathbf{x}_0$ Start location.

$n_{\textbf{start}}$ Start node.

$\mathbf{x}_i$ Surfacing location.

t_s Surfacing time, between consecutive stints.

t_t Travel time, from $\mathbf{x}_0$ to $\mathbf{x}_{\text{goal}}$.

$\mathbf{x}_{\textbf{goal}}$ Target waypoint location.

$n_{\textbf{goal}}$ Target (or goal) node.

α_T Thermal expansion coefficient.

U_c Ocean current speed.

U_e Glider effective speed $U_e = U_g + U_c$.

U_g Glider nominal surge speed $U_g = U\cos\gamma$.

V_g Glider volume at atmospheric pressure.

b Wingspan.

S Wing surface area.

Ω Wing sweep angle (at $1/4$ chord line).

ψ Yaw angle in the BODY frame.

Chapter 1

Introduction

We will begin justifying this work by describing one of the main scopes of application of its contributions and results. It is the vast scientific discipline called Oceanography. Also known as Marine Science, it is the science that studies the ocean. It was in 1872 when the departure of the H.M.S. Challenger (see Figure 1.1) from England on the historic four-year global expedition of exploration established the science of Oceanography (Earle, 2012). Given the complexity of the world ocean system, Oceanography actually covers a wide range of topics, including ocean currents, waves, and geophysical fluid dynamics; plate tectonics and the geology of the sea floor; marine organisms and ecosystem dynamics; and fluxes of various chemical substances and physical properties within the ocean and across its boundaries. Although we will later see that the contributions of the present work can assist several sectors in the industry and also the navy, here we will focus on which is probably the sector most favored from the improvements in marine technology and engineering.

1.1 Ocean Research

Probably, one of the most interesting oceanographic phenomena to study is Climate Change, since it affects globally both the ocean itself and the life it supports. The factors that cause Climate Change include oceanic processes, variations in solar radiation received by Earth, plate tectonics and volcanic eruptions, and human-induced alterations of the natural world. Clearly, dissimilar agents that are complex and difficult to measure and study. In the glossary of the National Snow and Ice Data Center (NSIDC), Climate Change is defined as the study dealing with variations in climate on many different time scales from decades to millions of years, and the possible causes of such variations. In order to study these processes to find statistical evidence of Climate Change for a particular period of time and validate hypotheses like the correlation of Global Warming and anthropogenic activities, a global monitoring and sampling network is required. Only with accurate temporal series of physical, chemical and biological data, it is possible to improve climate models, and the models of the subsystems they rely on (Bindoff et al., 2007). The more confident and predictable the state of the atmosphere and the oceans are, the greater the chance to monitor and sample processes that happen in the environment adequately —e.g. at the exact location, for their whole lifespan, with a higher or adaptive sampling resolution, etc. Otherwise, we will not be able to adapt our monitoring or sampling strategy until the phenomenon has reached a mature state in which it can be observed from the surface, and hence we miss all the previous subprocesses that take place underwater.

One of the most visible proofs of ocean warming and acidification is the growth in the frequency and intensity of bloom episodes. An algal bloom is a rapid increase or accumulation in the population of algae in an aquatic system. Also, the particular case of Harmful Algal Blooms (HABs) constitutes a remarkable problem for the ecosystem and human society. Ocean calcification and the impact on ocean calcifying organisms, is a similar and related process as well, with significant consequences on the biological and chemical characteristics

Figure 1.1: H.M.S. Challenger, a steam-assisted Royal Navy Pearl-class corvette launched on 13 February 1858 at the Woolwich Dockyard, that undertook the first global marine research expedition: the Challenger expedition (1872–1876). Picture taken from The NOAA Photo Library.

of the region. It turns out to be of extreme interest for the scientists to establish which are the physical, chemical and biological processes that originate HABs. It is also critical to observe the evolution of these structures, quantify the concentration of nutrients, classify the chemicals and organisms of the food chain that form the ecosystem of blooms, etc. The greatest difficulty lies in the fact that most of these processes happen beneath the surface, and might originate far beyond several days or weeks before their consequences are observable from the surface, as in the red tide shown in Figure 1.2. Since blooms depend on vertical transport, their spring exhibits a tight relation with upwelling zones.

Upwelling is a phenomenon in which a vertical wind-driven motion of dense, cooler, and usually nutrient-rich water goes towards the ocean surface, replacing the warmer, usually nutrient-depleted surface water. During upwelling episodes or in upwelling regions, the contribution of nutrient-rich water stimulates the growth of primary producers like phytoplankton. This is the basis for a food chain that supports some of the most productive fisheries in the world. Clearly, the study of this ecosystem demands methods that can monitor and take samples underwater. Furthermore, in those places where the episodes are sporadic, and generally unpredictable, large areas must be continuously sampled at different depths in order to detect the phenomenon since it begins. Vertical transport also happens in the other direction, going from the surface to the ocean's interior. In oceanic biogeochemistry there is a process known as the Biological Pump, or also Vertical Pump, which is illustrated in the diagram of Figure 1.3. Organic carbon is transported primarily by sinking particulate material —e.g. dead organisms or fecal pellets—, while Dissolved Organic Carbon (DOC) reaches the deep ocean by physical transport processes like downwelling. Also, inorganic carbon such as calcium carbonate ($CaCO_3$), a component of calcifying organisms, reaches the deep ocean by these means. This vertical pump takes CO_2 from the atmosphere and plays an important role in marine primary production.

Figure 1.2: Red Algal Bloom. Photo by Miriam Godfrey.

It is thought that Climate Change may affect the biological pump in the future, so it turns to be an important process to study as an indicator (Orr et al., 2005).

A key requirement for the study of any process and its temporal evolution in the ocean, is the knowledge or estimation of how the water masses move. That is exactly what Ocean Models are designed for, since they solve the oceanic primitives equations in order to obtain the ocean currents. The earliest class of Ocean Model, and still the most widely applied, was pioneered by Kirk Bryan using low-order finite difference techniques (Bryan and Cox, 2011). Since 1970, the computational methods have evolved and, in recent years, the usage of terrain-following coordinates (sigma, hybrid) Ocean Models has led to improvements of the numerical algorithms for time-stepping, advection, pressure gradient, and subgrid-scale parameterizations. Nowadays we can find Regional Oceanic Models (ROMs), which are based on models that can be adapted to relatively small areas. A good number of oceanographic institutions use these models to predict the ocean conditions. In order to run the models adequately, they must be correctly parameterized, tunned and, above all, initialized with the current conditions/state of the ocean system. This is far from being a trivial task, and for that reason ROMs are usually state-of-the-art ocean models tuned for a particular area, which turns to be a complex task in practice; see, as a ROM product example, the ocean currents map of Figure 1.4. A great and heterogeneous number of expensive technologies is required to achieve a proper tunning that allows to obtain sufficient accuracy in the forecast simulations. Such technologies comprises both observation or monitoring, and *in situ* sampling, e.g. satellites, HF radar, oceanographic buoys, among others.

Apart from the main world ocean currents, shown in Figure 1.5, the Ocean Model's solution must recover the existence of common ocean current structures. The world ocean currents might dominate the net flow in a particular region, but we also find a composition of eddies in general. Such eddies range in diameter from centimeters to hundreds of kilometers, and they may last a few seconds for the former, while the latter may persist several months. In oceanography, those eddies of large diameter and a persistence of days to months are known as Mesoscale eddies. In Figure 1.4 we can see several Mesoscale eddies resolved by the operational

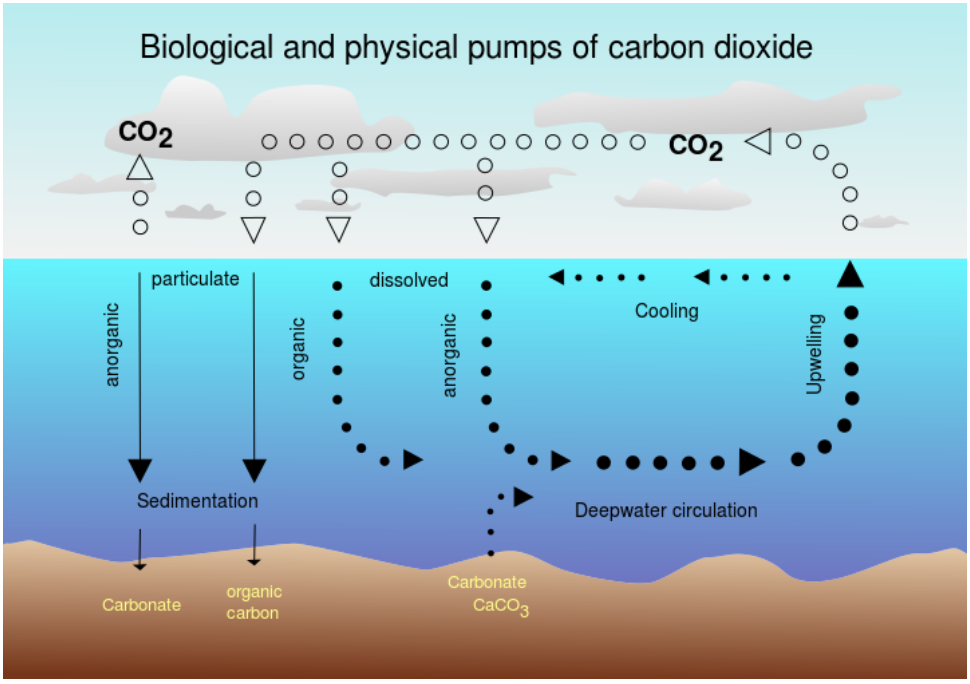


Figure 1.3: Biological and Physical (vertical) pumps of carbon dioxide, that operate in the ocean. Diagram by Hannes Grobe, from the Alfred Wegener Institute for Polar and Marine Research, Bremerhaven, Germany.

Iberian-Biscay-Ireland (IBI) ROM in the region of the Canary Islands. Also, in order to obtain an accurate modeling of ocean currents and conditions, river and fresh water discharges must be covered as well. River discharges produce plumes in the coastal ocean waters and provide a high contribution of sediments. Hence, both eddies and plumes must be accurately modeled in order to study a number of associated processes in the oceans.

Data Assimilation is the process by which observations or measurements are incorporated into a computer model of a real system. This process is critical to obtain a good estimation of the ocean state, and hence it improves the accuracy of the forecast outputs. In brief, observations of the current —and also past— state of a system are combined with the forecast to produce an analysis, which is considered as *the best* estimate of the current state of the system. Essentially, this analysis step tries to balance the uncertainty in the data and in the forecast. The model is then advanced in time and its result becomes the forecast in the next analysis cycle. In weather forecasting there are two main types of data assimilation (Ide et al., 1997). The first data assimilation methods where 3 dimensional (3DDA), in which only those observations available at the time of analysis are used. The other type of data assimilation is 4 dimensional (4DDA), in which the future observations are included, i.e. the time dimension is added. Consequently, several institutions offer products that approximate with relatively high accuracy the present and future state of the ocean, i.e. the nowcast and forecasts. This topic constitutes a core task in ocean research, usually known as Operational Research. It is also a capacitating element for other studies like path planning for marine vehicles, tracking evolving structures like eddies, transport and diffusion of contaminants, etc.

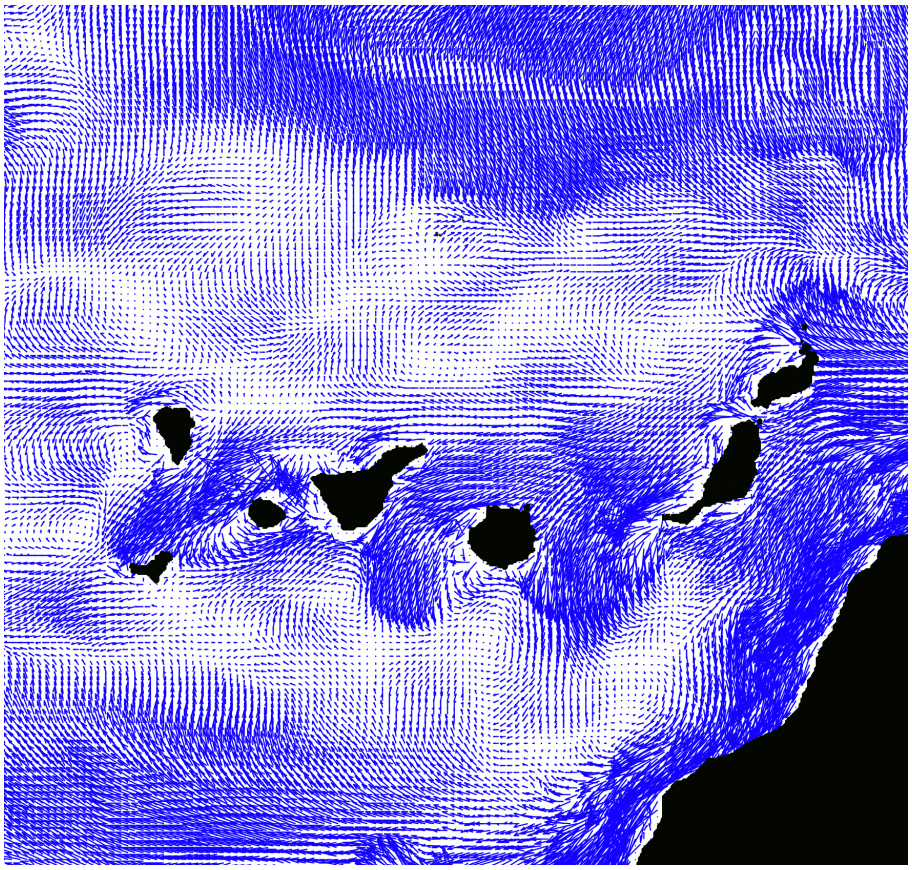


Figure 1.4: Ocean Currents forecast in the region of the Canary Islands, computed with the Regional Ocean Model (ROM) of MyOcean project for the Iberian-Biscay-Ireland (IBI) region. We can observer the presence of Mesoscale ocean eddies at the South of the islands.

1.2 Ocean Monitoring and Sampling

Consequently, in order to study and model the ocean from the surface to the seafloor, monitoring and sampling techniques are required. Since the successful launch on 27 June 1978 of SEASAT satellite by NASA's Jet Propulsion Laboratory (JPL), satellite monitoring have been an extremely powerful technique to observe the state of the Earth, and in particular the ocean. SEASAT was the first Earth-orbiting satellite designed for remote sensing of the Earth's oceans (Fu and Holt, 1982). As many modern oceanographic satellites, it was equipped with a RADAR altimeter, a microwave scatterometer, a multichannel microwave radiometer, a visible and infrared radiometer and a Synthetic-Aperture RADAR (SAR), as shown in Figure 1.6. Such instruments allow to remotely sense several oceanographic phenomena. For instance, the Sea Surface Height (SSH) or ocean surface topography, which is the height above the sea surface, is measured with the RADAR altimeter; the microwave scatterometer allows to measure the sea surface winds; the microwave radiometer can measure the Sea Surface Temperature (SST), as Figure 1.7 illustrates; meanwhile the visible and infrared radiometer identifies clouds, land and water features; and finally, the SAR can measure wave heights and sea ice conditions.

Among the most common instruments on board, most Earth-observing satellites are equipped with devices that rely on spectrophotometry and interferometry. The Moderate-Resolution Imaging Spectroradiometer

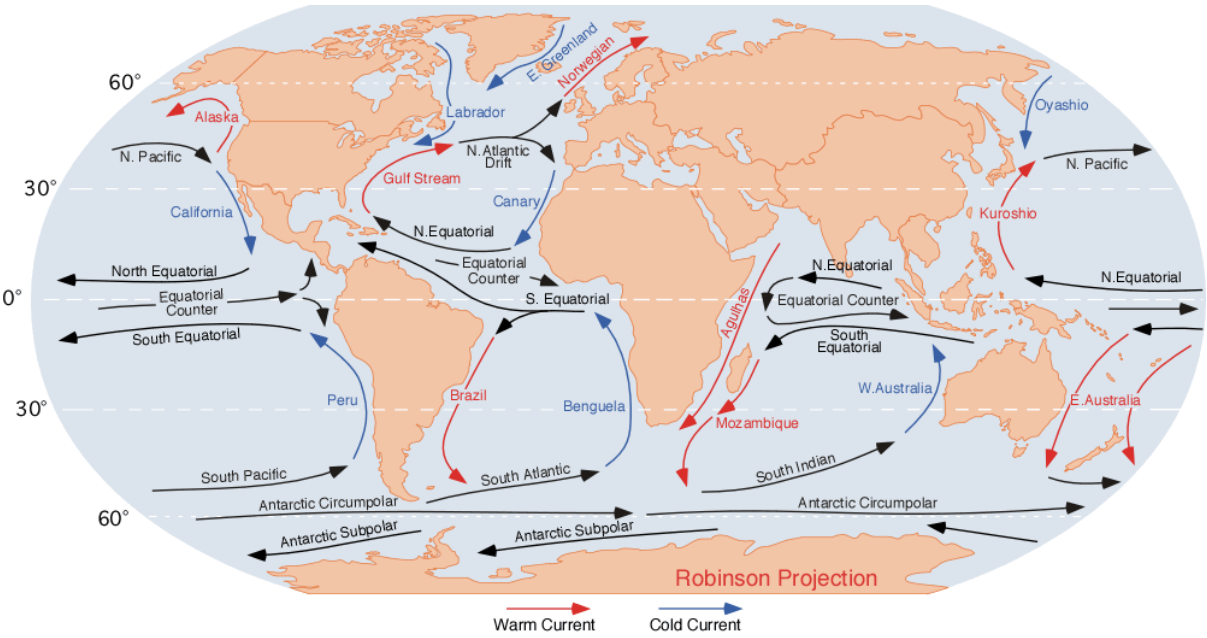


Figure 1.5: World Ocean Currents. Image made by Dr. Michael Pidwirny.

(MODIS) instrument, which is on board TERRA satellite, observes up to 36 spectral bands, including the visible spectrum as shown in Figure 1.8. These instruments allows to detect —after some calibration using data samples from the monitored area— visible objects or elements like oil spills, biological organisms or chemicals properties like phytoplankton, chlorophyll A, among others, as with the Sea-viewing Wide Field-of-view Sensor (SeaWiFS) instrument on board the SeaStar satellite launched in 1997. These instruments are not only used in spacecraft systems, but also go on board aircrafts. Although a geo-localization and an Attitude Heading Reference System (AHRS) are required to retrieve the aircraft position and attitude accurately, it is possible to use higher resolution instruments because we can get much closer to the water surface; for instance, these vehicles usually carry spectrometers that sample a greater number of frequency bands than satellites. Last but not least, the Coastal Ocean Dynamics Applications RADAR (CODAR) is a type of High Frequency (HF) RADAR that permits to measure surface ocean currents in coastal waters in almost real time, as well as waves heights and estimate local wind direction. Developed between 1973 and 1983 at NOAA's Wave Propagation Laboratory (Barrick et al., 1977), modern CODAR can measure out to 100–$200\mathrm{km}$ offshore with a rough resolution of 3–$12\mathrm{km}$, but its main advantage over spacecraft and aircraft is that using high frequencies, it achieves resolutions as fine as 200–$500\mathrm{m}$ —at the expense of shorten range around 15–$20\mathrm{km}$—, always in real time.

However, all the techniques based on remote sensing, as the ones discussed so far, have an important drawback. They cannot observe the state of the ocean and the life it supports beneath the surface; at most, they can see only a few meters depth with good visibility conditions. Both the water itself and Suspended Particulate Matter (SPM) are responsible of that fact. First, most of the sensors on board a spacecraft or an aircraft cannot penetrate more than a few meters ($\sim 20\mathrm{m}$) in the water column, due to the high attenuation of electromagnetic signals in water; this happens because of water's high permitivity and electrical conductivity, so plane wave attenuation is higher than in the air (Rhodes, 2006). Secondly, particles suspended in the water reflects the signal and does not allow it to go deeper; this phenomenon is known as backscattering, and it produces a diffuse reflection of the signal, which consequently goes back to the source. Also, some physical, biological or chemical properties of the ocean are not observable with remote sensing instruments, or at least

they are difficult to observe directly, like salinity, dissolved O_2, etc. Similarly, in many cases such instruments required a calibration phase using samples taken from the observed region; typically because they estimate the property indirectly, they perform an average or simply as a consequence of the noise introduced by the several atmosphere layers. Consequently, *in situ* data sampling is required to study water processes along the vertical, i.e. under the surface —in particular, at relatively large depths. The importance of sampling at different depths, beyond the ocean surface, resides on the fact that it is crucial to understand and characterize a great and important number of oceanographic processes. Most of these processes have been briefly discussed previously, which can be physical like upwelling, mesoscale ocean eddies, plumes produced by river discharges, or biological like the vertical pump of CO_2, stratification of the ecosystem, study of phytoplankton and vertical transport of organisms and organic components, among others.

In Figure 1.9 we have most usual classical sampling technologies. They range from arrays of sensors known as rosettes to weather buoys equipped with several sensors and even Research Vessels (R/Vs) that can carry large and extremely accurate instruments. An important advantage of sampling over remote sensing is the higher resolution and the better quality of the measurements; indeed, instruments on board a spacecraft are calibrated using samples taken from the monitored region. Apart from the old techniques used for data sampling in the oceans —as hemp ropes with a weight to measure the depth, or bottles with papers to estimate the ocean currents—, the first known proposal for surface weather observations at sea occurred in connection with aviation in 1927, with the deployment of weather stations along the ocean to assist seaplanes flights (Lee Dowd Jr., 1927). Nowadays, we can find many types of buoys used to measure a great variety of properties of the ocean surface. Basically, we have moorings, of large size and equipped with accurate

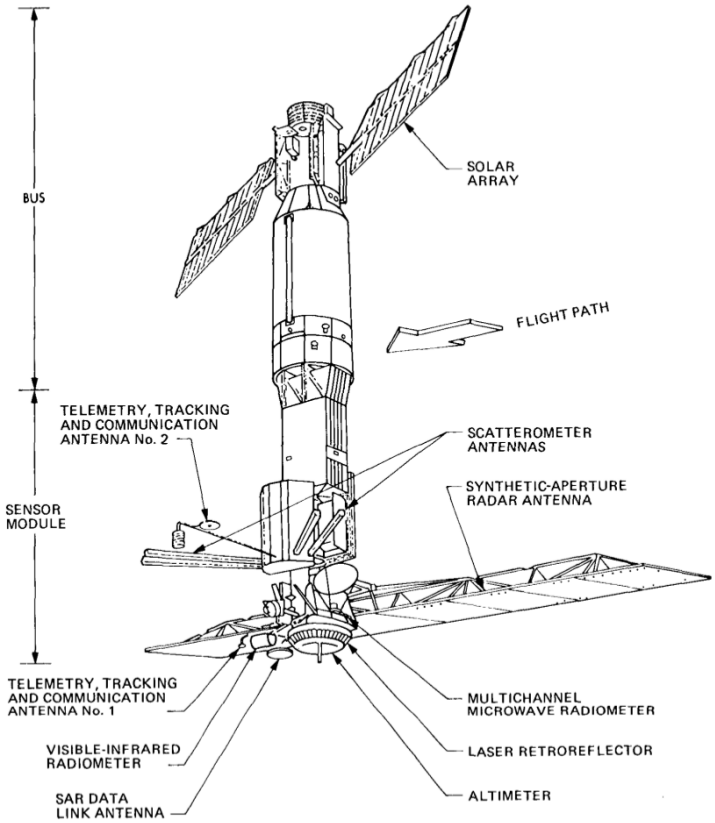


Figure 1.6: SEASAT bus and payload configuration. Image taken from Fu and Holt (1982).

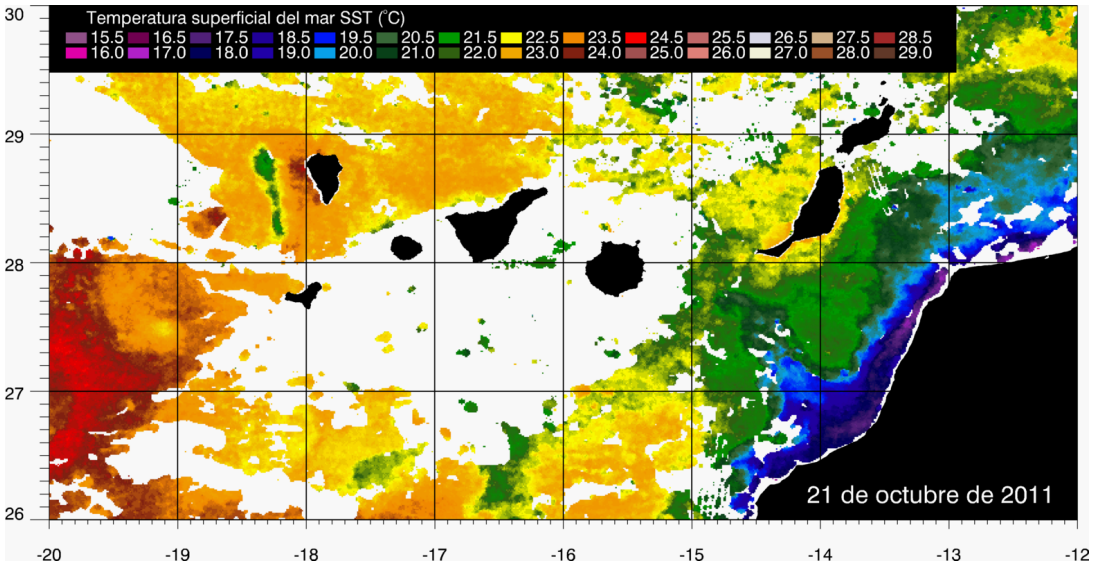


Figure 1.7: Sea Surface Temperature (SST) in the region of the Canary Islands, obtained with satellite monitoring systems for the 21 October 2011. SST image generated by the Division of Robotics and Operational Oceanography (DROC).

instruments, and (Lagrangian) drifters of smaller size and lower cost, which move with the ocean currents. Oceanographic buoys are like weather stations: they measure atmosphere properties like air temperature, barometric pressure, and wind speed and direction, as well as water properties like its temperature and salinity, wave height and dominant period, etc. Some drifters can also submerge and take samples underwater, which are later communicated when the buoy wakes at the surface again. ARGO, a global array of 3566[1] free-drifting profiling floats that measures the temperature and salinity of the upper $2000\mathrm{m}$ of the ocean, constitutes a good application example. It allows continuous monitoring of the ocean that helps to study Climate Change and its regional impacts. The main advantages of buoys are the large autonomy and the relatively low cost. On the other hand, the most remarkable disadvantage is that they cannot move, so they only drift with the ocean currents, as well as the fact that most of them only sample the surface, and sensors are less accurate than the ones carried on R/Vs. On the contrary, a R/V can move to any given point and carry extremely accurate instruments —usually very heavy, large and expensive—, and also resources or vehicles to explore the depths. However, R/V campaigns are extremely expensive; for instance a single day of formation or research with the R/V "Sarmiento de Gamboa", property of the Spanish research center CSIC (*Centro Superior de Investigaciones Científicas*), costs approximately $20000€$. Many institutions and governments invest on improving these technologies to obtain more efficient, accurate and cheaper devices, because of the pressing need to monitor the ocean, not only for Oceanography but also for military and civil applications, as ship transport.

1.3 Unmanned Underwater Vehicles

In recent decades, given the limitations of satellites, remote sensing instruments in general, and oceanographic buoys, the technological advances and the *know-how* learned have led to the development of Unmanned Underwater Vehicles (UUVs), which can move and take samples underwater with certain autonomy. Although

[1] Number of ARGO floats that have delivered data within the last 30 days, for March 10^{th}, 2013.

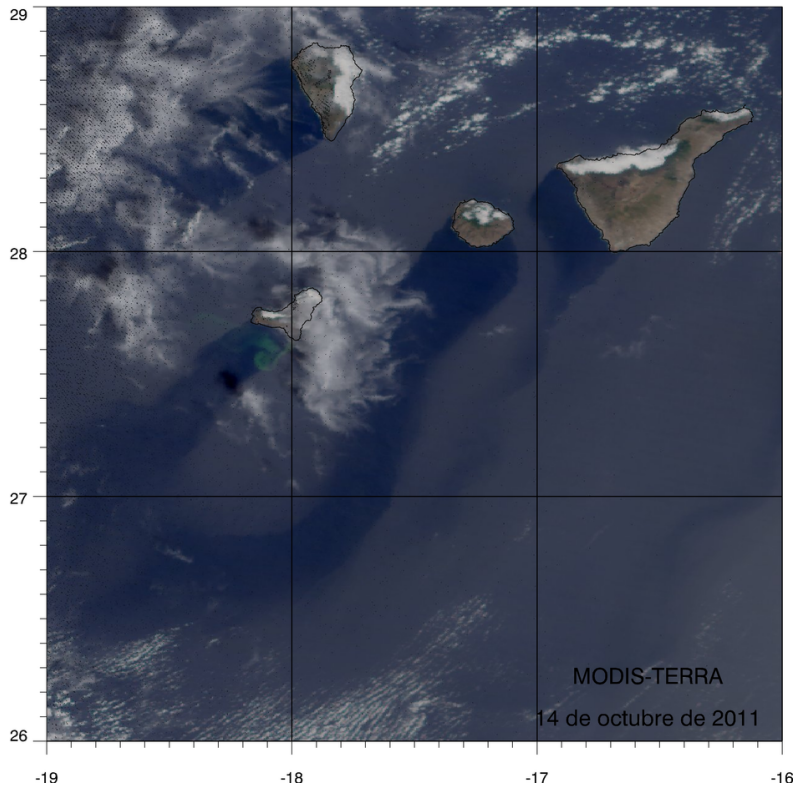
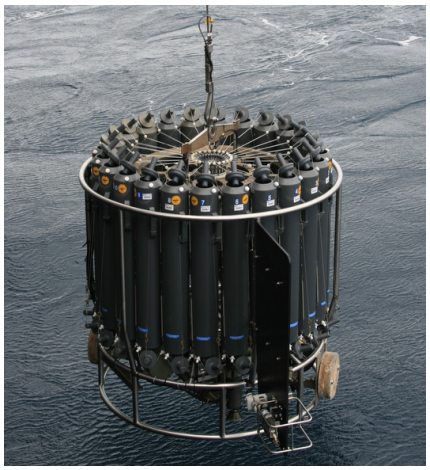


Figure 1.8: Visible band image of the Moderate-Resolution Imaging Spectroradiometer (MODIS) instrument on board TERRA satellite for the region of the Canary Islands on 14 October 2011.

there is a number of different types of UUVs depending on the target application and depth, for the sake of clarity here we mention the two main categories only; later, in Section 2.1 we will provide a more detailed taxonomy. First, we have Remotely Operated Vehicles (ROVs), which are tethered underwater vehicles linked to a ship by an umbilical cable, highly maneuverable and operated by a person aboard a vessel. Secondly, Autonomous Underwater Vehicles (AUVs) travel underwater without requiring input from an operator.

In the 1950s the Royal Navy used the ROV "Cutlet", shown in Figure 1.10, to recover practice torpedoes. Since this first ROV, a number of military and science applications have begun to be done more efficiently with this kind of vehicle. The US Navy uses ROVs for rescue operations and mainly as a mine countermeasure capability. On the other hand, the science community use ROVs to study the ocean. The main application is high quality video images retrieval, although additional sampling devices may be on board as well. ROVs rate from micro and mini vehicles to light or heavy workclass, which a propulsion of up to 220HP, the ability to carry several manipulators and a $\sim 3500\mathrm{m}$ depth rating. All of them are deployed from a surface vessel and remote controlled by a trained pilot. They allow to perform complex tasks underwater, so they can be used in place of professional divers in most scenarios. Such a capability allows for a large number of operations that are not possible using the classical sampling technologies seen so far. However, ROVs are still attached to a vessel and the pilot, so in this sense they are more a tool than a vehicle itself, which in the case of Ocean Research are used during R/V campaigns.

On the contrary, with AUVs we no longer need a remote operator —the ROV pilot— and a vessel for the deployment and recovery is usually unnecessary —or a cheap rubber boat may be enough. The first AUV was developed at the Applied Physics Laboratory at the University of Washington as early as 1957 (Widditsch,



(a) CTD-Rosette with CTD sensors and Niskin bottles.

(b) Buoy developed to calibrate the Soil Moisture and Ocean Salinity (SMOS) mission of the European Space Agency (ESA).

(c) Cutaway drawing of the R/V Hesperides, from the UTM-CSIC.

Figure 1.9: Classical Sampling Technologies, comprising: (a) rosettes with Conductivity Temperature Depth (CTD) sensors along with Niskin bottles to take samples of seawater; (b) weather (or oceanographic) buoys of any kind (moorings, floats, drifters); and (c) operations from a Research Vessel (R/V), like the deployment of CTD-rosettes, and data sampling with a great variety of accurate sensors. The buoy in (b) was built —with our collaboration on the remote and embedded control software— in 2008 at the ULPGC for the MIDAS-4 National Project to measure temperature and salinity with a SeaBird SBE 37-SIP, attached at the bottom of the buoy (partly shown in the figure).

1973). This pioneer AUV named Special Purpose Underwater Research Vehicle (SPURV) was used to study diffusion, acoustic transmission, and submarine wakes (see Figure 1.11). Now, the vehicle moves completely autonomously while it executes the mission, programmed and configured on surface before the deployment, until the recovery, which are the two only stages that required external intervention —from an human operator or another vehicle. It is also possible to communicate with the vehicle, specially when it surfaces, allowing to

Figure 1.10: The pioneer Cutlet ROV developed by the Royal Navy (the principal naval warfare service branch of the British Armed Forces) and first used in the 1950s to retrieve practice torpedos and mines.

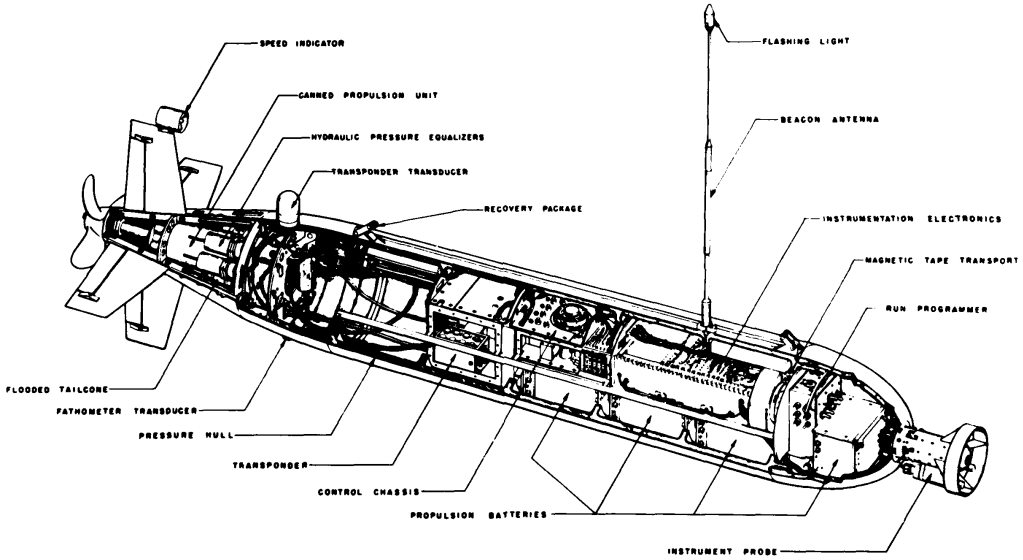


Figure 1.11: Self-Propelled Underwater Research Vehicle (SPURV) AUV. Cutaway drawing taken from Widditsch (1973).

retrieve mission data, check the system status, and even change the mission to some extent. Applications of AUVs are found in the oil and gas industry, military missions and ocean research. A survey of the seabed with an AUV is nowadays the most efficient way to obtain a map of the place where oil platforms or infrastructure is to be built; military missions with AUVs include underwater mine detection, inspection of protected areas and anti-submarine warfare; scientists used AUVs equipped with many sensors to study the ocean and the sea floor. However, with these moving and autonomous vehicles arises a number of issues that demand an intelligent management on the resources on board. There are three important problems that engineers try to alleviate or cope with: the autonomy, specially in comparison with buoys, which depends on the battery hotel and sensor payload, and does not last more than a day in the best cases; the difficulty to communicate

underwater using acoustic modems, which only allow for a low bandwidth and short distance, apart from other limitations; and localization, since the sensors on board are not precise enough for long term dead reckoning and an underwater acoustic positioning system like the Ultra-Short BaseLine (USBL) incurs an additional cost and the presence of a vessel at surface. On the other hand, the advantage of AUVs is that they cost much less than renting a R/V for a single day, and still can carry multiple sensors, although they may be less accurate and have lower resolution. ROVs and AUVs can move along the whole volume of water and perform complex maneuvers and tasks, and are far safer and cheaper than a bathyscaphe with a pilot or scientist on board, or even professional and scientific divers.

1.3.1 Autonomous Underwater Gliders

It was in 1978 when the first idea and applicability of a vehicle that steer to move slow consuming low power was envisioned by Doug Webb for SOund Fixing And Ranging (SOFAR) floats at small speeds such as $10\mathrm{km/day}$, in a Woods Hole Oceanographic Institution (WHOI) memo (Regier and Stommel, 1978). Later, in November 1991, an advanced, autopilot-controlled glider was designed and successfully tested at Seneca Lake (New York); it was the precursor of the modern SLOCUM gliders of Teledyne Webb Research and Bluefin's Spray gliders, as well as others like the Seaglider of iRobot, all of them shown together in Figure 1.12. The second original prototype used a hydraulic pump to inflate an external bladder as the means of changing buoyancy for ascent/descent, gliding along a saw-tooth trajectory underwater (Simonetti, 1992). In this second test, the average horizontal glide speed was improved to $0.20\mathrm{m/s}$, as well as the turn radius, which was of 7 – $13\mathrm{m}$. Small changes in its buoyancy and the center of mass, in conjunction with the wings allow gliders to convert vertical motion into horizontal, so they navigate gliding forward with zero power consumption. The idea of a flexible bladder to change the vehicle buoyancy resembles the swim bladder of lungfish (Hass, 1994) (see Figure 1.13). This system reduces dramatically the power consumption required for moving, since the pump that changes the buoyancy only has to be actuated a reduced number of times, at the inflexion points in the yo-yo profile performed by the glider. The yo-yo stands for the up-down vertical movement produced by the buoyancy change. This provides gliders with a large autonomy —in the order of months, depending on the scientific equipment—, and it is for this reason they have been extensively applied to Ocean Research in recent years (Rudnick et al., 2004).

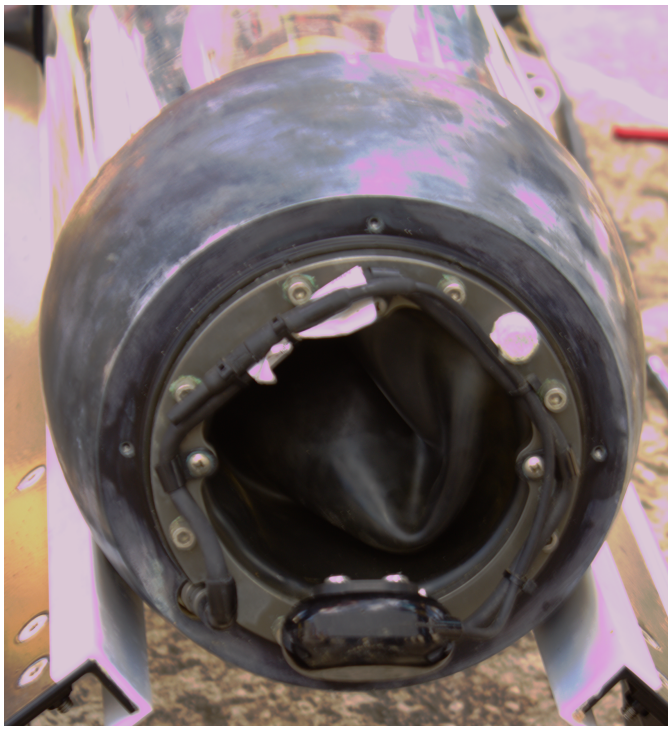
Along with the designed and field tests of the original gliders in 1991 by Stommel, also the specification of a 5-year endurance, $0.28\mathrm{m/s}$ ($> 1\mathrm{km/h}$) horizontal speed ocean glider with and ocean thermocline-driven buoyancy change engine was presented (Simonetti, 1992). Nowadays, although thermal gliders have been developed, most common gliders use an electrical engine. The first commercial sale of a glider happened relatively recently, in October 2001, at WHOI (Donaldson, 2007). They have become a very efficient way to obtain depth profiles of temperature, salinity or other ocean properties for long periods of time and around a given region or path. The large autonomy of ocean gliders beat AUVs, but they navigate at low speed and with less maneuverability. Consequently, the effect of the ocean currents turns to be considerable in the resulting trajectory followed by the glider; ocean currents range from 0 to 1 – $2\mathrm{m/s}$ generally, so they are usually close to the glider's horizontal speed or even faster. They produce a drift in the trajectory, which introduce a significant uncertainty on the glider's location while underwater. Only when the glider surfaces and obtains a Global Positioning System (GPS) fix —or from any other Global Navigation Satellite System (GNSS)—, it is able to establish its actual location —and the average ocean current for the last stint (underwater flight) as well.

1.4 Autonomous Underwater Glider Piloting and Path Planning

Three main tasks have to be done by human operators during a glider mission. In chronological order: deployment, piloting and recovery. The deployment and recovery are isolated procedures, which are done only once in a mission, when it starts and at the end, respectively. All the operations required to put a glider

Figure 1.12: Spray from Bluefin Robotics (front), SLOCUM from Teledyne Webb Research (middle) and Seaglider from iRobot (back), at the Gliders Laboratory in the installations of the Oceanic Platform of the Canary Islands (PLOCAN). Courtesy of PLOCAN. Please note that wings are removable to facilitate shipping and maintenance, and for that reason they have been removed from the hull in the vehicles shown here.

on the water, and all the checks to verify it is working properly, are part of the deployment. Similarly, the recovery comprises the operations to retrieve the vehicle from the water. Both are physically done from the surface using a vessel or a rubber boat. During the mission, however, the glider operates autonomously, but still requires some sort of piloting. The reason falls on the fact that ocean currents may be faster than the glider's. Hence, it can get trapped in a region of strong currents while trying to reach the current target waypoint. The basic control algorithm running in the glider's processor —a low power Persistor in the case of SLOCUM gliders, with low computational power— is not enough to find a path to the target in a reasonable time; it computes the dead reckoning using the depth and attitude (pitch and roll) sensors and incorporates a module for ocean current estimation to correct the vehicle bearing. Therefore, a glider pilot can change the current target waypoint temporarily to avoid strong opposite currents and then go to the desired location. In this thesis we will refer to manual piloting when a human communicates with the vehicle to change the mission parameters, in particular for the navigation subsystem, and manipulates the waypoints list to reach a given region or follow a certain trajectory, in accordance with the ocean currents that may interfere on the glider's motion. The SLOCUM Coastal Electric glider is commanded with user-generated missions executed from the glider shell named GliderDOS —a subset of picoDOS, which itself is a subset of the Disk Operating System (DOS). Mission files are composed of behaviors —with arguments— and sensor values. In SLOCUM, the `goto_list` behavior establishes the list of waypoints to reach and how to traverse it (see Kerfoot and Aragon, 2010). If required, the pilot can change the waypoint list and some arguments in a *mafile* —a file with the waypoints specification for the mission—, and re-compile the mission without exiting it. At the time of writing, the pilots have to apply these changes and re-compile the mission manually, from the Dockserver terminal that allows to communicate with the SLOCUM glider. Also, it is the pilot itself who analyzes the



(a) Bow nose of the SLOCUM glider with the bladder, of black rubber, partly empty.

(b) Swim bladder of a Rudd fish.

Figure 1.13: Analogy of the bio-inspired bladder that controls the glider buoyancy.

ocean currents maps, which may include forecasts computed with the Ocean Models discussed so far, and intelligently finds the list of waypoints required to reach a particular region, despite of the ocean currents. The RU27 glider was piloted this way in the trans-Atlantic mission, which final trajectory is shown in Figure 1.14.

A good glider pilot must have some knowledge of the ocean behavior, its structure and dynamics, apart from the technical details of the glider navigation and control system. Therefore, in order to analyze ocean currents forecasts appropriately, some oceanographic background is demanded. But not only that, since the temporal variability of ocean currents is high, particularly near the coast and at sub-mesoscale level, it is possible to exploit such information, as we explore in this thesis. However, for a pilot this task becomes quite complex under such conditions, because of the great volume of data and the great difficulty to establish the optimal path when currents vary drastically in short periods of time —e.g. in hours. In any case, a path planner applied to pilot gliders using dynamic ocean currents maps produced by Ocean Models, is always useful because it allows to automate the piloting. Also, in those regions of high variability and complexity, path planning algorithms can find a feasible path to the target waypoint, with the warranty of optimality, or

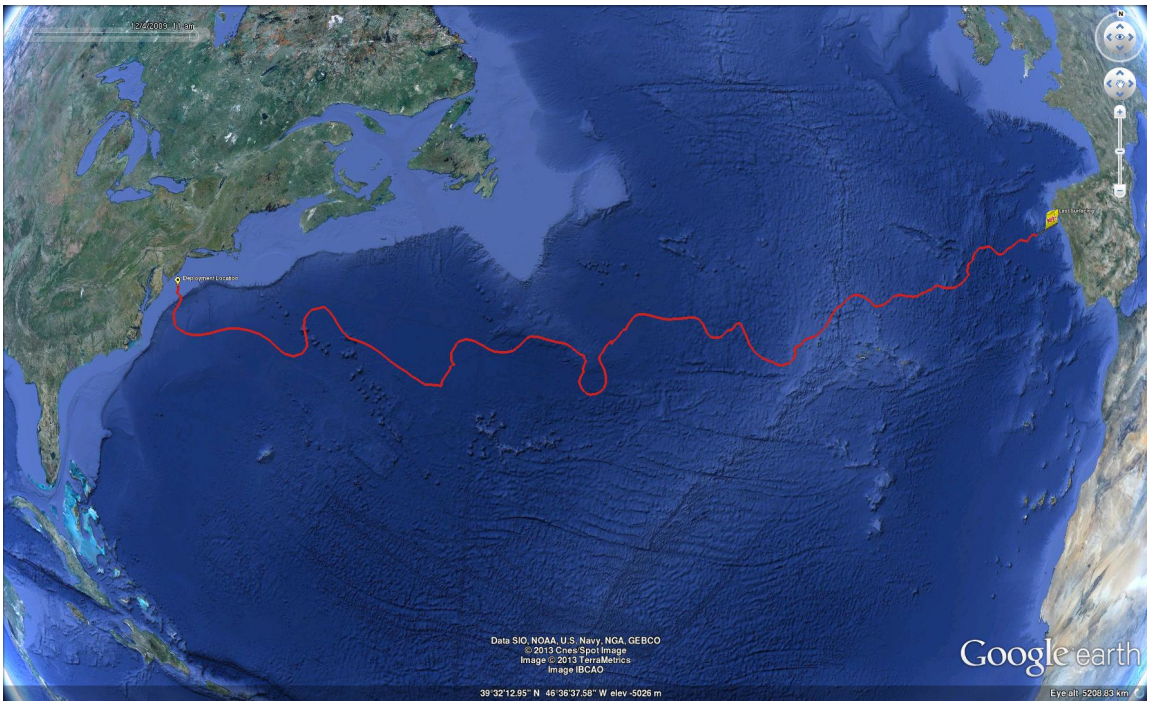


Figure 1.14: Path followed by the RU27 *Scarlet Knight* (SLOCUM) glider in the trans-Atlantic mission, accomplished in 2010. Mission headed by the Rutgers University, with the collaboration of the Division of Robotics and Computational Oceanography (DROC), SIANI.

at least it will be close to the optimal. In sum, the main advantages of glider path planning with respect to traditional manual piloting are the ability to consider all the high temporal dynamics of the ocean currents —as computed in Ocean Models forecasts— to obtain feasible and nearly optimal paths to the target waypoint automatically.

Path planning techniques applied to glider piloting have a great potential for various reasons. First, it is a key element for true automatic, intelligent and efficient piloting. In the case of SLOCUM gliders, some sort of automatic waypoint modification is required, since the current software does not support it, and still requires the intervention of the pilot. The integration of such tools and a path planner would allow for less human-assisted and more autonomous glider missions to be accomplished at the ocean. Secondly, when tracking evolving features or structures like eddies, path planning techniques can incorporate the logic to obtain adaptive sampling paths that help to characterize the phenomenon. Such path planner would take into account the temporal evolution and variability or dynamics of the structure, which is very difficult to manage manually by a glider pilot. And last but not least, gliders can be considered Lagrangian driven drifters that can contribute data samples to improve Ocean Models by means of data assimilation.

This thesis covers several problems —with a number of associated applications— that can be solved with path planning techniques. The most basic problem tries to find the optimal path from a start waypoint A to a target one B, which is classical path or motion planning problem in Robotics. The classical solution is the A* algorithm (Hart et al., 1968) that, based on Dijkstra's algorithm (Dijkstra, 1959), obtain better performance by means of using heuristics. It also allows for obstacle avoidance and there are many variants that deal with extensions of the original problem. However, even for the basic problem, when applied to ocean gliders, even the basic A* must be adapted to accommodate the particularities of glider path planning: in brief, the glider navigation scheme, and the effect of the ocean currents, which are function of the space (x, y, z) and time

t. In this dissertation, apart from the basic path planning problem, we also cover path planning for multiple vehicles with coordination constraints, path following and moving targets or evolving ocean structures like ocean eddies. Several state-of-the-art techniques have been extended to support glider path planning and novel approaches are also proposed to capture the particularities of glider's navigation. The algorithms analyzed range from graph-based techniques as A*, to probabilistic sampling methods like Rapidly-Exploring Random Trees (RRTs) (Lavalle, 1998), and optimization approaches like Genetic Algorithms (GA), Simulated Annealing (SA) and Iterative Optimization. We have tried to design and implement all the algorithms in this work with the glider navigation pattern in mind, because here we focus on contributions to the particular field of glider path planning. However, some proposals may be applicable to similar planning problems, which lie under a category that can be defined as path planning in variable asymmetric cost environments, although still with some peculiarities, as we will later discuss.

1.5 Thesis Hypothesis

This PhD work delves in the following thesis statement:

> Novel Path Planning algorithms designed for the problem of Autonomous Underwater Gliders (AUGs) navigation bring the possibility of computing (sub-)optimal paths automatically, and hence improve, simplify and, in some cases, enable missions in zones with strong ocean currents that exhibit a complex distribution or a high temporal variability.

Put differently, our hypothesis is that the current way of piloting ocean gliders can benefit from path planning algorithms. Such benefits will be evident and remarkable in certain scenarios, which are those of high temporal variability and strong ocean currents. Although the automation of glider piloting is of interest *per se*, a path planner can also help in the task of finding the optimal trajectory to reach a point, stay at a given position, among others tasks. Therefore, it can improve many factors of the mission —e.g. time, autonomy, safety, etc.— and what is best, it might turn into feasible some kind of missions that were particularly difficult to perform formerly.

Our intention is also to show that classical path planning techniques, and some of the concepts they are based on, can be exploited to adapt them or design novel approaches for ocean gliders. Indeed, we will introduce several methods that are inspired in graph-based methods, probabilistic search and optimization. This dissertation tries to show that these techniques manage the problem properly, since they consider the peculiarities of the problem, and hence are susceptible of being incorporated in real missions/conditions. Also, the experiments will allow to categorize the methods depending on the complexity of the scenario —obstacles, strong currents, high temporal dynamics, etc. Finally, a special effort is made to reduce the computational time of our techniques, so they can be run during the short time intervals the gliders are at surface to localize themselves and communicate. The path planners are therefore able to find a path within this time, and the resulting bearing command can be uploaded to the vehicle before it submerges again.

Similarly, we explore the applicability of these techniques to concomitant problems. This is the case of multi-glider coordination, obstacle avoidance, navigation under constraints, path following, area surveillance, tracking evolving ocean structures like ocean eddies, among others. Then, we discuss how these methods can be extended to support other problems, and further generalization concerns.

1.6 Limitations and Scope

In this thesis we try to answer the following questions, which together define the scope and extension of the present work.

- In the field of Robotics, the Path Planning literature gathers an important number of algorithms that are generally used to solve the problem of finding the (quasi-)optimal sequence of state configurations that allows a robot to move from one pose or position to another. It is of great interest to study whether these methods are applicable in the field of gliders path planning or not. For instance, is A* search applicable? Can it be adapted to deal with the problem? or is there some particularity that prevents it to be done, or does it turn to be an inefficient approach?
- We will see that classical or common path planning methods are not directly applicable to the problem at hand, unless they are adapted —sometimes significantly. At this point, we will try to answer which are the particularities of path planning for Autonomous Underwater Gliders (AUGs) that justify such need for adaptation. Consequently, it will be possible to give some insight to the following questions:
  - Which are the differences with respect to other problems that are solved with classical path planners? Here, we refer to problems like path planning in mazes, cost maps —e.g. relief maps—, etc.
  - What reasons explain why classical techniques are not applicable or useful in this case? Or what impede them to manage the problem efficiently?
- The study of the literature and the state-of-the-art should allow us to find out which kind of problem we are trying to solve, i.e. within which category is taxonomically classified by other authors, or at least to establish the most similar.
- From a more pragmatic point of view, we want to investigate and engineer several methods to address the problem. In the present work we will implement several novel techniques to give some evaluative assessment for a representative sample of glider path planning scenarios. This assessment will be twofold. First, evaluate the optimality of the solution or how far from the optimum is it —or how much does it improve other state-of-the-art methods. Secondly, the computation time and the usage of memory resources.
- In connection with the previous question, we will be able to compare our results with other approaches that might already exist in the literature of this topic. We could answer then, how other authors deal with the problem and which simplifications and assumptions do they make. And hence, we will try to see whether the algorithms designed for glider path planning scale and are generalizable or not, i.e. if they can be apply to more general path planning problems.
- Regarding the dimensionality of the problem, we will see that the ocean modeled processes have 4 dimensions from the point of view of glider path planning, neglecting the vehicle configuration and dynamics. Being underwater vehicles, gliders navigate through the three-dimensional water mass, which also moves significantly over time. Also, we could consider, or just neglect, some characteristics of the vehicle dynamics and the navigation, as the gliding angle, the minimum and maximum depths of the yo-yo profiles, the nominal speed, and so on. How to manage all this complexity and high dimensionality of the problem efficiently is also a concern of the present work. Similarly, we want to see whether it is possible to ensure the optimality of the solution found, or just to some extent and under certain assumptions. And also, analyze in which cases is more difficult to ensure such optimality.
- From the implementation and evaluation of different techniques we will obtain some intuition of which ones are more suitable for certain conditions or scenarios. Some categorization of the scenarios will reveal as well, e.g. clutter areas that demand obstacle avoidance logic, high temporal variability zones, strong currents, etc.

The questions above define the scope of the present thesis, since it tries to answer each of them. But it is also important to put clear the limitations or assumptions made in this work, which are there to reduce the complexity of the problem and tackle it adequately.

- Regarding the high dimensionality of the problem, we will have to reduce it to some extent. For this reason, we will assume a simplified motion model for the glider. Also, in some cases path planning will be done considering 2D surface currents, hourly spaced in time, while in others we will take the 3D ocean currents, although they will be daily means. We will see that this reduction in the dimensions of the problem is often acceptable, while it reduces the complexity substantially.

- Some of the particularities of typical glider missions and piloting are neglected often. In some cases it is not possible to observe some phenomena, like biofouling, which affects the glider speed and trimming. As a consequence, the glide angle, and even roll change, and it might produce some bias in the heading. Similarly, the weight, the mass center and the glider's buoyancy varies, with an important reduction on the glider nominal speed. In others we can only estimate with some error a certain parameter, like the time the vehicle stays at surface. Also, sometimes we do not model such aspects because we want to analyze the rest of the system, ignoring their effect. Indeed, we have implemented simple models for them, although it is still difficult to obtain the real values of a given mission in advance, before path planning.

- The embedded systems and software that state-of-the-art gliders have on board is still very restricted in capabilities. Hence, in practice, automatic piloting is not straightforward. Although not strictly a limitation, we have covered this fact in our simulations. Similarly, since the quality of the dead-reckoning is quite bad, communications are not possible, and the drift produced by ocean currents is not observable while submerged, a path planning approach with discrete commands widely separated on time is required. If that not were the case, we could formulate a different path planning scheme, in which we could vary the vehicle heading —or even other navigation parameters— more frequently. However, even in that case, the current vehicles would not be able toi have the ocean currents maps on board to update its heading accordingly, neither the computational power to handle them. Path planning would be of interest even in such case, hence.

1.7 Thesis Outline

Before we describe the outline of this dissertation, it is worth mentioning that this thesis is not about developing the differential equations of ocean models, neither the initialization and resolution of them. Here we simply take their results as inputs. We will call such results ocean model products, and we will describe the most common ones, what they represent, and what are they useful for, as well as their connection with path planning methods later in Appendix A. In brief, it will give an introductory overview of ocean modeling and data assimilation. Also, the discretization resolution of the output is relevant when doing path planning, so we will introduce an ocean model taxonomy based on their scale, and will discuss the importance of Regional Ocean Models (ROMs) in coastal zones.

The present document is composed of three main chapters that explain and explore the field of Ocean Research with Autonomous Underwater Gliders, along with the results and contributions of this thesis to the area. The first two chapters begin with a review of the state-of-the-art at the time of writing this dissertation. Then, a thorough analysis of some techniques suitable for the problem is included and validated with extensive simulations and experiments, as well as field trials.

Given the introduction to the field of Ocean Research of this chapter, we will start the next chapter by discussing the sampling technologies based on underwater vehicles. We connect the dissertation with the field of Underwater Robotics, and hence with Autonomous Underwater Vehicles (AUVs). Autonomous Underwater Gliders (AUGs) are briefly described in the first sections, bringing into focus their tight relation with ocean sampling and the great versatility they incorporate into this topic. This is a consequence of the great autonomy and the ability to move of these sampling platforms, as a main difference with respect to oceanographic buoys or propelled AUVs.

In Chapter 2 we also delve into the details of AUGs, that we will indistinctly called ocean gliders or just gliders in the sequel, as well. Since path planning methods have to model the vehicle that is going to navigate the maps, we must study thoroughly the principles that govern the kinematics and —to some extent— the dynamics. We will describe how gliders operate on water, the phases of a typical oceanographic mission, and the communication and navigation mechanisms. Regarding the glider's dynamic model —which also depends on the particular vehicle and its setup—, we will show different alternatives that we have considered; a punctual model, a more elaborated force balance model, and even dynamic models.

At the time of writing this thesis, there exists some remarkable models of gliders that deserve to be mentioned and discussed in detail. We will dedicate some lines to the description of Teledyne Webb Research's SLOCUM glider, Bluefin's Spray glider[2] and iRobot's Seaglider[3]. It is also worth mentioning the great interest on this vehicles, which have boosted up several glider network initiatives, like the European Glider Observatory (EGO), organizations and universities that are actively working with gliders, like the National Oceanography Centre Southampton (NOCS), University of Southern California (USC), Queensland University of Technology (QUT), Balearic Islands Coastal Observing and Forecasting System (SOCIB), Oceanic Platform of the Canary Islands (PLOCAN), among others.

Ocean gliders are nowadays a successful tool for long-term oceanographic surveys, they are used to retrieve ocean parameters that help in climate change studies, the industry and the navy are using them for their daily tasks, and much more. These vehicles have become a proven technology to operate in the sea and a robust, efficient and relatively low-cost tool for many tasks. We will enumerate some of them and how they are accomplished. In particular, we will elaborate on the piloting of ocean gliders with the software capabilities and applications provided by the manufacturers, like SLOCUM's DockServer. We will introduce the concept of automatic piloting in this area, what advantages it incorporates to the missions, describing which systems have still to be developed, and how it turns into feasible some specific types of missions that would not be possible otherwise.

In Chapter 3 we give a list of glider path planning problems and applications. We will start with different objective functions, for the two atomic optimization problems we identify in the field of glider path planning, being the minimal time path and the minimal distance to the target problems. Multiple gliders and constraint path planning is covered, since it is of great interest to study large ocean structures at several points simultaneously, or just for a cross-validation of the vehicle sensory suite. Apart from this sort of coordination, we also include the problems of path following or hold-track, and tracking of evolving ocean structures. An exhaustive analysis of ocean eddies is included, with their description, techniques to detect and characterize them, and their relevance and connection with Harmful Algal Blooms (HABs). From the point of view of glider path planning this introduces a variety of approaches to assist in the process of finding the centroid and boundary of the eddy over time, which requires a search for the optimal navigation pattern around the structure.

We will finish these first chapters with the list of the path planning techniques developed and a thorough description and comparison of them in Chapter 4. At that point, we will start with an introduction to Path Planning in the area of Robotics and Artificial Intelligence, in order to settle down into the general problem, and finally cover the specific case of AUGs. The different approaches designed to tackle the problem will be described in detail. Also, other techniques are explained and reasons are given to show why they are difficult to adapt for gliders.

The experimental results derived from the application of the techniques developed in this thesis is covered in Chapter 5. The Regional Oceanic Models (ROMs) used, and the scenarios or zones considered in the experiments are listed and described in terms of their temporal variability, ocean currents strength, presence of obstacles or low bathymetry depth, among other relevant features. Similarly, the glider motion model employed —from the ones available— in the path planners is defined and how the dead-reckoning is computed inside the planner kernel.

[2]The Spray glider was originally developed at Scripps Institution of Oceanography (SIO) under Dr. Russ Davis.

[3]The Seaglider was originally developed at the Applied Physics Laboratory (APL) of the University of Washington.

The results obtained with several techniques —novel and classic— appear in this document sorted by problem or application. That is, for each application —e.g. go to a target, path following, obstacle avoidance, multi-glider coordination, etc.— we show the solution found with different algorithms, which are compared then. Finally, we also discuss the field tests performed, starting with the mission setup, to later explain the data acquired and evaluate the path planning methods in real conditions. Also, some sort of validation of our path planners and the input products —i.e. ocean currents maps— computed by ROMs is included.

Finally, the last chapter of this thesis discusses the contributions, puts clear the conclusions and enumerates the conference papers and journal publications derived from this work. Also, further work is proposed to continue the research explored here. In brief, this covers more experiments, including field tests. Apart from the main matter of this dissertation, a few appendices are included at the end to provide complementary information and details.

Chapter 2

Ocean Gliders

In the introduction we have justified the great potential of ocean gliders for Ocean Research. They constitute an effective vehicle to monitor large regions for long periods of time, at a relatively low-cost investment. It is the particular propulsion system they use what confers on them such a large autonomy to operate at the oceans. However, it is also the origin of the slow speed they can generate to move underwater. Since the speed of the ocean currents can be equivalent —or even greater— to such speed, the vehicle drifts while it moves, and an intelligent and planned strategy is required to reach certain locations at the sea. In the next chapters we will address this particular task and the problems or applications it can help to solve.

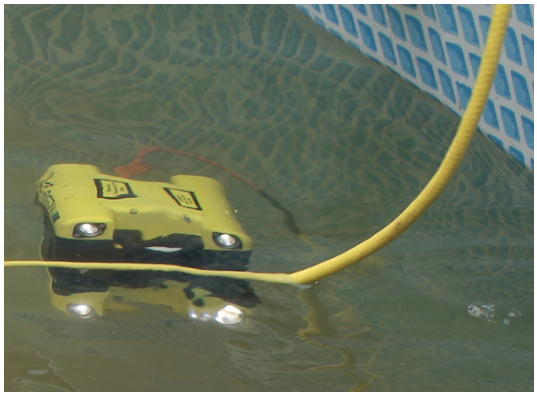
Before we described the most common glider path planning problems and the algorithms developed in this thesis, we will see here that the motion model of the vehicle plays an important role in the design of the path planner, particularly in the case of AUGs. For this reason, the next sections give a thorough description of ocean gliders, the piloting and navigation, and the mathematical formulation of different glider motion models. Three motion models are presented: a simple, but practical, point model; a force balance motion model; and references to dynamics motion models that, as we will see, are not required for path planning and, consequently, are out of the scope of this thesis.

2.1 Autonomous Underwater Gliders

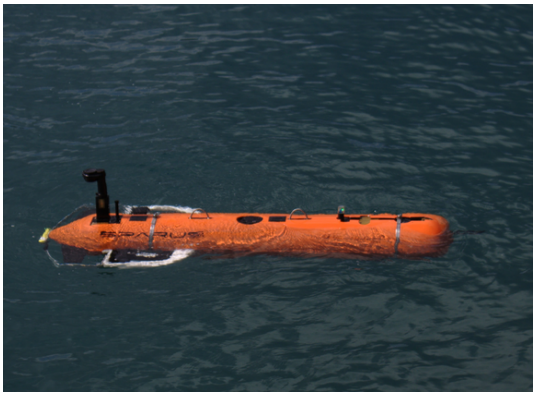
We have already described briefly ocean gliders in the introduction. Here we will give a more detailed explanation, from the perspective of Underwater Robotics. After a succinct categorization of underwater robots, we describe the principle of operation of ocean gliders. Later, this will be the basis of the glider motion models described in Section 2.2. These motion models are an essential element of the glider path planning algorithms developed in this thesis, since they will integrate the glider motion in order to evaluate the cost of traveling from one point to another.

In Underwater Robotics, we distinguishe different types of vehicles depending on their autonomy and target applications. On one side we find Remotely Operated Vehicles (**ROVs**), which are tele-operated vehicles that are connected to the surface vessel or pilot with an umbilical. The umbilical cable transmits the commands sent by the human operator, and sends back the data gathered underwater; it also provides power to the vehicle. On the other side, we have truly Autonomous Underwater Vehicles (**AUVs**), which navigate and perform some tasks autonomously. Both ROVs and AUVs are known as Unmanned Underwater Vehicles (UUVs), since there is no human on-board, reducing the risk, cost and size of the vessel. Two representative examples of ROVs and AUVs are shown in Figure 2.1 (a) and (b), respectively.

Depending on the target application, we find different types of AUVs. Generally, they are propelled by several thrusters, so their autonomy is limited to a few hours —e.g. $100\mathrm{h}$ at $4\mathrm{knot}$ for the Hugin 1000 AUV (Konsberg, 2013). However, we can distinguish between cruising and intervention vehicles. Torpedo-shape



(a) AC ROV.



(b) SPARUS AUV.

Figure 2.1: Examples of Unmanned Underwater Vehicles (UUVs): (a) AC ROV property of QSTAR, and (b) SPARUS AUV designed by the *Centre d'Investigació en Robòtica Submarina (CIRS)*. Images taken at (a) FIMAR 2013 and (b) the 2nd Field Training School in Autonomous Underwater Robotics Intervention.

AUVs operate offshore in long-range cruising or survey missions, while intervention vehicles can also perform tasks in confined environments and even do some sort of manipulation (Ribas et al., 2012). Autonomous Underwater Gliders (AUGs) lie within the first category, although they are not propelled vehicles. They incorporate an efficient propulsion system that allows them consume very low energy to navigate in the ocean, so the autonomy is increased to several months.

2.1.1 Description

AUGs, or ocean gliders, propel themselves by changing their buoyancy and using wings to produce forward motion. Here we follow the description of the principle of operation of gliders given in the work of Bishop (2008). The mechanism of propulsion is known as a **buoyancy engine**, which in the case of the SLOCUM glider (see Figure 2.2) can ingress or expel $\Delta V_{bp} = 250$cc of fluid. In Section 2.2.3 we will see the balance of forces in the vehicle and the resulting speed for ΔV_{bp} and other parameters like the pitch θ. In fact, a primary pitch control is achieved by placing the buoyancy engine in the nose cone of the vehicle. A more precise tuning of the pitch is given by the movement of the internal batteries.

Figure 2.2: SLOCUM glider designed by Teledyne Webb Research.

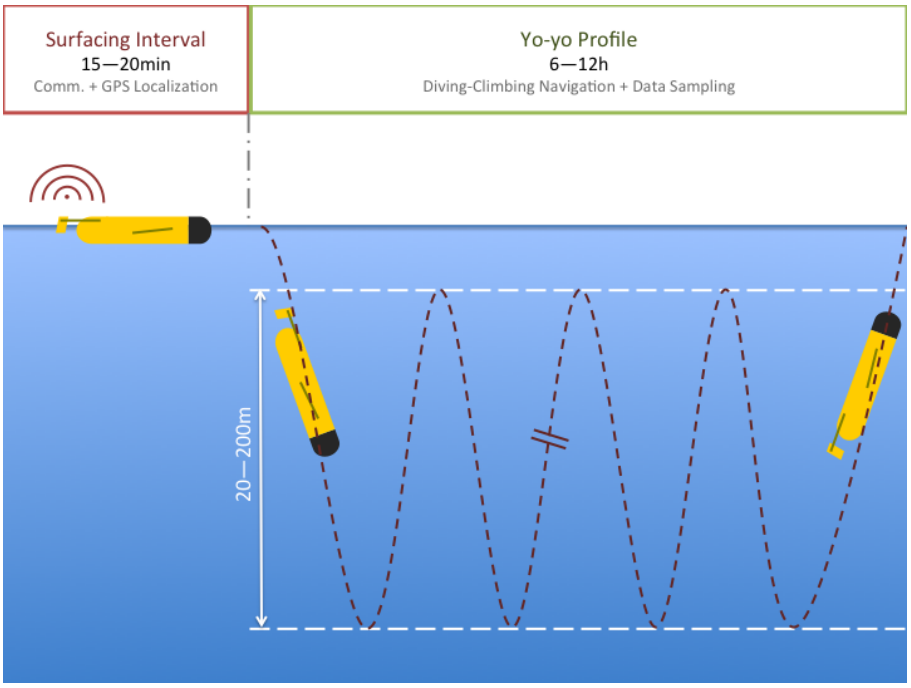


Figure 2.3: Diagram of the yo-yo profile and saw-tooth pattern followed by a glider while it navigates underwater using the oil bladder and pump. The most common depth interval for an underwater stint and the time spent at surface are given. Note that in the figure the dimensions are not proportional, and the pitch angle is exaggerated. The travel distance for a $6 - 12\mathrm{h}$ yo-yo profile is $\approx 10 - 20\mathrm{km}$, gliding at $U_g = 0.4\mathrm{m/s} = 1.44\mathrm{km/h}$, and the pitch angle is $\approx 35°$. Also, depending on the glider model, the depth interval can be $20 - 200\mathrm{m}$ or $20 - 1000\mathrm{m}$, in the most general case nowadays.

Basically, the pitch allows the wings and the hull to produce the hydrodynamic lift required to propel the glider horizontally, while it dives or climbs as a consequence of the change in buoyancy. As a result, the vehicle describes a vertical saw-tooth pattern while it moves, as shown in Figure 2.3. Each cycle of consecutive dive and climb movements is referred as a yo, hence the trajectory underwater is known as a yo-yo profile. This propulsion system produces an effective but low horizontal speed U_g, usually below $U_g \leq 0.4\mathrm{m/s}$. Consequently, gliders are strongly sensitive to ocean currents U_c, so they might have to adapt the bearing ψ_g to the ocean current velocity field in order to travel along the desired/commanded heading ψ_e.[1]

In terms of power consumption, the glider saw-tooth profile is very efficient, since the gravitational force is used as the power source for propulsion, that is the most critical task for AUVs persistence. Besides processing and communication, the batteries are only used intensively during a small fraction of the cycle time to change the vehicle buoyancy, by means of an electric buoyancy pump; and, much less demanding, to modify the vehicle attitude and bearing angle ψ_g while underwater using low consumption actuators. Going one step further, the thermal gliders replace the electric pump by thermo-active materials that react to the temperature gradient of the ocean, extending the vehicle range to allow, theoretically, planet circumnavigation.

[1] We follow the notation of Keay (1995) for the bearing and heading, where bearing is equivalent to course, being the orientation of the forward movement of the vehicle.

2.1.2 Piloting and Navigation

From the perspective of the glider pilot, gliders are semi-autonomous underwater vehicles that navigate autonomously underwater, but can be commanded when they surface. The navigation behavior of a glider during a mission is depicted in Figure 2.4. Periodically, the glider surfaces to localize itself using the GPS, and to communicate the mission data collected underwater via satellite to the ground station. It also stays a few minutes $\approx 15\text{min}$ at surface waiting for new orders, as the target waypoint $\mathbf{x}_{\text{goal}}$. The figure illustrates this surfacing interval and the yo-yo profile underwater. The ellipses represent the uncertainty on the vehicle position underwater. Gliders dead reckon when they navigate underwater in order to maintain the heading ψ_e to $\mathbf{x}_{\text{goal}}$. However, the navigation instruments and the processing hardware on board are very modest —e.g. SLOCUM gliders use a persistor for processing and only an inclinometer to measure the pitch θ and roll ϕ, which is used by the Inertial Navigation System (INS) that computes the vehicle dead-reckoning—, and hence the pose estimate uncertainty grows over time rapidly because of the low precision of the INS. The main source of uncertainty on the pose estimate comes from the drift caused by the ocean currents, which cannot be detected by the sensors on board. Since the gliders travel at a slow nominal speed U_g, they drift significantly from the expected trajectory, and only when they surface again the pose uncertainty collapses to the actual location with the first GPS fix.

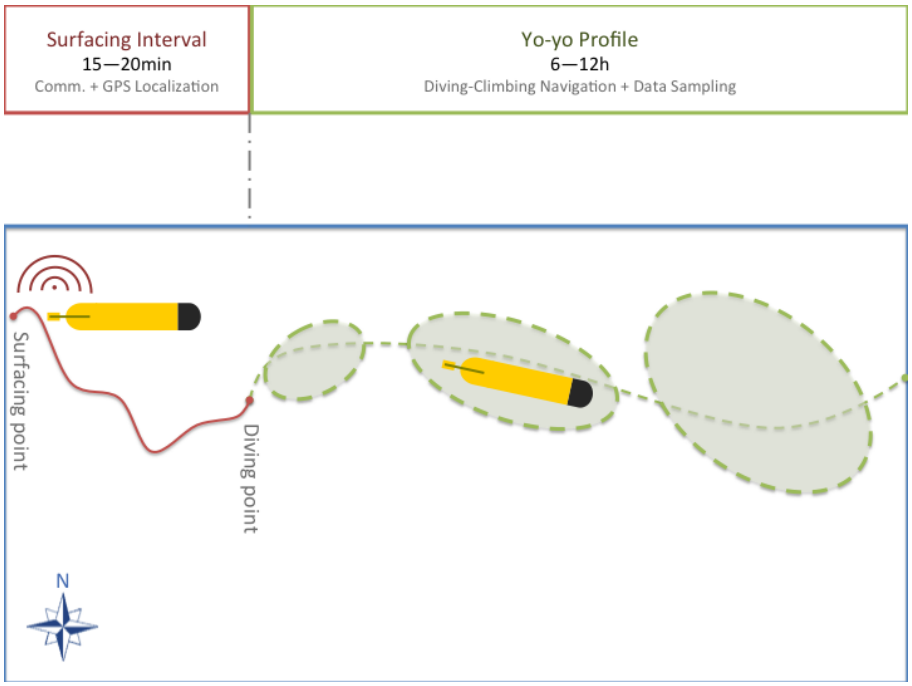


Figure 2.4: Diagram of a glider navigation stint, that comprises: the emersion and localization with the GPS, the drift produced by ocean currents and wind at surface for $t_s \approx 15\text{min}$, while it also communicates with the base station; the immersion; and the yo-yo profile during the underwater navigation period, with inflection points at a given depth interval $[z_{\min}, z_{\max}]$. On surface, the glider localization is solved using the GPS, but after the diving point the location is unknown. The ellipses ◌ represent the uncertainty of the position estimate underwater. The main source of uncertainty is the drift caused by the ocean currents. At each surfacing point such uncertainty collapses with the first GPS fix.

Figure 2.4 represents a cycle which is usually referred as stint. These cycles are repeated typically for 6 – 12h periods, a separated by surfacing intervals of 15 – 20min. During a stint, the glider tries to maintain

the bearing ψ_g that compensates the expected effect of the ocean currents in order to head to $\mathbf{x}_{\text{goal}}$. It sets a pitch θ that is generally fixed and close to the optimum in terms of the forward propulsion U_g. The pitch θ, along with the depth interval $[z_{\min}, z_{\max}]$, determine the number of yo-yos during the stint, which is typically small. Regarding the depth interval, the top level is usually $z_{\min} \approx 20\text{m}$ to avoid the effect of waves and winds near the surface. The bottom level depends on the mission, but it is also bounded by the maximum pressure the hull of the vehicle can stand underwater. In the case of SLOCUM gliders the are two models: one can reach $z_{\max} = 200\text{m}$, while the other can dive up to $z_{\max} = 1000\text{m}$.

At the time of writing this dissertation, ocean gliders are piloted manually. This means that when they surface, it is a human who connects to the vehicle to change the target waypoint $\mathbf{x}_{\text{goal}}$ or any other system parameter. In fact, in the case of SLOCUM gliders, this is done using the Dockserver and associated tools. With the potential of glider path planning tools to find the optimal path to the target waypoint $\mathbf{x}_{\text{goal}}$, it comes the demand of further automation of this process. This software cannot be accessed programmatically, so the resulting set of bearings $\mathcal{B}$ of the path found by the path planner, cannot be easily sent directly to the vehicle. A great effort is being made at this moment to develop the tools that will allow to integrate glider path planning algorithm in the loop. This way, the optimum path can be recomputed every time the glider surfaces, and the updated $\mathcal{B}$ can be sent before it dives again —or at least the bearing for the next surfacing point.

2.1.3 Commercial Models

A good selection of most remarkable glider systems up-to-date is covered by Barker (2012) in a Master Thesis that evaluates this technology. Here we will described some of them briefly, with special attention to the SLOCUM gliders, since we have used the glider path planning algorithms developed in this thesis to assist on their piloting.

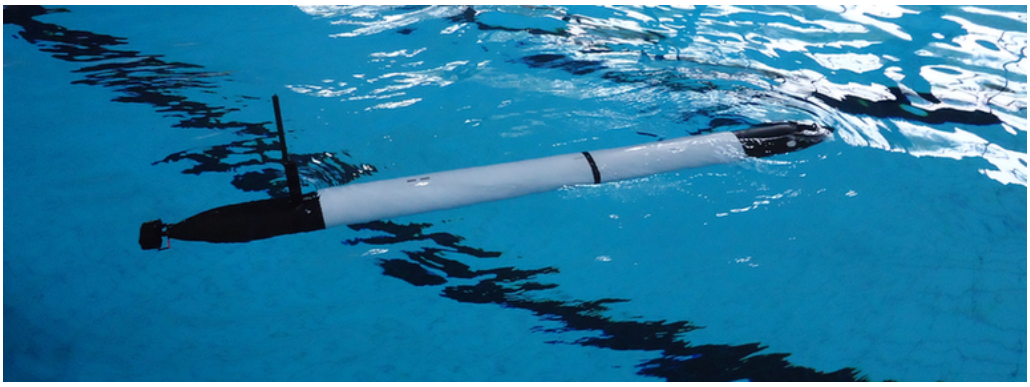


Figure 2.5: Folaga hybrid underwater glider in a pool test during the 2nd Co3-AUVs Summer School at Jacobs University in Bremen.

The SLOCUM gliders are manufactured by Teledyne Webb Research. In the electric SLOCUM an electrical pump inflates/deflates external bladder to change the vehicle buoyancy. There are two main variants, with depth ratings of 200m and 1000m. The company was awarded the Littoral Battlespace Sensing Glider contract from the US Navy, and since them they have became one of the main providers of this technology worldwide (Barker, 2012). They also develop the thermal SLOCUM glider, with a depth rating of 1200m. The buoyancy engine is controlled by the change of state of a oil mixture, so no electric power is used. However, it only operates if the temperature difference at surface and the at the maximum depth allow such change of state, which must be $\geq 12°\text{C}$ for the SLOCUM model.

We have already shown the electric SLOCUM, along with the Spray and the SeaGlider in Figure 1.12. On one hand, Bluefin Robotics manufactures the deep-diving Spray glider, originally designed by the Scripps Institution of Oceanography. On the other hand, there are two variants of the SeaGlider manufactured by iRobot. A two pump variant for the depth interval $[120, 1000]\mathrm{m}$, and a single pump variant for depths from the surface to $120\mathrm{m}$, which consumes less energy than the former.

The three ocean gliders mentioned above are widely used in Ocean Research, particularly the SLOCUM gliders. However they are also used in military applications, as the ANT glider developed under the sponsorship of the Office of Navel Research (ONR), and designed to meet the requirements of the US Navy Undersea Master Plan (Barker, 2012). This glider incorporates novel capabilities for military applications, like mine detection.

All these previous ocean gliders exhibit large wing-like surfaces to generate the forward motion. However, there are designs that replace them with horizontal finned appendages at the tail, like the SeaExplorer manufactured by ASCA-Alcen. Furthermore, the Folaga hybrid underwater glider, manufactured by the GRAAL Tech, completely eliminates the wings. It still has a buoyancy engine, but it uses electrical thru hull thrusters embedded in the vehicle to achieve the forward propulsion, and yaw and pitch corrections (see Figure 2.5).

2.2 Glider Motion Model and Dynamics

Any path planning algorithm requires a vehicle motion model in order to simulate how it moves or navigates throughout the search or configuration space. There exist three main types of motion models we can consider for a glider —or any vehicle in general. The **point** motion model is the simplest one, it represents the vehicle as an holonomic, moving point, and requires just a few parameters. The other two possible motion models can be kinematic or dynamic models. A more elaborated kinematic motion model for a glider is presented here in Section 2.2.3, which models the **force balance** of the yo-yo navigation profile of the glider. Finally, **dynamic** motion models consider the accelerations and more complex parameters that cover the vehicle hydrodynamic structure. Although more precise, the last two models require much more parameters, and many of them are usually extremely difficult to identify. Furthermore, for glider path planning, and particularly for long-term missions, the non-holonomic and differential constraints of the glider dynamics can be neglected. Consequently, a point model usually suffices to simulate the vehicle motion in glider missions; at most, it makes sense to use the force balance model —as long as we know how to parameterize it— to obtain accurate trajectories that include the effect of the temperature and water density, in order to use the output of the path planner with a real glider.

2.2.1 Notation

Before we explain each of the three types of glider motion models in the next sections, we are going to describe the notation used. In Marine Engineering, the most accepted notation to represent the pose of the vehicle is the so called **SNAME 1950** notation, which can be consulted in Fossen (2002). The attitude —i.e. the orientation of the vehicle **BODY** frame— is generally represented by the following Euler angles. They are defined as followed, for a rotation with respect to the frame axis:

x**-axis:** Roll ϕ, which can be discarded in most motion models for glider path planning. Indeed, it is not considered by the point and force balance motion models described here. Certainly, the roll can produce an effect on the motion, but it is generally assumed to be perfectly trimmed to $\phi = 0^\circ$ and stable during the motion.

y**-axis:** Pitch θ, which represents the angle used by the glider to dive and climb in the yo-yo profiles. As a consequence of the saw-tooth pattern, the pitch takes a constant value during a dive or a climb, but it

changes sign —and probably its magnitude slightly— at the inflection points of each yo. The pitch θ is considered positive when it goes down during the dives, and negative when it goes up during the climbs, so it must be modeled as a piecewise function. As long as the motion model is three-dimensional, it must consider the pitch θ. Otherwise —as it happens with some two-dimensional, simplified versions of the point motion models—, the glider nominal horizontal speed U_g accommodates the influence of θ. Also, the ocean currents are averaged in depth, to reflect their effect on the actual three-dimensional trajectory followed by the vehicle.

z-**axis:** Yaw ψ, which is the direction with respect to the North followed by the vehicle —the magnetic declination at the vehicle location and time is substracted, using the datum for the corresponding latitude, longitude and date. Note that ψ is part of the vehicle attitude, and is called bearing, rather than the actual course or heading, which is the angle of the resulting trajectory, affected by the drift produced by the ocean currents. We assume $\psi = 0^\circ$ at the North, growing clockwise, i.e. it is a north-based azimuth angle in the spherical coordinate system that represents the Earth.

Regarding the navigation frame, for glider missions is common to use the NED (North East Down) frame (Fossen, 2002) with the geographic coordinates latitude ϕ, longitude λ and depth z (growing downwards). Since, every time the glider surfaces, it geo-localizes with the GPS fixes, such position information is easily available. In the next sections, we adopt a slightly abuse of notation, since we represent the longitude λ with x, and the latitude ϕ with y —as in the x and y axes—, for the sake of clarity. Anyway, distances are computed in the appropriate projection, although some simplified distance metrics —including the Euclidean distance— are possible depending on the mission length (see Section A.2).

2.2.2 Point Motion Model

In a point model, the whole vehicle is represented by a single point in the space, with location $\mathbf{x} = (x, y, z)$, where x and y represent the longitude λ and latitude ϕ, respectively, and z is the depth with respect to the surface. The point motion model of a glider assumes this representation and also considers the attitude of the vehicle, in order to integrate its position according with the differential equations that model its motion, as well as the navigation pattern explained thus far (see Section 2.1). In the three-dimensional, general case, we take the yaw ψ —or bearing ψ_g— and pitch θ angles into account. However, in 2 dimensions —i.e. projecting the movement into the surface plane— only the bearing ψ_g is considered. Finally, the glider constant, nominal speed U_g is used to compute its location over the time.

We distinguish two different point motion models. First, the unconstrained, free motion model, in which we simply integrate the differential equations of the model. Secondly, a constrained motion model, where a given location $\mathbf{x}_1$ must be reached, starting from the current location $\mathbf{x}_0$. In this second model, the drift produced by the ocean currents must be compensated by setting the appropriate bearing ψ_g, so the glider trajectory follows the heading ψ_e defined by the angle between $\mathbf{x}_0$ and $\mathbf{x}_1$. The use of this constrained motion model is motivated by the search grid used in graph-based path planning methods like the A* algorithm used in this thesis and explained in Section 4.3. It requires to solve the navigation equations that take the vehicle from the current location $\mathbf{x}_i$ to the next $\mathbf{x}_{i+1}$ in the grid, given the ocean currents and U_g.

2.2.2.1 Unconstrained Motion Model

The most simple and intuitive point motion model is the unconstrained, free integration of the differential equations that model the glider. In this motion model, the ocean currents and the glider velocity vectors are simply composed to obtain the resulting velocity vector, at every step in the integration process. In order to highlight the importance of the velocity integration we will start by showing some figures that illustrate the drift effect produced by the ocean currents. The important point here is that in the presence of ocean currents we must integrate the point motion model with a sufficiently small integration step Δt, so we actually capture

the drifting. This happens for both static and dynamic ocean currents, although the effect is more noticeable for the latter.

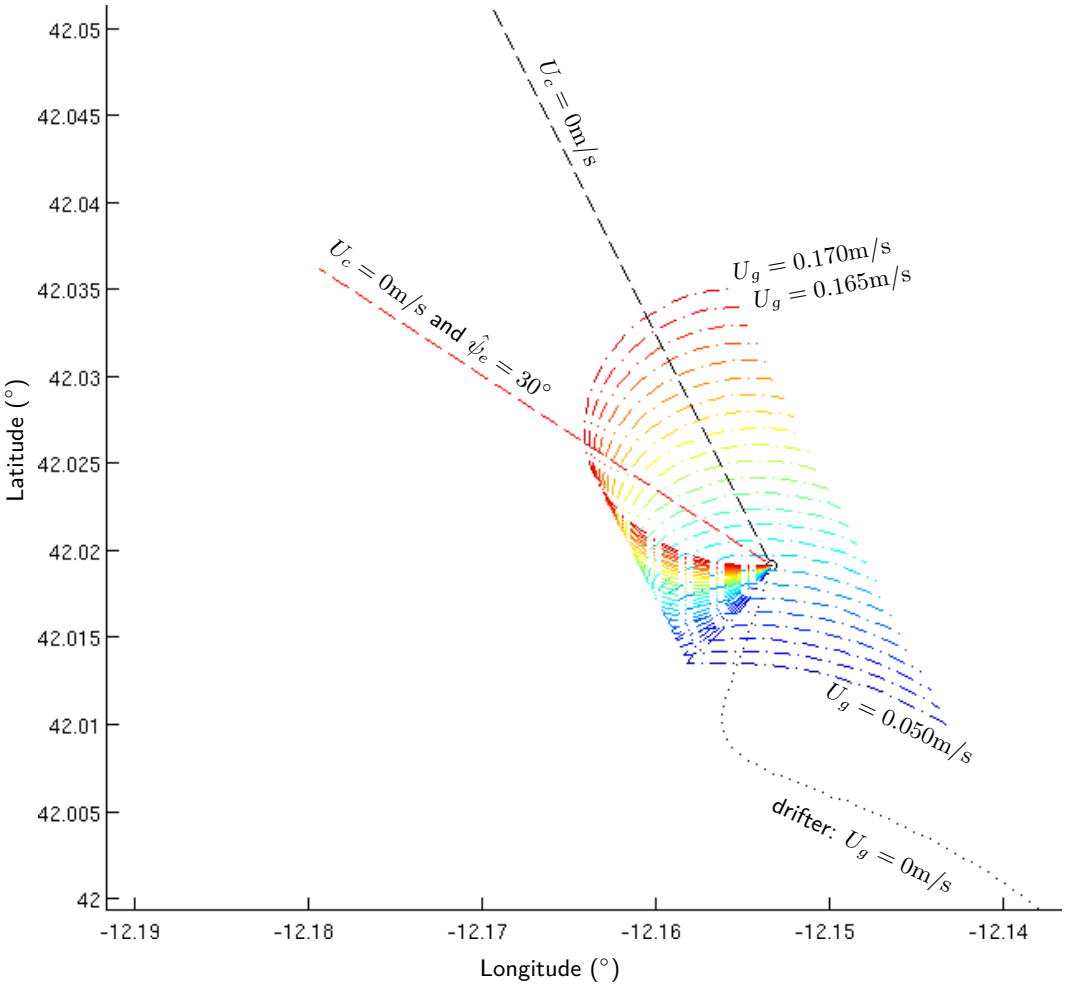


Figure 2.6: Unconstrained Point glider motion model run for different glider speeds U_g, for a given, constant heading error of $\hat{\psi}_e = 30°$N. The model is integrated for a surfacing time $t_s = 8\mathrm{h}$. We observe the drift in the trajectories due to the effect of the ocean currents, from $U_g = 0.050\mathrm{m/s}$ to $U_g = 0.170\mathrm{m/s}$ —the ocean currents are not shown for the sake of clarity, because they vary over the time and their effect can be observed in the drifter trajectory. As U_g reduces, they tend to the drifter trajectory with $U_g = 0\mathrm{m/s}$ ·····. On the contrary, the trajectory with no ocean currents $U_c = 0\mathrm{m/s}$ would be straight, as shown with no heading error - - - and $\hat{\psi}_e = 30°$ - - -.

In real glider missions it is common to observe small errors or some uncertainty associated with the expected heading ψ_e and nominal speed U_g. Therefore, it seems reasonable to evaluate the impact of small errors on these quantities. After all, they are the parameters or inputs of the unconstrained point motion model in 2 dimensions —in 3 dimensions we also have the pitch θ. Additionally, those errors degrade/grow systematically throughout the mission due to biofouling —e.g. during the final part of RU27 trans-Atlantic flight U_g reduced significantly —$\approx 0.15\mathrm{m/s}$, for a theoretical maximum of $0.4\mathrm{m/s}$.

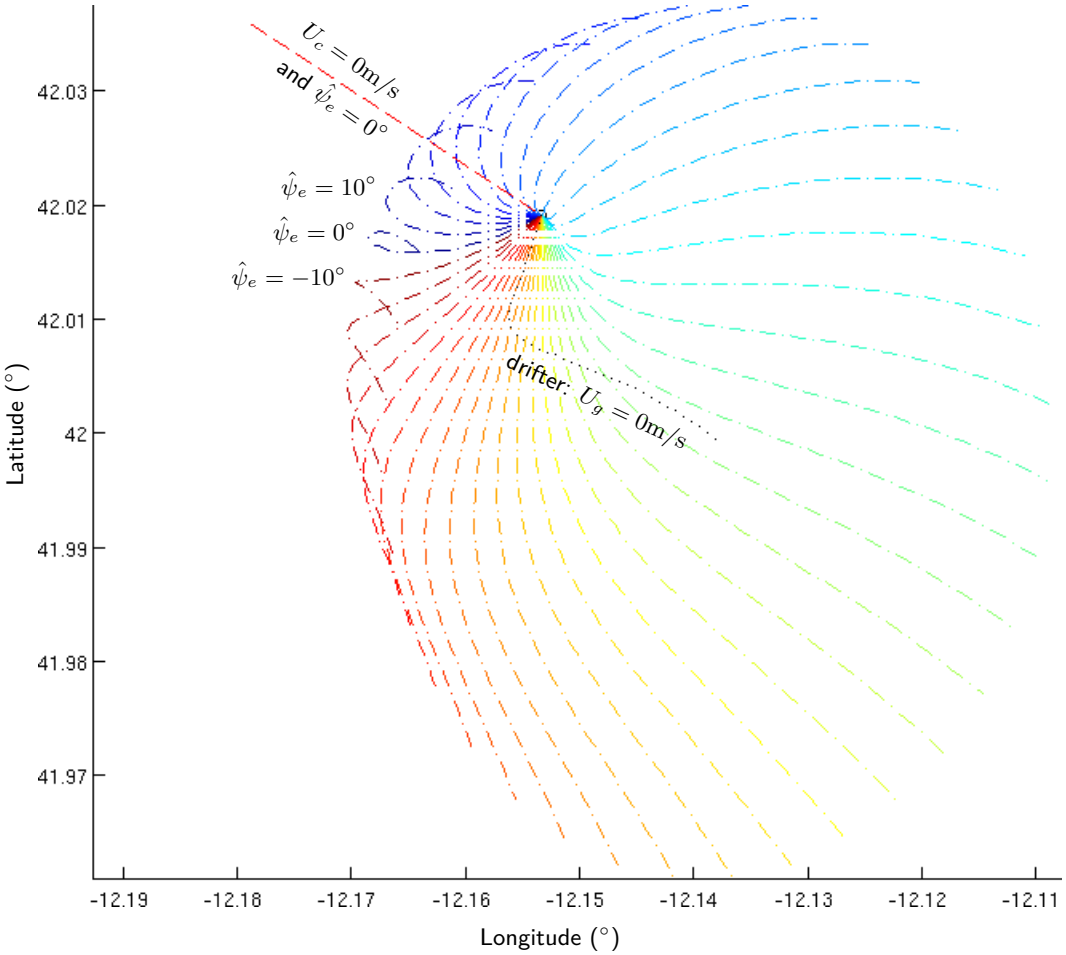


Figure 2.7: Unconstrained Point glider motion model run for different heading error angles $\hat{\psi}_e$, for a given, constant glider speeds $U_g = 0.15\mathrm{m/s}$. The model is integrated for a surfacing time $t_s = 8\mathrm{h}$. We observe the drift in the trajectories due to the effect of the ocean currents, for $\hat{\psi}_e = \{-10, 0, 10\}°$. The drifter trajectory with $U_g = 0\mathrm{m/s}$ and no heading error $\hat{\psi}_e = 0°$ ····· is shown for comparison. On the contrary, the trajectory with no ocean currents $U_c = 0\mathrm{m/s}$ would be straight, as shown with no heading error - - -.

The heading error $\hat{\psi}_e$ is a measure of the deviation from the theoretically expected heading ψ_e, probably induced by several coupled, nonlinear or unpredictable factors. Both $\hat{\psi}_e$ and U_g have a high impact in the resulting trajectory followed by the glider, as shown in Figure 2.6 and 2.7. For instance, there exists a strong influence of the ocean current field over time and a tight dependence on the heading error that affects the glider motion. The resulting trajectories are also affected by the glider nominal speed, since it determines whether the glider can beat the ocean currents or not. However, if we compare Figure 2.6 and 2.7, we observe that the effect of the heading error is clearly more significant. Thus, if $\hat{\psi}_e$ were known, it will be possible to estimate the actual glider speed U_g using the surfacing locations $\mathbf{x}_i$, as shown in Figure 2.8 for the RU27 surfacing points during the trans-Atlantic mission headed by the Rutgers University.

In Figure 2.6 and 2.7 we observe that for certain heading angles ψ_e is not possible to navigate within them because of the effect of the ocean currents U_c. This effect is clearer when the glider nominal speed U_g

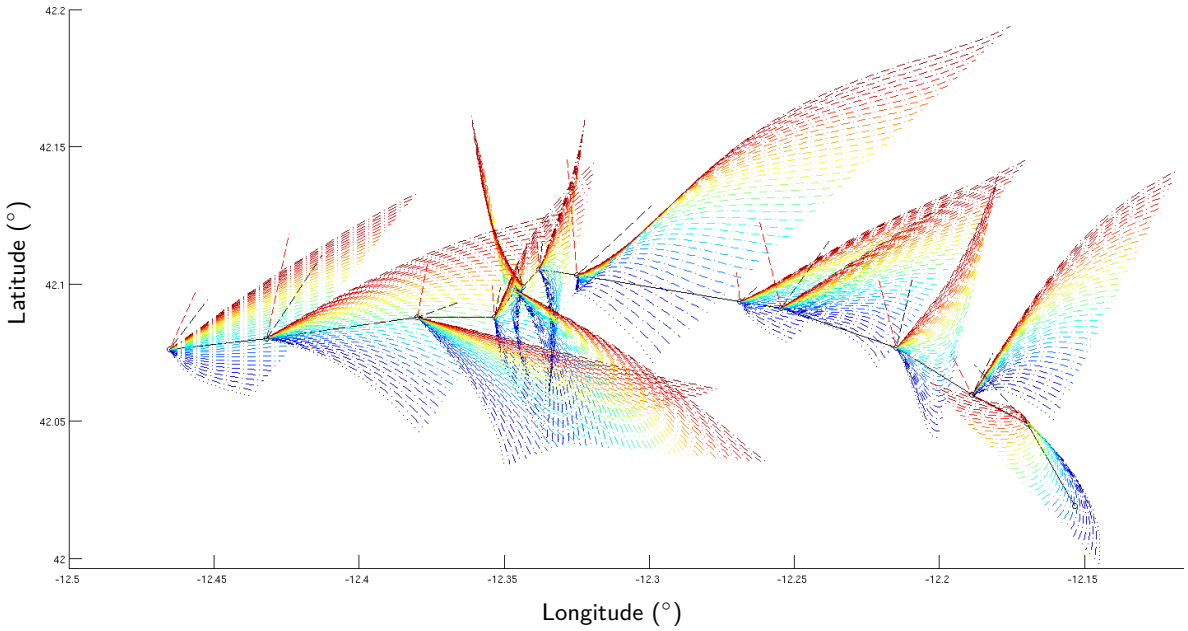


Figure 2.8: Unconstrained Point motion model applied to a series of trajectories varying U_g from $0.01\mathrm{m/s}$ to $0.40\mathrm{m/s}$ with $0.01\mathrm{m/s}$ increments. Given the heading error $\hat{\psi}_{e_i}$ for each stint i, we may analyze the effect of U_g for different surfacing points of the RU27 glider path (ground truth) $\mathbf{x}_i$ ——. The trajectories for $U_c = 0\mathrm{m/s}$ - - -, $U_c = 0\mathrm{m/s}$ and $\hat{\psi}_e = 0°$ - - - , and a drifter ($U_g = 0\mathrm{m/s}$) ····· are shown too.

is small, as shown in the first figure. As we will later show formally for the constrained motion model, if we impose the vehicle to follow a given heading ψ_e, in many cases that will be unfeasible because of the ocean currents. In the presence of strong ocean currents, this will lead to situations in which many trajectories are simply discarded because the heading cannot be followed. On the contrary, here with the unconstrained motion model we just compute the effective speed $\mathbf{U}_e$, regardless of the resulting trajectory and heading followed, which will probably be non-constant, as in the figures.

The effective speed $\mathbf{U}_e$ is computed for the unconstrained motion model using

$$\mathbf{U}_e = \mathbf{U}_g + \mathbf{U}_c \tag{2.1}$$

where $\mathbf{U}_g$ is the glider nominal speed and $\mathbf{U}_c$ is the speed of the ocean current through the path $\mathcal{P}$ described by the glider. Here, the speeds are in vector form, and correspond to the two-dimensional case, in which the overall effect of underwater ocean currents is projected onto the surface.

Since $\mathbf{U}_c$ is different at each point in the environment, we must integrate the velocity through $\mathcal{P}$. Hence, we have

$$\mathbf{U}_e = \mathbf{U}_g + \int_{\mathbf{x}_{i-1}}^{\mathbf{x}_i} \mathbf{U}_c(\mathbf{x})\, d\mathbf{x} \tag{2.2}$$

for a path $\mathcal{P}$ starting at $\mathbf{x}_{i-1}$ and ending at $\mathbf{x}_i$ locations, that will correspond to surfacing points.

The integral along the path $\mathcal{P}$ is typically discretized and integrated numerically for a fixed time t_s between surfacings. Such integration allows to obtain the glider trajectory under the ocean current field conditions. Therefore, we implement the following recursion

$$\mathbf{U}_e(t) = \mathbf{U}_g + \mathbf{U}_c(\mathbf{x}(t), t) \tag{2.3}$$

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \mathbf{U}_e(t) \cdot \Delta t \tag{2.4}$$

Algorithm 2.1 Trajectory integration.

Require: Surfacing time t_s and integration step dt. Glider nominal speed $\mathbf{U}_g$, ocean currents velocity field $\mathbf{U}_c(\mathbf{x})$ for all locations $\mathbf{x}$, and initial location $\mathbf{x}_0$.
Ensure: Final location $\mathbf{x}_{t_s}$ after integrating the trajectory for t_s time.
Algorithm: trajectory(t_s, dt, $\mathbf{U}_g$, $\mathbf{U}_c(\mathbf{x})$, $\mathbf{x}_0$) **return** $\mathbf{x}_{t_s}$
1: $\mathbf{x} = \mathbf{x}_0$ ▷ Initial location
2: **for all** $t = 0$ to t_s with step dt **do**
3: $\quad \mathbf{U}_e = \mathbf{U}_g + \mathbf{U}_c(\mathbf{x})$
4: $\quad \mathbf{x} = \mathbf{x} + \mathbf{U}_e \cdot dt$
5: **end for**
6: **return** $\mathbf{x}$

where Δt is the integration step and $\mathbf{x}(0)$ is initialized with the initial location of the vehicle. The integration algorithm is summarized in Algorithm 2.1. The time t is discretized according with Δt, and the position of the glider is integrated until we reach the surfacing time t_s. We actually do not have to select Δt, since this Ordinary Differential Equation (ODE) can be solved numerically using fourth and fifth order Runge-Kutta formulas (Dormand and Prince, 1980). In this process, we generally need to interpolate the ocean currents $\mathbf{U}_c(\mathbf{x}(t), t)$ at every location $\mathbf{x}(t)$ of the trajectory, and time t. Although we can simply use the Nearest Neighbor ocean current, a decent simulation of the glider trajectory requires at least some linear interpolation, in 2, 3 or 4 dimensions —including the time t, for dynamic ocean currents. If more accurate results are required, we can use more precise techniques like the tricubic interpolation in 3 dimensions of Lekien and Marsden (2005), which addresses the precise interpolation of ocean currents measured with CODAR.

It is straightforward to extend the unconstrained motion model to the 3 spatial dimensions, i.e. including the depth motion. In 3 dimensions the model also requires the pitch angle θ as an input. Therefore, a glider modeled as a moving point follows these differential equations

$$\dot{x} = u + U_g \cos\theta \cos\psi \tag{2.5}$$
$$\dot{y} = v + U_g \cos\theta \sin\psi \tag{2.6}$$
$$\dot{z} = w + U_g \sin\theta \tag{2.7}$$

where U_g, ψ and θ are the glider's nominal surge speed, heading and pitch, and where u, v and w are the x, y and z components of the ocean current speed at a given location (x, y, z) and time t. As usual, (x, y, z) represents the longitude, latitude and depth, respectively. The pitch θ can take any angle in the range $[-\pi/2, \pi/2]$, where positive values indicate a dive, and negative indicate a climb, in the yo-yo profile. Although the $\dot{z}$ differential equation above accommodates any θ, the pitch is actually a function of the time t. Indeed, it changes at the minimum $z_{\min}$ and maximum $z_{\max}$ depth points in which the glider's pump actuates on the oil bladder. At these points, the pitch changes its sign. Hence, we have a piecewise function

$$\theta = f_\theta(t) = \begin{cases} \theta_{\text{dive}} & \text{if diving} \\ \theta_{\text{climb}} & \text{if climbing} \end{cases} \tag{2.8}$$

Consequently, we must integrate each piece separately, and to do so we must know the integration time T for each piece —i.e. for each dive or climb. Although $\theta \in [-\pi/2, \pi/2]$, it is easier to assume that we have a flag to indicate whether the glider is diving or climbing, and we always have $\theta \in [0, \pi/2]$, i.e. $|\theta|$. This trick allows as to obtain T, but now the differential equations above are only valid when the vehicle goes down. On the contrary, when it goes up, (2.7) changes sign; meanwhile (2.5) and (2.6) remains the same, since the pitch θ only affects the depth z component.

$$\dot{z} = w - U_g \sin\theta \tag{2.9}$$

We still have piecewise differential equations, for each $\dot{z}$, so we must compute the integration time T for each, taken the solution of the previous as initial value. The integration time T is computed from the depth interval, which is

$$T = \frac{\Delta z}{\dot{z}} = \frac{z(t_0 + T) - z(t_0)}{w \pm U_g \sin\theta} \tag{2.10}$$

We assume null vertical ocean currents $w = 0$, which is generally the case, so

$$T = \frac{z(t_0 + T) - z(t_0)}{\pm U_g \sin\theta} \tag{2.11}$$

Hence, according with the yo-yo profile, the depth interval can only take two different values:

$$\Delta z = \begin{cases} \Delta z_{\text{out}} = z_{\max} & \text{for (outer) first and last yos} \\ \Delta z_{\text{in}} = z_{\max} - z_{\min} & \text{for inner yos, if any} \end{cases} \tag{2.12}$$

Now we can take the absolute value of the depth speed $\dot{z}$, and hence the integration interval is simply

$$T = \frac{\Delta z}{U_g \sin\theta} \tag{2.13}$$

Also, in order to ensure that the vehicle is at surface when the stint is done, the following equation must be used to compute the time T_u underwater, given the number n of yos and the depth interval $[z_{\min}, z_{\max}]$, assuming that we start at $z = 0$ and finish at the surface as well.

$$T_u = \frac{2\left(\Delta z_{\text{out}} + (n-1)\Delta z_{\text{in}}\right)}{U_g \sin\theta} \tag{2.14}$$

We can also estimate the position at surface given the time drifting t_d there, in order to simulate with more precision the trajectory of a glider during a real mission. In this case we use the equations of a drifter at surface, i.e. we set $\dot{z} = 0$ in place of (2.7). This is exactly a bi-dimensional unconstrained motion model with $U_g = 0\text{m/s}$, that can be included in the path planning algorithm to take into account the $t_d \approx 15\text{min}$ the glider spend at surface.

2.2.2.2 Constrained Motion Model

A constrained point motion model works radically different from the unconstrained one described in the previous section. Now we basically solve the navigation problem of going from a start location $\mathbf{x}_0$ to an ending location $\mathbf{x}_1$. The angle between $\mathbf{x}_0$ and $\mathbf{x}_1$ defines a heading ψ_e that we want the vehicle to follow. Therefore, given the ocean currents vector $\mathbf{U}_c$ and the glider nominal horizontal speed U_g, the navigation problem consists on finding the bearing angle ψ_g required to move in ψ_e, if possible. Indeed, when the ocean currents are stronger than U_g, there exists a range of heading angles ψ_e which are infeasible, i.e. there is not bearing ψ_g that allows the vehicle to reach $\mathbf{x}_1$ heading in ψ_e.

Given the glider nominal speed U_g, the desired course or heading ψ_e, and the ocean current speed U_c and direction ψ_c, we can compute the drift correction that the glider must apply to keep such heading. Thus, we obtain the bearing ψ_g with

$$\psi_g = \begin{cases} \psi_e + \arcsin s & \text{if } |s| \leq 1 \\ \nexists & \text{otherwise} \end{cases} \tag{2.15}$$

and the resulting speed U_e with

$$U_e = \begin{cases} U_g\sqrt{1 - s^2} + U_c \cos\psi_{ec} & \text{if } |s| \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.16}$$

where

$$s = \frac{U_c}{U_g} \sin \psi_{ec} \tag{2.17}$$

and $\psi_{ec} = \psi_e - \psi_c$. Regardless of the angle difference ψ_{ec}, we can defined the speed ratio SR as

$$SR = \frac{U_c}{U_g} \tag{2.18}$$

which allows to categorize ocean currents as strong or weak, depending on whether $SR > 1$ or $SR < 1$, respectively.

Consequently, if $U_g > U_c$ the set of feasible heading angles ψ_e is unbounded. Otherwise, this set is delimited by a so called *accessibility cone* with an arc angle ψ_r given by

$$\psi_r = 2 \arcsin \frac{U_g}{U_c} \tag{2.19}$$

For the special case of $U_g = U_c$ we have $\psi_r = \frac{\pi}{2}$. Thus, a heading ψ_e is physically feasible if and only if it falls in $\psi_e \in [-\frac{\psi_r}{2}, \frac{\psi_r}{2}]$.

Note that in the particular case of $U_g = U_c$ the condition $|s| \leq 1$ is always true, since $s = \sin \psi_{ec}$. Operating, (2.15) and (2.16) are reduced to

$$\psi_g = \begin{cases} 2\psi_e - \psi_c & \text{if } |\psi_{ec}| \leq \dfrac{\pi}{2} \\ \nexists & \text{otherwise} \end{cases} \tag{2.20}$$

and

$$U_e = \begin{cases} 2U_g \cos \psi_{ec} & \text{if } |\psi_{ec}| \leq \dfrac{\pi}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.21}$$

respectively, satisfying $U_e \geq 0$.

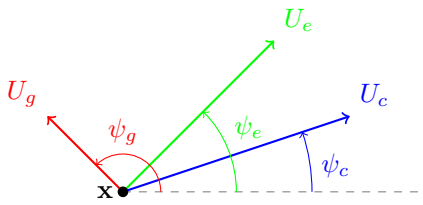


Figure 2.9: Vector composition for drift correction. A bearing ψ_g must be commanded to obtain a heading or course ψ_e that can beat a current of direction ψ_c. The resulting effective velocity U_e depends on the glider speed U_g and the ocean current speed U_c.

In Figure 2.9 is depicted the vector composition of the drift correction process. Notice that all this velocities $U \cdot$ are not vectors, but magnitudes $\|\mathbf{U}_\star\|$. We assume $U_e = 0$ when it is not possible to keep a particular heading ψ_e. This is usually a consequence of a low glider speed U_g relative to the ocean current speed U_c, which forces the glider to navigate following the ocean current direction ψ_c.

Soulignac et al. (2008) develop an alternative deduction of this motion model in his work, which does not use trigonometric functions. The resulting expressions are completely equivalent to the ones shown thus far, but are faster to compute in general.

According to Soulignac et al. (2008) the temporal cost t of the movement $\mathbf{d}$ verifies $\mathbf{d} = (\mathbf{U}_g + \mathbf{U}_c) \cdot t$. Projecting this equality on x and y axis yields

$$\begin{cases} d_x = (U_{g_x} + U_{c_x})t \\ d_y = (U_{g_y} + U_{c_y})t \end{cases} \implies \begin{cases} d_x - U_{c_x}t = U_{g_x}t \\ d_y - U_{c_y}t = U_{g_y}t \end{cases} \tag{2.22}$$

The relation $U_g^2 = U_{g_x}^2 + U_{g_y}^2$ allows us to eliminate U_{g_x} and U_{g_y} in (2.22), leading the relation

$$(d_x - U_{c_x}t)^2 + (d_y - U_{c_y}t)^2 = U_g^2 t^2 \tag{2.23}$$

that rearranging for t yields

$$(U_g^2 - U_c^2)t^2 + 2\langle \mathbf{d} \cdot \mathbf{U}_c \rangle t - d^2 = 0 \tag{2.24}$$

where $d^2 = d_x^2 + d_y^2$.

Solving this second degree equation of (2.24), we have

$$t = \begin{cases} \dfrac{\sqrt{\Delta} - \langle \mathbf{d} \cdot \mathbf{U}_c \rangle}{U_g^2 - U_c^2} & \text{if } \Delta \geq 0 \\ \infty & \text{otherwise} \end{cases} \tag{2.25}$$

where $\Delta = U_g^2 d^2 - (U_{c_x} d_y - U_{c_y} d_x)^2$ and $\langle \cdot \rangle$ is the scalar product. Note that t is defined only if $\Delta \geq 0$, in order to satisfied that $t \geq 0$.

Here, in the particular case of $U_g = U_c$ the second degree equation of (2.24) becomes the one degree equation

$$2\langle \mathbf{d} \cdot \mathbf{U}_c \rangle t - d^2 = 0 \tag{2.26}$$

and solving, we simply have

$$t = \frac{d^2}{2\langle \mathbf{d} \cdot \mathbf{U}_c \rangle} \tag{2.27}$$

The main application of this constrained motion model is concerned with the implementation of graph-based path planning algorithms. This type of methods uses a grid search space which must be explored for the optimal solution. In order to move from one node n_i —that represents a surfacing point— to its neighbor nodes n_{i+1} in the grid, we must apply the constrained motion model described here; see the A* algorithm in Section 4.3 for instance. It is possible to extend the motion model to 3 dimensions, but for gliders it is not really required because underwater they move non-holonomically following the saw-tooth pattern that restricts the vehicle maneuverability considerably. That is, we only need to look into the search space defined by the surfacing points. Anyway, in order to obtain a good simulation of the glider motion while underwater, we must average the ocean currents from n_i to n_{i+1}. In practice, if the search space grid resolution is sufficiently precise, the ocean currents are similar in the region around n_i and n_{i+1}, so even a single value may suffice, i.e. it is considered constant. Therefore, we can take the ocean current vector $\mathbf{U}_c$ at n_i, or the depth average from n_i to n_{i+1} in order to represent the fact that the gliders are actually affected by the underwater ocean currents while they navigate performing yo-yo profiles.

2.2.3 Force Balance Motion Model

A more elaborated —but still kinematic— motion model is the one proposed by Merckelbach et al. (2010). It is based on the equilibrium of the forces actuating in a glider while it dives or climbs. Figure 2.10 depicts these forces and the angles that describe the motion and the vehicle attitude in the xz plane for a climb motion. As we will see in the equations below, this model depends on a number of parameters associated with the vehicle shape and the navigation system, i.e. the buoyancy pump and the wings. With this model

we have a good approximation of the glider vertical and surge speed, which allow for a better dead-reckoning than the simple point model. However, the main drawback of this model is that we must identify a great number of parameters that depends on each vehicle and are usually unavailable. The constraints imposed by the surfacing points and vertical speed estimated from the depth measurements of the pressure sensor, help in such identification, but it is usually not enough, and flow simulations or water tank tests are generally required. Anyway, here we develop the equations of this force balance model and show some results that evaluate the model characteristics and dependence with respect to the most remarkable parameters, like the buoyancy change and the pitch angle.

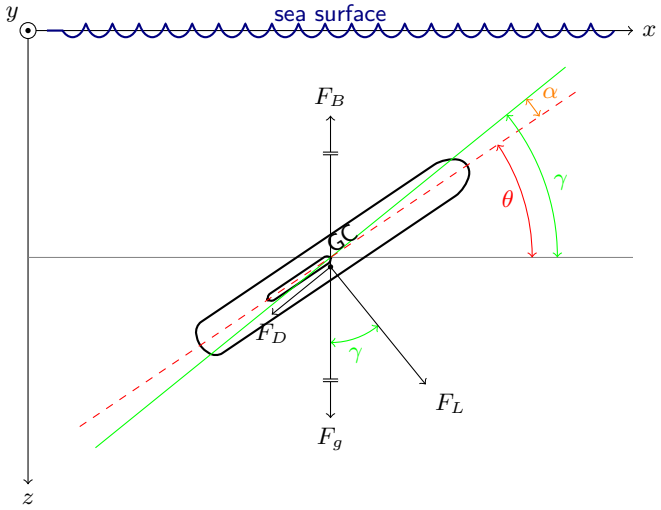


Figure 2.10: Force balance diagram that represents a glider moving in the xz vertical plane along with the forces involved and the definition of angles. This diagram follows the one in (Merckelbach et al., 2010).

The forces working on a glider vehicle are the buoyancy F_B, which goes up to the surface; the gravity F_g, which goes down in opposite direction to F_B; the lift F_L, which goes perpendicular to the movement, and consequently points down or up, when the glider climbs or dives, respectively; and the drag F_D, which goes backwards, against the movement. The lift is responsible of generating the forward movement of the vehicle, as long as it is enough to compensate for the drag force F_D. The pitch angle θ represents the angle of the vehicle's horizontal axis with respect to the horizontal plane. The glide angle γ represents the angle of the climb/dive plane with respect to the horizontal plane, and it includes the angle of attack α, which is produced by the wings and the hull design dynamics. Hence, the difference between the glide angle γ and the pitch θ is the angle of attack α.

According with the diagram of Figure 2.10, the vertical and horizontal force balances are

$$F_B - \cos\gamma F_L - \sin\gamma F_D - F_g = 0 \tag{2.28}$$

$$-\cos\gamma F_D + \sin\gamma F_L = 0 \tag{2.29}$$

respectively, where F_B is the (net) buoyancy force, F_g is the force due to gravity, F_L is the lift force, F_D is the drag force and γ is the glide angle, defined as the sum of the pitch angle θ and the angle of attack α, i.e. $\gamma = \theta + \alpha$.

The force due to gravity is given by

$$F_g = g m_g \tag{2.30}$$

where m_g is the mass of the glider and g is the acceleration due to gravity.

The net buoyancy force is given by

$$F_B = g\rho(V_g(1 - \varepsilon P + \alpha_T(T - T_0)) + \Delta V_{bp}) \tag{2.31}$$

where ρ is the *in-situ* density, V_g is the glider volume at atmospheric pressure, ε is the compressibility of the hull, P is the water pressure, α_T is the thermal expansion coefficient, T is the water temperature, T_0 is a reference water temperature and ΔV_{bp} is the buoyancy change produced by the buoyancy engine.

The drag and lift forces are given by

$$F_D = \frac{1}{2}\rho C_D S U^2 \tag{2.32}$$

$$F_L = \frac{1}{2}\rho C_L S U^2 \tag{2.33}$$

where C_D and C_L are the drag and lift coefficients, respectively, S is the wing surface area and U is the glider velocity through water along the glide path.

The total drag is the combined effect of the parasite drag and the induced drag, and can be parameterized as

$$C_D = C_{D_0} + C_{D_1}\alpha^2 \tag{2.34}$$

where C_{D_0} is the parasite drag and C_{D_1} a coefficient determining the induced drag. Decomposing the total drag into drag due to the hull and wings, respectively, we write $C_{D_0} = C_{D_{0,h}} + C_{D_{0,w}}$ and $C_{D_1} = C_{D_{1,h}} + C_{D_{1,w}}$.

The induced drag due to the wings can be parameterized as

$$C_{D_{1,w}} = \frac{a_w^2}{\pi \mathit{A\!R} e} \tag{2.35}$$

where $e \approx 0.8$ is the span efficiency parameter. The other drag coefficients can be measured directly.

The total lift is modeled as the sum of the lift generated by the hull and the lift generated by the wings. However, under normal flight conditions the angle of attack is generally small, so the lift coefficient can be parameterized as

$$C_L = (a_h + a_w)\alpha \tag{2.36}$$

where a_h and a_w are the lift slope coefficients for the hull and wings, respectively.

A semi-empirical formula for the lift slope coefficient for the wings, accounting for the lift reduction due to the wing sweep angle Ω and the aspect-ratio $\mathit{A\!R}$, simplifies to

$$a_w = \frac{2\pi \mathit{A\!R}}{2 + \sqrt{\mathit{A\!R}^2(1 + \tan^2\Omega) + 4}} \tag{2.37}$$

On the other hand, the slope coefficient for the hull a_h can be measured directly.

Substituting (2.34) and (2.36) into (2.32) and (2.33), respectively, yields

$$F_D = \frac{1}{2}\rho(C_{D_0} + C_{D_1}\alpha^2)SU^2 \tag{2.38}$$

$$F_L = \frac{1}{2}\rho(a_h + a_w)\alpha S U^2 \tag{2.39}$$

An implicit expression for the angle of attack is found by substituting (2.38) and (2.39) into the horizontal

force balance (2.29) and solving for α,

$$\begin{aligned} -\cos\gamma F_D + \sin\gamma F_L &= 0 \\ -\cos\gamma\frac{1}{2}\rho(C_{D_0} + C_{D_1}\alpha^2)SU^2 & \\ +\sin\gamma\frac{1}{2}\rho(a_h + a_w)\alpha SU^2 &= 0 \\ \alpha &= \frac{\cos\gamma\frac{1}{2}\rho(C_{D_0} + C_{D_1}\alpha^2)SU^2}{\sin\gamma\frac{1}{2}\rho(a_h + a_w)SU^2} \\ \alpha &= \frac{\cos\gamma(C_{D_0} + C_{D_1}\alpha^2)}{\sin\gamma(a_h + a_w)} \end{aligned}$$

that yields the implicit equation

$$\alpha = \frac{C_{D_0} + C_{D_1}\alpha^2}{(a_h + a_w)\tan(\theta + \alpha)} \tag{2.40}$$

from which α can be solved numerically.

Combining (2.28), (2.38), (2.39) and (2.40) gives

$$\begin{aligned} F_B - F_g - \cos\gamma F_L - \sin\gamma F_D &= 0 \\ F_B - F_g - \cos\gamma\frac{1}{2}\rho(a_h + a_w)\frac{C_{D_0} + C_{D_1}\alpha^2}{(a_h + a_w)\tan\gamma}SU^2 & \\ - \sin\gamma\frac{1}{2}\rho(C_{D_0} + C_{D_1}\alpha^2)SU^2 &= 0 \\ F_B - F_g - \frac{1}{2}\rho(C_{D_0} + C_{D_1}\alpha^2)SU^2\left(\frac{\cos\gamma}{\tan\gamma} + \sin\gamma\right) &= 0 \\ F_B - F_g - \frac{1}{2}\rho(C_{D_0} + C_{D_1}\alpha^2)SU^2\left(\frac{\cos^2\gamma + \sin^2\gamma}{\sin\gamma}\right) &= 0 \end{aligned}$$

that yields

$$F_B - F_g - \frac{\rho SU^2(C_{D_0} + C_{D_1}\alpha^2)}{2\sin\gamma} = 0 \tag{2.41}$$

from which, once α is solved, U can be evaluated with

$$U = \sqrt{\frac{2\sin\gamma(F_B - F_g)}{\rho S(C_{D_0} + C_{D_1}\alpha^2)}} \tag{2.42}$$

Given the glider heading ψ_e, as we had for the point motion model in Section 2.2.2, the horizontal and vertical components of the velocity vector follow from

$$u_g = U\cos\gamma\cos\psi_e \tag{2.43}$$
$$v_g = U\cos\gamma\sin\psi_e \tag{2.44}$$
$$w_g = U\sin\gamma \tag{2.45}$$

respectively, where $\gamma = \theta + \alpha$. The speed u_g corresponds to the x-axis or longitude, and v_g maps to the y-axis or latitude. Also note that the glider surge speed U_g is just the compound of the horizontal speeds u_g and v_g, which yields

$$U_g = U\cos\gamma \tag{2.46}$$

With this expression of the glider speed, the highest surge speed U_g is obtained at $\theta \approx 35°$. The maximization is not trivial because of the dependence of α in the drag coefficient C_D, but we can compute it numerically to obtain a more precise approximation of θ.

Given α, the pitch θ can be computed with

$$C_{D_0} + C_{D_1}\alpha^2 = \alpha(a_h + a_w)\tan(\theta + \alpha) \tag{2.47}$$

$$\tan(\theta + \alpha) = \frac{C_{D_0} + C_{D_1}\alpha^2}{\alpha(a_h + a_w)} \tag{2.48}$$

$$\theta = \arctan\left(\frac{C_{D_0} + C_{D_1}\alpha^2}{\alpha(a_h + a_w)}\right) - \alpha \tag{2.49}$$

Table 2.1: Parameters required by the glider force balance motion model.

Parameter	Description
γ	Glide angle $\gamma = \theta + \alpha$
θ	Pitch angle in BODY frame, usually measured with an IMU
α	Angle of Attack (AoA)
V_g	Glider volume at atmospheric pressure
m_g	Glider mass
ε	Compressibility of the hull
α_T	Thermal expansion coefficient
T_0	Reference temperature for α_T
∇V_{bp}	Buoyancy change due to the buoyancy engine
C_D	Drag coefficient (2.34)
C_L	Lift coefficient (2.36)
S	Wing surface area
U	Glider velocity through water along the glide path
a_h	Lift slope coefficient for the hull
a_w	Lift slope coefficient for the wings (2.37)
Ω	Wing sweep angle (at $1/4$ chord line)
AR	Aspect Ratio $\mathit{AR} = \frac{b^2}{S}$
b	Wingspan
$C_{D_{0,h}}$	Parasite drag for the hull
$C_{D_{0,w}}$	Parasite drag for the wings
$C_{D_{1,h}}$	Induced drag for the hull
$C_{D_{1,w}}$	Induced drag for the wings (2.35)
e	Span efficiency parameter $e \approx 0.8$

The three-dimensional speed vector defined by u_g, v_g and w_g must be compound with the ocean current speed at each point and time to obtain the actual vehicle speed underwater. Such computation is straightforward and follows the unconstrained three-dimensional point motion model explained in Section 2.2.2. Nevertheless, contrary to the point models, the force balance models requires much more parameters to be configured adequately, which are summarized in Table 2.1.

It is possible to create an spider plot for the force balance motion model, as shown in Figure 2.11 and 2.12 for the diving and climbing states, respectively. These glider plots —also known as glide polar plots— represent the horizontal surge speed U_g against the vertical speed w_g for several changes of buoyancy ΔV_{bp} of the bladder. Interestingly, along with the curves for different ΔV_{bp} we obtain iso-lines of same angle of attack

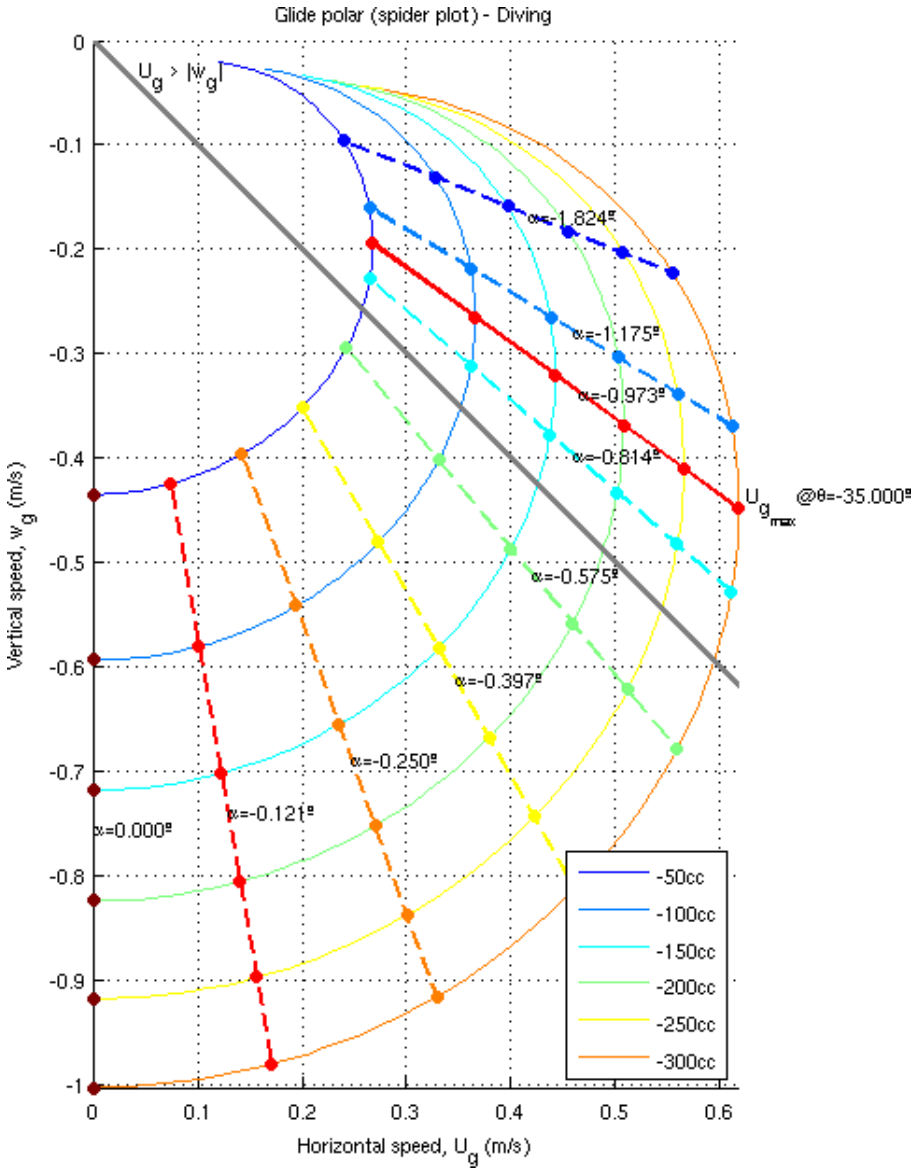


Figure 2.11: Spider plot for the force balance motion model during a dive. It shows the surge U_g vs. vertical w_g speeds of the glider for different changes of buoyancy ΔV_{bp} —typically the bladder or buoyancy pump capacity. The values above the $U_g > |w_g|$ line corresponds to the highest surge speeds U_g, and the most common values of pitch θ are picked up from that side. In fact, the greatest U_g is obtained at $\theta \approx -35°$, that yields $U_g \approx 0.5\mathrm{m/s}$ for $\Delta V_{bp} = -200$cc, which is in agreement with the usual gliding speed of SLOCUM gliders, for such pitch and buoyancy pump configuration.

α. Furthermore, we can also obtain the pitch θ using the fact that $\lambda = \theta + \alpha$. In both figures we have u_g and v_g for $\Delta V_{bp} \in [50, 300]$cc —which is negative during the dives. A typical common value of $\Delta V_{bp} = \pm 200$cc for the SLOCUM gliders gives the usual glider nominal surge speed U_g. Its value is generally measured at the optimal pitch θ angle, i.e. the one that provides the highest glider nominal surge speed $U_{g_{\max}}$, which

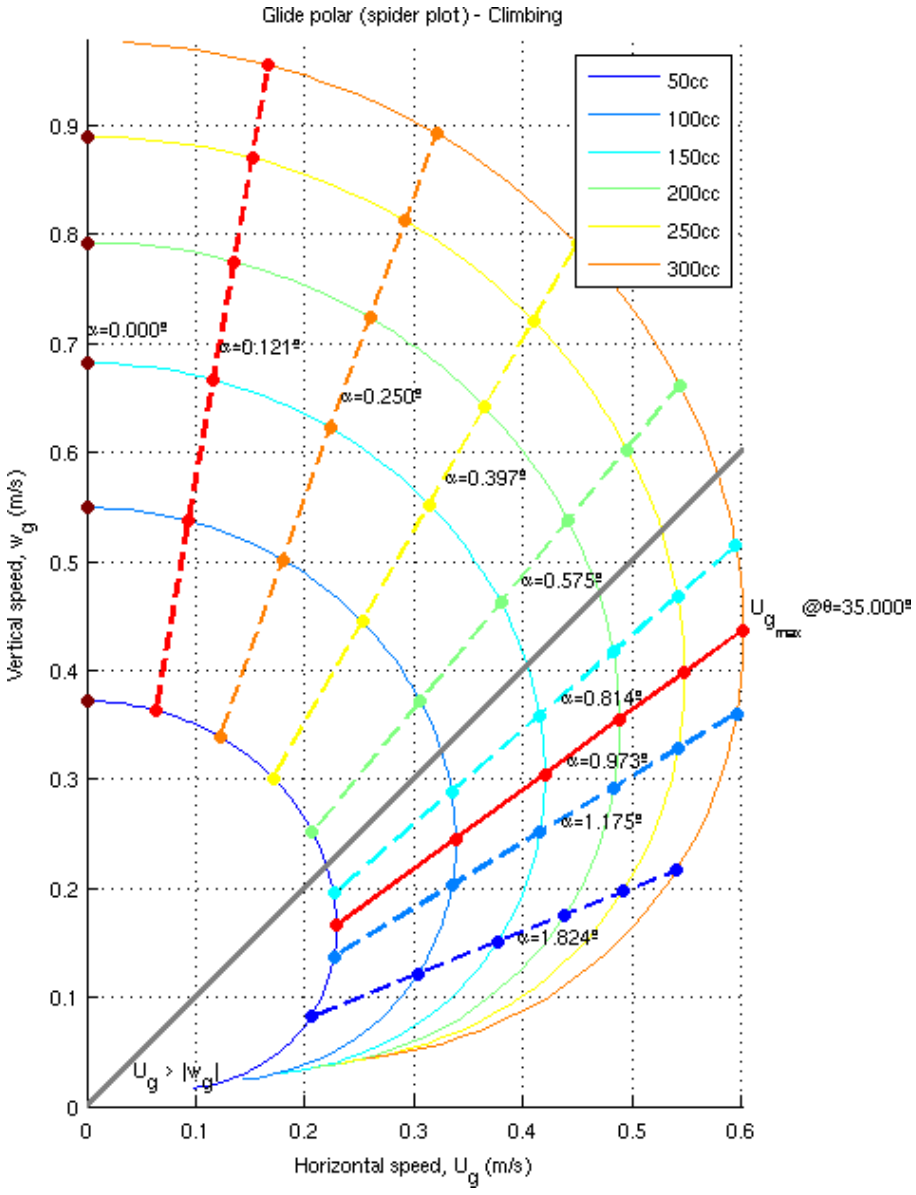


Figure 2.12: Spider plot for the force balance motion model during a climb. It shows the surge U_g vs. vertical w_g speeds of the glider for different changes of buoyancy ΔV_{bp} —typically the bladder or buoyancy pump capacity. The values below the $U_g > |w_g|$ line corresponds to the highest surge speeds U_g, and the most common values of pitch θ are picked up from that side. In fact, the greatest U_g is obtained at $\theta \approx 35°$, that yields $U_g \approx 0.5\mathrm{m/s}$ for $\Delta V_{bp} = -200$cc, which is in agreement with the usual gliding speed of SLOCUM gliders, for such pitch and buoyancy pump configuration. Also note that for all configurations the climb surge speeds U_g —and the vertical speeds w_g as well— are generally slower than the corresponding dive ones, shown in Figure 2.11.

in the figures is $\approx 0.5\mathrm{m/s}$. Note that in real conditions the θ is usually around $25°$, rather than $\theta = 35°$, which is the theoretical approximation to the optimal —actually, it must be solved numerically, as we do for

the figures. As expected, the plot shows clearly the increment on the glider speed with greater changes of buoyancy ΔV_{bp}, since it produces a higher buoyancy force F_B. Similarly, small angles of attack α have a clear effect, increasing the vertical speed w_g, but reducing the nominal surge speed U_g. For the figures, the highest nominal surge speed $U_{g_{\max}}$ is obtained with $\alpha \approx 1°$; meanwhile, with $\alpha = 0$ the models establishes the $U_g = 0$, i.e. no forward movement is produced, and the vehicle only goes down or up, with only a vertical speed of $w_g \approx \pm 0.8\mathrm{m/s}$ for $\Delta V_{bp} = \pm 200$cc.

From the glider plots we observe that the speed is slightly greater during the dives. This is a consequence of the balance of the gravity, lift and buoyancy forces, that favors the movement downwards. Above the line $U_g > |w_g|$ we have those glider setups that produce greater nominal surge speeds U_g. Below this line the glider is performing poorly and may be an indication, in real missions, of biofouling problems. With these graphics is quite easy to evaluate and identify some parameters of the glider during the mission. This is one of the strength of the force balance model, but it still requires some parameters to be set *a priori* that, as commented before, are difficult to measure.

Another important advantage over point motion models is that the force balance does not only consider the ocean currents, but also the water density, which affects the actual glider speed, as shown in the glider plots Figure 2.11 and 2.12. In essence, this model could be used in place of the unconstrained point motion model of Section 2.2.2.1, since it is also unconstrained; in fact, it will be difficult to make it constrained, to reach a given location $\mathbf{x}$, so it cannot be used in some graph-based path planning algorithm like A*.

This glider force balance model reduces to a similar force balance model of a profiler float or a drifter, for certain values of θ. This happens when the pitch angle θ becomes $\pm 90°$ for the dive and climb movements, in the case of a vertical profiler. Similarly, a drifter at a given depth is obtained with $\theta = 0$ and neutral buoyancy for such depths. The details of these variations of the force balance models are developed in Appendix B.

2.2.4 Dynamic Motion Models

There exists a number of works that model glider dynamics (Graver, 2005; Mahmoudian and Woolsey, 2008), but in general, the glider kinematics suffices to model the vehicle motion for all missions —not only for long-term ones. A dynamic motion models considers the forces and accelerations in the vehicle, and allow to characterize its motion accurately. However, for glider path planning problems, in which the vehicle navigates hundreds of miles or even in short-term missions, such accuracy is not required. Furthermore, due to the high uncertainty of the vehicle position underwater —especially x and y— and also the uncertainty of the ocean currents $\mathbf{U}_c$ and other physical parameter of water —e.g. density, temperature—, that accuracy is lost within the error introduced by the uncertainty of the localization and the ocean model.

According to Fossen (2002) the marine craft equations of motion can be written in the following *vectorial setting*,

$$\dot{\eta} = \mathbf{J}_{\mathbf{\Theta}}(\eta)\nu \tag{2.50}$$

$$\mathbf{M}\dot{\nu} + \mathbf{C}(\nu)\nu + \mathbf{D}(\nu)\nu + \mathbf{g}(\eta) + \mathbf{g}_0 = \tau + \tau_{\text{wind}} + \tau_{\text{wave}} \tag{2.51}$$

where

$$\eta = (x, y, z, \phi, \theta, \psi)^\top \tag{2.52}$$

$$\nu = (u, v, w, p, q, r)^\top \tag{2.53}$$

are the generalized pose η —composed of position $(x, y, z)^\top$ and attitude or orientation given by the Euler angles $\mathbf{\Theta} = (\phi, \theta, \psi)^\top$—, and velocities ν, for 6 DOF. Similarly, τ is a vector of forces and moments, which can be due to the wind τ_{wind} or waves τ_{wave}, which are usually neglected for gliders though. The model matrices $\mathbf{M}$, $\mathbf{C}(\nu)$ and $\mathbf{D}(\nu)$ denote inertia, Coriolis and damping, respectively, while $\mathbf{g}(\eta)$ is a vector of generalized gravitational and buoyancy forces. Static restoring forces and moments due to ballast systems

and water tanks —as the glider buoyancy pump and the actuated moving batteries to control the pitch θ— are collected in the term $\mathbf{g}_0$. Finally, $\mathbf{J}_{\mathbf{\Theta}}(\eta)$ is the Jacobian of the pose η with respect to the attitude $\mathbf{\Theta}$.

The differential equations of the marine craft for 6 DOF can be simplified for the particular case of a glider, in which we can consider only 5 DOF: x, y, z, θ and ψ; although the roll ϕ may still be worth modeling if we need high accuracy. Other simplifications are possible, as shown in the work of Graver (2005), but even in that case the model is still very complex and requires many parameters. In fact, Graver (2005) provides up to 10 parameters for the ROGUE glider, and similarly, for the SLOCUM glider, it performs several glide experiments for parameter identification. It is common to use flow simulation software to obtain hydrodynamic parameters like the drag and lift coefficients, or laboratory experiments. However, even during a glider mission these parameters may vary —e.g. biofouling—, and they change from one vehicle to another, or even for the same vehicle due to the trimming and on board equipment for a particular deployment. Hence, the efforts to parameterized a dynamic motion model correctly are far beyond the scope of this thesis. Additionally, this type of motion model would require much more computing time than the point motion model of Section 2.2.2, so the path planning algorithm would last more, and no significant benefit would be obtained from the simulation of the motion dynamics.

Chapter 3

Ocean Glider Path Planning Problems and Applications

Ocean gliders constitute an important advance in the highly demanding ocean monitoring scenario. Their efficiency, endurance and increasing robustness make these vehicles an ideal observing platform for many long-term oceanographic applications (Rudnick et al., 2004). Gliders are used for a great variety of tasks, not only in long-term missions, but also short-term ones. However, it is possible to reduce those tasks to a small number of basic problems, which can be arranged to solve more complex ones. In this section, we describe those basic problems and some applications that, in our humble opinion, are the most demanded by the Ocean Research community, where gliders are useful sampling platforms. Therefore, such basic tasks are presented here as optimization problems, which enable more elaborated tasks or applications. Among the latter we have covered path following problems, multi-glider coordination and constraint satisfaction, and sampling techniques for evolving structures on the ocean. All these applications are described in detail in the sequel, along with the oceanographic applications they relate with. Moreover, the obstacle avoidance problem is commented, since in many cases it is an implicit problem that must be solved by the glider path planning algorithm, as a basic feature. This taxonomy of path planning problems for ocean gliders will be later studied, to evaluate the different algorithms developed.

3.1 Path Optimization

We have identified two atomic glider path planning problems: time and distance optimization. Both of them consist in finding an optimal path by means of an objective function minimization, and consequently, here we explain these basic problems and the respective objective functions that model each of them. The first one looks for the minimal time path to reach a target waypoint $\mathbf{x}_{\text{goal}}$. This problem is equivalent to the Shortest Path Problem (SPP) for ground robots (LaValle, 2006). For gliders, navigating the ocean currents, the cost of traveling from one point $\mathbf{x}_i$ to another $\mathbf{x}_{i+1}$ is not only a function of the distance between those points, but also of the speed, which depends on the ocean currents profile. The second atomic problem in glider path planning consists on minimizing the remaining distance to the target waypoint $\mathbf{x}_{\text{goal}}$. In this case the travel time is asumed to be known and fixed beforehand, so this characteristic of the problem can be exploited by some path planning techniques.

Along with these atomic problems, we mention similar problems that can be seen as direct extensions of the atomic ones. This is the case of the minimization of the energy consumption, the optimal departure time and the minimal earliest arrival problems, which will be addressed by the minimal time path problem, and to some extent by the minimal distance to the target problem. Finally, we also highlight the differences

between propelled AUVs and gliders, in order to illustrate the importance of considering the constant time surfacing scheme performed by gliders. In fact, the difference between propelled AUVs and gliders is twofold. First, gliders navigate with a relatively low speed U_g with respect to the ocean currents U_c, i.e. $SR > 1$. Secondly, gliders cannot localize accurately while underwater, so the bearing angles ψ_g are commanded only every surfacing time t_s. This second limitation is also a crucial distinction between gliders and Unmanned Aerial Vehicles (UAVs), which may also travel with $SR > 1$ on the winds field, but they localize themselves continuously using the GPS and actuate the rudder often in order to set the appropriate heading ψ_e. It is precisely this low rate actuation in discrete intervals, usually only every surfacing time t_s, what makes a difference with respect to other vehicles performing almost the same task.

3.1.1 Minimal Time Path

The minimal time path problem is defined as the problem of finding the path $\mathbf{P}$ —made of a series of bearing angles $\mathcal{B}$— that takes the glider from a starting waypoint $\mathbf{x}_0$ to the target waypoint $\mathbf{x}_{\text{goal}}$ in the minimal travel time t_t. Each bearing is set at each surfacing point, so the number of bearings and surfacing points is the same. More formally, this minimization problem can be expressed as

$$\min_{\mathcal{B}} f(\mathcal{B}) \tag{3.1}$$

where $f(\mathcal{B})$ is a simulation function that applies —preferentially— the unconstrained motion model of Section 2.2.2.1 from $\mathbf{x}_0$ using each bearing in $\mathcal{B}$, so it reaches the ending point $\mathbf{x}_{n-1}$ that should lie within a radius around $\mathbf{x}_{\text{goal}}$ as Figure 3.1 shows. The number of bearings $|\mathcal{B}|$ to reach the target $\mathbf{x}_{\text{goal}}$ is computed in this optimization process, since $|\mathcal{B}|$ actually determines the travel time; recall that each bearing ψ_g is commanded after every surfacing time t_s.

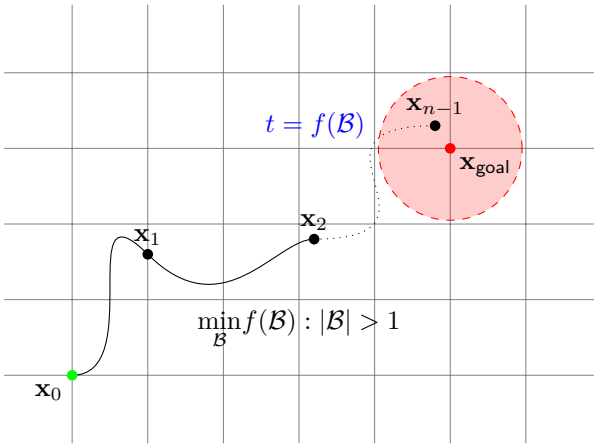
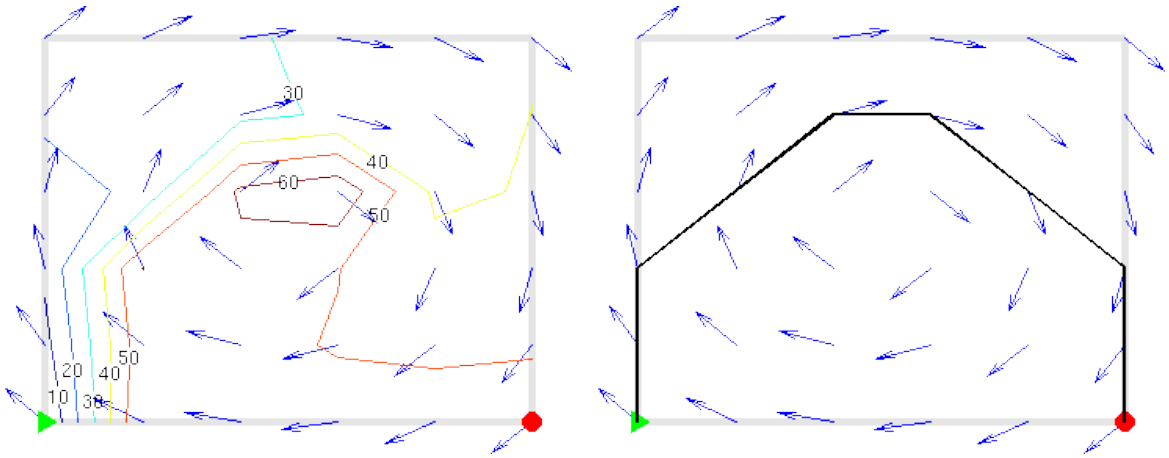


Figure 3.1: Diagram of the minimal time path problem. We look for the path $\mathcal{P}$ —made of a set of bearings $\mathcal{B}$— that minimizes the travel time $t_t = f(\mathcal{B})$ to go from the starting location $\mathbf{x}_0$ to the target one $\mathbf{x}_{\text{goal}}$. The number of bearings $|\mathcal{B}|$ is not fixed, since it actually determines the travel time. The target $\mathbf{x}_{\text{goal}}$ is considered as reached when the ending point $\mathbf{x}_{n-1}$ falls on a radius around it.

It is worth mentioning that in the field of Robotics, this problem is known as a single-pair SPP. However, some algorithms actually solve the more general problem of the single-source SPP. In the single-source problem, we have a starting point $\mathbf{x}_0$ and we compute the minimal time path —for gliders— to every point $\mathbf{x}_i$ in the map, not only $\mathbf{x}_{\text{goal}}$. It turns to be as efficient to solve the single-pair as the single-source SPP (LaValle, 2006). Therefore, the single-source problem can be solved with a wavefront approach as the one shown in



(a) Wavefront isolines. (b) Minimal Time Path.

Figure 3.2: Minimal Time Path problem as a wavefront expansion search in the vector field defined by the ocean currents and the glider nominal speed U_g. The optimal path search can be seen as a two steps procedure, in which we first obtain the (a) iso-lines of cost —which is time for our problem— in the vector field. Then, a gradient descend gives the minimal cost/time path $\mathcal{P}$ from the wavefront, as shown in (b).

Figure 3.2 (a), and the use a gradient descent method to retrieve the minimal time path to the target waypoint $\mathbf{x}_{\text{goal}}$, as shown in the figure (b).

In practice, the minimal time path problem is generally solved for long-term missions with durations from one to several weeks and trajectories length of several hundred kilometers. In large missions, we do not have dynamic ocean currents because the ROMs cannot predict further than a few days ahead. Therefore, although we can solve the minimal time path problem with both dynamic and static ocean currents, we will see than it is generally applied to static ones. Historic average ocean currents maps are used in these cases, for the seasonal period when the mission takes place. At most, this configures a quasi-stationary scenario for the glider path planning algorithm.

We have mentioned so far the equivalence between the SPP and the minimal time path problem. However, the SPP is thought for static environments, meanwhile we consider the minimal time path problem for both static and dynamic ocean currents. The Canadian Traveler Problem (CTP) is a generalization of the SPP to scenarios that are partially observable (Huang and Liao, 2012; Karger and Nikolova, 2008), i.e. the cost of moving is revealed while the environment is explored (Papadimitriou, 1991). The concept of partial observability may include the case of dynamic ocean currents, since we cannot know the actual cost of navigating from one point $\mathbf{x}_i$ to another $\mathbf{x}_{i+1}$ until we reach $\mathbf{x}_i$, i.e. until we determine the time t at which we arrive to $\mathbf{x}_i$, to take the ocean currents. This is regarded as a #P-hard problem addressed by relatively few algorithms (Papadimitriou, 1991). Equivalent problems are dynamic traffic routing, where the density of vehicles in the routes produces variable costs to travel within the routing network; and costmaps, which are typically associated with terrain-based path planning. In fact, we could build a costmap from the ocean currents vector field, but that would be equivalent to apply the constrained motion model. However, these problems still do not include the glider navigation and behavior features, like the constant time surfacing constraint.

The minimal energy consumption problem is tightly related to the minimal time path problem. Indeed, in the case of gliders, the lesser the travel time, the lesser the energy consumption. This is basically a consequence of the direct relation between the travel time t_t and the number of bearings $n = |\mathcal{B}|$, since the

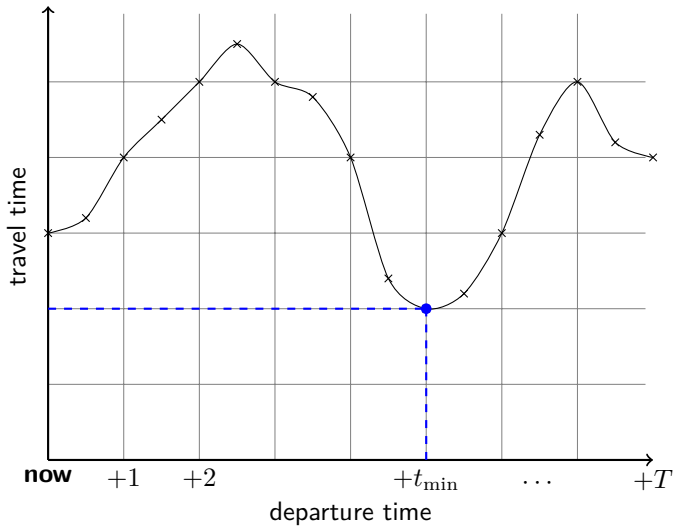


Figure 3.3: Diagram of the minimal departure time problem. Intuitively, the minimal time path problem is solved starting from $\mathbf{x}_0$ **now** and at different later times $t = \{+1, +2, \ldots, +T\}$, using some discretization step Δt. Then, the optimal departure time is just the global minimum of the resulting travel time curve, at $+t_{\min}$ in the figure.

surfacings lasts always almost the same time t_s, so

$$t = nt_s \tag{3.2}$$

Consequently, the number of inflections in the yo-yo profiles is also proportional to the travel time. Since most of the energy consumption employed in the navigation happens at those inflection points, it seems reasonable to say that we will obtain the same path for the minimal time and minimal energy consumption problems —for the same number of yos and depth limits. On the contrary, if the vehicle were a propelled AUV, since they have the ability to adjust their speed, the minimal energy consumption path is usually different from the minimal time one, clearly. For instance, Inanc et al. (2005) propose an objective function built as a weighted sum of the travel time and energy consumption for AUV path planning. Gliders, however, can only travel at a constant nominal surge speed U_g, and it is the number of inflection points what determines the energy consumption in terms of the navigation system. Therefore, only if we include the number of yos —or similarly, their depth limits— in the path planning algorithm, the minimal energy consumption path will differ from the minimal time one. However, it is very uncommon to change the number of yos during a glider missions. In fact, this could be in conflict with other requirements of the scientific mission.

In the literature of ocean glider piloting and planning there is a number of works that address additional optimization objectives and applications. The optimal departure time is covered in (Soulignac et al., 2009) with a Symbolic Wavefront Expansion (WE) algorithm. In Figure 3.3 a plot shows the travel time for several runs of the minimal time path problem. Each run starts at a different departure time, from now to a given maximum departure time $+T$ in discrete intervals Δt. As a result, we have several travel times that form a curve. Its minimum corresponds precisely to the optimal departure time, since it yields the minimal path time of all. Clearly, we can solve this problem by solving the minimal time path problem iteratively, although more efficient solutions may exist, as the Symbolic WE.

3.1.2 Path of Minimal Distance to the Target

The minimal distance to a target problem pursues the minimization of the remaining distance $d = \|\mathbf{x}_{\text{goal}} - \mathbf{x}_{n-1}\|$ from the ending point $\mathbf{x}_{n-1}$ to the target $\mathbf{x}_{\text{goal}}$ for a given travel time t_t, as depicted in Figure 3.4. Since the travel time t_t is given for this problem, the number of bearings $|\mathcal{B}| = n$ is fixed and known in advance as well, being

$$|\mathcal{B}| = n = \left\lceil \frac{t_t}{t_s} \right\rceil \tag{3.3}$$

where t_s is the surfacing time. More formally, this minimization problem can be expressed as

$$\min_{\mathcal{B}} \|\mathbf{x}_{\text{goal}} - \mathbf{x}_{n-1}\| \tag{3.4}$$

where $|\mathcal{B}| = n$. As we will see later in this chapter, using optimization methods (see Section 4.6), this particular problem has a more direct solution than the minimal time path problem discussed thus far.

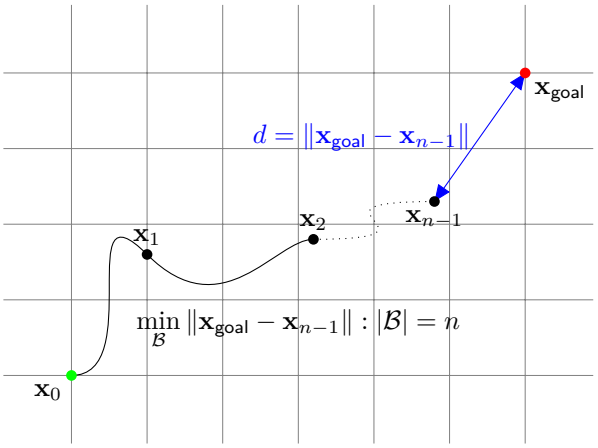


Figure 3.4: Diagram of the minimal distance to the target problem. We look for the path $\mathcal{P}$ defined by a set of n bearings $\mathcal{B}$ that minimizes the distance $d = \|\mathbf{x}_{\text{goal}} - \mathbf{x}_{n-1}\|$ from the ending point $\mathbf{x}_{n-1}$ to the target $\mathbf{x}_{\text{goal}}$. In this problem the number of bearings $|\mathcal{B}| = n$ is fixed and known in advance.

In practice, the minimal distance to the target is usually required in short-term missions, with a maximal duration of 3 to 5 days and a typical trajectory length below 150km. For this temporal horizon, state-of-the-art ROMs provide ocean currents in the form of predictions with a high space and temporal resolution. Indeed, most ROMs solve the ocean model hourly, and compile forecasts from now up to 3 days ahead. Consequently, the problem is generally solved using dynamic ocean currents, although it is also valid for the simpler case of static ocean currents.

A concomitant problem was addressed by Thompson et al. (2010) using a three-dimensional grid to optimize an *earliest valid arrival* criterion. The idea here is to reach the target waypoint $\mathbf{x}_{\text{goal}}$ not before a given time $t_{\min}$. This problem can also be solved by compiling several solutions of the minimal distance to the target problems, or by imposing a constraint such that the valid arrival time condition $t \geq t_{\min}$ is satisfied. The actual implementation will depend on the path planning algorithm. Indeed, a possible approach consists on solving the minimal distance to the target problem for the number of bearing $|\mathcal{B}|$ such that the path lasts at least $t_{\min}$. Then we iteratively test paths of greater length; although the inverse approach is also possible, i.e. we can start with a larger $|\mathcal{B}|$ and reduce it up to $t_{\min}$ as long as the distance to the target is within a radius that establishes the vehicle arrives to it. To some extent, and depending on the approach, this is actually a solution of the minimal time path (see Section 4.6.1).

3.2 Obstacle Avoidance

Any of the previous optimization problems are supposed to perform adequately not only in offshore, obstacle free environments, but also in troublesome scenarios where the obstacle avoidance problem must be solved. We find this kind of environments mainly in coastal areas, where we have static and dynamic/moving obstacles such as land areas, shallow waters, and heavy traffic shipping routes. Also strong ocean currents against the heading to the target waypoint $\mathbf{x}_{\text{goal}}$ can be regarded as obstacles, since gliders —with their low surge speed U_g— cannot beat them; note that with the inclusion of strong ocean currents as possible obstacle, even in offshore environment the obstacle avoidance must be solved.

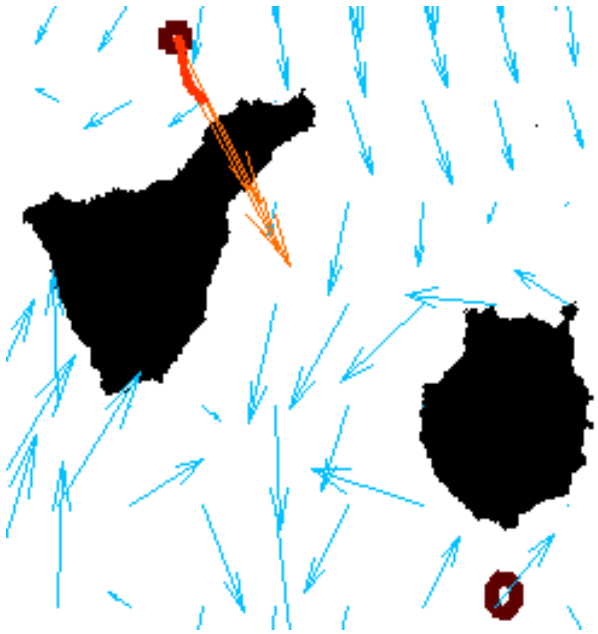


Figure 3.5: Obstacle Avoidance in glider path planning optimization problems. Obstacles in the ocean can appear in different forms: strong ocean currents against the desired heading, land areas —like the islands in the figure—, shipping routes, shallow waters, and basically any area where the glider cannot navigate doing its characteristic yo-yo profiles underwater.

In Figure 3.5 we have a typical coastal scenario with land obstacles —e.g. islands. Without any sort of obstacle avoidance logic in the glider path planning algorithm, the path gets easily trapped. In other words, by solving the obstacle avoidance problem we are actually trying to avoid local minima in the optimization process, which is also a consequence of the large number of optimization variables involved and the complexity of the environment. In general, a wide exploration of the search space allows to avoid the obstacles and find an initial rough path that can be optimized later. However, regarding ocean currents, it is also possible to segment the environment to avoid unstable regions. We will address this problem in Section 3.5, where we study different techniques to extract structures from the ocean currents, so we can reduce the high dimensionality of the problem.

3.3 Path Following

The path following problem was introduced by Eichhorn (2010) with the equivalent name of Hold Track problem. In brief, it consists on finding the path with the lowest deviation from an user-defined trajectory

$\mathcal{P}_d$, but still reaching the target waypoint $\mathbf{x}_{\text{goal}}$. Regarding the objective of reaching $\mathbf{x}_{\text{goal}}$, it is solved as the minimal time path problem discussed thus far. Meanwhile, the path following requirement can generally be integrated as a constraint in the optimization framework.

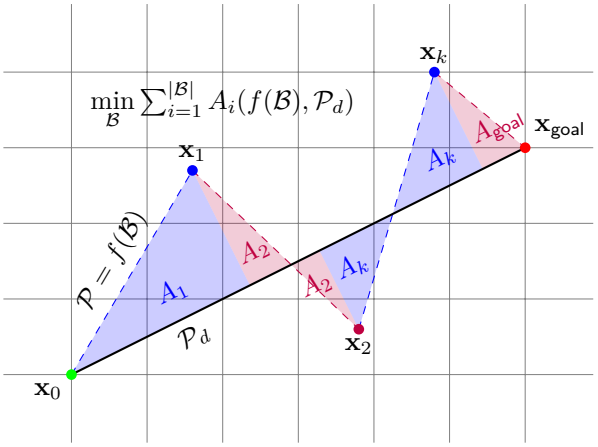
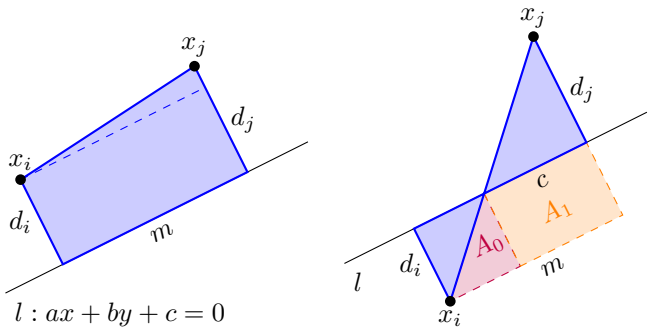


Figure 3.6: Diagram of the path following problem. We look for the path $\mathcal{P}$ defined by a set of bearings $\mathcal{B}$ that minimizes the deviation from a desired path $\mathcal{P}_d$. Such deviation can be measured as a distance d or an area function A. This path following problem is solved as a constraint on the minimal time path problem, that allows to reach the target waypoint $\mathbf{x}_{\text{goal}}$.

Figure 3.6 illustrates the path following problem, where the desired path $\mathcal{P}_d$ goes from $\mathbf{x}_0$ to $\mathbf{x}_{\text{goal}}$. The problem can be easily understood as the minimization of the area covered by the resulting path found $\mathbf{P}$ around $\mathcal{P}_d$. This is a piecewise area made of right trapezoids areas A_i for $i = 1, \ldots, |\mathcal{B}|$, i.e. for each stint —that is one unit less than the number of bearings $|\mathcal{B}|$, or surfacing points. Note that there are different ways to compute these areas, but the right trapezoid formula is valid also when the path $\mathcal{P}$ crosses $\mathcal{P}_d$.



(a) Right trapezoid. (b) Right trapezoid (crossing).

Figure 3.7: Right trapezoid diagram, for the computation of the area between the desired path $\mathcal{P}_d$ and the path found $\mathcal{P}$, in the path following problem. The right trapezoid area is computed for every two consecutive surfacing points $\mathbf{x}_i$ and $\mathbf{x}_j$. In (a) $\mathbf{x}_i$ and $\mathbf{x}_j$ lie in the same semi-plane, divided by the line l that defines $\mathcal{P}$. When they lie in different semi-planes (b), $\mathcal{P}$ crosses $\mathcal{P}_d$ and two sub-areas must be subtracted from the right trapezoid area.

Given the equation of the desired path $\mathcal{P}_d$ simplified as a line l for 2 dimensions in the following form

$$l : ax + by + c = 0 \tag{3.5}$$

and 2 consecutive surfacing point locations $\mathbf{x}_i$ and $\mathbf{x}_j$ that define the two-dimensional coordinates

$$\mathbf{x}_i = (x_i y_i)^\top \tag{3.6}$$
$$\mathbf{x}_j = (x_j y_j)^\top \tag{3.7}$$

we can compute the distance from the points $\mathbf{x}_i$ and $\mathbf{x}_j$ to the line l as follows

$$d_i = \frac{ax_i + by_i + c}{\sqrt{a^2 + b^2}} \tag{3.8}$$
$$d_j = \frac{ax_j + by_j + c}{\sqrt{a^2 + b^2}} \tag{3.9}$$

assuming an Euclidean metric space.

Using the normal vector $\mathbf{n}$ of the line l,

$$\mathbf{n} = (ab)^\top \tag{3.10}$$

and the distances d_i and d_j of the points $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, we can compute their projection onto l with

$$\mathbf{p}_i = \mathbf{x}_i - \mathbf{n}d_i \tag{3.11}$$
$$\mathbf{p}_j = \mathbf{x}_j - \mathbf{n}d_j \tag{3.12}$$

The distance of the projected points $\mathbf{p}_i$ and $\mathbf{p}_j$ can now be computed as the norm of their difference, which yields the base m of the right trapezoid of Figure 3.7. That is,

$$m = \|p_j - p_i\| \tag{3.13}$$

The distances shown thus far are signed distances, so we take their absolute value to obtain the heights h_i and h_j of the right trapezoid,

$$h_i = |d_i| \tag{3.14}$$
$$h_j = |d_j| \tag{3.15}$$

Consequently, the area of the trapezoid is defined as

$$A = \frac{1}{2}mh \tag{3.16}$$

where

$$h = h_i + h_j \tag{3.17}$$

When two consecutive surfacing points $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\mathcal{P}$ cross $\mathcal{P}_d$, we must subtract the two sub-areas A_0 and A_1 depicted in Figure 3.7 (b), from the right trapezoid area. In this case, the trapezoid polygon is complex. We can detect whether the trapezoid is a simple or complex polygon by checking the signed distances d_i and d_j. If they have different sign, the trapezoid is complex, and we must do the subtraction correction. We can see the following similar triangles equivalence in the figure,

$$\frac{a}{h} = \frac{c}{h_j} \tag{3.18}$$

where $h_j = |d_j|$.

Hence,

$$c = \frac{ah_j}{h} \tag{3.19}$$

We have two areas to subtract. First, a triangle of area

$$\begin{aligned} A_0 &= \frac{1}{2}(a-c)h_i \\ &= \frac{1}{2}\left(a - \frac{ah_j}{h}\right)h_i \\ &= \frac{1}{2}ah_i\left(1 - \frac{h_j}{h}\right) \\ &= \frac{1}{2}\frac{ah_i^2}{h} \end{aligned} \tag{3.20}$$

and, secondly, a rectangle of area

$$\begin{aligned} A_1 &= ch_i \\ &= \frac{ah_ih_j}{h} \end{aligned} \tag{3.21}$$

They sum

$$\begin{aligned} A_0 + A_1 &= \frac{1}{2}\frac{ah_i^2}{h} + \frac{ah_ih_j}{h} \\ &= \frac{ah_i}{h}\left(\frac{1}{2}h_i + h_j\right) \\ &= \frac{ah_i}{h}\left(h - \frac{1}{2}h_i\right) \\ &= ah_i\left(1 - \frac{h_i}{2h}\right) \end{aligned} \tag{3.22}$$

which is one of the several arrangements of this equation. Therefore, for complex right trapezoid, we compute the right trapezoid area A and then subtract $A_0 + A_1$, so we actually have $A - (A_0 + A_1)$.

Instead of the right trapezoid areas, we can simply use the distance from each surfacing point $\mathbf{x}_i = (x_i y_i)^\top$ to the line l,

$$d = \frac{ax_i + by_i + c}{\sqrt{a^2 + b^2}} \tag{3.23}$$

or a pseudo-distance,

$$d = ax_i + by_i + c \tag{3.24}$$

which is defined as a simplified version to speed up the computations. We can also choose between computing these distances for the two consecutive surfacing points $\mathbf{x}_i$ and $\mathbf{x}_j$, or just to the last one, i.e. $\mathbf{x}_j$. In Section 5.7 we will compare all these deviation metrics, in order to show that the simplified versions can be used in place of the actual right trapezoid area or the XTE formulas of Eichhorn (2010).

3.3.1 Relaxed Path Following

The problem of reaching a target waypoint $\mathbf{x}_{\text{goal}}$ in the minimal travel time while keeping the path $\mathcal{P}$ as close as possible to the desired path $\mathcal{P}_d$ is shown in Figure 3.8. A glider holds the track of a straight line along an anti-cyclonic (clockwise) eddy, where the minimal time path will not follow the desired path. It is also

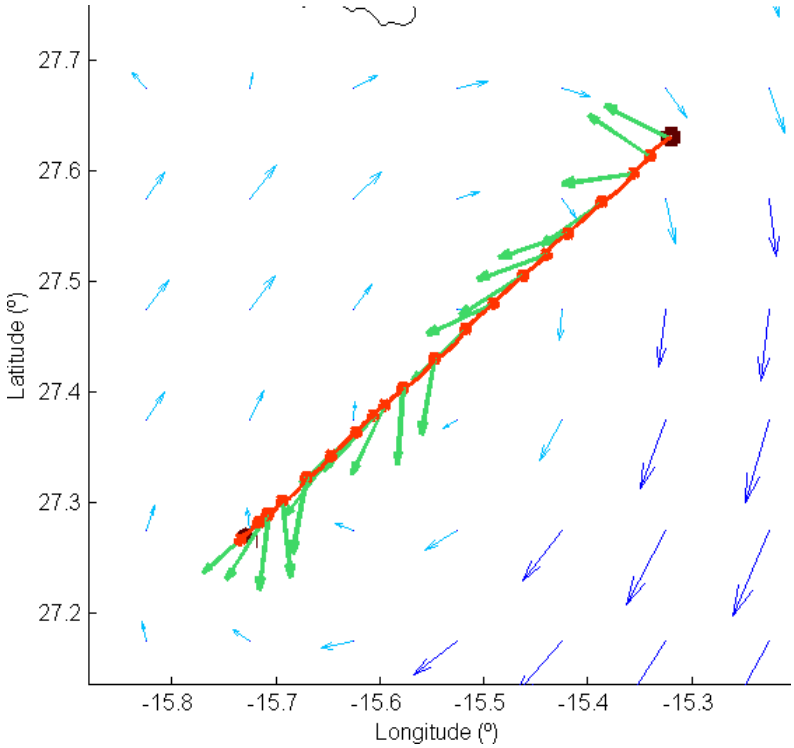


Figure 3.8: Path following example for gliders. The figure shows the closest path $\mathcal{P}$ found to follow the desired path $\mathcal{P}_d$ from the start $\mathbf{x}_0$ to the target $\mathbf{x}_{\text{goal}}$ waypoint. The glider bearings $\mathcal{B}$ ⟶ of the path $\mathcal{P}$ at each surfacing point are shown along with the ocean currents ⟶ (ocean currents that exceed the glider speed $U_g = 0.4\text{m/s}$ are highlighted ⟶) from the start ■ to the target waypoint ○.

possible to relax the path following constraint. This means that we can allow the path found $\mathcal{P}$ to deviate from $\mathcal{P}_d$ up to some given distance. This way we define a corridor of a certain width, so it is possible to find paths of less travel time to reach $\mathbf{x}_{\text{goal}}$, but still close to $\mathcal{P}_d$, as shown in Figure 3.9. Therefore, the travel time can be reduced at the expense of a small deviation from $\mathcal{P}_d$. Indeed, this relaxation of the constraint allows us to obtain a set of different solutions that define a Pareto curve, as we will show in Section 5.7.

3.3.2 Summary

In short, by solving the path following problem it is possible to obtain paths that reach the target waypoint, while they also remain close to an arbitrary trajectory defined, for example, according to a certain scientific interest. It is straightforward to extend the problem to arbitrary curves by a piecewise linear approximation, using the line desired paths shown before as a basis. It is also possible to use other functions directly, as far as we are able to compute the deviation between it and the path found —the area enclosed, or a distance from each surfacing point $\mathbf{x}_i$ to the curve.

In oceanography, gliders are useful for long-term missions and data assimilation campaigns, which are demanded by climate change research, to improve our knowledge of ocean processes and so on (Dobricic et al., 2010). In this kind of missions it is crucial that the vehicle samples the ocean across a given transect, i.e. it follows an usually linear path that goes from the start $\mathbf{x}_0$ to the target $\mathbf{x}_{\text{goal}}$ waypoint. For instance, it is common to cross the eddy from one side to another passing through the center, in order to estimate its dimensions and measure its properties (Martin et al., 2009). With gliders, due to their relative slow nominal

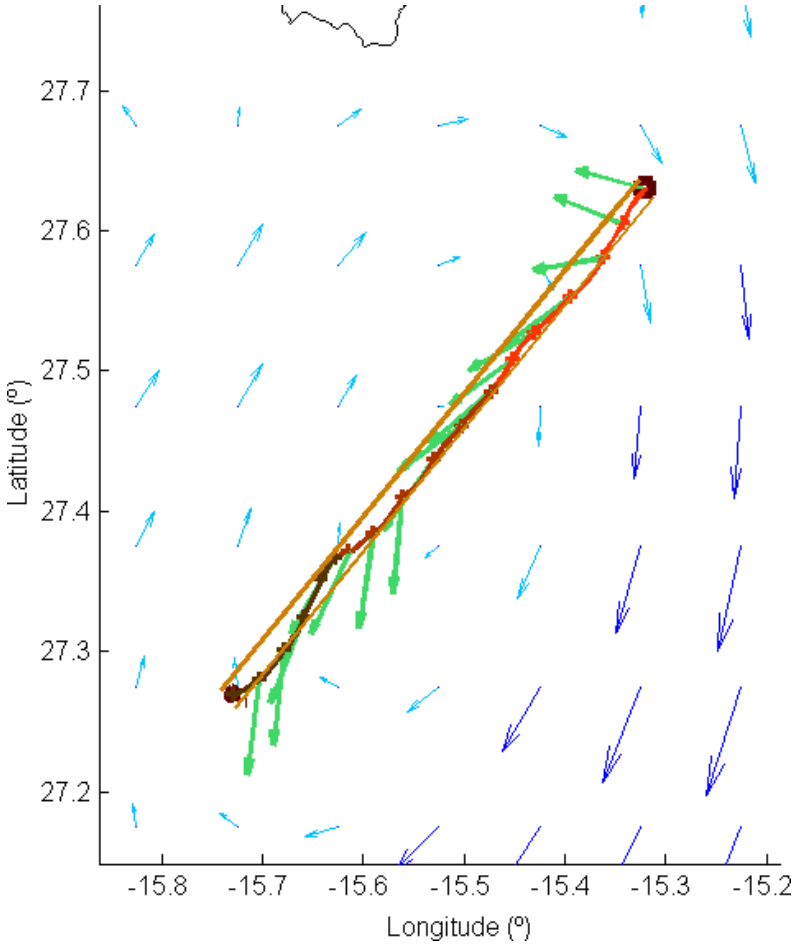


Figure 3.9: Path following example for gliders. The figure shows the closest path $\mathcal{P}$ found to follow the desired path $\mathcal{P}_d$ from the start $\mathbf{x}_0$ to the target $\mathbf{x}_{\text{goal}}$ waypoint, within a corridor around $\mathcal{P}_d$ —i.e. the path following problem is relaxed. The glider bearings $\mathcal{B}$ → of the path $\mathcal{P}$ at each surfacing point are shown along with the ocean currents → (ocean currents that exceed the glider speed $v_g = 0.4\text{m/s}$ are highlighted →) from the start ■ to the target waypoint ○.

speed, the path following problem described here cannot be solved with control theory techniques. The problem is such that it requires a global vision of the environment and some memory in order to take the best path and avoid situations in which the vehicle could get trapped and not reach the target.

3.4 Multi-Glider Path Planning and Coordination

In the last years a line that has received a good deal of attention by many researchers is the use of multiple vehicles in coordinated missions. Some relevant examples include the works of Zhang et al. (2007) and Bhatta et al. (2005), which face the problem of adaptive sampling of oceanic variables by means of a glider fleet, maximizing the area covered and hence improving the spatial and temporal resolution. The multi-glider problem is concerned with the coordination of multiple vehicles at the path planning phase to accomplish coverage tasks that are solved more efficiently by a fleet of gliders navigating in certain formation or according

to some constraints. In fact, it is possible to implement a glider path planning algorithm by imposing coordination constraints to the objective function solved for each vehicle in the fleet.

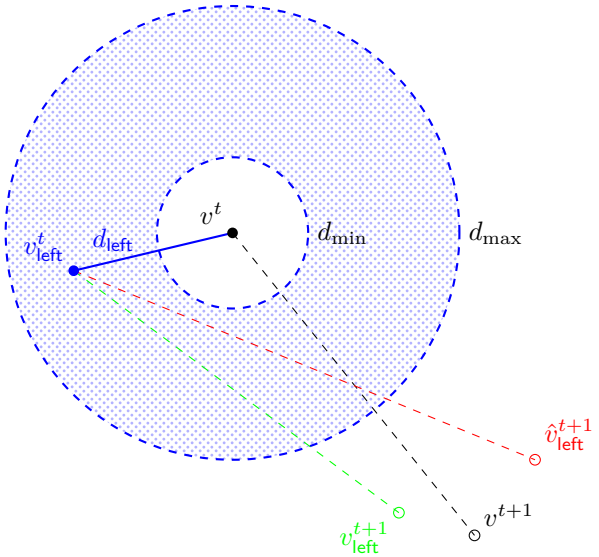


Figure 3.10: Multi-glider coordination constraint diagram, in which each glider sets its trajectory according to a constraint. In this case, at a given time t, for a particular glider or vehicle v^t in the fleet, we impose that the distance d_{left} with respect to the left glider v^t_{left} must fall within the range $[d_{\min}, d_{\max}]$. Also, note that for a given time t we check that the stints do not intersect —i.e. they do not cross each other. For this reason, the stint from v^t_{left} to $\hat{v}^{t+1}_{\text{left}}$ is discarded. We will proceed similarly for a right glider v^t_{right}, if present.

Basically, this task can be defined as an extension of the atomic problems explained thus far to multiple gliders. In the example shown in Figure 3.10, these gliders are forced to travel within a distance range among them. To be more specific, given a glider v, at each surfacing point we should check that its neighbors lie in the range $[d_{\min}, d_{\max}]$. Only those paths in which that condition is satisfied are considered as valid. Also, another check for segment intersections is required, to prune stints that cross others. In the example of the figure, only the stint in which the glider v_{left} surfaces at v^{t+1}_{left} is valid then. These constraints must be applied during the optimization phase, so the paths found for each glider v are still optimized for the given objective function, but they also satisfy the coordination constraint. This constitutes a form of coordinated path planning, that solves a multi-glider problem for the whole fleet simultaneously. In the example of the figure we have considered a particular constraint that keeps the vehicles close to each other navigating in formation, but we might have adopt other constraints. In fact, in the next section we address a common problem in real missions that consists on minimizing the time to recover a fleet of gliders.

3.4.1 Efficient Recovery

The efficient recovery is a multi-glider path planning problem in which a fleet of k gliders must be recovered by a ship s. The problem is depicted in Figure 3.11, where the starting location of the gliders at $t = 0$ is represented by the surfacing points v^0_i, and the ship s is waiting at the harbor for a time T before it goes to recover them. Therefore, the vehicles have a time T to reach the locations v^T_i that will allow to minimize the recovery time. That is, the locations v^T_i such that the ship recovery route $\mathcal{P}_r$ will be the shortest —or minimal time one, which are generally the same because we assume the ship travels relatively fast with respect to the winds and ocean currents.

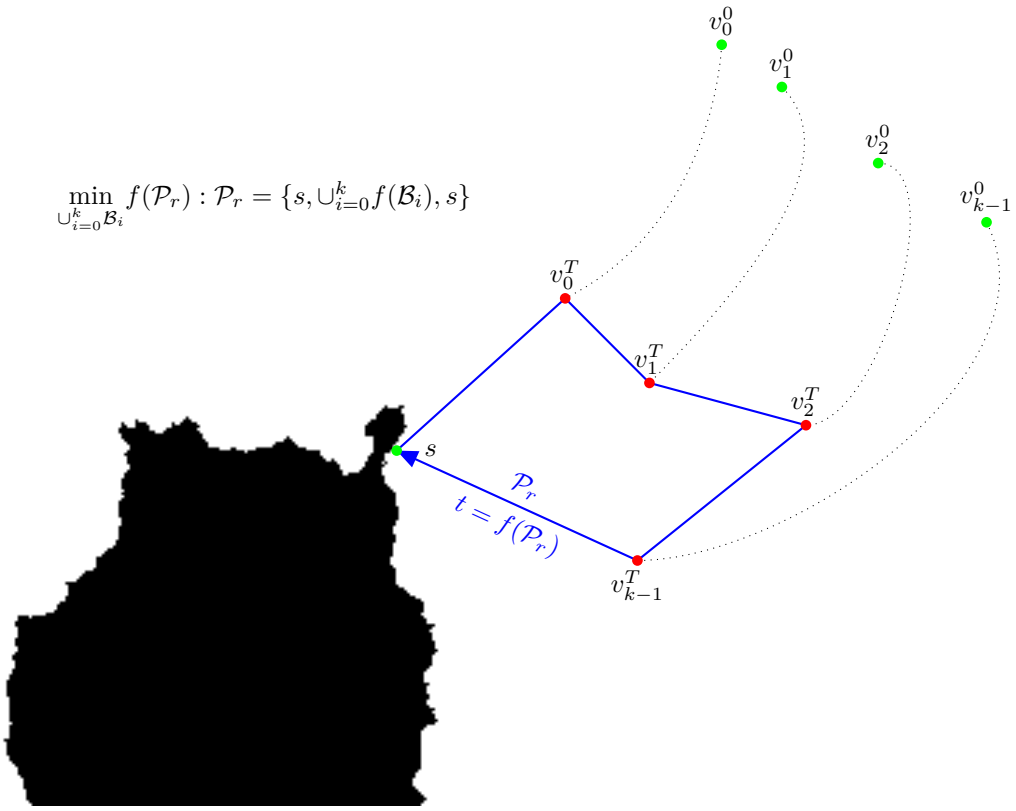


Figure 3.11: Diagram of the efficient recovery problem. A fleet of k vehicles moves to the most appropriate location v_i^T during a given time T, for each vehicle $i = 0, \ldots, k-1$, so the path $\mathcal{P}_r$ to recover them with a ship s takes the minimal travel time $t_t = f(\mathcal{P}_r)$.

There exist different ways to solve the efficient recovery problem. Apart from the individual path planning case, in which each glider solves the minimal distance to the target problem independently for the given time T, trying to reach the harbour or the ship location s, we can identify two coordinated, multi-glider path planning approaches. First, we can impose a constraint such the vehicle end almost together after the time T. This way we have a meeting point —which is found by the path planner, not given by the user—, so the ship only have to go there and back to the harbour. However, the optimality of this approach is usually dominated by the glider that is further from s, as we will see in the experiments of Section 5.8. A second approach actually minimizes the recovery time, i.e. the travel time $t_t = f(\mathcal{P}_r)$ required by the ship to departure from the harbour s and recover each glider v_i in sequence at their ending locations v_i^T, and comes back to s. This is exactly the problem we want to solve, so this approach will provide the optimal solution, and will produce a solution similar to the one shown in Figure 3.11.

3.5 Tracking Evolving Ocean Structures

Ocean gliders have proven to be also useful in the opportunistic short-term characterization of dynamic structures. Among these, mesoscale eddies are of particular interest due to the relevance they have in many

oceanographic processes. In the ocean we find different evolving structures, which can be defined as blocks or regions of water that exhibit certain characteristics as a whole. The mass of water that constitutes these structures usually moves over time, so it is not trivial to track and sample them. The problem of tracking evolving ocean structures requires path planning algorithms that allows a glider to track them, while it also takes samples in order to characterize it. This problem can usually be solved by combining several path following sub-problems.

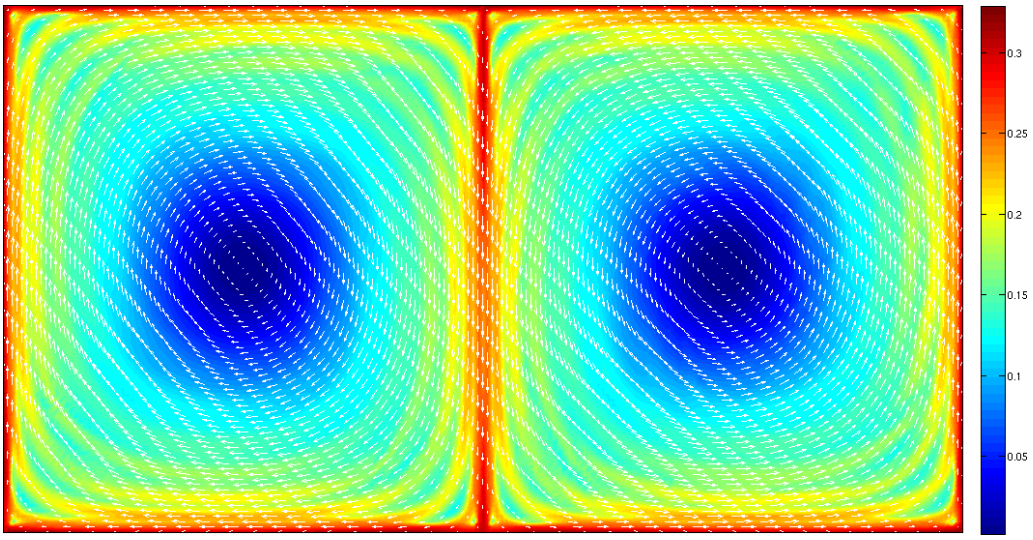


Figure 3.12: Finite Time Lyapunov Exponents (FTLE) computed on a double eddy synthetic ocean currents field —overlaid with white arrows. The FTLE values represent the stability of every region in the field, such that unstable regions —like the eddies boundaries— are in red and stable ones are in blue —as the centroid of the eddies.

The most common evolving structures in the ocean are ocean eddies. The centroid and boundary points of eddies associated with bloom episodes are tracked in the work of Smith et al. (2010). However, we can also define regions according with their temporal stability. In fact, there is a number of works that use Finite Time Lyapunov Exponents (FTLE) —or Direct Lyapunov Exponents (DLE)— to analyze the ocean currents stability (Kasten et al., 2012; Robel, 2010; Shadden et al., 2005). In Figure 3.12 we have the FTLE for a double eddy synthetic ocean currents field. The centroid c of the eddies are stable regions, because any drifting object that starts there, will end up there as well after some time T. On the contrary, the interface between the eddies is very unstable, because the particles there may end up at very different locations after T.

This analysis produces Lagrangian Coherent Structures (LCS) that can simplify the glider path planning problems to some extent, since we can reduce the regions explored for the optimal path to those LCSs. In a work of Inanc et al. (2005), the FTLE is computed for the ocean currents of the region where some glider missions will take place. The LCSs are used to obtain some insight of the stability of certain zones of the basin, and that information can be plugged into the path planning techniques, in order to avoid unstable structures, since the uncertainty in the location of the vehicle may lead to bad results on the field tests. LCSs are also evolving structures, since depending on the integration time T, they cover different regions or change their intensity. Therefore, we can also track the regions they define —or the ones they enclosed—, in order to characterize the actual stability of such regions on the field.

3.5.1 Ocean Eddies

In the particular case of eddies, the complexity of the path planning scenario is aggravated by the high spatio-temporal variability of these structures and their specific sampling requirements (Hátún et al., 2007). Indeed, even in the work of Garau et al. (2005), the A* search algorithm is used to find optimal paths over a set of eddies with variable scale and dynamics. Smith et al. (2010) propose an iterative optimization method based on the Regional Ocean Modeling System (ROMS) predictions to generate optimal tracking and sampling trajectories for evolving ocean processes. More specifically, a fleet of multiple vehicles is used to track the centroid and boundary of evolving ocean processes. Their scheme includes near real-time data assimilation and has been tested both in simulation and real field experiments.

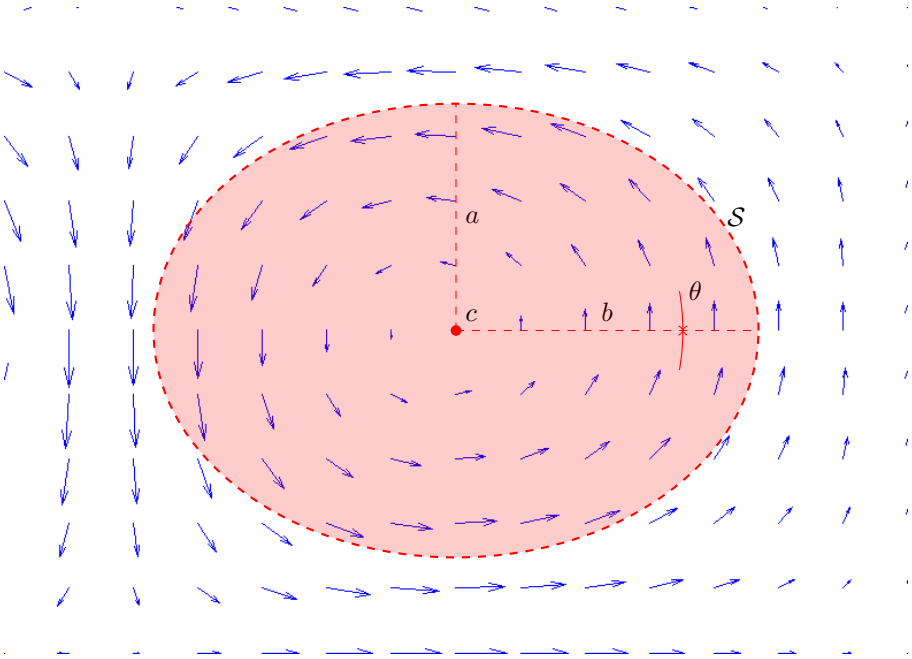


Figure 3.13: An eddy is an evolving structure that can be modeled as an ellipse of half axes a and b, and orientation θ. The center of the ellipse lies on the eddy centroid, while the perimeter of the ellipse approximates its boundary.

In order to track an ocean eddy we need some sort of model of it. In Figure 3.13 we have an ellipse model for the surface footprint of mesoscale ocean eddies. The ellipse is centered at the centroid c of the eddy, and the half axes a and b, along with its orientation θ, allows to represent the eddy boundary with the ellipse perimeter. We could also model the eddy volume in 3 dimensions, as we will see in the algorithm designed in this thesis (see Section 4.8). The problem that it solves consists on generating the glider trajectories that optimize the sampling of the evolving eddy structure, according with the given objective functions and the ROM forecast maps. Similarly, the problem can be extended to multiple vehicles, in which the problem scales to the coordination of glider fleets to define optimal sampling strategies (Leonard et al., 2010; Smith et al., 2010).

It is worth mentioning the tight relation between ocean eddies and ocean processes that take place beneath the surface. For instance, in some cases they may originate Harmful Algal Blooms (HABs) because of the vertical transport of inorganic materials from the seabed (Lopez et al., 2008). Consequently, a good sampling strategy of this evolving structures should consider their three-dimensional volume. The temporal analysis of HABs and their formation constitutes an important application of this problem in Ocean Research. It has

a huge interest in data assimilation and ocean modeling, as well as in the study of biological processes that occur associated with mesoscale eddies. This kind of eddies has $\approx 50 - 100\text{km}\varnothing$, so a glider traveling at $U_g = 0.4\text{m/s}$ will spend $\approx 1.5 - 3$ days to do a transect from one side to another crossing the centroid of the eddy. This means that a typical mission of at least 4 transects usually lasts more than 1 week, which is time enough to appreciate how the structure evolve or move spatially over time.

3.5.2 Sampling patterns

Although we can sample evolving structures with multiple vehicles, in this work we have only explored the case of a single vehicle. In particular, we consider the windmill pattern of Figure 3.14 to sample mesoscale eddies —which also resemble Foucaults's pendulum curve. This pattern starts at any point $\mathbf{x}_0$ at the eddy boundary, and crosses the eddy centroid c traveling in a straight line to goes to the opposite side of the eddy boundary, as the curve of the figure shows. After crossing the eddy, the vehicle moves on the boundary, so the next transect crossing the structure traverse a different area. This way, we can obtain a better characterization of the centroid c and the boundary S of the eddy. Note that this pattern is actually a concatenation of path following problems, where the path is the windmill pattern compose of line transects crossing c and arcs in the boundary, which can be simplified to lines as well.

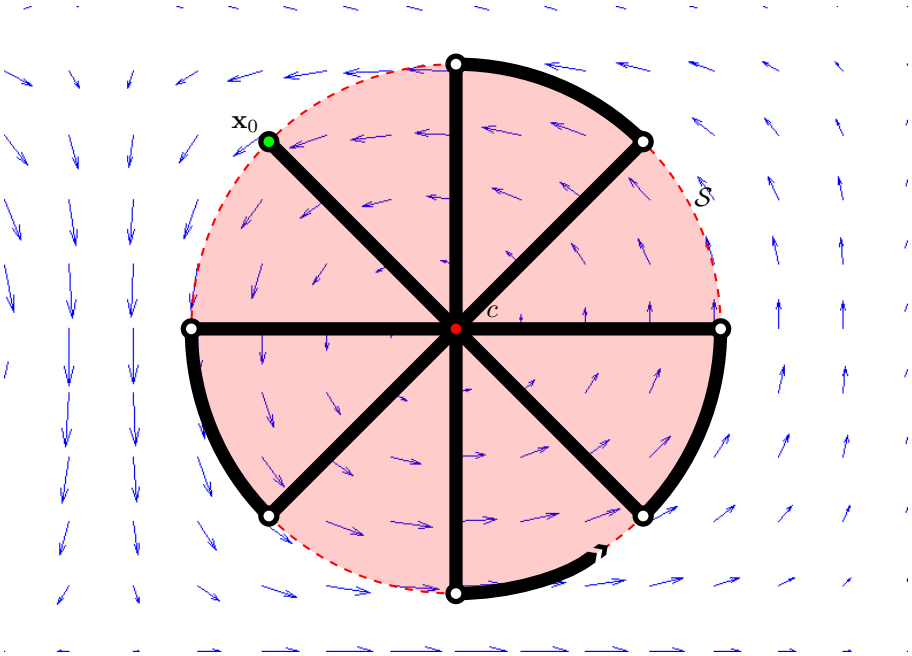


Figure 3.14: Windmill eddy sampling pattern. For a circular eddy, the windmill pattern defines a path that successively crosses the center c of the eddy with a series of transects starting at $\mathbf{x}_0$. Each transect is a linear path that goes from one side of the eddy to the opposite. There, it travels some distance through the eddy boundary before crossing the eddy again.

Although different patterns can be defined, this windmill pattern provides a good coverage of the eddy and it is thought to focus on the two main features of the structure: the centroid c and the boundary S. With another pattern, like a lawnmower, we would have good coverage, but c would be crossed only once. Since the structure is moving over time, it is important to sample the centroid at the higher possible rate, so we can track the evolving structure adequately. Therefore, this problem focus on the exploration and identification or

characterization of the evolving structures. Furthermore, it is also in connection with data assimilation tasks, where sampling complex regions is important to feed the ocean models.

Chapter 4

Ocean Glider Path Planning

In this chapter we address the glider path planning topic for the problems and applications of the previous chapter, using the glider motion models of Section 2.2. In the robotics literature, Path Planning covers the topic of finding a feasible and usually optimal path to allow a robot to move autonomously from one location to another in the environment (Lavalle, 1998). Here we will discuss the state-of-the-art path planning techniques and our proposals to deal with the motion model of ocean gliders.

A general view of Path Planning in Robotics precedes a survey of the state-of-the-art glider path planning techniques, discussing their assumptions and drawbacks. The detailed explanation of the glider path planning techniques developed in this thesis follows. The last sections are allocated to discuss feature-based path planning approaches for AUGs, which rely on the detection of physical ocean structures, like mesoscalar eddies, or the extraction of unstable regions computed by means of stability analysis.

4.1 Glider Path Planning Survey

In Robotics, path planning addresses the problem of getting a robot from one point to another. It is one of the three major tasks in any robotic system, as depicted in the diagram of Figure 4.1. This thesis focus on path planning and, to some extent, in exploration or coverage path planning. This task is very challenging when it is to be solved under the influence of ocean currents and for slow speed vehicles. The dynamic ocean currents velocity field directly affects the movement of the vehicle, producing a drift from the expected trajectory. Compared to ground mobile robotics, the ocean environment is much more challenging, since operating conditions can vary notably even on reduced areas and over a relatively short period of time. In terms of path planning, the current field conforms an asymmetric and variable cost map.

In the particular case of ocean gliders, all the mentioned difficulties are magnified, because of their low surge speed and the low rate of actuation only at the surfacing points. Although this problem is very similar to path planning Unmanned Aerial Vehicles (UAVs) affected by winds, there is a key difference. Ocean gliders can only modify their bearing ψ_g at discrete times, i.e. when they are on surface. Also, at least this single Degree of Freedom (DOF) ψ_g must be considered within the configuration space.

For these reasons, most of classical approaches in the path planning field are not directly applicable to this problem. Many path planners apply a certain form of discretization, either on the trajectory or the command/configuration space, to reduce the computational cost. However, the downside of discretization lies in the presumably degradation of the quality of the results, that might lead to unrealistic trajectories. The computing time is another factor which is often understated due to the typical long duration of glider missions and immersion periods. Although this is generally true, it is not the case when the path planner must respond within a reduced time interval to face an unforeseen situation. Therefore, short-term scenarios

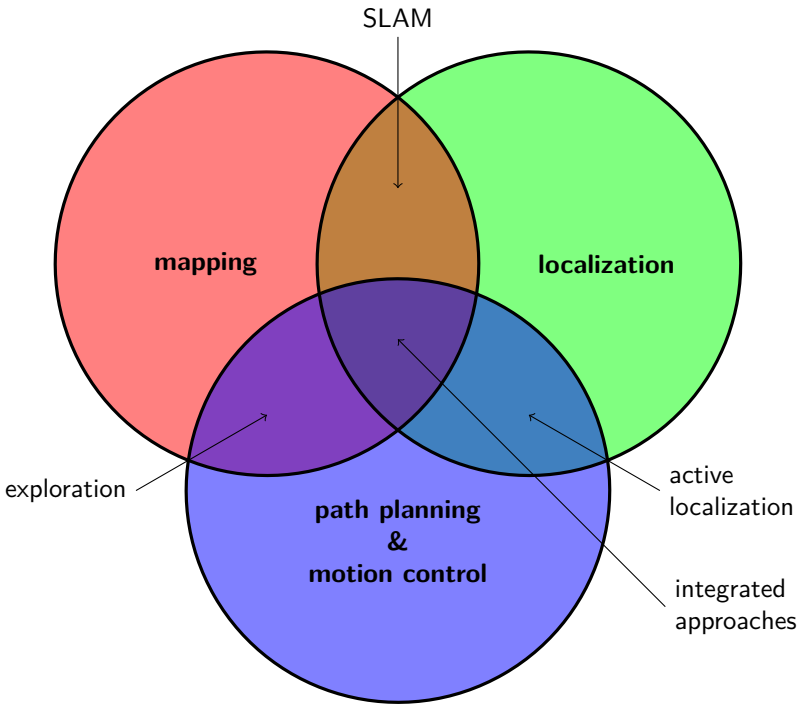


Figure 4.1: Diagram of the tasks that must be solved by a robot in order to acquire accurate models of the environment; figure reproduced from (Stachniss, 2009). The overlapping areas represent combinations of the main tasks, which cover mapping, localization and path planning.

with dynamic ocean currents are also covered by glider path planning techniques, as we will see along this chapter.

For the sake of space below we only explain in detail the techniques applicable to glider path planning, which are actually, the ones analyzed in this thesis, including adaptations or extensions for some of them and even some novel proposals. Path planning for Autonomous Underwater Gliders (AUGs) has been a subject of interest for researchers since the introduction of these robotic platforms. Different approaches have been developed applying techniques that include searching algorithms based on Artificial Intelligence, potential field modeling, multi-objective optimization, etc. Some of the most relevant, in our opinion, are summarized in the following sections.

4.1.1 Graph-based methods

Graph-based methods are a classical path planning approach that lies in the category of Discrete Optimal Planning (LaValle, 2006). A grid-shape graph represents the search space with the edges labeled indicating the cost of traveling from a vertex to one of its neighbors. A* (Hart et al., 1968) is probably the first graph method adapted to the problem at hand (Garau et al., 2005), with the strong assumption of static ocean currents; although it can be run with dynamic ocean currents at the expense of losing the guarantee of optimality in the general case. Also Pêtrès et al. (2005) and Soulignac (2010) apply A*, with a thorough analysis of the cost function in the latter. Note that some authors call this method Wavefront Expansion (WE), which actually does not use any heuristic and therefore it is Dijkstra's algorithm basically (Dijkstra, 1959). In this paper, our implementation of A* is equivalent to Garau et al. (2005, 2009). In another paper, Soulignac et al. (2009)

address time-varying winds, combining WE and local optimization for each path explored.

As most graph-based path planning methods, A* discretizes the search space using an uniform grid. As a consequence, the cost function must compute the travel time to go from one node n_i to its neighbors, using the constrained glider motion model of Section 2.2.2.2. The inconvenient of this approach is twofold. First, we only find the optimal path that lies on the search grid, while the actual optimum lives in the continuous space defined by the differential equations of the unconstrained glider motion model described in Section 2.2.2.1. Secondly, in the presence of strong ocean currents, many neighbors are unreachable, and the search gets easily trapped or discards many paths prematurely. Although we can increase the resolution of the search grid, the problem is actually associated to the heading angles ψ_e defined by the topology of the grid, as the set of headings is restricted to $\psi_e \in M\pi/4$ for $M = 0, \ldots, 7$ in an 8-connected grid. Carroll et al. (1992) apply A* on a quad-tree search space, which is adapted to the ocean currents field, i.e. it has higher resolution where the ocean currents vary more spatially; or more formally, where the gradient of the ocean currents is greater. Similarly, the Fast A* of Walsh and Banerjee (2010) claims to mitigate the discretization problem as well, by varying the search map resolution as needed and calling A* repeatedly; it has not been applied for glider path planning though.

There exists a number of variants of A* that are worth to mention. Any-angle methods, like Theta* (Daniel et al., 2010; Nash et al., 2007), try to obtain shorter paths alleviating the angle discretization problem caused by the search grid. However, a visibility concept require by this technique is not applicable in the presence of ocean currents, as their effect on the trajectory must be taken into account. Incremental methods, like D* (Stentz and Mellon, 1993), D* Lite (Koenig and Likhachev, 2002) or Incremental Phi* (Nash et al., 2009), aim to reduce the exploration time using previous solutions when partial changes are produced in the map. In the case of ocean currents the change is global and therefore it is not possible to exploit already computed solutions. Finally, post-smoothing techniques have the drawback that small changes in the path found might produce very different, even unfeasible, paths due to the influence of ocean currents. Indeed, with different locations in the smoothed path, the whole path must be integrated again using the glider motion model from the start location $\mathbf{x}_0$.

4.1.2 Probabilistic Sampling-based methods

Probabilistic Sampling-based methods have been applied to overcome the combinatorial explosion of the optimal path planning search to some extent. Rapidly-exploring Random Trees (RRT) (Lavalle, 1998; LaValle and Jr., 1999; Simmons and Urmson, 2003) constitute an incremental sampling and searching approach that has been applied to both AUVs (Tan et al., 2004) and gliders (Rao and Williams, 2009). The main drawback of this method in our context is the inability to handle time-dependent ocean currents (see Section 4.5). The RRT algorithm is very fast, but it does not assure the optimality of the paths found and often requires further refinement, which is not straightforward in the problem at hand. More precisely, we cannot separate the path planning procedure into a coarse search followed by a post-processing refinement, as it is common practice in classical Robotics. Post-smoothing techniques rely on some sort of continuity in the distance or cost within the search space. However, the presence of ocean currents in the glider path planning problems produce important changes in the vehicle resulting trajectory if we modify it, since it becomes influenced by different ocean currents —when it travels through different locations. This happens because the ocean currents field is defined on a continuous space, so any change on a given path —as the one found with the RRT algorithm— implies that the path is affected by different cost values, which are neither homogeneous, nor predictable without actually recomputing the whole path. That is, we must run the path planning algorithm again. In the case of RRT this is a serious drawback, since this algorithm is thought to find paths fast, but they are generally sub-optimal, rough solutions.

4.1.3 Optimization methods

Bio-inspired optimization solutions can also be found for this problem. Genetic Algorithms (GAs), for example, are used in (Alvarez et al., 2004) for AUV path planning in the presence of variable currents. The main problem is that it has execution time limitations, however, prevent the use of a high number of generations, reducing the quality of the path found. Ant Colony Optimization (ACO) algorithms constitute another interesting alternative used for ship route obstacle avoidance (Tsou and Hsueh, 2010), but is not directly extensible to highly dynamic environments. Simulated Annealing (SA) is a global optimization technique applied to path planning, usually as a post-processing refinement over previously generated results (Kirkpatrick et al., 1983). It is very sensitive to the initial guess, which must be selected wisely. As an example, in (Witt and Dunbabin, 2008) it is incorporated as part of a swarm optimization method.

Some authors have addressed the problem using **System Theory** and Analytical Methods (Techy et al., 2010). In this field, optimal paths are obtained solving a Boundary Value Problem (BVP). Ocean gliders can be modeled as a Dubins car, so Dubins curves are suitable in the case of steady flow —i.e. static ocean currents—, with some variants allowing for turn constraints. Furthermore, Zermelo's optimal navigation formula is applicable for unsteady flow —i.e. dynamic ocean currents—, but without turn constraints. In either case, these are local path planning methods that perform poorly with strong currents, so only propelled vehicles might benefit from them. These techniques are thought for short-term motion control, rather than long-term path planning, and they do not reflect the concept of constant time surfacings, since they control the vehicle heading at a higher rate.

The approaches based on minimization of **Energy Functions** are also worth commenting. As good examples, we can cite the work of Kruger et al. (2007), that includes the time as an extra dimension in the search space, or Witt and Dunbabin (2008), which incorporate modeling of time-varying obstacles using **Potential Fields**. The main drawback of potential fields is that they often get trapped in local minima. The problem of local minima has been tackled by means of strategies based on particle swarms, simulated annealing, or genetic algorithms. Another problem with potential field methods is their adaptation to dynamic ocean currents. It is very inefficient to re-compute the potential field for the whole map for each time instant. Furthermore, we must select a time step Δt equal to the surfacing time t_s to do so, and even in that case it will not cover the ocean currents variability while the glider navigates underwater.

In other proposals, the currents are modeled as continuous time functions, as is the case of the Non-linear Trajectory Generation or NTG method (Milam et al., 2000) applied over B-Splines of Zhang et al. (2008). These approaches use **numeric optimization** techniques to solve the differential equations that model the glider motion under the effect of ocean currents. This models are extensions of the three-dimensional unconstrained motion model described in Section 2.2.2.1. Similarly, Yilmaz et al. (2008) models problem in such a way that Mixed Integer Linear Programming (MILP) is applied to find the optimum path. Also, He et al. (2009) propose an iterative optimization process for glider path planning. However, the focus of that work is centered on the waypoint precision enhancement, and not in optimal path planning. Furthermore, only static ocean currents are considered.

Special mention require the Level Set Methods (LSMs) (Lolla et al., 2012), which are a numerical technique for tracking interfaces and shapes using an Eulerian approach. LSMs are applicable to many different problems, including robotic path planning, where the interface or shape they track is the wavefront expanded by the cost function from the starting location $\mathbf{x}_0$. By solving a Partial Differential Equation (PDE) it finds the cost from the source $\mathbf{x}_0$ to any location in the search space, so a gradient descend procedure can be used to retrieve the optimal path.

The Fast Marching (FM) method, which lies in the LSM family, have been recently applied for AUV path planning by Pêtrès et al. (2007, 2005). FM can be regarded as a continuous version of Dijkstra's algorithm. Indeed, from the algorithmic point of view, it belongs to the class of Breadth-First Search (BFS) algorithms. However, the neighbors are updated according to a numerical scheme based on the Eikonal equation $\|\nabla u\| = \tau$, where u represents the costmap —e.g. the ocean currents— and τ is the cost of moving

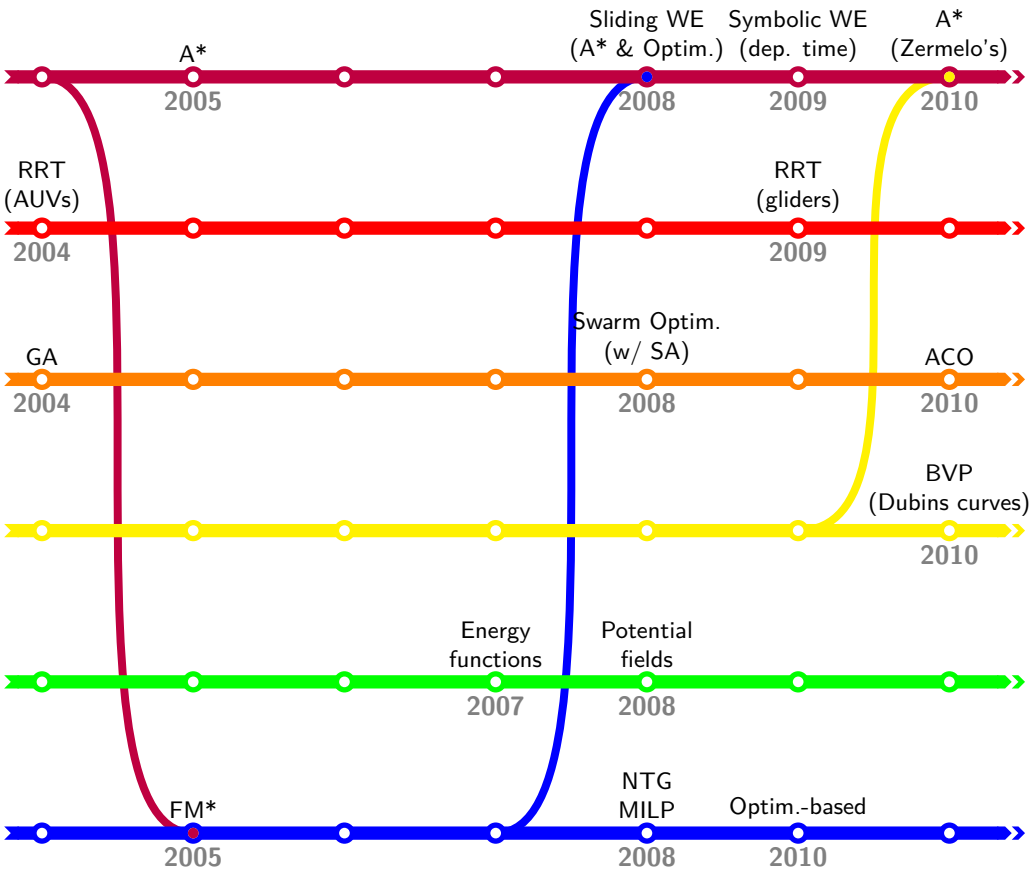


Figure 4.2: Temporal diagram with some of the most relevant glider path planning approaches contributed to the field thus far. The year of publication and the technique are assembled in the diagram, which is divided by categories according with the type of path planning approach used (in top-down order): Graph-based methods ; Probabilistic Sampling-based methods ; Bio-inspired optimization methods ; System Theory approaches ; Energy functions ; and numeric optimization methods . The thin lines that connect different categories represent algorithms that combine ideas from different types of approaches.

to a neighbor location. Therefore, the discretization problems disappear with this continuous motion model. In its original form, the FM only addresses static isotropic costmaps, as in the SPP. In his PhD Thesis, Pêtrès (2007) extended this technique to dynamic anisotropic —or asymmetric— costmaps, as in the Minimal Time Path problem. Also, an heuristic version called FM* is proposed, equivalent to A*. This approach, however, has not been applied to gliders yet, in which the bearing ψ_e can only be changed every surfacing time t_s.

Many authors use some sort of graph-based method for glider path planning, usually with A* at its core, but with some additional methods to alleviate the problems derived from its discretization (Sathyaraj et al., 2008; Thompson et al., 2009, 2010; Wang, 2007). Indeed, one approach to the problem consist on combining the graph-based search with some kind of local optimization (Soulignac, 2010). However, all these techniques still do not represent adequately the restriction on the modification the bearing ψ_g at discrete times because the travel time between neighbor nodes is still variable, rather than constant. Similarly, Eichhorn (2010) combines A* and Zermelo's optimal navigation formula to the path following problem. Other alternatives that also make use of continuous models are described on the works of Pêtrès et al. (2007, 2005). Later, this

line of work was extended to deal with strong currents in the works of Soulignac et al. (2008).

4.1.4 Summary

The time lines in Figure 4.2 show the most representative techniques in the glider path planning field. The figure summarizes the state of the art by arranging the original contributions for each type of path planning approach. Some categories are connected in order to illustrate the relationship between certain techniques. In fact, FM* (Pêtrès et al., 2005) can be seen as the continuous version of the graph-based method A* (Garau et al., 2005); the Sliding Wavefront Expansion (WE) (Soulignac et al., 2008) combines A* with local optimization methods; and Eichhorn (2010) also combines a variant of A* known as Time Varying Environment (TVE) with Zermelo's optimal navigation formula, from System and Control Theory.

In the next sections we will explain the different approaches developed in this thesis. They are described in detail, highlighting their advantages, drawbacks and assumptions. From the survey of path planning techniques above, we have selected the most promising ones, including those already in use by others, as well as novel approaches. This comprises graph-based methods like A*, the probabilistic sampling method RRT, and optimization techniques. Along with the description of the algorithm, we try to put clear how certain techniques solve or alleviate the drawbacks or assumptions imposed by others. For instance, we will see how we address the discretization problem of A* with a novel approach that integrates probabilistic sampling techniques, or with optimization methods. Later, in the experimental results (see Chapter 5), we evaluate each of these approaches for the glider path planning problems discussed thus far in Chapter 3, so we can compare the computing time and path optimality against a naive Direct to Goal strategy that resembles the manual glider path planning.

4.2 The Direct to Goal Approach

Before we start with the glider path planning techniques developed in this thesis, we are going to discuss a simple approach that can be considered equivalent to the control algorithm on board the glider, which is used to reach a waypoint. We have termed it the Direct to Goal —DtG for short— approach, since it tries to go directly to the goal/target waypoint from any surfacing point, regardless of the ocean currents. In fact, it is a blind technique, in the sense that it does not take into account the speed and direction of the ocean currents nearby. It is extremely fast to compute, but the path is usually far from the optimal, as we will see in Chapter 5. Even worse, in many cases, it does not find any path to the target waypoint, at all.

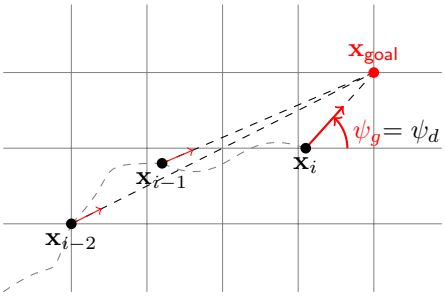


Figure 4.3: Diagram of the Direct to Goal approach, which sets a bearing ψ_g equal to the heading ψ_d to the goal/target $\mathbf{x}_{\text{goal}}$ waypoint from the current location $\mathbf{x}_i$.

The basic operation of the DtG approach is depicted in the diagram of Figure 4.3. At each location $\mathbf{x}_i$, it commands the glider to set the bearing $\psi_g = \psi_d$, where ψ_d is the direction from $\mathbf{x}_i$ to the target waypoint $\mathbf{x}_{\text{goal}}$. The locations $\mathbf{x}_i$ correspond to the surfacing points, because only when the vehicle surface it is possible

to reconfigure its bearing, since the new precise GPS position is available and we can communicate with it. In fact, once the pilot has set the target waypoint $\mathbf{x}_{\text{goal}}$, gliders operate exactly this way. Every time they emerge, the bearing is recomputed in order to head to $\mathbf{x}_{\text{goal}}$. Additionally, some gliders —e.g. SLOCUM— can estimate the ocean current in the last stint and assume it will be the same for the next, so it is used to correct the next bearing in order to actually head to the target. This behavior can also be activated in the DtG. However, in sum it does not make any significant difference, because if we do not perform such correction at $\mathbf{x}_{i-1}$, in the next surfacing point $\mathbf{x}_i$ the drift will be reflected in the next bearing set to head to the target from $\mathbf{x}_i$.

In practice, glider pilots usually set a waypoint for some days, according with their knowledge of the region and the ocean currents dynamics for the season, as well as some intuition. A waypoint may not be changed during days, and during this slot of time the vehicle tries to reach it by running a Direct to Goal approach as the one described here, with or without ocean currents corrections. For this reason, DtG will be used as the basis for comparison, since in this work we try to contribute practical approaches to improve current path planning or piloting methods for ocean gliders. Consequently, while the DtG strategy sets the bearing $\psi_g = \psi_d$ in the straight direction to the target, path planning algorithms explore several paths starting from the current location $\mathbf{x}_i$, trying to find an optimal plan for the particular problem at hand.

The next sections describe the glider path planning algorithms developed in this thesis: the adaptation of A* using costmaps, i.e. the ocean currents velocity field; our novel CTS-A* (Constant Time Surfacing A*) algorithm that uses the unconstrained motion model within the A* search in order to model the problem adequately by ensuring a constant time between consecutive surfacing points; and Adaptive Bearing Sampling (ABS) method to focus the CTS-A* search and hence reduce its computing time; the RRT algorithms; optimization-based methods, including our novel Optimization scheme on the configuration space of the bearing angles $\mathcal{B}$, the Iterative Optimization approach, and an Intelligent Initialization based on a coarse version of the CTS-A* for obstacle avoidance. We also explain how to model the problem in favorable scenarios with as a Boundary Value Problem (BVP), and give some insight in glider path planning using ocean evolving structures.

4.3 Adaptation of the classical A* algorithm

One of the most common path planning algorithms in Artificial Intelligence is the graph-based algorithm A* (Hart et al., 1968), shown in Algorithm 4.1. In its original form, it solves the Shortest Path Problem (SPP). More precisely, A* solves the single-pair SPP, in which we have to find the shortest path from the single source/start node n_{start} to the single destination/goal/target node n_{goal}. This is exactly the kind of problem described thus far in Section 3.1, but requires some adaptation of the cost function to deal with a time cost instead of distances. Here, we explain the algorithm, how we model the search space, the cost function and the heuristic, as well as some implementation details, extensions, and important implications of the nature of A* in glider path planning.

A* finds the optimal path from n_{start} to n_{goal} in a graph, which in our case consists on a $N \times M$ grid of nodes with 8-neighbor vicinity. Or, in other words, a directed graph $G = \langle V, E \rangle$, with $|V| = N \times M$ vertices/nodes that represent a location in a metric space, and $|E| = 8\,|V|$ —ignoring the grid boundaries— edges/arcs that connect the nodes. The nodes disposition is a grid in which they are connected to their surrounding nodes with a weighted edge. That weight $c(n_i, n_{i+1})$ represents the cost of traveling from the current node n_i to the nearby one n_{i+1}. In the SPP, $c(n_i, n_{i+1})$ represents the distance from n_i to n_{i+1}, but in the minimal time —or the minimal distance to a target problems— explained in Section 3.1, that cost is the travel time. Such cost is computed with the constrained glider motion model of Section 2.2.2.2. By using the unconstrained motion model, we are computing the bearing required to travel from the current node n_i to any of its neighbors n_{i+1}. The gridded 8-vicinity topology establishes a set of fixed headings $\psi_e = k\pi/4$ for $M = 0..7$, that gives the 8 neighbors. In glider path planning problems, we are actually looking for such

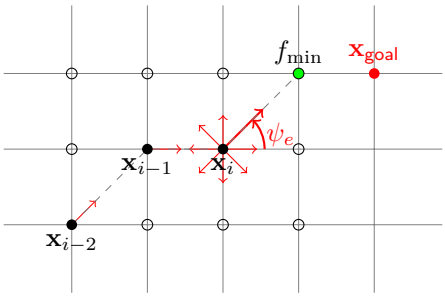


Figure 4.4: Classical A* algorithm operating on an uniform grid that defines the search space domain Ω. The heading angles ψ_e are constrained by the grid, producing straight line trajectories between nodes and non-constant surfacing times for glider path planning problems. The distance between nodes can be parameterized, but a high resolution would increase the computing time. Also note that the nodes do not represent surfacing points because they are not separated a constant time, so we must compute the actual surfacing points afterwards, setting the bearing found for the previous node.

list of bearings, rather than the location of each node (see Figure 4.4), since the ocean currents at each node affect this computation.

To focus its search, A* uses h-values $h(n)$ that approximates the distances to the target node from each of the nodes $n \in V$ using an heuristic function. The heuristic function $h(n)$ is admissible if it never overestimates the actual minimal cost of reaching the target node n_{goal} —otherwise, we cannot guarantee that the path to the target is the optimal one. In the special case where $h(n) = 0\ \forall n$ we have Dijkstra's algorithm (Dijkstra, 1959), whose main drawback is the fact that it does a blind search; indeed, for this reason it is considered as a greedy algorithm. Therefore, the ultimate purpose of the heuristic is to reduce the computational cost of the search.

According with Sniedovich (2006), Dijkstra's algorithm can be seen as a successive approximation that solves the dynamic programming functional equation for the SPP by means of the Reaching method (Denardo and Fox, 1979). In fact, Dijkstra (1959) uses the fact that, "if R is a node on the minimal path from P to Q, knowledge of the latter implies the knowledge of the minimal path from P to R", which is in accordance with Bellman's Principle of Optimality. Bellman (2003), widely known as the father of Dynamic Programming, establishes that this programming method is applicable to problems that exhibit overlapping subproblems and optimal substructure. This means that Dijkstra's algorithm actually solve the single-source SPP, not just the single-pair one. In the single-source SPP, we find the best path from the source node n_{start} to all other nodes in the graph. And therefore, it turns out that there is no algorithm more efficient for the single-pair SPP, than for the single-source SPP. Similarly, the Wavefront Expansion method of Dorst and Trovato (1989) —repeatedly used in the glider path planning literature—, operates like Dijkstra's algorithm.

One possible implementation of Dijkstra's algorithm uses a node set $\mathcal{Q}$ that contains all the undiscovered nodes so far. Initially it has all but the start node n_{start}, and the algorithm proceed by exploring each node at a time —starting with n_{start}. The next node selected for expansion is extracted from $\mathcal{Q}$, being the one with the minimum cost. The computational cost of the Dijkstra's algorithm is defined as $\mathcal{O}\left(r\left|E\right| + m\left|V\right|\right)$, where r and m are the number of operations to insert and extract the minimum from the node set $\mathcal{Q}$, respectively. By using a min-priority queue implemented by a Fibonacci heap the algorithm runs in $\mathcal{O}\left(\left|E\right| + \left|V\right|\log\left|V\right|\right)$ (Fredman and Tarjan, 1984, 1987). Although Fibonacci heaps are theoretically better than binary heaps, in practice the latter exhibit less overhead, and they are common in standard libraries, like the C++ STL `std::priority_queue` template; in our implementation we have used a binary heap, indeed.

As it has been previously said, A* introduces an heuristic function $h(n)$ to speed up the search of Dijkstra's algorithm. Basically, $h(n)$ is added to the cost of the nodes inserted into the node set $\mathcal{Q}$, so their priority

Algorithm 4.1 A* algorithm pseudo-code for implicit graph search.

Require: Start n_{start} and target n_{goal} nodes. Heuristic function $h(n)$ that estimates the cost from node n to n_{goal}. The heuristic $h(n)$ is assumed to be consistent, so we use a closed set $\mathcal{C}$ to make the search more efficient. Function **parents**(n_i) that retrieves the path $\mathcal{P}$ by extracting the parents of $n_i = n_{\text{goal}}$ from the parent($\cdot$) map recursively, until it reaches n_{start}.
Ensure: Optimal path $\mathcal{P}$ found from n_{start} to n_{goal}.
Algorithm: A*(n_{start}, n_{goal}) **return** $\mathcal{P}$

```
1:  g(n_start) = 0                                   ▷ assume g(n_i) = ∞ for the rest
2:  parent(n_start) = ∅                              ▷ assume parent(n_i) = ∅ for the rest as well
3:  O = C = ∅                                        ▷ empty open and closed sets
4:  O.insert(n_start, g(n_start) + h(n_start))
5:  while O ≠ ∅ do
6:      n_i = O.pop                                  ▷ extract node n_i with minimum cost
7:      if n_i = n_goal then                         ▷ target n_goal reached
8:          return parents(n_i)                      ▷ path found (extracted from parent( · ))
9:      end if
10:     C = C ∪ {n_i}                                ▷ close node n_i
11:     for all n_{i+1} ∈ successors(n_i) do         ▷ for each successor n_{i+1}
12:         g' = g(n_i) + c(n_i, n_{i+1})            ▷ tentative g-value
13:         if (n_{i+1} ∉ C ∪ O) ∨ (g' < g(n_{i+1})) then   ▷ undiscovered or better g'
14:             g(n_{i+1}) = g'                      ▷ update
15:             parent(n_{i+1}) = n_i
16:             O.insert(n_{i+1}, g(n_{i+1}) + h(n_{i+1}))   ▷ decrease key (update, or insert if undiscovered)
17:         end if
18:     end for
19: end while
20: return ∅                                         ▷ no path found
```

is changed. At this point is important to say that our A* implementation manages implicit graphs. It is common to use implicit graphs when the search graph is such huge that it is better to generate the adjacency list of each node during the exploration. In Algorithm 4.1, the **successors** method is responsible of retrieving all the children nodes n_{i+1} of the current node n_i, according with the 8-neighbor topology. As the algorithm shows, A* maintains for every node n:

1. The g-value $g(n)$, that is the distance of the shortest path from the start node n_{start} to the node n found so far.

2. The parent node of n, which is used to retrieve the path after the search ends. Recall that for the problem on hand, the path consists on a sequence of bearing angles ψ_g that are to be commanded to the glider, and the computed surfacing locations $\mathbf{x}$. This is a consequence of using the constrained motion model of Section 2.2.2.2, as mentioned before.

There are two general implementations for AI search methods: tree and graph search algorithms. A* can be implemented with any of them. In a tree search, the topology of the graph already guarantees that no node is rediscovered during the search. On the contrary, searching on a graph may expand nodes already explored. As it was previously mentioned, for glider path planning problems we use a directed graph with a gridded topology. Therefore, we use the graph search version of A*. In this case, since we can rediscover nodes already explored from alternative paths, we need to use an additional set, not only $\mathcal{Q}$. Indeed, we have to maintain two data structures:

1. The open set $\mathcal{O}$, which consists on a priority queue that contains the nodes to be considered for expansion, along with the expected cost $f(n) = g(n) + h(n)$ to reach the target n_{goal}. The open set $\mathcal{O}$ is equivalent to Dijkstra's node set $\mathcal{Q}$.

2. The closed set $\mathcal{C}$, which contains nodes that have already been expanded in order to ensure that each node is expanded only once, unless we find a better cost $g(n)$ from n_{start}, coming from an alternative path.

Since there are different ways of interpreting the A* algorithm, below we provide the equivalence between node states, coloring and the open and closed sets. The nodes can take three different states during the A* search:

undiscovered: It is a **white** node, which is neither in the open $\mathcal{O}$, nor in the closed $\mathcal{C}$ set, i.e. $n \notin \mathcal{O} \cup \mathcal{C}$.

discovered: It is a **gray** node, which is in the open $\mathcal{O}$ set, i.e. $n \in \mathcal{O}$.

explored: It is a **black** node, which is in the closed $\mathcal{C}$ set, i.e. $n \in \mathcal{C}$.

Note that the case of $n \in \mathcal{O} \wedge n \in \mathcal{C}$ is not valid. The implementation with the open and closed sets attends to the fact that we search an implicit graph, so they are used to mark the nodes discovered and explored thus far. As in Dijkstra's algorithm, for the open set $\mathcal{O}$, we use a binary heap to obtain the best efficiency. For the closed set $\mathcal{C}$ we need a hashmap to store the information of whether the node is closed/explored or not; note that we can use a single hashmap to store the nodes with further information, including an attribute to indicate whether the node has been explored or not, which is what we do in our implementation.

A* updates the g-value and parent of an unexpanded successor n_{i+1} of node n_i by considering the distance $g(n_i)$ of the path from n_{start} and the distance $c(n_i, n_{i+1})$ from n_i to n_{i+1}, resulting in $g(n_i) + c(n_i, n_{i+1})$. It updates the g-value and parent of n_{i+1} if the new path distance $g(n_{i+1})$ is shorter than the shortest path from n_{start} to n_{i+1} found so far (see line 13 of Algorithm 4.1); undiscovered nodes are also updated.

The algorithm minimizes the cost to the target, which in our case consists on a temporal cost based on the maximum speed a glider can achieve within the ocean current field and the great circle distance between a pair of locations $\mathbf{x}_i$ and $\mathbf{x}_j$ for $i \neq j$, represented by the nodes of the gridded search graph. The great circle distance is the shortest distance on a sphere or ellipsoid that in this case approximates the Earth surface, but a simpler metric distance might be used (see Section A.2). In spherical geometry, the shortest distance between two points is an arc of a great circle. In this case, the triangle inequality holds provided the restriction that the segment covers less than half a great circle (Brock et al., 2009; Ramsay and Richtmyer, 1995). Even for large missions, we can assume this restriction is met. Consequently, since the triangle inequality holds, in glider path planning we are still working in a metric space, as in our experiments. This is important in order to guarantee that the heuristic is consistent, and the problem has optimal substructure.

To compute the heuristic we estimate the maximum velocity $U_{c_{\max}} = \max_{\mathbf{x} \in \mathcal{P}}(\|\mathbf{U}_c(\mathbf{x}, t)\|)$ of the ocean currents through the path $\mathcal{P}$ the glider will follow. Since $\mathcal{P}$ is unknown in advance, we estimate $U_{c_{\max}}$ using a domain Ω spatial and temporally centered at the current location and time. Given $U_{c_{\max}}$ and the glider nominal speed U_g, we compute an underestimated temporal cost to reach the target from the current location $\mathbf{x}$, of node n, using the straight distance $d = \|\mathbf{x} - \mathbf{x}_{\text{goal}}\|$ to the target,

$$h(n) = \frac{d}{U_g + U_{c_{\max}}} \tag{4.1}$$

which is a conservative but admissible heuristic function. Furthermore, as long as we use the same estimate of $U_{c_{\max}}$ for every node n in the graph, this heuristic is also consistent.

Given the 8-connected grid of A*, Rao and Williams (2009) proposes a simplified heuristic, where d is not computed as an Euclidean distance, but rather as an almost Manhattan distance considering also the $M\pi/4$

connections. For a given location $\mathbf{x} = (x, y)$ and target waypoint $\mathbf{x}_{\text{goal}} = (x_{\text{goal}}, y_{\text{goal}})$, in 2 dimensions, we compute the width w and length l values as

$$w = \min |x_{\text{goal}} - x|, |y_{\text{goal}} - y| \tag{4.2}$$

$$l = \max |x_{\text{goal}} - x|, |y_{\text{goal}} - y| \tag{4.3}$$

Now, the distance d turns into

$$d = \Delta x \left((\sqrt{2} - 1)w + l \right) \tag{4.4}$$

which can be directly plugged into the the previous heuristic $h(n)$ definition (4.1). Also note, that for the Euclidean distance, it is sufficient to use the squared distance $d = \|\mathbf{x} - \mathbf{x}_{\text{goal}}\|^2$, which makes (4.1) as efficient in terms of operations as (4.4), and hence it is what we use in our implementation.

Regarding the time complexity of A*, Pearl (1984); Russell and Norvig (2010) establish that in the worst case the number of nodes expanded is exponential on the length of the optimal path, but it is polynomial if the search space is a tree, there is a single goal, and the heuristic meets the following condition:

$$|h(n) - h^*(n)| = \mathcal{O}(\log h^*(n)) \tag{4.5}$$

where $h^*(n)$ is the optimal heuristic, i.e. an admissible heuristic that gives the exact cost to get from the node n to n_{goal} (Russell and Norvig, 2010). Unfortunately, in our case the search space is not a tree, but a gridded graph, and the heuristic $h(n)$ (4.1) is presumably far from the optimal. Therefore, the time complexity of A* applied to glider path planning problems is not polynomial. However, it expands less nodes than Dijkstra's algorithm, since it is not an informed search as A* —that uses an admissible $h(n)$.

So far, we have said that the heuristic used for the search must be admissible. This is required to guarantee that we find the optimal path. However, we can impose a more strict condition on the heuristic. It is said that a consistent, or monotone, heuristic function is defined as

$$h(n_i) \leq c(n_i, n_{i+1}) + h(n_{i+1}) \tag{4.6}$$

$$h(n_{\text{goal}}) = 0 \tag{4.7}$$

Indeed, (4.6) establishes that it is necessary and sufficient for a heuristic to obey this triangle inequality to be consistent (Pearl, 1984).

By definition, a consistent heuristic is also admissible, i.e. it never overestimates the cost of reaching the target from any node n_i. However, the opposite is not always true. Although it is unusual to have an admissible heuristic h which is not consistent, it can be made into a consistent heuristic h' using the pathmax equation (Russell and Norvig, 2010):

$$h'(n_{i+1}) = \max \{h(n_{i+1}), h(n_i) - c(n_i, n_{i+1})\} \tag{4.8}$$

A* is complete and optimal provided that $h(n)$ is admissible (for a tree search) or consistent (for a graph search) (Russell and Norvig, 2010). Indeed, the pseudocode of Algorithm 4.1 assumes that the heuristic function is consistent, which is a frequent case in most practical problems. However, if the assumption is not true, nodes in the closed set may be rediscovered and their cost improved. In other words, the closed set can be omitted —yielding the tree search algorithm— if a solution is guaranteed to exist; or if the algorithm is adapted so that new nodes are added to the open set only if they have a lower $f(n)$ value than at any previous iteration. For static ocean currents, the heuristic $h(n)$ (4.1) is consistent, for the reason of the analysis presented here. However, for dynamic ocean currents, since the cost of going from any node to another changes, the triangle inequality is violated, and $h(n)$ is no longer consistent. Consequently, we may rediscovered explored nodes from alternatives paths with an improved cost $g(n)$. Fortunately, $h(n)$ remains an admissible heuristic in these scenarios as well, so we can apply A*.

According with Dechter and Pearl (1985), if the heuristic is not admissible, A* is not guaranteed to expand fewer nodes than another search algorithms with the same heuristic. However, at the expense of the optimality, we can speed up the search by relaxing the admissibility criterion. Indeed, for large search spaces, it is common to overestimate the heuristic cost, to reduce the exploration time. Formally, we can use an ϵ-admissible heuristic, which guarantees that the solution is no worse than $(1+\epsilon)$ times the optimal. There exist several ϵ-admissible algorithms, being the static and dynamic weighting two reasonable approaches. Static weighting (Pohl et al., 1969) changes the cost function into

$$f(n) = g(n) + (1+\epsilon)h(n) \tag{4.9}$$

On the other hand, the dynamic weighting adapts the heuristic weight to the node depth (Pohl, 1973), as follows

$$f(n) = g(n) + (1+\epsilon\omega(n))h(n) \tag{4.10}$$

where

$$\omega(n) = \begin{cases} 1 - \dfrac{d(n)}{N} & \text{if } d(n) \leq N \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

being $d(n)$ the depth of node n in the search graph, and N the anticipated length of the solution path. We have experimented with static weighting, but for the results show in Chapter 5 we focus on the solutions and computing time of A* using the admissible heuristic (4.1).

There exists several approaches to estimate the maximum velocity $U_{c_{\max}}$ of the ocean currents for the heuristic function $h(n)$ (4.1). Since taking the global maximum of all ocean currents in the map gives an admissible $h(n)$, it is a very conservative value. Alternatively, we can take the mean, or more appropriately, estimate its value only for a region around the current node n_i —both in space and time. Although this may violate the admissibility of $h(n)$, in practice —for relatively large regions—, it provides a reasonable improvement for the heuristic.

With the heuristic $h(n)$ we try to reduce the computing time of the algorithm, at the expense of finding a suboptimal solution. However, the A* method also has several drawbacks that come from the fact of using a grid. These limitations are more noticeable in the case of glider path planning, because of the effect of ocean currents in the constrained motion model used for the nodes expansion/relaxation. First of all, the grid resolution must be sufficiently high, so the nodes lie on areas with almost constant ocean currents. Otherwise, some sort of interpolation is required during the node expansion process, in the **successors** method. More important than the resolution is the topology of the grid, because it determines the heading angles ψ_e that will be expanded. When ocean currents are very strong, it happens that the range of feasible headings gets extremely narrow (see the accessibility cone of Section 2.2.2.2). It becomes smaller than the angular resolution between the headings, which in our case —for an 8-vicinity— is of $45°$. In those cases, if the ocean current direction falls inside the bisector of two successive heading angles, it happens that no node is expanded, since the constrained motion model establishes that it is impossible to reach them. The Figure 4.5 shows a case in which this happens, and the optimal path is not found. In order to alleviate this problem, in some experiments we have used a greater vicinity radius, i.e. we have included the children of the neighbor nodes, and so on. Consequently, new heading angles are introduced —indeed, in the successors relaxation, only those nodes that produce different, new headings are considered. Therefore, we get nodes expanded in the presence of strong ocean currents, in these type of situations.

The A* algorithm have been used in the context of glider path planning by several authors previously (Garau et al., 2005; Pêtrès et al., 2005; Soulignac, 2010). Here, we have a thorough analysis of the algorithm and its main drawbacks when applied to the minimal time and minimal distance to the target problems (see Section 3.1). The most remarkable limitation of this approach is the fact that the algorithm uses the constrained motion model explained in Section 2.2.2.2, to compute the cost $c(n_i, n_{i+1})$ of traveling from a node n_i to each of its neighbors n_{i+1}. This imposes a set of heading angles ψ_g that in the presence of strong

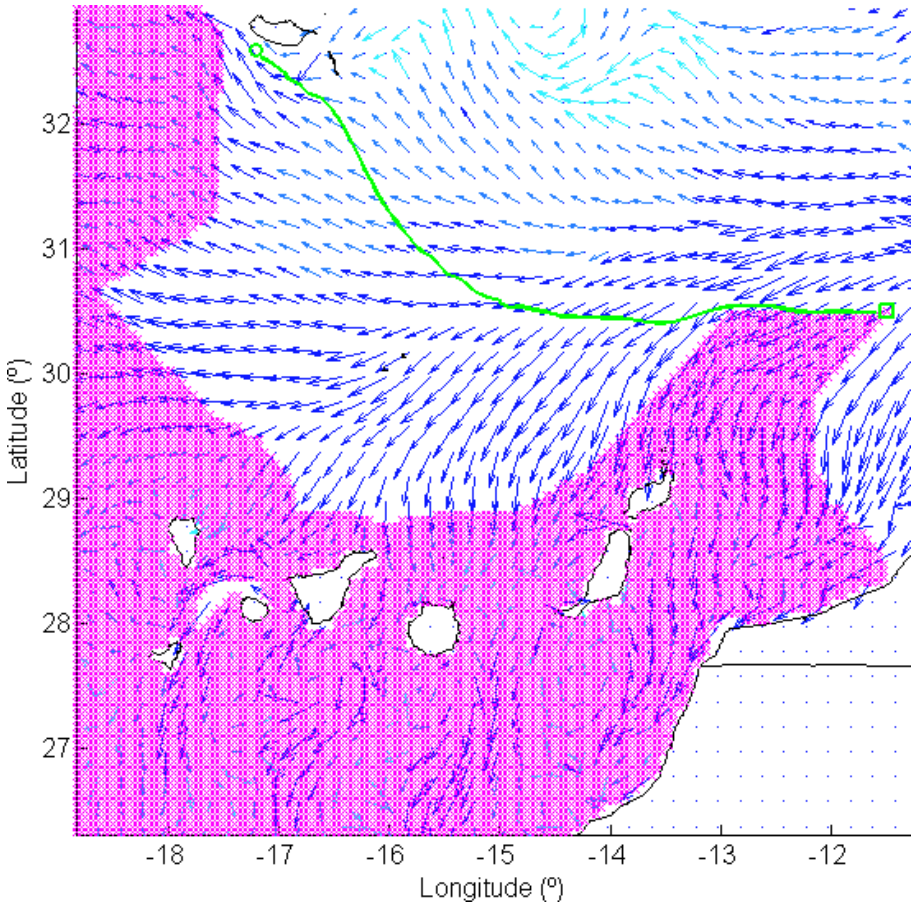


Figure 4.5: Example of optimal path not found by the A* algorithm. The figure shows the nodes explored from the start to the target waypoint. At a certain node, the constrained motion model does not allow to explored the optimal path trajectory, because its heading does not fall into the 8-vicinity grid of the search graph.

ocean currents are often unreachable. Furthermore, with this scheme we obtain non-constant time surfacings between nodes. This does not reflect the glider navigation pattern explained in Section 2.1, which exhibits constant time surfacing. For this reason, we have explored other approaches to overcome such limitations, as we will see in the sequel.

4.4 A novel Constant Time Surfacing A* (CTS-A*) approach

In this thesis we have developed a graph-based search algorithm named Constant Time Surfacing A* (CTS-A*). This algorithm alleviates the two main drawbacks of A* discussed thus far: the difficulties to reach the neighbor nodes in regions of strong ocean currents, and the non-constant travel time required to reach them, in any case. First, in the A* adaptation of Section 4.3 we use the constrained motion model explained in Section 2.2.2.2 to go from one node n_i to its neighbors n_{i+1} in the gridded search graph. This motion model forces the vehicle to reach the neighbor nodes n_{i+1}, by setting the bearing ψ_g that yields the heading ψ_e defined by each n_{i+1} —the $M\pi/4$ angles for $M = 0, \ldots, 7$, in the 8-vicinity grid—, which allows to beat the

ocean currents. When the ocean currents are very strong, most of the neighbors n_{i+1} are unreachable, as it was explained in Section 4.3. Consequently, the algorithm cannot expand more nodes from n_i, and in some cases this prevents from finding a path to the target waypoint, as the Figure 4.5 illustrates. Secondly, even when it is actually possible to reach the neighbors n_{i+1} with the constrained motion model, it happens that the time required to arrive is not constant, because of the design of the algorithm and the motion model. This represents a clear diversion from the glider navigation pattern, with stints of constant time between successive surfacing points (see Section 2.1).

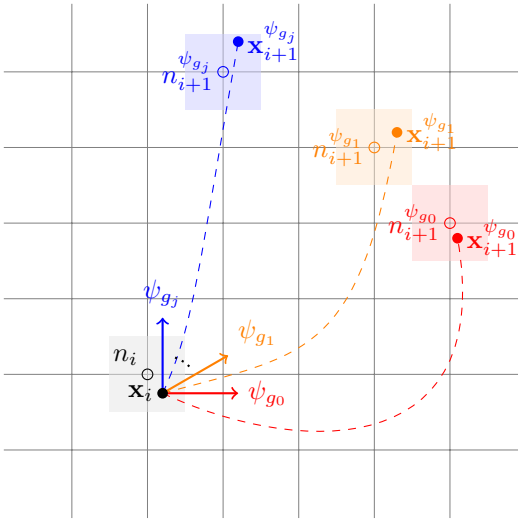


Figure 4.6: Successors generation in the CTS-A* approach proposed. Starting at a node n_i several trajectories are integrated for surfacing time t_s considering different bearings ψ_{g_j} starting from location $\mathbf{x}_i$. The ending points $\mathbf{x}_{i+1}^{\psi_{g_j}}$ are mapped to the nearest neighbor node $n_{i+1}^{\psi_{g_j}}$.

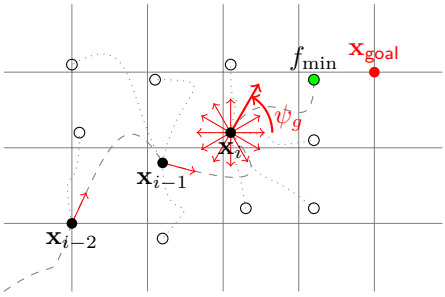


Figure 4.7: Diagram of the operation of the Constant-Time Surfacing A* algorithm. At each surfacing location $\mathbf{x}_i$ different bearing angles ψ_g are considered, integrating the glider trajectory for the surfacing time t_s, or a multiple $s \cdot t_s$.

The Constant Time Surfacing A* algorithm —CTS-A* for short— does not impose the strict condition of reaching the adjacent neighbors n_{i+1}. On the contrary, it integrates numerically the unconstrained motion model of Section 2.2.2.1 for a constant surfacing time t_s —or a multiple $s \cdot t_s$. The ending location is then mapped into the gridded search graph, but proceeding this way, we do not impose the vehicle to navigate in a certain heading ψ_e, which might be unfeasible. Therefore, all the neighbors can be expanded even in areas

with strong ocean currents. Furthermore, we keep the surfacing time constant. The integration process and the mapping to the nodes of the gridded search graph is depicted in Figure 4.6 for a subset of bearing angles ψ_{g_j}; note that contrary to A*, we set the bearing directly, instead of computing it from a given heading angle ψ_e.

4.4.1 Successors generation

Therefore, the CTS-A* approach includes a notable modification to the original A* algorithm. As a consequence of forcing constant time surfacings, the process of generating successors changes:

1. For each bearing ψ_{g_j} we integrate the glider trajectory for the surfacing time t_s, i.e. we compute the trajectory followed by a glider that keeps a bearing ψ_{g_j} under the influence of ocean currents, with the unconstrained motion model. We also allow for multiples $s \cdot t_s$ of the surfacing time, so in the presence of strong ocean currents we give enough time to the vehicle to move to the area covered by a neighbor node n_{i+1}; otherwise, the ending location would still lie into the current node n_i, and it would be discarded —since it would have a higher cost $g' = g(n_i) + t_s$. With multiple surfacings s, the cost $c(n_i, n_{i+1})$ is given by $s \cdot t_s$, which is considered to compute $g(n_{i+1})$ in the line 10 of the algorithm.

2. The ending location of each trajectory is mapped into the Nearest Neighbor node $n_{i+1}^{\psi_{g_j}}$ of a grid, using the **NN** method in the algorithm. Special care is taken when that node already represents another location, and consequently an alternative path that reaches it. This requires the **successors** generation procedure to be hard-coded alongside the open list $\mathcal{O}$ update step, which is almost reproduced in this method.

3. Therefore, if two locations fall into the same node —i.e. if the node was discovered or explored previously—, we take and map the one with lower cost $g(n)$. If both have the same cost $g(n)$, we take the best location according to some function $d(\mathbf{x}, n)$, which can be the Nearest Neighbor to the node n centroid or the heuristic $h(n)$. If we use $h(n)$, we can approximate all this logic with a single comparison on the cost $f(n)$. Additionally, the bearing is also stored in the node.

This new generation of successors scheme is concerned with the node labeling and the open $\mathcal{O}$ and close $\mathcal{C}$ sets management. The path finding basis of this approach is depicted in Figure 4.7. For each surfacing location $\mathbf{x}_i$, starting with $n_{\text{start}} = n_0$, the glider trajectory is integrated for the surfacing time t_s —or a multiple $s \cdot t_s$— for different bearing angles ψ_{g_j}. This process is repeated with the new surfacing locations generated, $\mathbf{x}_{i+1}^{\psi_{g_0}}$, $\mathbf{x}_{i+1}^{\psi_{g_1}}$ and $\mathbf{x}_{i+1}^{\psi_{g_j}}$ in the figure, which are selected according to the cost function $g(n)$ that guides the searching to the target node n_{goal}. The ending locations $\mathbf{x}_{i+1}^{\psi_{g_j}}$ are not discretized, since they are mapped into the Nearest Neighbor node $n_{i+1}^{\psi_{g_j}}$ of the gridded search graph —i.e. the nodes are labeled with that ending location. In the next iteration, the glider trajectory is computed starting at the location $\mathbf{x}_{i+1}^{\psi_{g_j}}$ saved.

4.4.2 Discussion

We may argue that using constant time surfacings is equivalent to say that all the edges E in the graph have the same constant weight. Clearly, if all the weights are the same, we can omit them and find the optimal path in the unweighted version of the graph. In this type of graphs, the priority queue —that implements the open set $\mathcal{O}$— degenerates into a FIFO queue, and the problem can be solved with a Breadth-First Search (BFS), instead of Dijkstra's algorithm. However, there are two reasons that hold us from doing this for glider path planning. First, we allow to perform more than a single surfacing to travel from one node n_i to its neighbors n_{i+1}, in order to deal with regions of strong ocean currents. Therefore, the weights $c(n_i, n_{i+1})$ are

Algorithm 4.2 Successors generation in CTS-A*.

Require: Current node n, the number k of bearings to sample and the surfacing time t_s to integrate the trajectory. The trajectory integration requires the integration step dt, the glider nominal speed $\mathbf{v}_g(\psi_g)$ for a bearing ψ_g, and the ocean currents velocity field $\mathbf{v}_c(\mathbf{x})$. If a neighbor node m has equal cost g' than the new trajectory $\mathbf{x}_{\psi_g}$, a $d(\mathbf{x}, n)$ function is used. If $d(\mathbf{x}, n) = h(n)$, we can use $f(n)$ instead of $g(n)$, with a single condition.

Ensure: Successors list $\mathcal{S}$ and node labeling update considering the cost $g(\mathbf{x})$ and the distance $d(\mathbf{x}, m)$ of the location $\mathbf{x}$ paired with the node m.

Algorithm: successors(n, k, t_s) **return** $\mathcal{S}$

$\mathcal{S} = \varnothing$
for all 1 to k **do**
  $\psi_g \sim [0, 2\pi)$ ▷ sample bearing
  $s = 0$
  repeat
    $\mathbf{x}_{\psi_g} =$ **trajectory**$(t_s, dt, \mathbf{v}_g(\psi_g), \mathbf{v}_c(\mathbf{x}), \mathbf{x}_n)$
    $m =$ **NN**$(\mathbf{x}_{\psi_g})$ ▷ Nearest Neighbor node m to $\mathbf{x}_{\psi_g}$
    $s = s + 1$
  until $m \neq n$
  $g' = g(n) + s \cdot t_s$
  if $m \notin \mathcal{C} \cup \mathcal{O}$ **then** ▷ m undiscovered so far
    $m_{\mathbf{x}} = \mathbf{x}_{\psi_g}$
  else
    if $g' < g(m)$ **then** ▷ lower cost
      $m_{\mathbf{x}} = \mathbf{x}_{\psi_g}$
    else if $g' = g(m)$ **then** ▷ equal cost, but
      if $d(\mathbf{x}_{\psi_g}, m) < d(m_{\mathbf{x}}, m)$ **then** ▷ better location
        $m_{\mathbf{x}} = \mathbf{x}_{\psi_g}$
      end if
    end if
  end if
  $\mathcal{S} = \mathcal{N} \cup \{m\}$ ▷ add m to successors list $\mathcal{S}$
end for
return $\mathcal{S}$

not always the constant surfacing time t_s, but a multiple $s \cdot t_s$, which is not constant. Now, let us assume there are not such regions in a particular test case, i.e. the ocean currents are weak and it suffices with a single surfacing to travel from n_i to another node n_{i+1} in t_s time. Under this assumption, we can run a BFS, which has a time complexity of $\mathcal{O}(|E| + |V|)$. Nevertheless, the informed search of A*, using the heuristic $h(n)$ is faster in practice, because the number of nodes explored is reduced. With $h(n)$ comes the second reason we cannot use a FIFO queue, since the cost $f(n) = t_s + h(n)$ —where $g(n) = t_s$ for our assumption— is no longer a constant weight.

Regarding the time complexity of the CTS-A* approach, since it only changes the implementation of the **successors** method, it remains the same as for the A* (see Section 4.3). However, we have to pay attention to the cost of integrating the unconstrained motion model for each bearing angle ψ_g. Indeed, for each node relaxed in the generation of the successors we incur in a constant cost τ, which depends on the integration step Δt or the tolerance configured in the solver (see Section 2.2.2.1). This constant τ multiplies for all the nodes expanded, which is in the order of $\mathcal{O}(|V| \log |V|)$, yielding $\mathcal{O}(|E| + \tau |V| \log |V|)$, which in terms of complexity is equivalent to $\mathcal{O}(|E| + |V| \log |V|)$, the same as in A*. Note that for A*, with the 8-vicinity grid,

$|E| = 8\,|V|$. Here, although we use the same 8-vicinity grid, the number of edges E depends on the number M of bearing angles ψ_g sampled, so $|E| = M\,|V|$. In our experiments we have analyzed different values, from $M = 8$ to 18, where more samples M yield better results, but at the expense of a higher computing time.

The fact that all the neighbor nodes are expanded does not mean that the time complexity of the algorithm is exponential. It is easy to see that the mapping into a fixed gridded search graph allows us to keep the computational cost bounded. After the trajectory integration with the unconstrained motion model, it is common that the ending location gets mapped into a node already used to map another neighbor location or path. In this cases, only the best trajectory is chosen to be represented by the node, according with its cost $g(n)$ and heuristic $h(n)$, as it is shown in the line 13 of Algorithm 4.2, for the process of generating the list of **successors** nodes.

Ideally, all bearing angles would be considered for the trajectory integration. Unfortunately, it is computationally expensive to test so many bearings, and therefore we must take a subset of samples intelligently. It is expensive not only because the number of nodes expanded grows exponentially, but because we also incur in the integration time τ mentioned above, for each node expanded as well. The sampling process to obtain M bearings, which takes place in line 3 of Algorithm 4.2, can be uniformly distributed or it might follow a Probability Density Function (PDF) distribution, which would be a more intelligent approach. Indeed, in the next section we explore a novel informed distribution for this sampling process, which will be evaluated against the uniform sampling in the experiments shown in Section 5.4.3. It is also worth mentioning that the uniform samples can be centered at the bearing angle in the direction to the target ψ_d, which constitutes a first improvement on the sample set selection.

4.4.3 An intelligent Adaptive Bearing Sampling (ABS)

The Adaptive Bearing Sampling (ABS) algorithm discussed in this section is a variant of the CTS-A* algorithm discussed thus far. In the CTS-A*, a given number M of bearing angles ψ_g are sampled uniformly, to expand the **successors** of the current node n_i, by numerical integration of the glider unconstrained motion model explained in Section 2.2.2.1. This uninformed approach requires a significant number M of bearings to obtain an optimal path —according with our experiments $M \approx 18$, i.e. $\Delta^\circ \approx 20^\circ$ (see Section 5.4). Unfortunately, this is computationally expensive, as explained in the previous section. Therefore, a more intelligent, and informed distribution of the bearing angles samples can allow keeping M small without compromising the optimality of the result.

Instead of an uniformly distributed bearing sampling, the ABS algorithm follows a PDF distribution. It is possible to model such a PDF taking the factors below into account:

1. The effective speed U_e that a glider will be able to achieve, which is a function of the glider nominal speed U_g, the ocean currents speed U_c, and the alignment of their orientation angles ψ_g and ψ_c, respectively.

2. The heading ψ_d direct to the goal, that gives the shortest, and minimal time, path to the goal under null ocean currents conditions.

Firstly, the effective speed U_e is modeled as in the constrained motion model of Section 2.2.2.2. Given U_g, U_c, ψ_c and the desired heading ψ_e, it computes the bearing ψ_g required to compensate for the drift produced by the ocean currents as

$$\psi_g = \begin{cases} \psi_e + \arcsin s & \text{if } |s| \leq 1 \\ \nexists & \text{otherwise} \end{cases} \tag{4.12}$$

which yields the effective speed U_e

$$U_e = \begin{cases} U_g\sqrt{1-s^2} + U_c \cos\psi_{ec} & \text{if } |s| \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

where

$$s = \frac{U_c}{U_g} \sin \psi_{ec} \tag{4.14}$$

and $\psi_{ec} = \psi_e - \psi_c$.

Consequently, if $U_g > U_c$ the set of feasible heading angles ψ_e is unbounded. Otherwise, this set is delimited by a so called *accessibility cone* with and arc angle ψ_r given by

$$\psi_r = 2 \arcsin \frac{U_g}{U_c} \tag{4.15}$$

For the special case of $U_g = U_c$ we have $\psi_r = \frac{\pi}{2}$. Thus, a heading ψ_e is physically feasible if and only if it lies in $\psi_e \in [-\frac{\psi_r}{2}, \frac{\psi_r}{2}]$. This constitutes a summary of the constrained motion model explained in Section 2.2.2.2. It is used here to compute U_e, which is employed to build part of the informed PDF distribution of the ABS strategy, as we will see in the sequel.

Figure 4.8 (a) and (b) show the effective speed vectors $\mathbf{U}_e$ for different heading angles ψ_e, computed with (4.13). The speed values U_e obtained for each $\psi_e \in [-\pi, \pi]$ might be thought as weights w or importance factors that constitute a PDF distribution. This PDF $f_{\psi_c}(\psi_e)$ establishes the heading angles with a higher probability of producing a greater U_e —i.e. those which are aligned with ψ_c. Indeed, the highest probability is always assigned to $\psi_e = \psi_c$, with a value of $U_e = U_g + U_c$. Depending on the relation between U_g and U_c, defined as the speed ratio $SR = 100\frac{U_c}{U_g}$, we have different distributions (see Figure 4.8 (c)).

However, it is not enough to follow the directions ψ_e with greater U_e, since this information is local. We have to take the direction ψ_d to the goal into account, which constitutes a global guess. An additional PDF distribution $f_{\psi_d}(\psi_e)$ is built accounting the difference between the heading ψ_e and ψ_d. We propose the following Normal distribution to model it,

$$\mathcal{N}(\psi_e;\ \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(\psi_e - \mu)^2}{2\sigma^2}\right\} \quad \psi_e \in [0, 2\pi] \tag{4.16}$$

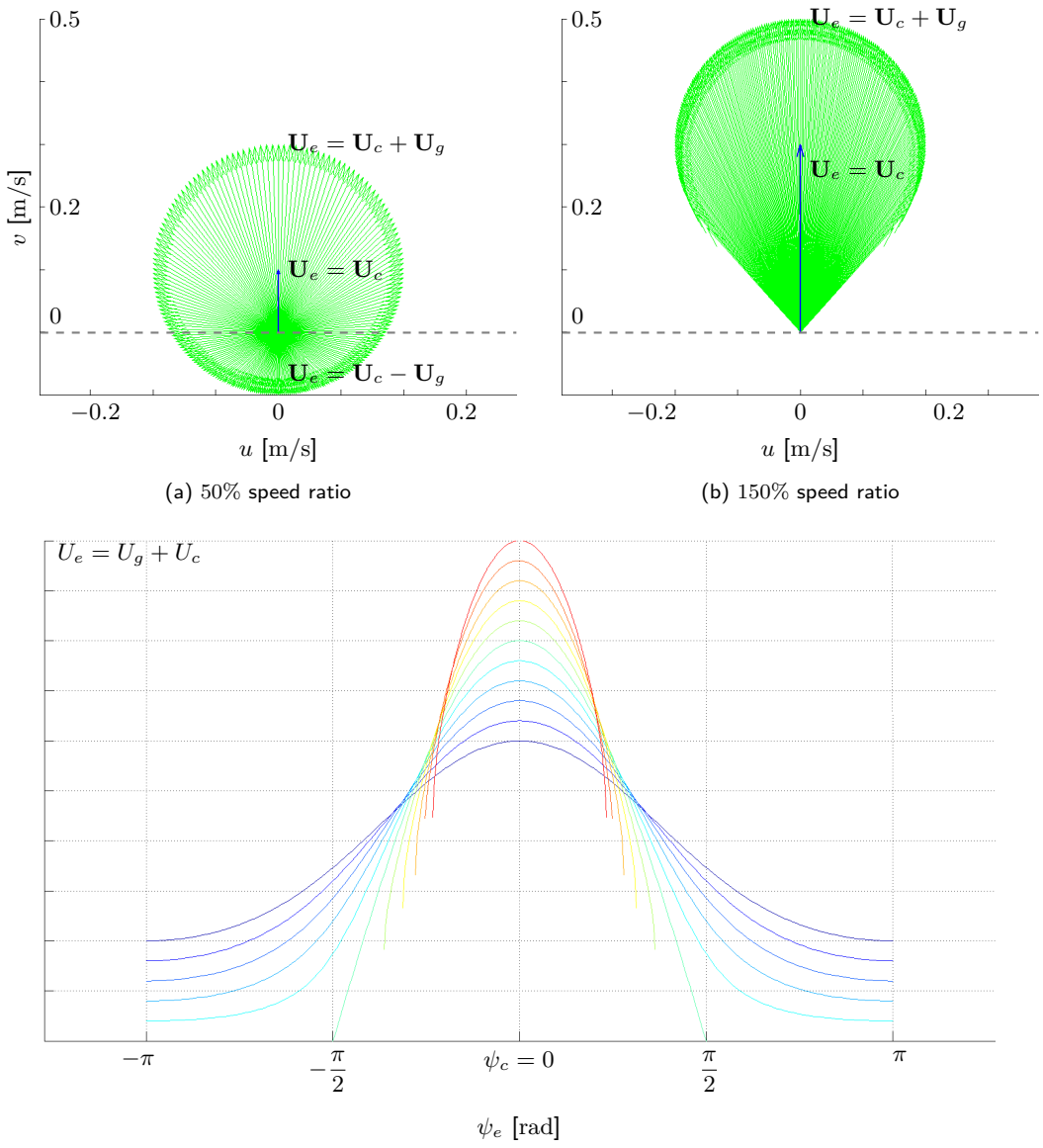
with mean $\mu = \psi_d$ in rad and variance σ^2. A realization of this Normal distribution on ψ_d is shown in Figure 4.9.

The composition of the PDF distributions discussed so far yields a PDF $f_\Theta(\psi_e) = f_{\psi_c}(\psi_e) + f_{\psi_d}(\psi_e)$ that models both factors simultaneously (see Figure 4.10). This new PDF $f_\Theta(\psi_e)$ allows to sample several heading angles ψ_e directly. However, the PDFs used so far, being discrete distributions, require a normalization satisfying $\sum_{i=1}^{k} w_i = 1$ to ensure a correct sampling —i.e. one that produces samples distributed according to the PDF distribution.

A powerful way to sample from nonparametric distributions is the Sequential Importance Sampling (SIS) algorithm, a Monte Carlo (MC) method that forms the basis for most sequential MC filters (Doucet and Johansen, 2008; Fishman, 1996; Hammersley and Handscomb, 1964). Developed by Madow (1949, 1953); Madow and Madow (1944) in a series of papers, the Sequential sampling algorithm is a method commonly used for this purpose. Figure 4.11 depicts the operation of the SIS algorithm. Instead of selecting samples independently of each other, the selection involves a sequential stochastic process. This is achieved by drawing a random number $r \in [0, M^{-1}]$, where M is the number of samples to draw. Samples are selected by repeatedly adding M^{-1} to r and choosing the sample that corresponds to the resulting number. Any number $U = r + (m-1)M^{-1} \in [0, 1]$ points to exactly one sample, namely the sample i for which

$$i = \arg\min_j \sum_{k=1}^{j} w_k \geq U \tag{4.17}$$

The advantage of the SIS algorithm is threefold. First, it covers the space of samples in a more systematic fashion than an independent random sampler, and it tends to not provide ill-suited samples that an uniformly



(a) 50% speed ratio

(b) 150% speed ratio

(c) PDF distributions $f_{\psi_c}(\psi_e)$ given by U_e

Figure 4.8: Effective speed vectors $\mathbf{U}_e$ for heading angles $\psi_e \in [-\pi, \pi]$, considering different speed ratios in (a) and (b); (c) PDF distributions $f_{\psi_c}(\psi_e)$ for $SR \in [50, 150]\%$ centered at ψ_c.

distributed set of samples would. Secondly, if all samples have the same importance factors, the resulting sample set will follow the same PDF distribution. Third, this low-variance sampler has $\mathcal{O}(M)$ complexity.

The SIS algorithm might produce repeated headings, as Figure 4.12 shows. Duplicated headings are replaced by uniformly distributed samples using the interval $(\psi_{e_i} - d_l, \psi_{e_i} + d_r)$ around the duplicated heading

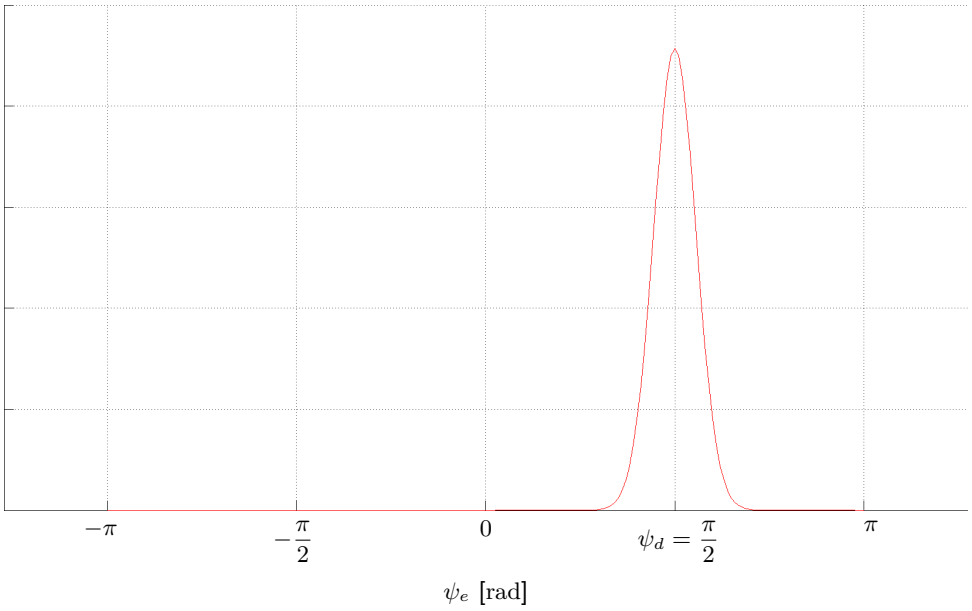


Figure 4.9: PDF distribution $f_{\psi_d}(\psi_e)$ that considers the heading $\psi_d = \frac{\pi}{2}$ direct to the goal, modeled with (4.16).

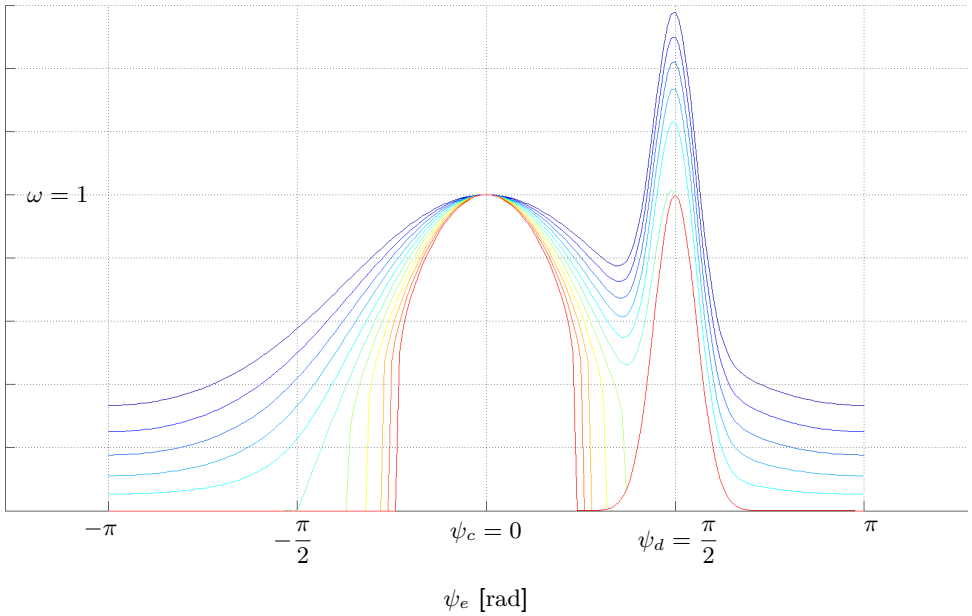


Figure 4.10: Combination of the PDF distributions $f_{\psi_c}(\psi_e)$ shown in Figure 4.8 (c), and a PDF distribution $f_{\psi_d}(\psi_e)$ that considers the direct heading $\psi_d = \frac{\pi}{2}$ to the goal, modeled with (4.16). $SR \in [50, 150]\%$.

ψ_{e_i}, where

$$d_l = \frac{\psi_{e_i} - \psi_{e_{i-1}}}{2} \tag{4.18}$$

$$d_r = \frac{\psi_{e_{i+1}} - \psi_{e_i}}{2} \tag{4.19}$$

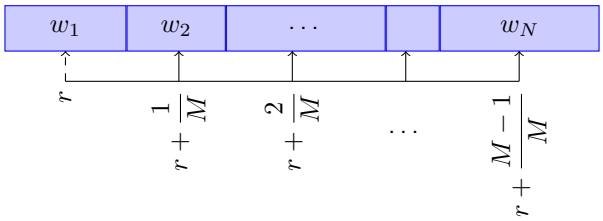


Figure 4.11: Sequential Importance Sampling (SIS) of M samples from N original weighted samples, where $M < N$, i.e. subsampling. When $M = N$ it is usually known as resampling, while it performs an oversampling when $M > N$.

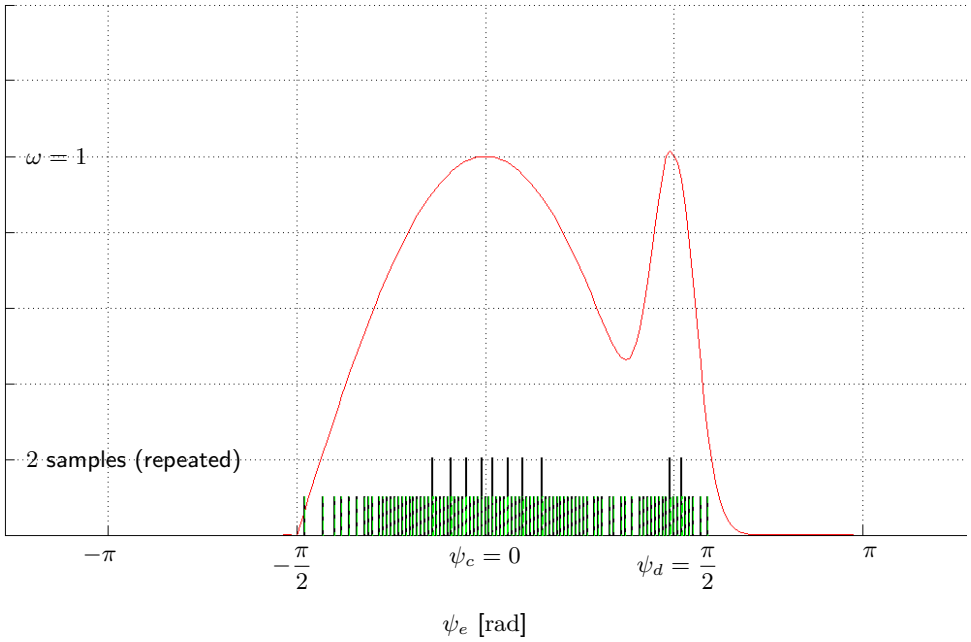


Figure 4.12: Proposed PDF distribution $f_\Theta(\psi_e)$ for $SR = 100\%$ —— and samples obtained after applying the SIS algorithm for $M = 100$ samples. Samples are shown with an histogram count of M bins —— to show repetitions, which are eliminated creating new samples - - - in a post-processing step.

are the midpoints of each side of ψ_{e_i}.

Once the set of heading samples ψ_{e_i} is drawn from $f_\Theta(\psi_e)$ in Figure 4.10 with the appropriated speed ratio, the bearing angles ψ_{g_i} for each ψ_{e_i} are given by (4.12), for $i = 1, \ldots, M$. Recall that (4.12) is undefined for $|s| > 1$, where s, given by (4.14), depends on SR. On those cases, we assume $U_g \gg U_c$, so $s \to 0$ and therefore $\psi_{g_i} = \psi_{e_i}$.

In Section 5.4.3 we will compare the uniform sampling with the ABS sampling discussed here. We consider the PDF distribution proposed here, although different distributions may be defined as well, and the influence of the number M of bearing samples is analyzed. With the ABS strategy we pretend to mitigate the exponential growth of the computing time with M, by keeping it small using an informed, intelligent sampling strategy that allows to obtain nearly optimal paths in better time using less M bearing samples.

4.5 Extension of Rapidly-Exploring Random Tree algorithm

The Rapidly-exploring Random Tree or RRT technique (Lavalle, 1998) is based on a random generation of test cases on the problem domain, building up an exploring tree with nodes that tend to cover the search space. RRT is included in the Rapidly-exploring Dense Tree (RDT) family of methods, which were originally developed for handling differential constraints (LaValle, 2006). In fact, the glider path planning problems can be modeled by the differential equations and constraints of the vehicle motion model, considering the ocean currents (see Section 2.2). Contrary to other algorithms, which focus on finding an optimal path, RDTs are aimed at returning only feasible trajectories, not necessarily optimal. Their main advantage is that they produce a solution trajectory with relatively less sampling than other methods, and consequently they run much faster. Indeed, this performance gain is enabled in part by the lack of concern for optimality.

Algorithm 4.3 RRT algorithm pseudo-code.

Require: Start $\mathbf{x}_0$ and target $\mathbf{x}_{\text{goal}}$ locations, and number of iterations K.
Ensure: Tree or graph $\mathcal{G}$, that will connect $\mathbf{x}_0$ with $\mathbf{x}_{\text{goal}}$.
Algorithm: RRT($\mathbf{x}_0$, K) **return** $\mathcal{G}$

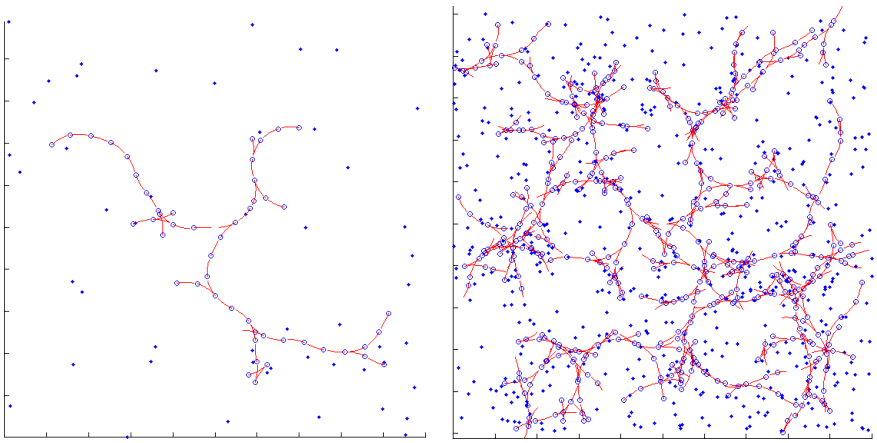
1: $\mathcal{G}$**.init**($\mathbf{x}_0$) ▷ add start location
2: **for all** 1 to K **do**
3: $\mathbf{x}_{\text{rand}} =$ **rand** ▷ new random location
4: $\mathbf{x}_{\text{near}} =$ **NN**($S(\mathcal{G})$, $\mathbf{x}_{\text{rand}}$) ▷ Nearest Neighbor location in the *swath* $S(\mathcal{G})$
5: $\mathbf{x}_{\text{new}} =$ **new**($\mathbf{x}_{\text{near}}$, $\Delta\mathbf{x}$) ▷ new location
6: $\mathcal{G}$ **.add_vertex**($\mathbf{x}_{\text{new}}$)
7: $\mathcal{G}$ **.add_edge**($\mathbf{x}_{\text{near}}$, $\mathbf{x}_{\text{new}}$)
8: **end for**
9: **return** $\mathcal{G}$

In Algorithm 4.3 the basic RRT algorithm is presented; note the the RDT algorithms are considered as RRT when they use random samples in the configuration/search space, rather than a deterministic sequence. In brief, the algorithm builds a graph $\mathcal{G}$, which is actually the exploring tree. The root of the tree is the start location $\mathbf{x}_0$. The pseudo-code shown in the algorithm is the original version of Lavalle (1998), but adapted to the notation of the space configuration of glider path problems. Then, for a given number of iterations K, the RRT algorithms sample randomly a new location $\mathbf{x}_{\text{rand}}$. In the general case, this random location must be validated, in order to check that it is a feasible location —e.g. it does not collide with an obstacle. Fortunately, in the case of glider missions, we usually do not have this problem because they operate offshore. Then, the nearest neighbor **NN** node in the *swath* $S(\mathcal{G})$ of $\mathcal{G}$ is chosen; the *swath* of a graph $\mathcal{G} = (V, E)$ is simply the reachability of all edges $e \in E$, since they define a path, that is

$$S(\mathcal{G}) = \bigcup_{e \in E} e([0, 1]) \tag{4.20}$$

where $e([0, 1])$ is the image of the path e (LaValle, 2006). The nearest neighbor node $\mathbf{x}_{\text{near}}$ is used to generate a new location $\mathbf{x}_{\text{new}}$ by moving from $\mathbf{x}_{\text{near}}$ an incremental distance Δq in the direction of $\mathbf{x}_{\text{rand}}$. That new node $\mathbf{x}_{\text{new}}$ is added to $\mathcal{G}$ along with and edge that connects $\mathbf{x}_{\text{near}}$ with it.

After a certain number of iterations, the RRT algorithm exhibits its innate biased towards exploration, filling the configuration space. In Figure 4.13 (a), after $K = 50$ iterations, we observe an example run of the RRT algorithm on a bi-dimensional search space. Even at this early stage, we observe how the search space is almost uniformly sampled. It is after $K = 500$ iterations in (b), when we see a more dense tree that covers almost all the search space, with some gaps that would be filled in further iterations. It is also easy to observe that despite of the good exploration capabilities of this approach, the solution —if found— will not be optimal and will require further refinement or some sort of smoothing procedure. However, it is important



(a) 50 iterations. (b) 500 iterations.

Figure 4.13: Example of an RRT tree after (a) 50 and (b) 500 iterations.

to note that in the case of differential problems like glider navigation, a small change in the location of any node of the path found will result in a different trajectory, because the ocean currents may be different at the new locations, and hence the motion model will yield a new ending location. This is an important drawback of this approach for glider path planning.

When applied to navigation, a bias is used to direct trajectories towards the goal location (Rao and Williams, 2009; Tan et al., 2004). In fact, our implementation is based on the work of Rao and Williams (2009), which biases the RRT growth. They point out that the probability distribution for any RRT implementation is a critical aspect. On one hand, the growth of the tree must be extensive enough to ensure optimal solutions are not overlooked, but on the other hand, it must be small enough to avoid expanding into unnecessary areas. They propose applying the adjusted Iterative k-Nearest RRT (IkRRT) algorithm of Simmons and Urmson (2003) which weights the Voronoi regions of each node by some heuristic cost, and adjusts the RRT implementation to consider the k-nearest neighbors rather than a single neighbor. Additionally, to apply the RRT algorithm to glider path planning, they advice to generate RRTs bidirectionally, i.e. one from the start $\mathbf{x}_0$ and one from the target waypoint $\mathbf{x}_{n-1}$; the latter is generated in reverse to simulate the actual glider motion.

In our preliminary experiments we have observed that at least two RRTs must be generated simultaneously, in order to obtain good results. With a single RRT from the start waypoint $\mathbf{x}_0$ the path cost is very poor, or we simply do not reach the target. The drawback of requiring at least 2 RRTs is that one of them must be run from the target waypoint $\mathbf{x}_{n-1}$, and in the case of dynamic ocean currents we need to know the time when the glider will reach it. Clearly, we do not know this information, unless we solve the problem. Therefore, we cannot generate a RRT from $\mathbf{x}_{n-1}$, since we do not know the time to retrieve the ocean currents corresponding to it. For this reason, the RRT algorithm is only used with static ocean currents in the experiments of this thesis.

4.6 Analysis and Applicability of Optimization techniques

Apart from the graph-based search methods discussed thus far, it is possible to apply optimization methods to minimize an objective function that models the glider path planning problem at hand. This objective function will clearly depend on the particular problem we are solving. In this thesis, we have seen that it is possible

to solve all the problems enumerated in Chapter 3 using a single objective function that solves the minimal distance to the target problem (see Section 3.1.1). For the rest of the problems, we build a more elaborated algorithm on top of this one.

To solve the minimal distance to the target problem we define an objective function based on the ending point $\mathbf{x}_{n-1}$ of the integration of the glider trajectory for all the bearing angles $\mathcal{B}$ commanded from the starting waypoint $\mathbf{x}_0$,

$$\mathbf{x}_{n-1} = f(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G}) \tag{4.21}$$

where $\mathcal{C}$ is the ocean currents map and $\mathcal{G}$ is the glider setup —i.e. the nominal speed U_g and other parameters used in the unconstrained motion model of Section 2.2.2.1. Since we want to minimize the distance to the target waypoint $\mathbf{x}_{\text{goal}}$, we build our objective function as the remaining distance between $\mathbf{x}_{n-1}$, given by (4.21), and $\mathbf{x}_{\text{goal}}$. That is, the objective function f' is

$$f'(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G}) = \|\mathbf{x}_{\text{goal}} - f(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G})\| \tag{4.22}$$

where $\|\cdot\|$ is a distance metric, which in our case is the length of the arc of the great circle that passes through $\mathbf{x}_{n-1}$ and $\mathbf{x}_{\text{goal}}$, since we are navigating the spherical geometry that approximates the Earth; for short-term missions, with small distance, we can use another approximation or even an Euclidean distance (see Section A.2) though. Indeed, to avoid computing square roots during the optimization process, we can take the squared distance $\|\cdot\|^2$, so the final objective function implemented is

$$f'(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G}) = \|\mathbf{x}_{\text{goal}} - f(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G})\|^2 \tag{4.23}$$

It is important to note that the number of bearing angles in $\mathcal{B}$ must be known in advance. For the minimal distance to the target problem, this is given *a priori*, since we want to leave the glider as close as possible to the target $\mathbf{x}_{\text{goal}}$ in a given time T. The number of bearings $|\mathcal{B}|$ is obtained as T/t_s, where t_s is the so called surfacing time, also known in advance hence. On the contrary, for the minimal time problem $|\mathcal{B}|$ is unknown *a priori* —indeed, it is the output of the objective function. Therefore, we will use an iterative scheme (see Section 4.6.1) on top of the Optimization method explained here.

Regarding the optimization method kernel, we have tried several techniques, in order to analyze how they perform with glider path planning problems. Most of these kernels solve the following minimization problem,

$$\min_x f(x) \tag{4.24}$$

which in some cases can be subject to constraints. Another alternative formulation is the Least Squares (LS) one, which solves this slightly different problem,

$$\min_x \|f(x)\|^2 \tag{4.25}$$

where $f(x)$ might be a non-linear function.

It is easy to adapt our objective function (4.23) for both formulations. In the first case, we minimize our objective function directly. That is,

$$\min_{\mathcal{B}} f'(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G}) \tag{4.26}$$

which yields the best path as a sequence of bearings $\mathcal{B}$. We can observe that by plugging the value of (4.23), we actually have the formulation of a LS optimization problem,

$$\min_{\mathcal{B}} \|\mathbf{x}_{\text{goal}} - f(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G})\|^2 \tag{4.27}$$

Therefore, if we use this kind of optimization kernels, we only have to pass the difference between the target waypoint $\mathbf{x}_{\text{goal}}$ and (4.21),

$$\mathbf{x}_{\text{goal}} - f(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G}) \tag{4.28}$$

which provides the remaining distance to $\mathbf{x}_{\text{goal}}$. It is important to highlight that in any case, the optimization is in the space of the bearing angles $\mathcal{B}$, which makes the problem scale well for long-term missions.

All optimization kernels require an initial guess, in order to iterate numerically to the optimal solution. Therefore, we have an **initialization phase** concerned with the generation of a feasible path that can be used as an initial guess. We need this phase to be extremely fast, so it does not introduce an excesive overhead on the whole algorithm. Therefore, we use very simple approaches. In fact, we could use two strategies. First, we simply set all the bearings $\mathcal{B} = \psi_d$, where ψ_d is the direction direct to the goal, regardless of the actual glider trajectory if it would follow them. With this first strategy we assume that $\mathcal{B}$ would produce a straight line trajectory from $\mathbf{x}_0$ to $\mathbf{x}_{\text{goal}}$. The main advantage of this approach is that it is extremely fast, but the solution may not be feasible in some cases, since it can lead the optimization kernel to local minima, far from the actual global minimum. A second strategy consists on applying the DtG approach discussed in Section 4.2, which in most cases gives a sub-optimal, but feasible solution. Only when the scenarios are very complex —with strong ocean currents and many obstacles—, this approach fails, and we need a more intelligent and elaborated initialization phase, as it will be explained in the sequel in Section 4.6.2.

The next Section 4.6.1, extends the Optimization method explained in this section, to the minimal time path problem. With this problem, contrary to the minimal distance to the target problem covered here, the number of bearings $\mathcal{B}$ is unknown. However, an iterative approach uses the optimization kernel to solve the problem by successive optimization runs. Then, we improve the initialization phase explained above in order to avoid obstacles. The details of this intelligent initialization phased are explained in Section 4.6.2. Finally, we evaluate different optimization kernels with the Iterative Optimization method in Section 4.6.3; note that this iterative approach calls the kernels several times, so it allows for a more representative evaluation of the kernels.

4.6.1 A novel Iterative Optimization scheme

When the number of bearings $|\mathcal{B}|$ is unknown in advance, the previous Optimization approach is not sufficient. However, we can build an iterative approximation in which we basically start with an optimistic, small $|\mathcal{B}|$ that is incremented in successive calls to the optimization kernel, until we reach the goal $\mathbf{x}_{\text{goal}}$. Consequently, we will end up reaching $\mathbf{x}_{\text{goal}}$ in the minimal time. In fact, with this Iterative Optimization approach we solve the minimal time path problem described in Section 3.1.1.

The iterative approximation of this method is described in the pseudo-code of Algorithm 4.4. As in the previous Optimization method, in a first initialization step performs the computation of a coarse, optimistic underestimation of the number of bearings $\left|\hat{\mathcal{B}}\right|$ required to reach the target waypoint $\mathbf{x}_{\text{goal}}$. The function **length** in line 4 provides $\left|\hat{\mathcal{B}}\right|$ by taking the line-of-sight trajectory from the current ending location $\mathbf{x}$ — which initially is the starting location $\mathbf{x}_0$— to $\mathbf{x}_{\text{goal}}$. This function establishes the estimate number of stints —and consequently $\left|\hat{\mathcal{B}}\right|$— of duration t_s, required to reach $\mathbf{x}_{\text{goal}}$ assuming an ocean current $U_{c_{\max}}$. In our implementation we consider the maximum of all ocean currents around the mission area, to set $U_{c_{\max}}$. Since this might be quite pesimistic, in many cases we have reduced the area used to compute it, to a patch around the start and target waypoints.

Once we have the estimate $\left|\hat{\mathcal{B}}\right|$, the problem is solved as with the Optimization approach. Therefore, we obtain an initial guess with the function **init** of line 5. This function includes two options to obtain the list of bearings $\hat{\mathcal{B}}$; that is, it considers the bearing angles in the direction from $\mathbf{x}$ to $\mathbf{x}_{\text{goal}}$, or it applies the DtG approach for them. Then we extend the list of bearings $\mathcal{B}$ from $\mathbf{x}_0$ to the ending location $\mathbf{x}$, with the guess $\hat{\mathcal{B}}$ from $\mathbf{x}$ to the target $\mathbf{x}_{\text{goal}}$ (see line 6).

Once the initial bearings $\hat{\mathcal{B}}$ have been computed, the optimization kernel is run with this initial guess, starting at the last ending location $\mathbf{x}$. If the target waypoint $\mathbf{x}_{\text{goal}}$ is not met with the desired precision $t_{\max}$, the loop iterates again, expanding the list of bearings $\mathcal{B}$ with a new guess starting at the ending location $\mathbf{x}$

Algorithm 4.4 Iterative Optimization algorithm pseudo-code.

Require: The start $\mathbf{x}_0$ and target $\mathbf{x}_{\text{goal}}$ points, the ocean currents map $\mathcal{C}$, the glider setup $\mathcal{G}$ and the target precision time radius $t_{\max}$.
Ensure: List of bearing angles $\mathcal{B}$ for the path of minimal cost found.
Algorithm: iterative_optimization($\mathbf{x}_0$, $\mathbf{x}_{\text{goal}}$, $\mathcal{C}$, $\mathcal{G}$, $t_{\max}$) **return** $\mathcal{B}$

$\mathbf{x} = \mathbf{x}_0$ ▷ last surfacing point so far
$\mathcal{B} = \varnothing$ ▷ set of bearing angles that represent the path solution
repeat
  $\left|\hat{\mathcal{B}}\right| = \textbf{length}(\mathbf{x}, \mathbf{x}_{\text{goal}}, \mathcal{C}, \mathcal{G})$ ▷ # bearings required to reach $\mathbf{x}_{\text{goal}}$ from $\mathbf{x}$
  $\hat{\mathcal{B}} = \textbf{init}(\mathbf{x}, \mathbf{x}_{\text{goal}}, \left|\hat{\mathcal{B}}\right|)$ ▷ initialize $\left|\hat{\mathcal{B}}\right|$ new bearing angles
  $\mathcal{B} = \mathcal{B} \cup \hat{\mathcal{B}}$
  $\langle \mathcal{B}, \mathbf{x} \rangle = \textbf{optimize}(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G})$
  $U_{c_{\max}} = \max_{\mathbf{x}_i \sim \mathbf{x}} \mathcal{C}(\mathbf{x}_i)$ ▷ max current around $\mathbf{x}$, e.g. a rectangle
  $t = \dfrac{\|\mathbf{x}_{\text{goal}} - \mathbf{x}\|}{U_{c_{\max}}}$ ▷ time distance from last surfacing point to the target
until $t < t_{\max}$
return $\mathcal{B}$

of the last optimization iteration.

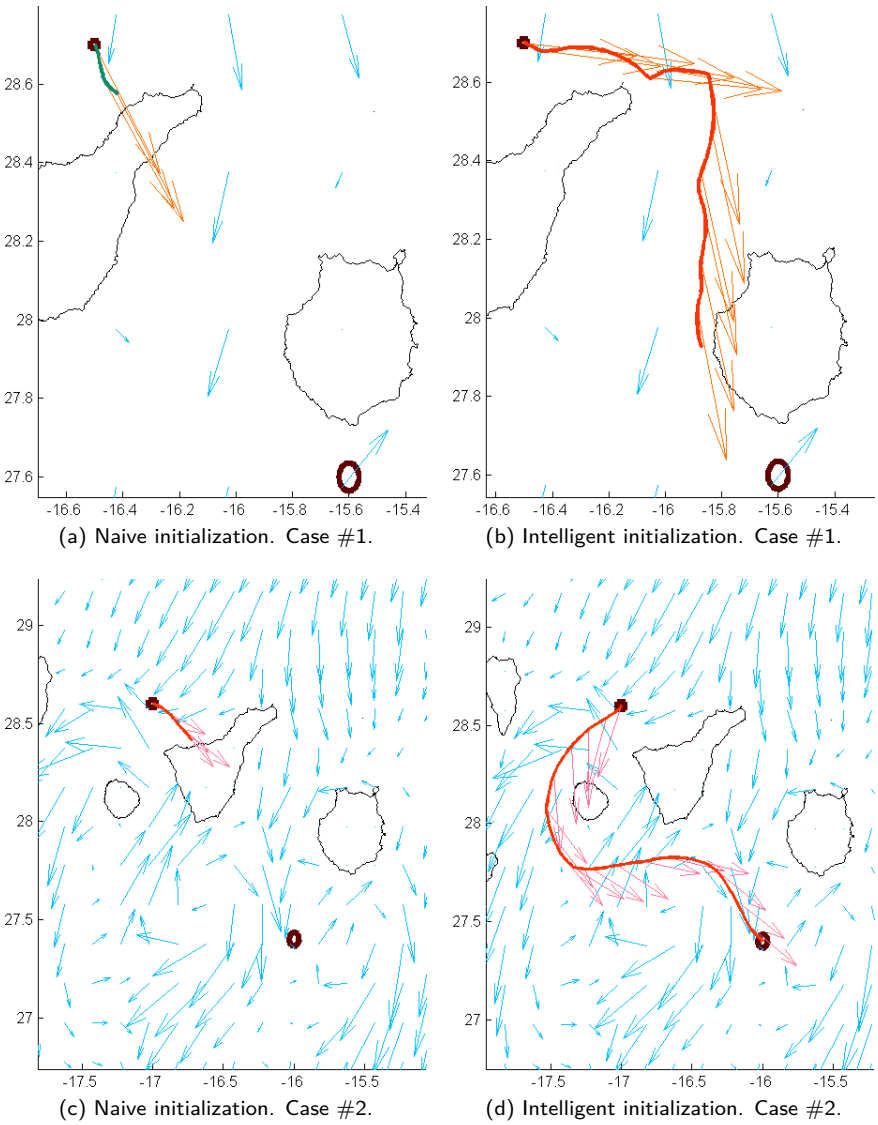
The termination condition is computed on the remaining time t. To calculate t, we divide the distance between the last ending location $\mathbf{x}$ and the target $\mathbf{x}_{\text{goal}}$, by $U_{c_{\max}}$. In the algorithm, the distance computation is represented by $\| \cdot \|$, which can be a great circle distance for long-term missions, or the Euclidean distance for short-term ones, or another distance metric approximation (see Section A.2). This process iterates until the precision time radius $t_{\max}$ is met, yielding the optimal path in the list of bearings $\mathcal{B}$, and its ending location $\mathbf{x}$. Since the stints have a constant time duration t_s, the termination condition must be defined for a time radius $t_{\max} \geq t_s$; otherwise the optimization will probably oscillate around the minimum.

4.6.2 Intelligent Initialization and Obstacle Avoidance

The initialization approaches described thus far, produce acceptable results for offshore areas, which are only affected by moderately strong ocean currents at most, and where we do not have any obstacles. However, as indicated previously, for short-term coastal navigation, the naive initialization using a straight line to the target or the DtG approach fails in these scenarios. As a consequence of the complexity of these environments, and the coupled nature of the process variables, the Optimization method can easily get trapped in local minima or lead to wrong, non-optimal solutions, including paths with collisions.

The Figure 4.14 illustrates this problem for a couple of test cases. For both of them we have run the naive initialization phase described thus far, and then the optimization method, for the figures (a) and (c), on the left. In both cases, the path does not reach the target waypoint because it gets trapped in an obstacle —the land, in the examples. On the contrary, with the intelligent initialization phase described here in the sequel, we obtain a rough feasible path which is successfully optimized later, as shown in the figures (b) and (d), on the right. Here, our intention is only to show that the naive initialization approach fails to find a path, so an intelligent initialization phase is required. The quality of the results will be later evaluated in the Chapter 5 by experimental comparison with graph-based path planning algorithms.

To overcome the limitation of the naive initialization when it is required to have some sort of obstacle avoidance logic in the path planning algorithm, we have developed an intelligent initialization phase which is run before the optimization method, in order to provide a feasible and good initial guess. Of course, being



(a) Naive initialization. Case #1.

(b) Intelligent initialization. Case #1.

(c) Naive initialization. Case #2.

(d) Intelligent initialization. Case #2.

Figure 4.14: Path found for 2 test cases with obstacles, using the Optimization method for the minimal distance to the target problem during 4 days. Each figure shows the resulting path for the glider bearing angles $\mathcal{B}$ found → for each surfacing point, along with the dynamic ocean currents →, from the start ● to the target waypoint ◎. The naive initialization approach is shown in (a) and (c), where no path that reach the target is found; meanwhile, (b) and (d) shows the successful result with an optimal path that reaches the target, by using the intelligent initialization phase.

an initialization phase, it must run fast. The advantage is that the initial guess does not have to be optimal, so a rough path is enough. For the sake of clarity, we termed our method (Iterative) Optimization with Intelligent Initialization, or just (Iterative) Optimization-CTS-A* for short. Hence, the initialization phase can be thought as a bootstrap module, which is inspired on a coarse version of CTS-A* Section 4.4 and ND algorithms (Minguez et al., 2004), and that generates an appropriate initial set of bearing angles $\mathcal{B}$ to start

the optimization phase described thus far. There, that initial guess is refined by the (iterative) optimization method until no improvement is obtained, generating the final solution —the optimal path.

We have said that our intelligent initialization phase is based on the CTS-A* algorithm. However, the problem of CTS-A* is its time complexity. In order to alleviate the high computing time, the following simplification strategy is applied during the initialization phase. On one hand, a scale factor is applied on the stint duration to reduce the resolution of the search space grid. This factor is computed as a function of the straight-line trajectory length. On the other hand, the bearing rose used for the node expansion is discretized and pruned according to the most probable trajectory orientation. The trajectories are then expanded recursively according with this procedure, starting from the initial point $\mathbf{x}_0$ and ending when the target $\mathbf{x}_{\text{goal}}$ is reached. An Euclidean distance metric is used as the cost function to guide the search, as well as an heuristic, optimistic estimation of the composition of the glider and ocean currents speed. The whole intelligent initialization phase comprises 4 steps, which are explained below.

1. First, the initialization process considers a set of evenly spaced radial vectors emanating from the starting point $\mathbf{x}_0$ at different bearing angles, and simulates the glider trajectory with that fixed bearing for each one, inside the temporal horizon T of the mission (see Figure 4.15 (a)). For the minimal distance to the target problem T is unknown, but for the minimal time path problem we assume some estimation; note that the Iterative Optimization method used for the minimal time problem applies an incremental iterative approximation of the number of bearings $|\mathcal{B}|$, so it is not a problem if T is underestimated, and the initial guess does not really reach the target $\mathbf{x}_{\text{goal}}$.

2. Secondly, a set of nodal points is selected for each trajectory. These points are candidate nodes n_i that are selected at evenly spaced surfacing points, as shown in Figure 4.15 (b). Generally, this is equivalent to evenly spaced time instants, because of the constant time surfacings t_s.

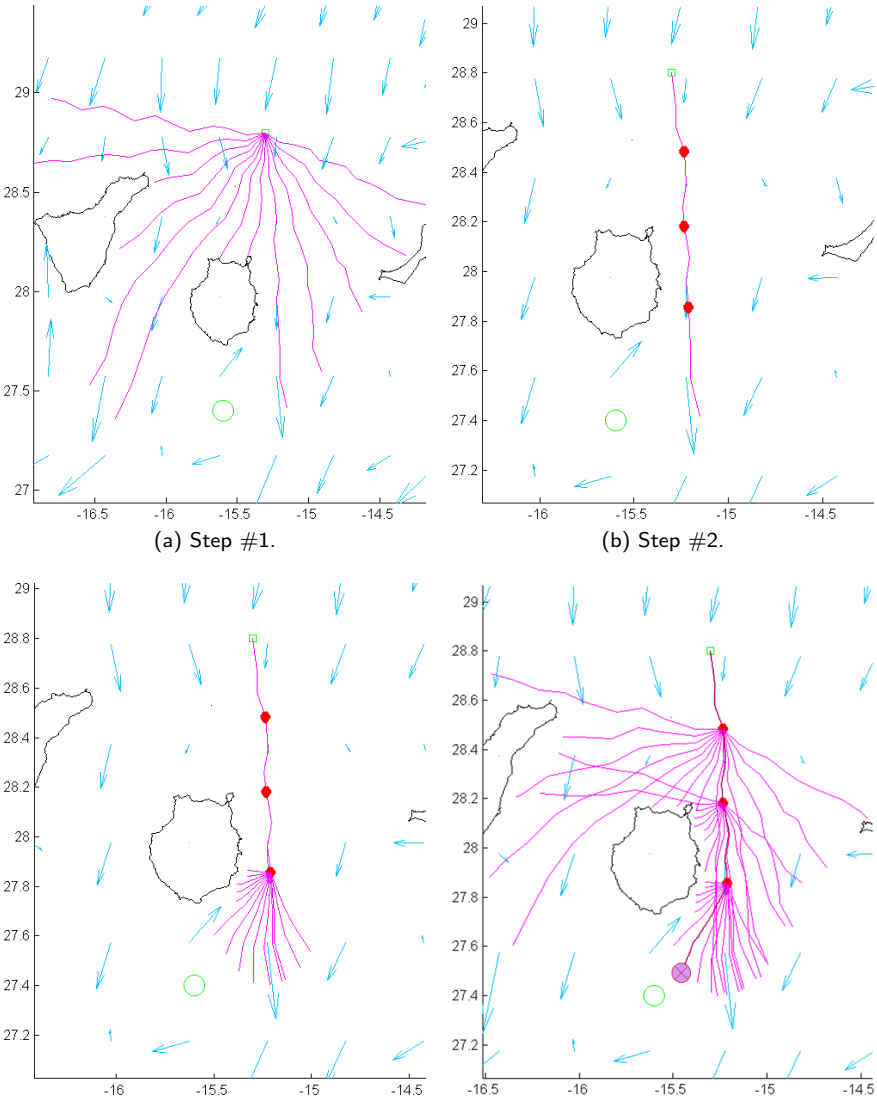
3. Thirdly, the algorithm considers a set of evenly spaced radial vectors again, emanating from every node n_i and simulates the glider navigation with a fixed bearing for each trajectory during the remaining temporal horizon. That is, a new set of trajectories is generated for each candidate node n_i recursively, simulating them for the remaining mission time (see Figure 4.15 (b)).

4. Finally, the bearing angles $\mathcal{B}$ of the trajectory that reaches the nearest location to the target point $\mathbf{x}_{\text{goal}}$ are selected as initial guess value for the optimization process (see Figure 4.15 (d)).

In practice, we have observed that it suffices to divide the trajectory in a single turning point —i.e. one node n_i— for short-term mission of $T = 4$ days, where the glider travels up to $\approx 100 - 150\text{km}$. For long-term mission we will need more nodal points, so the initialization phase will take slightly more time to finish.

In brief, the initialization process makes a division of candidate trajectories in 2 or more stretches. These candidate trajectories use a fixed bearing angle for all the stints into one stretch; note that one stretch simulate multiple glider navigation stints of a constant time surfacing t_s. The nodal division points between stretches, in the initialization algorithm, are allowed to have a flexible location n_{i+1} within the stretch defined by the nodes n_i and n_{i+2}, according with the diagram of the Figure 4.16.

Once the intelligent initialization phase finishes, the set of bearings $\mathcal{B}$ that defines a trajectory whose ending location has the minimal distance to the target waypoint $\mathbf{x}_{\text{goal}}$ is used as the initial guess in the next optimization phase. For the minimal distance to the target problem, the Optimization method of Section 4.6 tries to minimize the objective function from the ending point of that initial guess to the target, in order to find the optimal path. In the case of the minimal path problem, we use the Iterative Optimization method of Section 4.6.1, which tries to minimize the number of stints by reducing progressively the bearing set $\mathcal{B}$, by removing the last element. This reduction optimization loop is repeated until the optimal result is found. Note that in this case, instead of incrementing the number of bearing $|\mathcal{B}|$ (see Algorithm 4.4), we decrement them, because the initial guess is a non-optimal path, i.e. it has more bearings than the optimal one. However,



(a) Step #1. (b) Step #2.

(c) Step #3. (d) Step #4.

Figure 4.15: The Intelligent initialization for Obstacle Avoidance comprises the following steps: (a) First step of the initialization process: Radial vectors emanating from the starting point. (b) Second step of the initialization process: Selection of candidate nodes. (c) Third step of the initialization process: Radial vectors emanating from each candidate node. (d) Fourth step of the initialization process: Selection of the best trajectory.

depending on the approximation of the mission time horizon T used, the opposite it may happen, so we actually check whether we have reached the target or not, to disambiguate the actual case.

In conclusion, as it was initially shown in Figure 4.14 (b), this intelligent initialization approach is able to find a rough, non-optimal, but feasible path to the target waypoint $\mathbf{x}_{\text{goal}}$, which avoids obstacles like the coastline, land areas, or strong currents zones. Meanwhile, under the same conditions, the previous naive

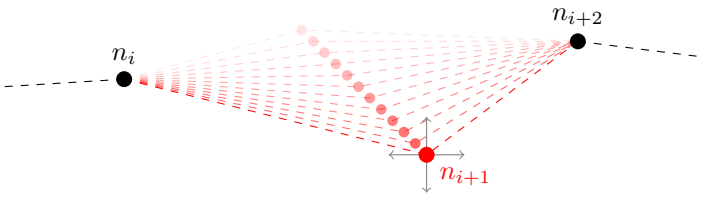


Figure 4.16: Diagram of the partition of the intelligent initialization stretches. For a given stretch from the node n_i to n_{i+2}, a new node n_{i+1} can be inserted between them. Its location is flexible, since we allow to move it in any direction in the plane.

initialization approach was not able to find a good solution (see Figure 4.14 (a)), and consequently, the optimization phase fails to find any solution, or it gets trapped in a local minima, so the optimal path is not found.

4.6.3 Evaluation of Optimization kernels

Regarding the optimization kernel used in both the Optimization and Iterative Optimization methods, we have compiled a set of test cases to evaluate the path cost of the solution found, and the computing time, which are shown in Table 4.1 and 4.2, respectively. Since we have done the evaluation for the minimal time path problem —with the Iterative Optimization method—, the path cost is the travel time. As we already said in the previous section, there are two formulations for the optimization problem. For the LS formulation we have tested a single kernel, called **lsqnonlin** in the tables, which supports non linear problems, as in our case. Then, for the general formulation, we have tested three kernels: **fminsearch**, **fminunc** and **fmincon**. The actual optimization techniques implemented by these kernels are mentioned below:

fminsearch: It uses the Nelder-Mead simplex direct search of Lagarias et al. (1998).

lsqnonlin: It solves non-linear LS problems by means of the Trust-Region-Reflective method (Coleman and Li, 1996) or Levenberg-Marquardt (Levenberg, 1944; Moré, 1978), with a default $\lambda = 0.01$, where λ is the so called damping parameter. The results in the tables are for the Trust-Region-Reflective method.

fminunc: It solves unconstrained minimization problems with the medium scale Quasi-Newton algorithm (Fletcher and Powell, 1963).

fmincon: It solves minimization problems, which can be subject to constraints. It implements the Active-Set (Gill et al., 1981) or the Interior-Point (Byrd et al., 1999) algorithms. Here, we have used the Active-Set method. It also supports Sequential Quadratic Programming (SQP) (Biggs, 1975; Nocedal and Wright, 2006) and Broyden-Fletcher-Goldfarb-Shanno (BFGS) methods (Broyden, 1970), which were tested in some preliminary experiments without acceptable results, so it is not included in the tables here.

For the 10 test cases run with a glider nominal speed of $U_g = 0.4\mathrm{m/s}$, and shown in Table 4.1 and 4.2, the best kernel is **fmincon**; we use it as a reference, and the speedup is computed for the other kernels, which is always below the 100%. This means that it is the kernel that provides the best results for both the minimal time path and minimal distance to the target problems. It also has the lowest average computing time, so in the experiments of Chapter 5 it will be the kernel used for any setup of the Optimization and Iterative Optimization methods. The drawback of other methods with similar optimality or computing time, is that in some test cases they get trapped in local minima, while **fmincon** does not. Furthermore, with **fmincon** we can solve optimization problems which are subject to constraints, like the path following and multi-glider problems of Section 3.3 and 3.4.

Table 4.1: Comparison of the travel time for different optimization kernels used in the Optimization method, for the minimal time path problem. The glider nominal speed was set to $U_g = 0.4\mathrm{m/s}$. The table shows the speedup of the travel time with respect to the **fmincon** kernel, i.e. $< 100\%$ is worse.

Case	$d[\mathrm{km}]$	**fminsearch**	**lsqnonlin**	**fminunc**
1	831.33	99.46%	99.64%	99.64%
2	1007.50	100.00%	100.00%	100.00%
3	974.34	27.94%	39.48%	110.21%
4	500.18	99.87%	100.00%	100.00%
5	861.88	100.00%	100.00%	100.00%
6	1050.70	18.74%	11.36%	60.86%
7	1158.70	108.04%	72.82%	79.29%
8	151.40	100.00%	100.00%	100.00%
9	1197.80	99.80%	100.00%	91.84%
10	136.07	100.00%	100.00%	100.00%
Mean	786.99	85.38%	82.33%	94.18%

Table 4.2: Comparison of the computing time for different optimization kernels used in the Optimization method, for the minimal time path problem. The glider nominal speed was set to $U_g = 0.4\mathrm{m/s}$. The table shows the speedup of the computing time with respect to the **fmincon** kernel, i.e. $< 100\%$ is worse.

Case	**fminsearch**	**lsqnonlin**	**fminunc**
1	3.24%	6.26%	16.06%
2	9.40%	48.36%	51.92%
3	14.21%	55.83%	620.20%
4	6.15%	47.70%	45.20%
5	7.47%	51.08%	38.54%
6	0.58%	0.83%	9.26%
7	6.70%	28.77%	5.86%
8	26.74%	49.29%	77.61%
9	17.79%	33.78%	25.43%
10	20.42%	75.00%	65.00%
Mean	11.27%	39.69%	95.51%

4.7 Boundary Value Problem models and methods

The minimal time path problem in the context of ocean currents have been traditionally solved as a two-point Boundary Value Problem (BVP). This method gives the optimal solution in the continuous domain. It basically set the start $\mathbf{x}_0$ and target $\mathbf{x}_{\text{goal}}$ waypoints as the boundary conditions of the problem and finds the optimal trajectory solving numerically the differential equations that model the problem. The interest of this method relies on the fact that it provides the real optimal solution without any sort of discretization —the ocean currents are simply interpolated. However, its main drawback resides on the fact that it fails to solve the problem for strong and opposite ocean currents. Therefore, in most cases it is only useful for validation purposes using test cases with synthetic ocean currents, instead of real ones.

The general form of the two-point BVPs is a system of ODEs (Ordinary Differential Equations)

$$\dot{y} = f(x, y, p) \tag{4.29}$$

and a set of boundary conditions at the two points a and b

$$0 = g(y(a), y(b), p) \tag{4.30}$$

where p is a vector of unknown parameters (Shampine et al., 2003). In the next sections we describe the two main optimal formulas to solve this problem in the context of the minimal time path problem for glider path planning. These optimal formulas are Zermelo's optimal navigation formula, and the more general Hamilton's equations. The basic difference among them is that Zermelo's considers 3 DOF, being the x and y position, and the bearing ψ. On the other hand, Hamilton's equations models the bearing in a more general way with two ODEs. Clearly, Zermelo's optimal navigation formula represents the glider unconstrained motion model in 2 dimensions, although the notation here is different. Also, the BVP usually requires the ODEs system to be augmented with an additional ODE $\dot{\tau}$, as it happens in the equations below.

4.7.1 Zermelo's Optimal Navigation Formula

It solves the adimensional equations

$$(x(t),\ y(t),\ \psi(t),\ \tau(t))^{\top} \tag{4.31}$$

subject to the ODE system

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{\tau} \end{pmatrix} = \tau \mathbf{A}_{4\times 1} \tag{4.32}$$

where

$$\mathbf{A}_{4\times 1} = \begin{pmatrix} u + U_g \cos\psi \\ v + U_g \sin\psi \\ v_x \sin^2\psi + u_x \sin 2\psi - u_y \cos^2\psi \\ 0 \end{pmatrix} \tag{4.33}$$

and with boundary conditions

$$x(a) = x_0 \quad y(a) = y_0 \quad x(b) = x_1 \quad y(b) = y_1 \tag{4.34}$$

where $a = 0$ and $b = 1$. These boundary conditions correspond to the start $\mathbf{x}_0$ and target $\mathbf{x}_{\text{goal}}$ waypoint of any glider path planning problem.

The Jacobian of the ODE system is

$$\mathbf{J} = \begin{pmatrix} u_x & u_y & -U_g \sin\psi & \\ v_x & -u_x & U_g \cos\psi & \mathbf{0}_{3\times 1} \\ \frac{\partial\dot{\psi}}{\partial x} & \frac{\partial\dot{\psi}}{\partial y} & \frac{\partial\dot{\psi}}{\partial\psi} & \\ & & \mathbf{0}_{1\times 4} & \end{pmatrix} \tau + (\mathbf{A}_{4\times 1})_{1\times 4} \tag{4.35}$$

where

$$\frac{\partial\dot{\psi}}{\partial x} = u_{xx}\sin 2\psi + v_{xx}\sin^2\psi - u_{xy}\cos^2\psi \tag{4.36}$$

$$\frac{\partial\dot{\psi}}{\partial y} = u_{xy}\sin 2\psi - u_{xx}\sin^2\psi - u_{yy}\cos^2\psi \tag{4.37}$$

$$\frac{\partial\dot{\psi}}{\partial\psi} = (u_y + v_x)\sin 2\psi + 2u_x\cos 2\psi \tag{4.38}$$

and $being v_y = -u_x$.

Finally, the Jacobian of the boundary conditions is

$$\mathbf{G}(a) = \begin{pmatrix} \mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 2} \\ & \mathbf{0}_{2\times 4} \end{pmatrix} \tag{4.39}$$

$$\mathbf{G}(b) = \begin{pmatrix} & \mathbf{0}_{2\times 4} \\ \mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 2} \end{pmatrix} \tag{4.40}$$

4.7.2 Hamilton's Equations

It solves the adimensional equations

$$(x(t),\ y(t),\ p_x(t),\ p_y(t),\ \tau(t))^\top \tag{4.41}$$

subject to the ODE system

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{p}_x \\ \dot{p}_y \\ \dot{\tau} \end{pmatrix} = \tau \mathbf{A}_{5\times 1} \tag{4.42}$$

where

$$\mathbf{A}_{5\times 1} = \begin{pmatrix} u + U_g \cos\psi \\ v + U_g \sin\psi \\ -p_x u_x - p_y v_x \\ p_y u_x - p_x u_y \\ 0 \end{pmatrix} \tag{4.43}$$

and

$$\psi = \arctan\left(\frac{-p_y}{-p_x}\right) = \pi + \arctan\left(\frac{p_y}{p_x}\right) \tag{4.44}$$

using `atan2` to compute $\arctan$. Alternatively, we might use

$$\sin\psi = \alpha p_y \tag{4.45}$$

$$\cos\psi = \alpha p_x \tag{4.46}$$

where

$$\alpha = \frac{1}{\sqrt{p_x^2 + p_y^2}} \tag{4.47}$$

and with boundary conditions

$$x(a) = x_0 \quad y(a) = y_0 \quad x(b) = x_1 \quad y(b) = y_1 \tag{4.48}$$

and

$$p_x(a)(u + U_g \cos\psi) + p_y(a)(v + U_g \sin\psi) = 1 \tag{4.49}$$

where $a = 0$ and $b = 1$.

The Jacobian of the ODE system is

$$\mathbf{J} = \begin{pmatrix} \mathbf{L}_{2\times 2} & \mathbf{M}_{2\times 2} & \mathbf{0}_{4\times 1} \\ \mathbf{N}_{2\times 2} & -\mathbf{L}_{2\times 2}^\top & \\ & \mathbf{0}_{1\times 5} & \end{pmatrix} \tau + (\mathbf{A}_{5\times 1})_{1\times 5} \tag{4.50}$$

where

$$\mathbf{L}_{2\times 2} = \begin{pmatrix} u_x & u_y \\ v_x & -u_x \end{pmatrix} \tag{4.51}$$

$$\mathbf{M}_{2\times 2} = \alpha^2 U_g \begin{pmatrix} -p_x \sin\psi & p_y \sin\psi \\ p_x \cos\psi & -p_y \cos\psi \end{pmatrix} \tag{4.52}$$

$$\mathbf{N}_{2\times 2} = \begin{pmatrix} -p_x u_{xx} - p_y v_{xx} & -p_x u_{xy} + p_y u_{xx} \\ p_y u_{xx} - p_x u_{xy} & p_y u_{xy} - p_x u_{yy} \end{pmatrix} \tag{4.53}$$

Finally, the Jacobian of the boundary conditions is

$$\mathbf{G}(a) = \begin{pmatrix} & \mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 3} & \\ & \mathbf{0}_{2\times 5} & & \\ \frac{\partial \dot{g}_5}{\partial x(a)} & \frac{\partial \dot{g}_5}{\partial y(a)} & \frac{\partial \dot{g}_5}{\partial p_x(a)} & \frac{\partial \dot{g}_5}{\partial p_y(a)} & 0 \end{pmatrix} \tag{4.54}$$

$$\mathbf{G}(b) = \begin{pmatrix} & \mathbf{0}_{2\times 5} & \\ \mathbf{I}_{2\times 2} & & \mathbf{0}_{2\times 3} \\ & \mathbf{0}_{1\times 5} & \end{pmatrix} \tag{4.55}$$

where g_5 refers to (4.49) and

$$\frac{\partial \dot{g}_5}{\partial x(a)} = p_x(a) u_x + p_y(a) v_x \tag{4.56}$$

$$\frac{\partial \dot{g}_5}{\partial y(a)} = p_x(a) u_y - p_y(a) u_x \tag{4.57}$$

$$\frac{\partial \dot{g}_5}{\partial p_x(a)} = u + U_g \cos\psi + \alpha^2 p_x(a)(p_y(a) U_g \cos\psi - p_x(a)\sin\psi) \tag{4.58}$$

$$\frac{\partial \dot{g}_5}{\partial p_y(a)} = v + U_g \sin\psi + \alpha^2 p_y(a)(p_x(a) U_g \sin\psi - p_y(a)\cos\psi) \tag{4.59}$$

4.7.3 BVP solvers

Both Zermelo's and Hamilton's equations can be solved numerically with the following BVP methods or solvers:

Collocation: Requires reformulation in order to have a fixed time interval $t \in [a, b]$, so a trick is used, by adding an additional ODE function τ.

Shooting: This solver is direct, i.e. it requires no reformulation of the ODE system.

We have accomplished a preliminary study of these methods with ocean currents generated with anaylitic streamfunction equations. The main drawback of the BVP solvers is that it fails with complex environments. Only with weak ocean currents we are able to find a solution. For this reason, we cannot include this approach in the experiments of this work, since we usually have test cases with strong ocean currents and obstacles. Anyway, for some synthetic test cases used for the path following problem (see Section 5.7), we have observed that for this particular problem and test, the Collocation solver performs better than the Shooting one. Also, Zermelo's optimal navigation formula works better than Hamilton's, since the latter fails before when the glider nominal speed U_g is reduced. In any case, optimal control (BVP) is very sensitive to the start/target waypoint location, U_g, the ocean currents function or model, and the BVP solver parameters. Thus, it is difficult to find a solution in many cases. Therefore, this family of methods are only useful to obtain the optimal solution in those cases were $SR < 1$, i.e. with weak ocean currents, and also without obstacles.

4.8 Glider Path Planning on Structures Extracted from the Environment

Apart from the glider path planning algorithms discussed thus far, it is also possible to simplify the complexity and dimensionality of the problem by means of extracting features or structures from the environment, or by finding regions which are unstable, dangerous or for some reason it is better to avoid navigating on them. Then, we can run the path planning algorithms only on the regions outside such structures —or on the contrary, only on those structures if they are the interesting regions.

In the next sections we discuss some intelligent ways to segment the ocean in different regions, so we can use them for feature-based path planning. One way to segment the ocean is by the stability analysis of the ocean currents vector field. Also, we can track Lagrangian structures like eddies, by detecting and characterizing them. Here we will present an algorithm that detects, characterize and track mesoscale eddies, which is an illustrative example and also an important application in Ocean Research, modeling and data assimilation.

4.8.1 Stability Analysis

The stability analysis is concerned with the temporal stability of every region on the ocean domain. The ocean currents define a dynamic system, in which we study the transport of particles within it. In fact, the stability analysis measures how these particles separate after a given interval time T. When applied to fluids, these particles are known as tracers. Clearly, the ocean currents define the vector field $\mathbf{v}(\mathbf{x}, t)$ of the ocean flow at any given point $\mathbf{x}$ and time t. This is exactly the output of the ROMs, which computed the ocean model equations in order to obtain such time-dependent velocity field $\mathbf{v}(\mathbf{x}, t)$. In time-dependent flows, there exist emerging patterns that influence the transport of the tracers, according with the work of Shadden et al. (2005), which is followed here.

The emerging patterns that are detectable in $\mathbf{v}(\mathbf{x}, t)$ are often referred as coherent structures. When they are studied in terms of fluid trajectories, they are named Lagrangian Coherent Structures (LCS). Formally, they are local extrema of the Finite-Time Lyapunov Exponent (FTLE) field, which quantifies the asymptotic behavior of infinitesimally close particles in a dynamic system (Shadden et al., 2005). Hence, it provides the stability analysis of the ocean flow on the ocean currents vector/velocity field. Then, LCSs are simply ridges in the FTLE field. Shadden et al. (2005) define such ridges as special gradient lines of the FTLE field that are transverse to the direction of minimum curvature. Here we will reproduce the mathematical background of FTLEs and will show how to compute them in order to extract LCSs, which being unstable regions should be avoided by glider path planning algorithms.

Paraphrasing Shadden et al. (2005), the FTLE is a finite time average of the maximum expansion rate for a pair of particles advected in the flow. Using the notation of their paper, when a point $\mathbf{x}$ is advected, it moves to $\phi_{t_0}^{t_0+T}(\mathbf{x})$ after the time T, starting a t_0. This advection is simply the unconstrained motion model of a drifter, described in Section B.1.1. Although here we consider bi-dimensional FTLEs for surface ocean currents, it is easy to extend this stability analysis to 3 dimensions (Lekien et al., 2007).

Consider the temporal evolution of a perturbed point $\mathbf{y} = \mathbf{x} + \delta\mathbf{x}(0)$ where $\delta\mathbf{x}(0)$ is an infenitesimal perturbation at $t = t_0$, i.e. we have a point nearby $\mathbf{x}$. After the advection for the time T, this pertubation becomes

$$\delta\mathbf{x}(T) = \phi_{t_0}^{t_0+T}(\mathbf{y}) - \phi_{t_0}^{t_0+T}(\mathbf{x}) = \frac{\partial\phi_{t_0}^{t_0+T}(\mathbf{x})}{\partial\mathbf{x}}\delta\mathbf{x}(0) + \mathcal{O}\left(\|\delta\mathbf{x}(0)\|^2\right) \tag{4.60}$$

using *Landau* notation (Marsden and Hoffman, 1993). The growth of linearized perturbations is hence

$$\|\delta\mathbf{x}(T)\| = \sqrt{\langle\delta\mathbf{x}(0), \Delta\delta\mathbf{x}(0)\rangle} \tag{4.61}$$

where

$$\Delta = \frac{\partial \phi_{t_0}^{t_0+T}(\mathbf{x})}{\partial \mathbf{x}}^{*} \frac{\partial \phi_{t_0}^{t_0+T}(\mathbf{x})}{\partial \mathbf{x}} \tag{4.62}$$

being M^* denotes the adjoint (transpose) of M.

The maximum stretching occurs when $\delta\mathbf{x}(0)$ is aligned with the eigenvector associated with the maximum eigenvalue of Δ, i.e. if $\lambda_{\max}(\Delta)$ is the maximum eigenvalue of Δ, then

$$\max_{\delta\mathbf{x}(0)} \|\delta\mathbf{x}(T)\| = \sqrt{\lambda_{\max}(\Delta)}\, \|\delta\bar{\mathbf{x}}(0)\| \tag{4.63}$$

where $\delta\bar{\mathbf{x}}(0)$ is aligned with the eigenvector associated with $\lambda_{\max}(\Delta)$. We can recast this equation to make explicit the Lyapunov exponent,

$$\max_{\delta\mathbf{x}(0)} \|\delta\mathbf{x}(T)\| = e^{\sigma_{t_0}^{T}(\mathbf{x})|T|}\, \|\delta\bar{\mathbf{x}}(0)\| \tag{4.64}$$

where

$$\sigma_{t_0}^{T}(\mathbf{x}) = \frac{1}{|T|} \ln \sqrt{\lambda_{\max}(\Delta)} \tag{4.65}$$

which represents the largest FTLE for a finite integration time T, associated with the point $\mathbf{x}$ at time t_0. Interestingly, we can use both positive and negative integration times T. We can use backward-time integration to locate *attracting* LCSs, and forward-time integration for *repelling* LCSs.

Once we have the FTLE field $\sigma_{t_0}^{T}(\mathbf{x})$, LCSs are defined as ridges on the field. A possible definition uses the second derivative ridge (Shadden et al., 2005), which uses the Hessian of the FTLE field,

$$H = \frac{\partial^2 \sigma_{t_0}^{T}(\mathbf{x})}{\partial \mathbf{x}^2} \tag{4.66}$$

Algorithm 4.5 FTLE field computation pseudo-code.

Require: Start location $\mathbf{x}_0$ of a set of particles $\mathcal{X}_0$ at time t_0, to be advected for the integration time T.
Ensure: FTLE field $\sigma_{t_0}^{T}(\mathbf{x})$.
Algorithm: FTLE($\mathcal{X}_0$, t_0, T) **return** $\sigma_{t_0}^{T}(\mathbf{x})$
1: **for all** $\mathbf{x}_0 \in \mathcal{X}_0$ **do**
2: $\quad \mathbf{x}_T = \phi_{t_0}^{t_0+T}(\mathbf{x}_0) \quad : \quad \mathbf{x}_T \in \mathcal{X}_T$ ▷ advect particle $\mathbf{x}_0$ for time T
3: **end for**
4: $J = \nabla \mathcal{X}_T$ ▷ gradient of flow map $\mathcal{X}_T$
5: $\Delta = J^* J$
6: $\sigma_{t_0}^{T}(\mathbf{x}) = \frac{1}{|T|} \ln \sqrt{\lambda_{\max}(\Delta)}$ ▷ apply (4.65)
7: **return** $\sigma_{t_0}^{T}(\mathbf{x})$

The computation of the FTLE field for a list of particles or tracers $\mathbf{x}_0 \in \mathcal{X}_0$ at time t_0 is summarized in Algorithm 4.5. All the particles are advected with $\phi_{t_0}^{t_0+T}(\mathbf{x})$ for an integration time T, i.e. their trajectories on the ocean currents velocity field are integrated with the unconstrained motion model of a drifter described in Section B.1.1. As a result, we obtain the location of each particle $\mathbf{x}_T \in \mathcal{X}_T$ after the time T. Then, we compute its gradient, which yields the Jacobian J that is used to compute Δ. Finally, the FTLE is a straightforward evaluation of (4.65). In Section 3.5 we have already seen the output of the FTLE field of a dynamic double eddy, which can be generated with the analytical ocean model of Section 5.1.2.4. Here we illustrate it again for a Stuart vortex (see Section 5.1.2.5), for an integration time $T = 10$. This produces a forward-time FTLE field, shown in Figure 4.17. Recall that the forward-time FTLE field shows the *repelling*

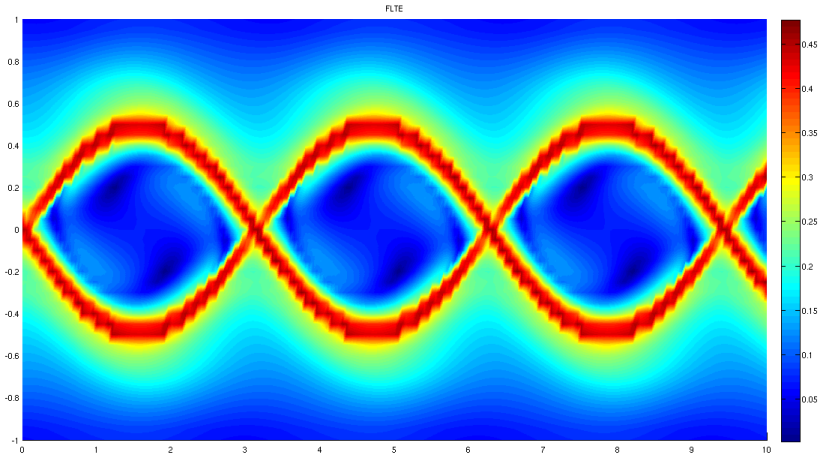


Figure 4.17: Stability analysis by computing the FTLE field of a Stuart vortex. The unstable regions correspond to LCSs, which appear in red tones at the boundaries of the eddies/vortexes.

LCSs of the ocean currents velocity field, so we observe that the unstable regions localize at the boundary of the eddies/vortexes.

In terms of glider path planning, the LCSs define regions which are inherently unstable. Due to the uncertainty of the ocean model and the vehicle localization, if the vehicle crosses an unstable region, a small error on the estimation of its position may yield a resulting trajectory very different from the expected one. Therefore, we can incorporate LCSs in the path planner as obstacles, in order to avoid such regions. The great interest of the LCSs is that they can be computed for both static and dynamic ocean currents, revealing unstable regions that are not evident at first sight.

4.8.2 Feature-based Path Planning

Instead of using the whole vector field defined by the ocean currents, we can find optimal paths on a low dimensional search space populated by features. Such features can be extracted from the ocean currents, with some algorithm. For instance, we may want track ocean evolving structures, like the LCSs discussed previously, or the aforementioned mesoscale eddies. Here, for feature-based path planning we simply referred to include information from such structures within the path planning algorithm in order to reduce the search space. In the next section we will see an example of this sort of feature-based path planning. We track mesoscale eddies, a particular kind of evolving structures very common on the ocean.

4.8.2.1 Detection, Characterization and Tracking of Mesoscale Eddies

Here we propose a path planning algorithm meant to track eddies or similar evolving structures. The approach comprises three phases: detection and identification of eddy model parameters, definition of objective function and path generation. The algorithm uses a discretized version of the eddy model proposed by Jimenez et al. (2008) for the Canary Islands eddies system, which originates from a combination of wind and topographic forcing. The eddy model divides the structure volume into several sectors rotating at different velocities. Each sector is defined by its min/max limits in depth, radius and angle dimensions (see Figure 4.18).

As a first phase, the eddy parameters are identified by combining the ocean currents, altimetry and temperature maps in a semi-automatic process. The centroid and the boundary of the eddy are annotated

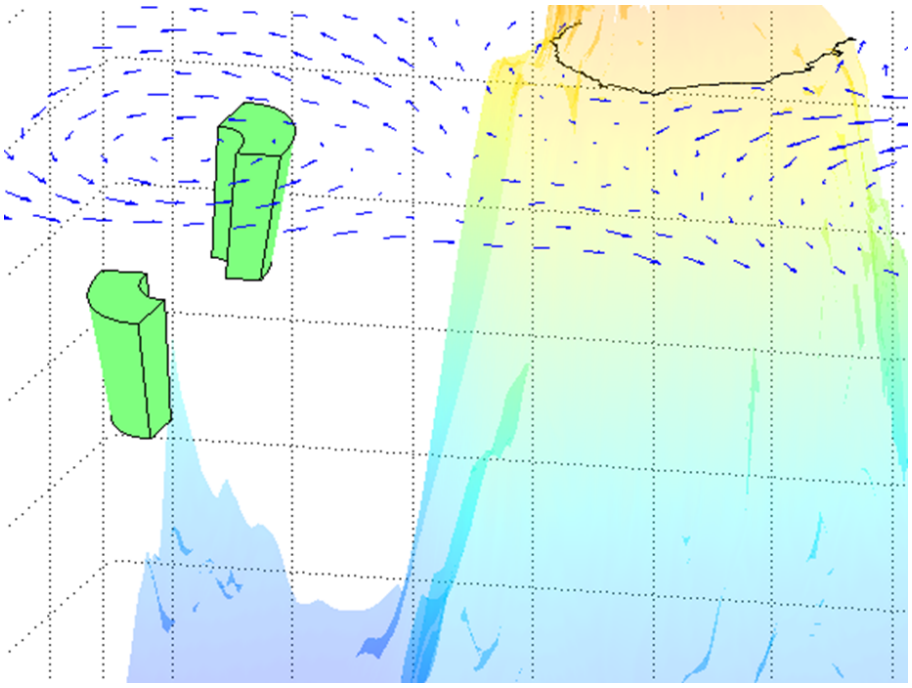


Figure 4.18: Mesoscale eddy model used for the path planning algorithm developed to detect, characterize and track this kind of evolving ocean structures. The figure shows the volume segmentation of an eddy.

manually on the SSHA maps and then refined using the ocean currents and temperature maps. Additionally, the eddy annotation is assisted by an automatic detection run at pre-processing step. It highlights potential eddy candidates, using the vorticity of the vector field or an eddy detection algorithm (Nencioli et al., 2010).

Once the eddy has been detected and modeled, the objective function is expressed in terms of the eddy sectors (see Figure 4.18) that should be sampled and the desired mission duration. This way it is possible to define different sampling strategies: track the centroid of the eddy, navigate at the boundary, maximize the sampled volume, flight at a given depth, and so on. Finally, in the last phase, a Genetic Algorithm optimization is run in order to find the trajectory that best meets the specifications. At this step, we use the three-dimensional unconstrained point motion model of Section 2.2.2.1.

4.9 Glider Path Planner Library and Tools

Apart from the algorithmic contributions of this thesis to the problem of glider path planning, as a result of this work we have developed as set of tools and a library. We have termed this library and tools *Pinzón*, after the Spanish navigator and explorer *Vicente Yáñez Pinzón*, who came back from America and faced strong opposing ocean currents near the North coast of Spain, as we faced at the end of the RU27 trans-Atlantic mission, when we assisted on the piloting with our path planning tools.

The library developed comprises all the path planning algorithms presented in this chapter, along with the point and force balance motion models. We can also use *Pinzón* to simulate/integrate the glider trajectory, given the real surfacing points $\mathbf{x}_i$ of a glider at the ocean. Additionally, we can simulate other vehicles like drifters and floats, under both motion models (see Appendix B).

The software allows to run and compare the DtG approach, A*, (ABS) CTS-A*, RRT, (Iterative) Optimization —with or without the intelligent initialization phase for obstacle avoidance. The Figure 4.19

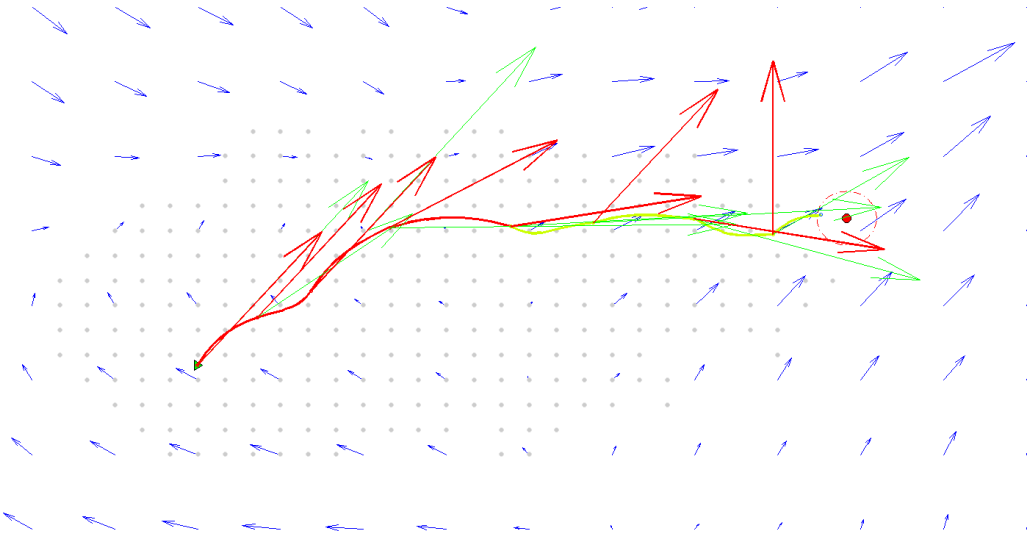


Figure 4.19: Output that shows the path found with the CTS-A* algorithm. The figure draws the glider trajectory from the start ▶ to the target ● node/waypoint. The blue arrows ⟶ on the background show the ocean currents vector field at the starting time, while **thick red** ⟶ and green ⟶ arrows are the bearing ψ_g and heading ψ_e angles, respectively.

depicts a typical path planning output. It shows the path found, which consists on a sequence of waypoint locations $\mathbf{x}_i$, and the bearing angles ψ_{g_i} that must be commanded to the glider. The bearing angles, along with the resulting heading angles ψ_{e_i}, are shown as arrows emanating from each surfacing point located at $\mathbf{x}_i$. The glider trajectory due to the drift is drawn to illustrate the important effect of time-varying ocean currents. Bear in mind that the background ocean currents vector field of Figure 4.19 corresponds to the first time snapshot, because it is not possible to show it over time in a single frame. Thus, *Pinzón* produces movie clips to sort this issue out. Finally, for graph-based algorithm the explored nodes —or even discovered— by the algorithm are marked, for debugging and evaluation purposes as in the figure for the CTS-A* algorithm. This gives some insight on the searching process, that might be exhaustive or not depending on the problem on hand. Equivalent debugging capabilities are offered for the rest of the algorithms supported.

The tools in *Pinzón* also allow to visualize the output of the ROMs. We can download the nowcast and forecast maps for a given date and analyze the different parameters computed by the ocean model. This include scalar water properties like the Sea Surface Temperature (SST), the Sea Surface Height Anomaly (SSHA), which are both defined for the surface, and three-dimensional data like the temperature and salinity —from which we can compute the density at every depth. And also include the ocean currents vector field, typically with the u and v components for the surface or at depth; see Appendix A for more information on Ocean Models and their products. Furthermore, our *Pinzón* path planner is capable of producing a movie to illustrate the temporal dynamics of the ocean currents vector field and how the glider trajectory is affected, for the path found. *Pinzón* also generates Google™ Earth animations in Keyhole Markup Language (KML) format, as the one in Figure 4.20 that allows to analyze the path found overlaid on the glider mission data, or even the weather, on the map where the mission takes place.

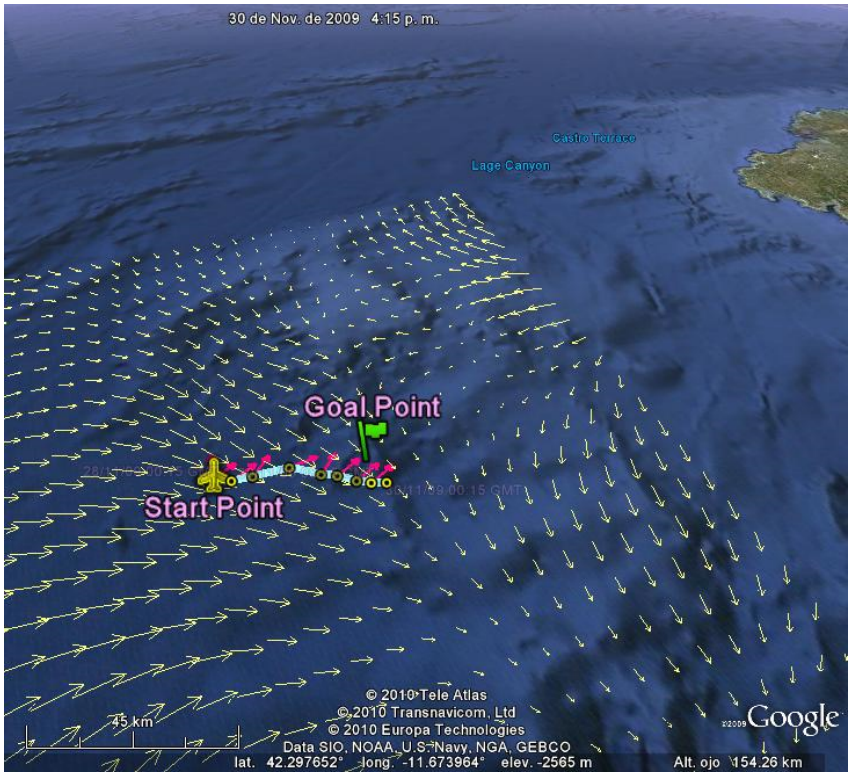


Figure 4.20: Example of path planned by *Pinzón* in the ESEOAT region of the ROM of the ESEOO project using the RU27 surfacing points, to reach a goal specified as crossing a meridian. The output is a KML animation that can be run in Google™ Earth; in the figure we see a snapshot.

Chapter 5

Experimental Results

This chapter shows the experimental results obtained for the glider path planning algorithms developed in this thesis. They are evaluated and compared with each of the problems or applications explained in Chapter 3. Here, we start by describing the default experimental setup, which is used for most of the algorithms and problems; any minor variation with respect to this base configuration will be explicitly mentioned though, along with any particular setup of a given experiment. The experimental setup gives an overview of the ocean model products and the glider motion model used for path planning. Although we mostly use real ROM products in our experiments, we have also developed a number of analytic ocean model equations. They constitute a complementary contribution of this work, being useful to evaluate the algorithms in environments that exhibit a particular structure or dynamics —e.g. constant ocean currents, sinks, sources, eddies, etc.

Before we show and discuss the results, the evaluation metrics are described in detail. In brief, they comprise the computational time and path optimality assessment. The first should be in agreement with the algorithm complexity, as described in Chapter 4. The path optimality depends on the particular problem though. If a clear and simple objective function is available, such function itself gives the path optimality directly, as in the minimal time and minimal distance to the target problems. The path following problem is evaluated using a Pareto curve for the path following and minimal time path objectives. In the case of multi-glider path planning problems, we evaluate the proximity of the gliders flying in formation, or the time required for their recovery, which are the two problems covered here.

For the atomic problems minimal time path and minimal distance to the target, all the path planning algorithms developed in this thesis are evaluated. For the rest of the problems —which are build as an aggregation of the atomic ones— we only want to check and assess the applicability of the most promising technique for such problem. This is the case of multi-glider coordination using constraints, path following —to some extent—, and tracking evolving ocean structures.

For the sake of clarity, we present the experimental results organized by problem or application. Note that some techniques are not applicable for some problems, so this seems a reasonable way to compare them. Furthermore, this way we can evaluate each method with its best configuration for a given problem, and answer one of the questions addressed in this thesis, i.e. which technique performs better for a certain problem/application. Finally, along with that comparison, we discuss several field glider missions in which our algorithms have been used aiding piloting and path planning.

5.1 Experimental Setup

Before we show the experimental results, we are going to indicate here the default and basic configuration of the experiments. This comprises the glider motion model, some assumptions or simplifications on the glider navigation behavior, and the characteristics of the ocean currents provided by the ocean model employed in

the test cases. Unless stated explicitly, the test cases later presented for each glider path planning problem will adopt the experimental setup described here.

In Section 2.2 we explored the three main options to model the glider motion. First, the point model, which considers the glider as a 0-mass point moving with a certain nominal speed u_g in a given bearing ψ_g. Secondly, the force balance model, which compounds the vehicle buoyancy, gravity, drag and lift forces to obtain the glider speed, which also depends on the angle of attack α of the wings with respect to the hull, and the pitch θ and glide γ angles. And third, dynamic motion models that solve the differential equations of the vehicle dynamics. In our experiments we use the point motion model described in Section 2.2.2, since for the accuracy obtained with the other models is not really required given the large errors and the uncertainty of some parameters of the glider motion model and the ocean model itself. Furthermore, for the relatively large missions performed, it makes no sense to use precise motion models in most of the cases.

Moreover, given the large dimensionality of the problems, it is common to consider certain assumptions. A common simplification consists in using only surface ocean currents to find the optimal path. However, the glider navigates underwater doing yo-yo profiles, and affected by the ocean currents at the depth it is flying. We can compute the depth average ocean currents though, if we want to work in 2 dimensions but still reflect the effect of underwater ocean currents. In fact, it is straightforward to use such depth average ocean currants as in Thompson et al. (2010), where the total ocean current force experienced by the glider underwater is taken to be a constant $C_{xy\tau}$ within each grid cell. Anyway, it will be better to consider the ocean currents in 3 dimensions, in order to obtain the most accurate estimation of the glider trajectory underwater, and hence the next surfacing point location $\mathbf{x}_{i+1}$. We have detected that the decision depends on the algorithm designed to solve the problem.

So far we have mentioned that the A* algorithm uses the constrained motion model of Section 2.2.2.2, while others like the CTS-A*, RRT or the Optimization method integrate the unconstrained motion model of Section 2.2.2.1. Recall that in the constrained motion model we do not integrate the glider trajectory while it navigates underwater. We simply compute the bearing ψ_g to follow a given heading ψ_e, using the glider nominal surge speed U_g and the ocean currents vector $\mathbf{U}_c$. Since the trajectory is not integrated, we do not really use the depth ocean currents. Therefore, for the constrained point motion model we use the two-dimensional surface ocean currents in our experiments; we use depth average ocean currents if the ROM provides the ocean currents at different depths, but this depends on the temporal resolution, as we will later see in Section 5.1.1.

On the other hand, for the unconstrained point motion model we integrate the glider trajectory underwater, which is affected by the ocean currents at each depth computed by the integration method. Therefore, if the ocean currents below the surface are provided by the ROM, we use three-dimensional ocean currents with the algorithms that use the unconstrained motion model, i.e. the CTS-A*, RRT and (Iterative) Optimization methods —and the DtG approach as well.

Notice that we do not simulate the drift during the surfacing interval (see Section 2.1.2). We have chosen to isolate its effect, assuming it distributes as a constant added cost for all the algorithms evaluated. Only when we are assisting in the glider piloting we will include the surfacing intervals of $\approx 15\mathrm{min}$ in the path planning algorithm —stating it explicitly—, so the path $\mathcal{P}$ found can actually be followed on the field. Note that to simulate the drift produced by the surface ocean currents for this slot of time, we use the drifter motion model, which is simply runs the two-dimensional unconstrained point motion model, or its three-dimensional version with no pitch $\theta = 0^\circ$ (see Section B.1.1).

In our experiments we generally set the glider pitch to $\theta = 25^\circ$, which is the typical value set for the SLOCUM glider. However, we could also set the optimal value $\theta = 35^\circ$, which give the maximum glider nominal speed $U_{g_{\max}}$ according with the equations of the force balance motion model (see Section 2.2.3). Anyway, for the same group of test cases we use the same pitch θ. Moreover, we do not include θ in the path planning optimization, since it is rarely changed during real glider missions. Similarly, we neither modify the depth limits $[z_{\min}, z_{\max}]$, which are set to $[20, 200]\mathrm{m}$ in most of the test cases run here. The number of yo-yo profiles done for the surfacing time t_s —while the glider navigates underwater— is not set and neither

change in the path planner. Its value is determined from the depth limits and t_s, as shown in Section 2.2.2.1.

Although we will indicate it explicitly, we consider two main values for the glider nominal surge speed U_g in our experiments. We set $U_g = 0.2\mathrm{m/s}$ as a slow glider speed, which is equivalent to navigate with strong ocean currents, i.e. $SR > 1$; and $U_g = 0.4\mathrm{m/s}$ as a fast glider speed, which is equivalent to navigate with weak ocean currents. This constitutes a sub-division of the test cases in the experiments as well. We also categorizes them as offshore or coastal, in order to separate obstacle free environments from the ones populated by many of them, respectively. And similarly, we distinguishes between long-term and short-term missions, which are associated with the use of static and dynamic ocean currents, respectively. Also, the former is generally the type of mission for the minimal time path problem, while the latter type of mission if more common for the minimal distance to the target problem. Unless stated explicitly, these categories obey this criteria.

5.1.1 Regional Ocean Model Scenarios

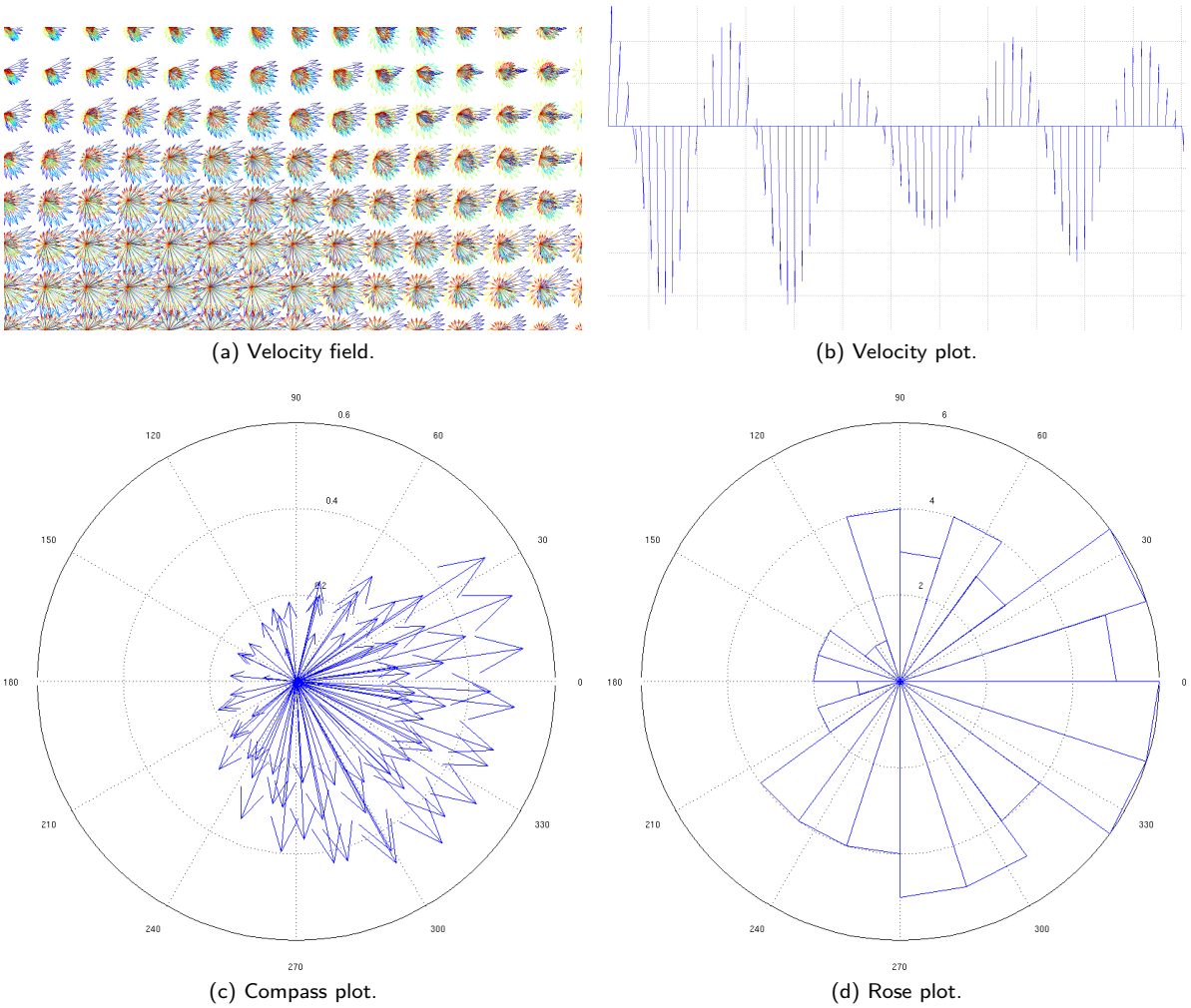
In the present work, we have fed our path planning algorithms with the ocean currents computed by Regional Ocean Models (ROMs). Although we have also used non regional Ocean Models, this ones have less resolution than ROMs, both in space and time. Therefore, they cannot capture many phenomena that take place in the ocean. Indeed, for our firsts experiments during the last stings of the RU27 trans-Atlantic mission, when the glider traveled the Spanish EEZ, we had to use a ROM that modeled that region. We used the ROM developed by Puertos del Estado (Ministerio de Fomento, Gobierno de España) as part of the project ESEOO, that covers the Spanish Atlantic shore (Sotillo et al., 2007, 2008). The model is run separately for three domains in the Spanish territory. First, the ESEOAT covers the North Atlantic area close to the west coast of the Iberian Peninsula. Secondly, the ESEOMED models the Mediterranean sea nearby Spain. And third, the ESEOCAN region comprises the Canary Islands.

In our experiments we have used extensively the ESEOAT model, and for the obstacle avoidance test cases we also use ESEOCAN, since the Canary Islands take the role of land obstacles in the ocean. This ROM provides temporal series hourly with 3-day predictions —or forecast maps— at 0.05° spatial resolution for latitude and longitude. The ROM output is validated with samples taken *in situ* by oceanographic sensor networks. Thus, the ROM is meant to provide accurate data, capturing the high temporal dynamics of the area. It provides two types of data in regards of their dimensionality and temporal resolution. First, bi-dimensional surface ocean currents for every hour. We consider them dynamic ocean currents in the experiments. Secondly, three-dimensional ocean currents for every day; more precisely, the model provides the daily means. For long-term missions we usually need more days than the 3-day forecast maps generated by the ROM. We use a single, representative map of the seasonal period when we are running the experiment. We use three-dimensional ocean currents, which represent the average ocean currents in such period of time.

Although the ROM provides several data, it is possible to plan paths using ocean currents only, because they are not geostrophic ocean currents simply, i.e. they incorporate the effect of other phenomena, like tides. Anyway, additional physical parameters might be incorporated easily, as the density —obtained from the depth, temperature and salinity scalar maps— if we were to use the force balance motion model, the Sea Surface Height Anomaly (SSHA) —tightly related with the ocean currents vector field— or the Chlorophyll concentration (CHL) to avoid dangerous areas in regards of the bio-fouling and its impact on the vehicle autonomy.

Apart from the ESEOAT and ESEOCAN regions, for the next experimental results we also use the output of the IBI ROM, which also covers the North Atlantic and the Canary Islands. Nevertheless, we have also run our algorithms for several path planning problems on different regions in the oceans. We have used global, mesoscale ocean models of less resolution, like NCOM, NLOM, and HYCOM, to mention some. All these ocean models are enumerated and described briefly in Appendix A.

In Figure 5.1 several plots illustrate the high temporal variability of ocean currents in the area covered by the ESEOAT ROM, which was crossed by the RU27 glider in its trans-Atlantic mission. Paths planned



(a) Velocity field.

(b) Velocity plot.

(c) Compass plot.

(d) Rose plot.

Figure 5.1: Ocean currents temporal dynamics in the ESEOAT region for a whole day, showing (a) the ocean currents field —of a patch of $\approx 6^\circ \times 12^\circ$ (latitude, longitude) at the ESEOAT region (North Atlantic)— and (b)-(d) the ocean currents speed and direction at a particular location $\mathbf{x} = (42^\circ\text{N}, -13^\circ\text{E})$, using hourly data for the 21-Nov-2009 —the intensity of this ocean currents is given in (c). The oscillations observed in (b) are common in hourly ocean currents, since they incorporate the tides.

without considering the change of ocean currents over time will be less close to reality, not matter if they are more or less expensive. In fact, depending on the temporal variability in the area, path planning over time will be more or less significant. As already said, static ocean currents are assumed to be daily means, instead of hourly series. Notice that for daily means, ocean currents are only interpolated for the last and first hour of two consecutive days when we integrate the unconstrained motion model of Section 2.2.2.1. As Figure 5.2 shows, dynamic ocean currents yield glider trajectories that cannot be reproduced by simply using daily means. The zones covered by the ESEOAT —as in the figure— and ESEOCAN regions are precisely very dynamic, so for short-term missions we always use dynamic ocean currents up to the 3 days of the forecast maps of the ESEOO ROM. In the case of IBI we have up to 5 days, but we generally do not have more than

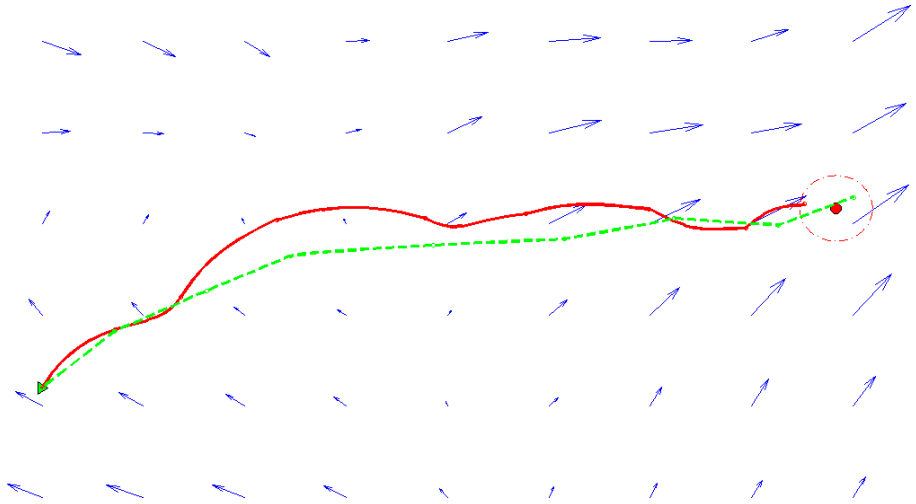


Figure 5.2: Trajectories of the path found by running a glider path planner on dynamic —— and static (daily means) - - - ocean currents, using the ESEOAT region from 21-Nov-2009 onwards. We solve the minimal time path problem from the staring waypoint $\mathbf{x}_0$ ▶ to the target waypoint $\mathbf{x}_{\text{goal}}$ ●, within the radius ◌. The dynamic case shows non-linear trajectories and takes 7 stints —or surfacing points—; meanwhile, the static case takes 1 stint more, because it does not take the ocean currents variability into account.

such number of predictions in most ocean models.

5.1.2 Analytical Ocean Model Equations

Apart from the ocean currents provided by ROMs for a particular region of the oceans, we can also generate them using analytic equations. Here we show some useful examples of these equations and a procedure to generate realistic ocean currents synthetically. The idea is to have ocean currents to reflect the typical structures on the ocean, which are typically eddies. Also, we can model areas with constant ocean currents in a given direction, sinks or sources, circular or elliptical eddies, and spiral vortex structures. All this kind of structures, can be modelled to some extend with the analytic equation shown in the sequel. Also, we can tile the space with multiple instances of them.

We can define the ocean currents vector field directly, but it is common to obtain it from a **streamfunction** $\Psi(x, y, t)$. To some extend, the streamfunction is equivalent to the SSHA, from which we can also apply the next procedure to compute its vector field, i.e. the ocean currents it defines. Therefore, for each location $\mathbf{x} = (x, y)$ and time t, the bi-dimensional vector field is obtained from the streamfunction applying the following equations

$$u(x, y, t) = -\frac{\partial \Psi(x, y, t)}{\partial y} \tag{5.1}$$

$$v(x, y, t) = \frac{\partial \Psi(x, y, t)}{\partial x} \tag{5.2}$$

that hold the incompressibility of the flow, i.e. the null divergence condition

$$\frac{\partial u(x, y, t)}{\partial x} + \frac{\partial v(x, y, t)}{\partial y} = 0 \tag{5.3}$$

According with the Coriolis effect, the relations (5.1) and (5.2) are valid for the North hemisphere solely. For the South hemisphere the sign of each velocity component changes

$$\hat{u}(x,y,t) = -u(x,y,t) = \frac{\partial \Psi(x,y,t)}{\partial y} \tag{5.4}$$

$$\hat{v}(x,y,t) = -v(x,y,t) = -\frac{\partial \Psi(x,y,t)}{\partial x} \tag{5.5}$$

In the next sections we describe the equations of several analytical ocean model equations, along with an example of the resulting vector field. We focus on those analytical models that provide vector fields of the same dimensionality that we usually need for glider path planning. The ideal would be to have the 3 vector components (u, v, w), but the vertical ocean currents w are generally very small and can be neglected. Indeed, most of the ROMs do not compute them, and there are few analytical ocean model equations that solve them. Therefore, the analytical ocean models shown here only compute the (u, v) vector fields. Then, in the ideal case we will have one of these vector at each location in the three-dimensional space (x, y, z) and also for each time t. However, the analytical models for 3 dimensions are very complex, so we only have models for the 2 surface dimensions (x, y). Finally, some of the models described are static, but most of them are dynamic or can be extended easily to be time-varying.

It is worth mentioning that the analytical ocean models operate in the metric space (x, y), rather than with the longitude λ and latitude ϕ coordinates of the spherical reference system of the Earth. This is an important difference with the ocean currents provided by the ROMs. We could use a Sinusoidal Projection to convert from (x, y) to (λ, ϕ) but it is not advisable because it is more efficient to work with the ocean currents represented on an evenly spaced grid. ROMs usually provide the ocean currents for evenly spaced (λ, ϕ), so it makes no sense to project them because we will not obtain an evenly spaced grid as a result.

Anyway, we need to project the locations $\mathbf{x}_i$ to compute the distance between two points in the glider path planner. Hence, we have tried different projections embedded in the distance equations, with different levels of accuracy —from the simple Sinusoidal projection to the exact great circle distance. These distances are computed many times during the path planning search, so they must run fast. Since we operate far from the poles, the flat Earth assumption is valid for the navigation frame during the glider mission. Therefore, we can use the Sinusoidal projection as a good approximation, since it runs faster than any of the other projections used in Section A.2 —note that we only use this for the vehicle location, but not for the ocean currents.

Finally, another difference between the analytical ocean models and the ROMs, is the discretization inherent to the output of the ROMs. The ocean currents are only given at certain locations in the space. On the contrary, the analytical ocean models can be solved for any location. Therefore, we do not need any kind of interpolation with them. But for the ocean currents maps provided by the ROMs we need a fast and accurate interpolation method. In the experiments shown here we use linear interpolation —trilinear for bi-dimensional surface ocean currents with time, and for static three-dimensional ocean currents as well—, although we have also taken the Nearest Neighbor ocean current in the grid with acceptable results. Note that more accurate methods —like the tricubic interpolation of Lekien and Marsden (2005)— are not generally required for glider path planning missions.

5.1.2.1 Power Spectrum with Random Phases

Despite of having several ROMs available, we have also developed a synthetic current field generator to assist the production of controlled test cases —e.g. current strength, eddies size. It is possible to generate a synthetic current field from a streamfunction $\Psi(\mathbf{x})$ —that resembles the Sea Surface Height Anomaly (SSHA)— as shown in Figure 5.3. Such a streamfunction might be randomly generated from a specific power spectrum with random phases, as proposed by Garau et al. (2005). In other words, it will consist on a Gaussian mixture

model given by the following sum of N bivariate Gaussians

$$\Psi(\mathbf{x}) = \sum_{i=1}^{N} \omega_i \cdot \mathcal{N}(\mu_i, \Sigma_i) \quad \mathbf{x} \in \mathbb{R}^2 \tag{5.6}$$

with a random mean μ_i and covariance matrix Σ_i each. The spectrum might be peaked at a particular spatial scale in order to obtain eddies with homogeneous length scale, which is parameterized by means of Σ_i. Finally, the weight ω_i controls the rotation direction of eddies and the current strength. Once we have the streamfunction, which defines a scalar map, we compute the vector field using (5.1) and (5.2).

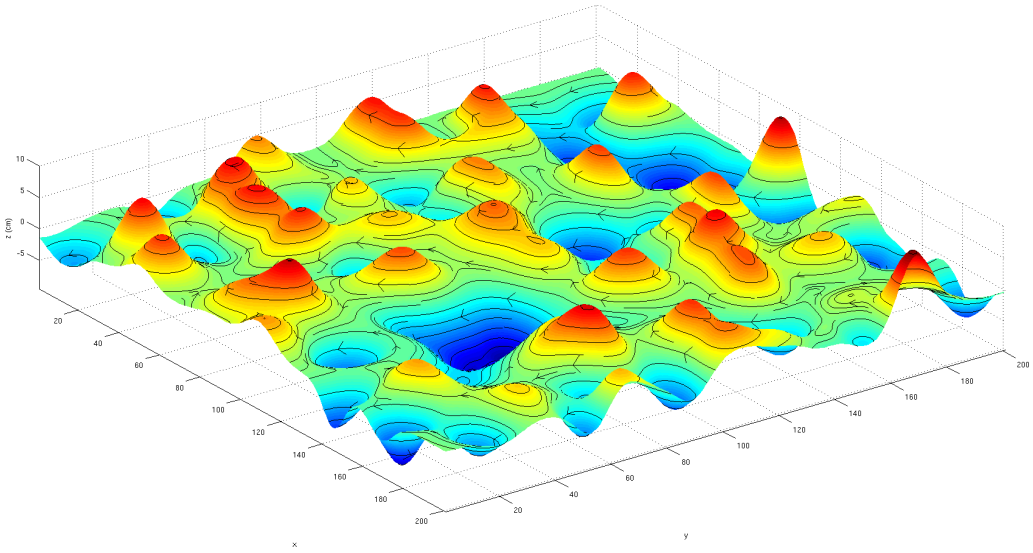


Figure 5.3: Streamfunction $\Psi(\mathbf{x})$, computed as a sum of N bivariate Gaussians that generate random synthetic eddies, and resulting ocean current field $\mathbf{U}_c(\mathbf{x})$ overprinted, which resembles a typical SSHA map. This synthetic streamfunction is defined in the domain $\mathbf{x} = (x, y)\mathrm{km}$.

This approach provides realistic ocean currents vector field realizations, where we can appreciate eddies with different characteristics. Furthermore, we can parameterize some of those characteristics, such as the number of eddies, their size and the current strength. In fact, Figure 5.4 shows some examples of synthetic ocean currents field realizations where the size of eddies has been tuned. This way, we can generate illustrative cases which are particularly interesting for glider navigation, that might be hard to retrieve from ROMs.

This approach does not generate realistic time-varying ocean currents. However, moving the Gaussians that model the streamfunction $\Psi(\mathbf{x})$ spatially over time gives high temporal changes that can be useful to evaluate path planning strategies.

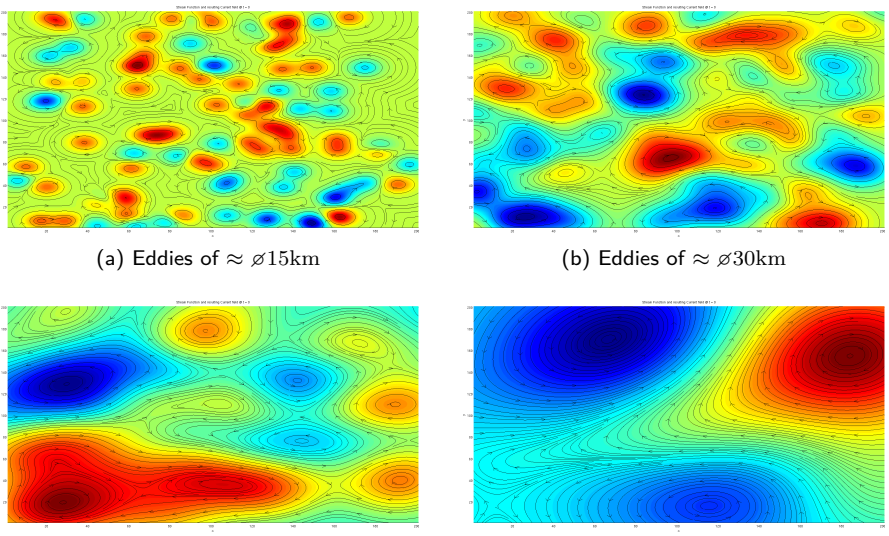
5.1.2.2 Constant vector field

Given a vector (u_c, v_c) representing a constant ocean current speed, we define a constant vector field for each location $\mathbf{x} = (x, y)$ as

$$u(x, y) = u_c \quad \forall \mathbf{x} = (x, y) \in \mathbb{R}^2 \tag{5.7}$$

$$v(x, y) = v_c \quad \forall \mathbf{x} = (x, y) \in \mathbb{R}^2 \tag{5.8}$$

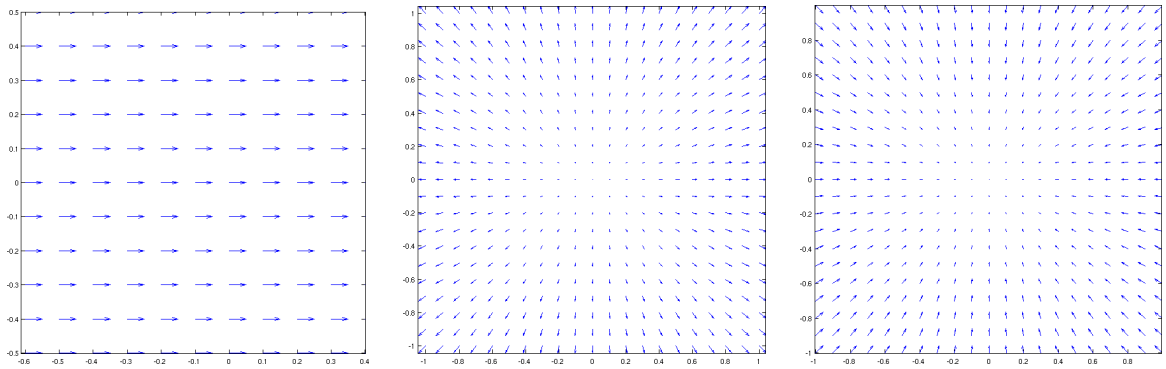
In Figure 5.5 (a) we have a constant vector field generated with $(u_c, v_c) = (0.05, 0)\mathrm{m/s}$ in the region defined by $x \in [-2, 2]$ and $y \in [-2, 2]$; note that we have zoomed in.



(a) Eddies of $\approx \varnothing 15\text{km}$ (b) Eddies of $\approx \varnothing 30\text{km}$

(c) Eddies of $\approx \varnothing 50\text{km}$ (d) Eddies of $\approx \varnothing 100\text{km}$

Figure 5.4: Examples of synthetic ocean current field realizations with eddies of different size. (a)-(d) The figures show the top view of the ocean current field $\mathbf{U}_c(\mathbf{x})$ and the generative streamfunction $\Psi(\mathbf{x})$.



(a) Constant. (b) Source. (c) Sink.

Figure 5.5: Basic analytical ocean currents for (a) constant ocean currents $(u_c, v_c) = (0.05, 0)\text{m/s}$, (b) source and (c) sink centered at $\mathbf{x}_c = (x_c, y_c) = (0, 0)$.

5.1.2.3 Source and Sink

Given the center $\mathbf{x}_c = (x_c, y_c)$ of the desired **source**, the vector field (u, v) for each location $\mathbf{x} = (x, y)$ is defined as

$$u(x, y) = r \cos \alpha \tag{5.9}$$

$$v(x, y) = r \sin \alpha \tag{5.10}$$

where r and α are the distance and angle with respect to $\mathbf{x}_c$, which are computed as

$$r = \sqrt{\Delta x^2 + \Delta y^2} \tag{5.11}$$

$$\alpha = \arctan\left(\frac{\Delta y}{\Delta x}\right) \tag{5.12}$$

being $\Delta x = x - x_c$ and $\Delta y = y - y_c$. Note that the $\arctan$ should be computed with `atan2`.

In order to obtain a **sink**, we can compute it from the vector field of the source defined above, just by changing the sign of both components $u(x, y)$ and $v(x, y)$, so we have

$$u(x, y) = -r\cos\alpha \tag{5.13}$$

$$v(x, y) = -r\sin\alpha \tag{5.14}$$

In Figure 5.5 (b) and (c) we have a source and a sink vector field, respectively. Both are centered in $\mathbf{x}_c = (x_c, y_c) = (0, 0)$ and are defined in the region $(x, y) \in [-2, 2] \times [-2, 2]$; note that we have zoomed in.

5.1.2.4 Double eddy

A dynamic double eddy can be defined with the streamfunction of Shadden et al. (2005) as

$$\Psi(x, y, t) = A\sin\left(\pi f(x, t, \epsilon, \omega)\right)\sin\left(\pi y\right) \tag{5.15}$$

where A is the amplitude of the stream and $f(x, t, \epsilon, \omega)$ is a time-varying function

$$f(x, t, \epsilon, \omega) = x\left(a(t, \epsilon, \omega)x + b(t, \epsilon, \omega)\right) \tag{5.16}$$

being

$$a(t, \epsilon, \omega) = \epsilon\sin\left(\omega t\right) \tag{5.17}$$

$$b(t, \epsilon, \omega) = 1 - 2a(t, \epsilon, \omega) \tag{5.18}$$

where ϵ and ω are two parameters that allow to configure the streamfunction and, consequently, the shape of the double eddy.

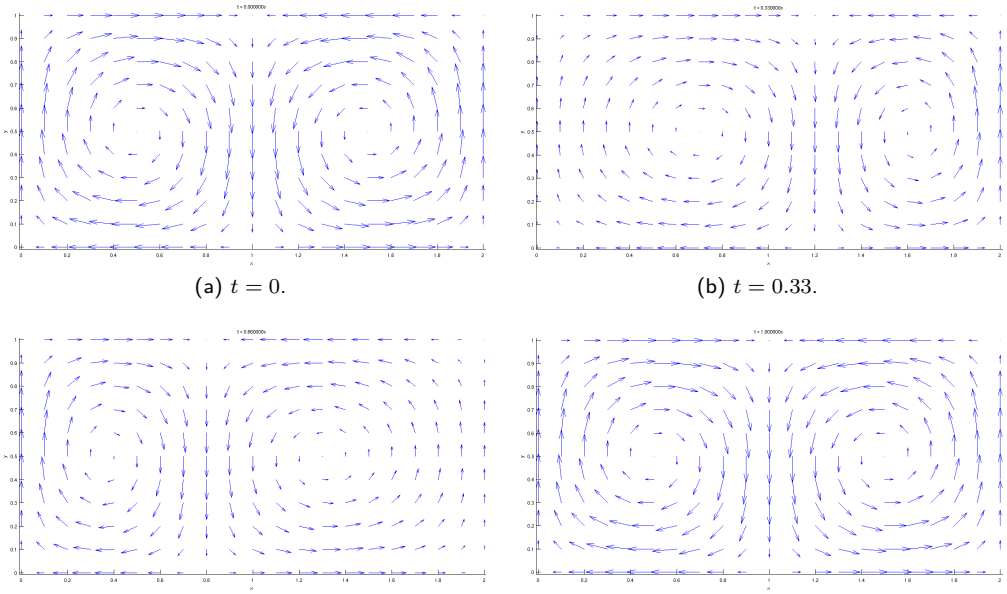
In order to compute the vector field (u, v) from the streamfunction, we need the first partial derivatives of (5.15), i.e. the Jacobian

$$\mathbf{J}_\Psi = \begin{pmatrix} \dfrac{\partial\Psi(x, y, t)}{\partial x} \\ \dfrac{\partial\Psi(x, y, t)}{\partial y} \end{pmatrix} = \pi A \begin{pmatrix} \cos\left(\pi f(x, t, \epsilon, \omega)\right)\sin\left(\pi y\right)\mathbf{J}_f \\ \sin\left(\pi f(x, t, \epsilon, \omega)\right)\cos\left(\pi y\right) \end{pmatrix} \tag{5.19}$$

where $\mathbf{J}_f$ is the Jacobian of (5.16), given by

$$\mathbf{J}_f = \frac{\partial f(x, t, \epsilon, \omega)}{\partial x} = 2a(t, \epsilon, \omega)x + b(t, \epsilon, \omega) \tag{5.20}$$

In Figure 5.6 (a)-(d) we have 4 snapshots at $t = \{0, 0.33, 0.66, 1\}$ for the double eddy vector field in the region $(x, y) \in [0, 2] \times [0, 1]$. We observe how the eddies stretch and expand together over time, conforming a dynamic vector field. We have set $A = 0.1$, $\epsilon = 0.25$ and $\omega = 2\pi$.



(a) $t = 0$. (b) $t = 0.33$.

(c) $t = 0.66$. (d) $t = 1$.

Figure 5.6: Snapshots for the analytical double eddy ocean currents at (a) $t = 0$, (b) $t = 0.33$, (c) $t = 0.66$, and (d) $t = 1$.

5.1.2.5 Stuart vortex

A Stuart vortex is a dynamic eddy-like structure (Kasten et al., 2012) defined for any location $\mathbf{x} = (x, y)$ and time t as follows

$$u(x, y, t) = 1 + \frac{\sinh(2y)}{A} \tag{5.21}$$

$$u(x, y, t) = \frac{\sin(2(x - t))}{4A} \tag{5.22}$$

where

$$A = \cosh(2y) - \frac{\cos(2(x - t))}{4} \tag{5.23}$$

In Figure 5.7 (a)-(d) we have 4 snapshots at $t = \{0, 0.33, 0.66, 1\}$ for the double eddy vector field in the region $(x, y) \in [0, 2] \times [0, 1]$. We observe how the eddies stretch and expand together over time, conforming a dynamic vector field. We have set $A = 0.1$, $\epsilon = 0.25$ and $\omega = 2\pi$.

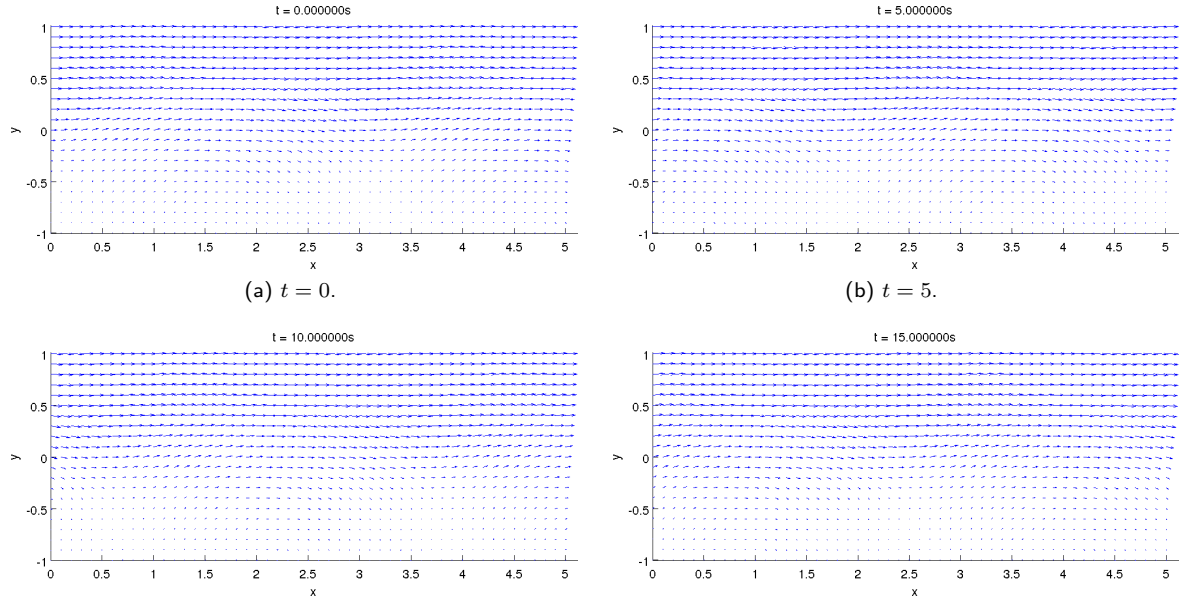
5.1.2.6 Tiled eddies

Interestingly, we can create a tile of eddies of the same size for any location $\mathbf{x} = (x, y)$ by using these simple equations,

$$u(x, y) = -\sin y \tag{5.24}$$

$$v(x, y) = \sin x \tag{5.25}$$

Similarly, we can make the tiled eddies a dynamic vector field by simply moving one of the coordinates x or y (Robel, 2010). In fact, we are just translating the equations of the tile in such direction. For instance, if



(a) $t = 0$. (b) $t = 5$.

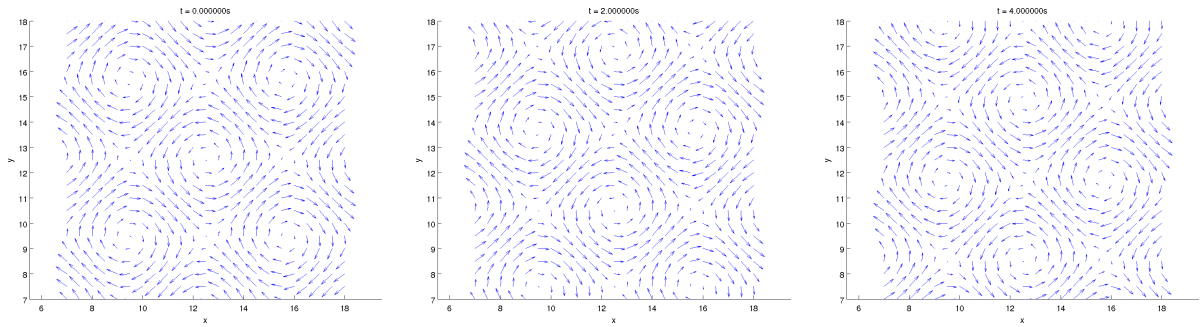
(c) $t = 10$. (d) $t = 15$.

Figure 5.7: Snapshots for the analytical Stuart vortex ocean currents at (a) $t = 0$, (b) $t = 5$, (c) $t = 10$, and (d) $t = 15$.

we make such translation to depend on the time t, we have dynamic tiled eddies. Therefore, in the case of a translation on y we have

$$u(x, y, t) = -\sin(y + t) \tag{5.26}$$

$$v(x, y, t) = \sin x \tag{5.27}$$



(a) $t = 0$. (b) $t = 2$. (c) $t = 4$.

Figure 5.8: Snapshots for the analytical dynamic tiled eddies ocean currents at (a) $t = 0$, (b) $t = 2$, and (c) $t = 4$.

In Figure 5.8 (a)-(c) we have 3 snapshots at $t = \{0, 2, 4\}$ for the dynamic tiled eddies vector field in the region $(x, y) \in [7, 18] \times [7, 18]$.

5.1.2.7 Sinusoidal

Eichhorn (2010) uses a time-varying ocean flow streamfunction that has a dynamic sinusoidal shape (Alvarez et al., 2004; Cencini et al., 1998), to test the path following problem. Indeed, we also use this same streamfunction in part of our preliminary experiments (see Section 5.7). The streamfunction is defined as

$$\Psi(x, y, t) = 1 - \tanh(Q) \tag{5.28}$$

where the quotient $Q = N/D$ is defined by

$$N = y + A \tag{5.29}$$

$$D = \sqrt{1 + A_x^2} \tag{5.30}$$

where, with a slightly abuse of notation for the sake of clarity,

$$A = A(x, t, k, c) = -B(t, \epsilon, \omega, \theta)\cos(k(x - ct)) \tag{5.31}$$

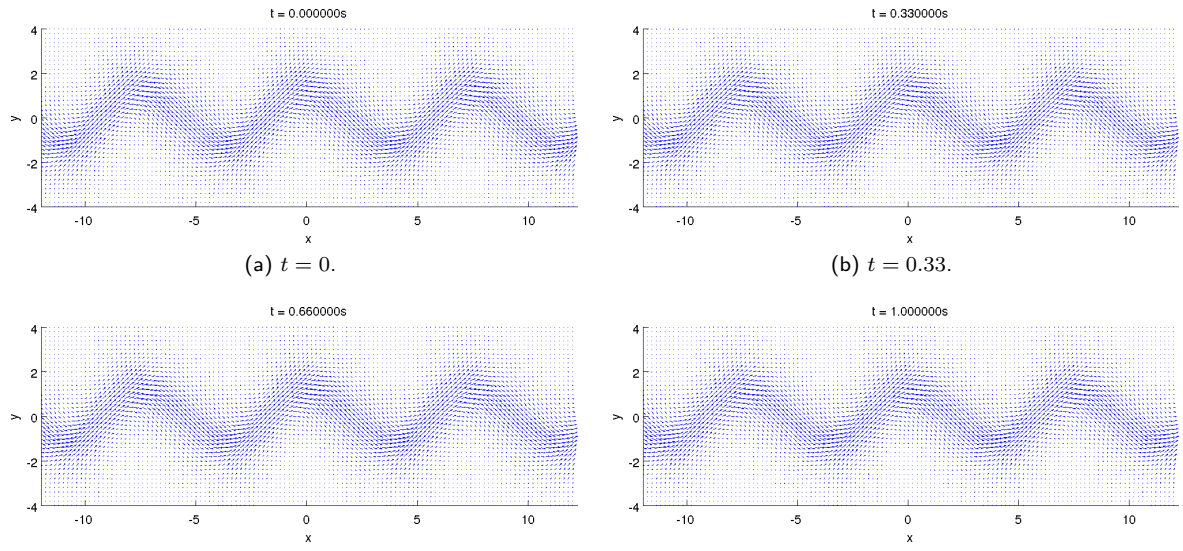
and

$$A_x = \frac{\partial A(x, t, k, c)}{\partial x} = kB(t, \epsilon, \omega, \theta)\sin(k(x - ct)) \tag{5.32}$$

where

$$B(t, \epsilon, \omega, \theta) = B_0 + \epsilon\cos(\omega t + \theta) \tag{5.33}$$

with $B_0 = 1.2$, $\epsilon = 0.3$, $\omega = 0.4$, $\theta = \pi/2$, $k = 0.84$ and $c = 0.12$. This are the values used in (Eichhorn, 2010) and in our tests in Section 5.7 as well.



(a) $t = 0$. (b) $t = 0.33$.

(c) $t = 0.66$. (d) $t = 1$.

Figure 5.9: Snapshots for the analytical sinusoidal ocean currents at (a) $t = 0$, (b) $t = 0.33$, (c) $t = 0.66$, and (d) $t = 1$.

In order to compute the vector field (u, v) from the streamfunction, we need the first partial derivatives

of (5.28), i.e. the Jacobian

$$\mathbf{J}_{\Psi} = \begin{pmatrix} \dfrac{\partial \Psi(x,y,t)}{\partial x} \\ \dfrac{\partial \Psi(x,y,t)}{\partial y} \end{pmatrix} = -\frac{\operatorname{sech}^2 Q}{D} \begin{pmatrix} A_x - \dfrac{Q A_{xx}}{D} \\ 1 \end{pmatrix} \tag{5.34}$$

where

$$A_{xx} = \frac{\partial^2 A(x,t,k,c)}{\partial x^2} = -k^2 A(x,t,k,c) \tag{5.35}$$

In Figure 5.9 (a)-(d) we have 4 snapshots at $t = \{0, 0.33, 0.66, 1\}$ for the dynamic sinusoidal vector field in the region $(x, y) \in [-12, 12] \times [-4, 4]$.

5.2 Evaluation Metrics

In order to evaluate the different algorithms developed in this thesis we will use a series of metrics. Here we enumerate and describe them in detail, since they will be used intensively in the following sections. Although these metrics are generally applicable to all the glider path planning problems addressed, in some cases the evaluation is rather qualitative. This is the case of problems like path following, some multi-glider problems and tracking evolving ocean structures, where the assessment of the technique depends tightly on the objective function that the problem aims to optimize. In these problems, we simply try to validate it can solve the problem adequately in a reasonable amount of computing time.

We distinguish two type of metrics. First, those that evaluate the quality of the result, that in the case of path planning is the optimality of the path found. Secondly, the computing time on average or the worst case. In order to obtain representative values for these evaluation metrics we will run a number of test cases with representative scenarios, divided in different categories. Such categories allow for a fine evaluation at different types of environments, like weak or strong ocean currents, static or dynamic ocean currents, offshore or coastal regions —which are usually populated by more obstacles—, and long-term or short-term missions.

Regarding the optimality of the path found, the actual evaluation metric depends on the glider path planning problem at hand, since it defines a specific objective function to optimize. However, for the problems discussed in this thesis, we identify two main optimality metrics, which are associated with each of the atomic glider path planning problems (see Section 3.1). On one hand, for the minimal time path problem we evaluate the **travel time**, which is the time required to go from the starting waypoint $\mathbf{x}_0$ to the target one $\mathbf{x}_{\text{goal}}$, for a given glider nominal surge speed U_g and the ocean currents U_c —of the map used in the tests— along the path $\mathcal{P}$ found. On the other hand, for the minimal distance to the target problem we evaluate the **remaining distance** to the target $\mathbf{x}_{\text{goal}}$, from the ending point $\mathbf{x}_{n-1}$ of the path $\mathcal{P}$ found for the travel time t_t given. In the case of the minimal time path problem, we have also used an alternative evaluation metric. This metric is the glider **effective speed** U_e of the path found. It provides a more relative value, rather than the travel time t_t, which is an absolute value. In fact, U_e is computed as

$$U_e = \frac{\|\mathbf{x}_{\text{goal}} - \mathbf{x}_0\|}{t_t} \tag{5.36}$$

which gives a quasi-normalized metric of the distance $d = \|\mathbf{x}_{\text{goal}} - \mathbf{x}_0\|$ from $\mathbf{x}_0$ to $\mathbf{x}_{\text{goal}}$ divided by the travel time t_t. However, this metric still depends on the distance d of a given test case, so we must categorized the test cases according to d in order to facilitate the comparison of the algorithms, as we do in Section 5.4 and 5.6

In glider path planning, the computing time is often ignored as a priority due to the typical long duration of glider missions and immersion periods of t_s time between surfacings. For this reason, the computing time of the results shown in the sequel is given simply as a relative reference. We have tried to use the best

implementation and configuration for each algorithm in our programming environment, but the computing time still depend strongly on the codification, compiling and executing efficiency, and it is not the central focus of this thesis.

Although this is generally true, in some cases the path planner must respond inside a short temporal window to face an unforeseen situation, and provide the glider with a new path before it dives again, during the short time it stays on surface for $\approx 15\text{min}$. In fact, this is very common in short-term missions, where the minimal distance to the target problem is usually solved at each surfacing point $\mathbf{x}_i$ in order to update the actual $\mathbf{x}_i$ in the path planner and obtain the optimal path for the next stints using the —possibly updated— dynamic ocean currents. Consequently, we evaluate the **computing time** of the techniques for a sufficient number of tests, and adopt the best configuration for each algorithm at the problem at hand.

5.3 Experiments

In the next sections we show the experimental results of the present thesis work. For each application or glider path planning problem we evaluate the algorithms discussed in Chapter 4. The results shown here have been obtained in simulation, but using ocean currents provided by ROMs that actually represent —up to some uncertainty— the state of the ocean, by means of the computation of ocean models fed and tuned with data assimilation. Therefore, the test cases run can be considered close to real field tests. Indeed, we have used some of the results to assist on the piloting of several gliders operating in long-term mission. This is the case of the RU27 trans-Atlantic missions, as we will see in Section 5.4.

In Chapter 3 we have described the glider path planning problems addressed in this work. They cover the atomic optimization problems of finding the minimal time path and the minimal distance to the target, along with a series of more elaborated problems. These two atomic problems are analyzed first, in order to validate the applicability of each technique in the basic problems that are later required to implement the others. Being the minimal time path problem the first one covered, we include the preliminary results we obtained at the start of this work. Similarly, we explain how we applied it for the piloting of the RU27 using the ocean currents of the ESEOAT region provided by the ROM of the ESEOO project. Apart from the feasibility of the techniques and the optimality of the path found, we also analyze the computing time, as mentioned previously. For this reason, we study the influence of the ABS strategy (see Section 4.4.3) for the CTS-A* algorithm at the beginning. Then, all the methods presented in the thesis are compared in long-term missions using static ocean currents, as this is the common scenario for minimal time path problems. On the other hand, for the minimal distance to the target problem, we address short-term missions, and we use dynamic ocean currents provided by the ROM forecast maps; note that in this case some methods are not applicable, like the RRT.

Then, we analyze how all the techniques perform in the presence of obstacles. We have run test experiments for both atomic problems in both types of missions —i.e. long-term and short-term, with static and dynamic ocean currents, respectively. Then, for the rest of the glider path planning problems, the most promising techniques are evaluated. For the path following problem we validate the different objective functions developed to measure the deviation of the path found $\mathcal{P}$ from the desired path $\mathcal{P}_d$ (see Section 3.3). Next, we apply our Iterative Optimization method to solve this problem as a constraint imposed on the minimal time path problem. We also relax the path following constraint with a corridor around $\mathcal{P}_d$ in order to obtain several non-dominated solutions for each objective. And finally, we illustrate how to scale this problem to more complex paths $\mathcal{P}_d$, as a windmill pattern to sample mesoscale eddies. Later, we coordinate multiple gliders to solve several multi-glider path planning problems. Here we cover the navigation in formation, and the efficient recovery of the glider fleet with a ship. Finally, we conclude our experiments by showing our preliminary results on tracking evolving ocean structures. In particular, we model mesoscale eddies as described in Section 4.8, and apply a Genetic Algorithm to optimize the sampling strategy of certain regions of the eddy.

5.4 Minimal Time Path

As we have previously mentioned, we are going to start the presentation of our experimental results with the minimal time path problem. This basic problem, in the field of glider path planning, is the equivalent to the shortest path problem for ground robots. It constitutes a minimal capacity required to perform other sorts of glider path planning, and it will be used in several applications at the ocean. Here, we compare the algorithms proposed in this thesis with the manual direct to goal piloting approach, and later among them. We consider two main cases for evaluation. The first case is the short-term path planning with dynamic ocean currents, where we have used the ROM forecast maps as input. The other one is the long-term path planning —in the order of ≈ 1 month traveling— using static ocean currents, since no ROM provides predictions for such number of days in advance. Indeed, we consider test cases in which the path does not last longer than the seasonal period, which can be represented with an average map.

In the first case, with dynamic ocean currents, we illustrate the viability of graph-based methods. The A* algorithm adapted to glider path planning (see Section 4.3) is compared with the novel CTS-A* approach (see Section 4.4). With our first experiments we do not want to evaluate these methods in accord with the metrics shown thus far. On the contrary, our intention is to put clear that the A* adaptation of authors like Garau et al. (2005); Pêtrès et al. (2005); Soulignac (2010) —and ours as well—, still does not reflects the glider motion model adequately. Later, using static ocean currents we will evaluate all the techniques presented in this dissertation using the evaluation metrics of Section 5.2, for the minimal time path problem in long-term scenarios.

At this point, we want to draw the attention to the central difference between A* and CTS-A*. Such difference comes from the fact that the cost function is based on the time rather than on the distance traveled. Furthermore, the effect of ocean currents on the glider motion model —whatever it is— creates a manifold that is confined to the accessibility cone shown in Section 4.4.3, i.e. the region where the vehicle is able to move according with the speed ratio SR. The A* algorithm defines an uniform grid in advance, where the nodes —or vertex— of the grid are placed at a given constant **distance** from each other. Consequently, the time required to reach a node is not constant, and it may even be impossible to follow the bearing to go from the current node to certain adjacent ones. Therefore, the cost function, which incorporates the constrained glider motion model and yields a time value, has a twofold negative impact on the A* algorithm.

First, in the presence of large ocean currents —i.e. a high speed ratio— the number of feasible bearings plummets. The path search is then reduced to a small number of adjacent nodes in the graph. Sometimes, it may even happen that the feasible bearings neither fall in any of these nodes, and no path is found. Clearly, we need to search in the bearing configuration space. This is a first consequence we conclude from the analysis of A* algorithm. The second negative impact is on the fail to remain faithful to the glider navigation pattern (see Section 2.1), in particular, regarding the constant time between successive surfacing points. Trying to reach the nodes on the grid, the time underwater is not constant anymore.

Our CTS-A* approach, although still based on A* search, solves the two problems of A* aforementioned. CTS-A* does it by exploring different bearing angles starting from the current node, and applying the unconstrained glider motion model of Section 2.2.2 for a constant integration time. Consequently, all bearings are explored and they produce a constant time stint underwater. Recall that the final location of these trajectories is then taken to a gridded search space, as explained in Section 4.4.

5.4.1 RU27 trans-Atlantic mission

In Figure 5.10 we show the output of A* and CTS-A*, along with the DtG approach and the path followed by the "Scarlet Knight" RU27 glider in the trans-Atlantic mission (Glenn et al., 2009). We developed our first path planning techniques —A* and CTS-A*— to assist in the glider piloting of RU27 when it entered in the area of the ocean modeled by the ESEOAT region of the ROM maintained by the ESEOO project, highlighted in Figure 5.11, which includes the Spanish EEZ (Exclusive Economic Zone) and the coast of Portugal and

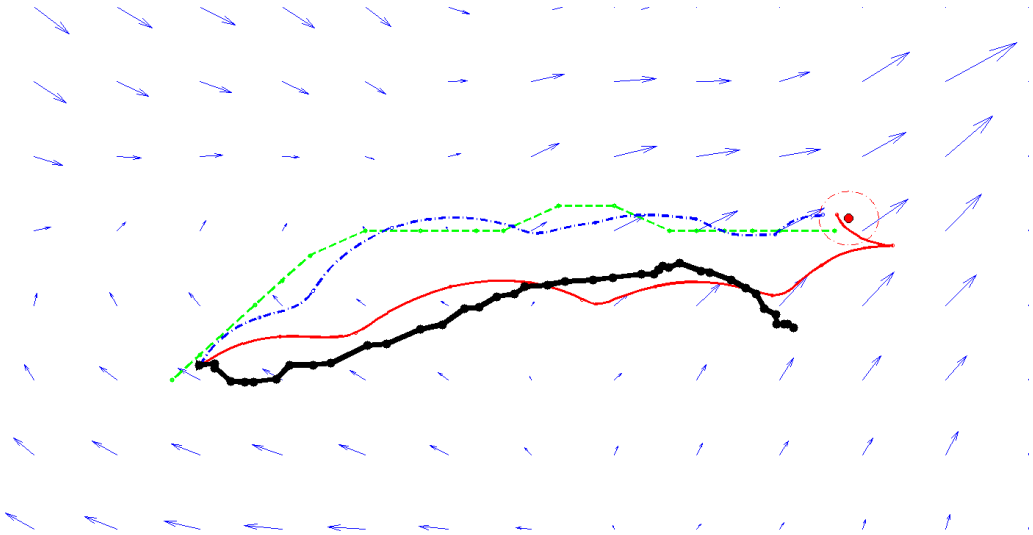


Figure 5.10: Comparison of paths planned with different path planning strategies from the start node ▶ to the target node ● for the 21-Nov-2009: DtG ——, A* - - - and CTS-A* -·-·-, along with the RU27 path ——; the ocean currents provided by ESEOAT in an area of $\approx 100\text{km}$, close to the coast of Galicia (Spain), were used by the path planning algorithms.

some part of France and Marocco. We run our algorithms to obtain the path plan that would allow the glider to reach the waypoint of Figure 5.10 starting at its actual surfacing point location on November 21, 2009. At this time —after almost 221 days of travel, as it took finally—, the ocean currents where very strong and highly time varying nearby. After several months flying the Atlantic Ocean, barnacles had attached to the glider's hull, as shown in Figure 5.13. Such accumulation of organisms on wet surfaces is known as biofouling. Its impact on the glider dynamics is an added mass and extra drag that reduces the vehicle horizontal speed; recall the spider plots shown for the force balance motion model in Section 2.2.3. More than 500 years ago, in 1493, a similar historic episode happens when the Spanish navigator *Pinzón* was coming back from the discovery of America. His caravel class ship named the *Pinta* was very damaged and the weather conditions were bad, as they were in our experiments with RU27. Our path planner proposed, for the glider, a path similar to the one *Pinzón* followed to reach the coast of Spain. For that reason, we coined *Pinzón* to our path planner.

For our experiments with RU27, we used the ROM forecast maps of ESEOAT, which predicts up to 3 days. The paths found with DtG, A* and CTS-A* techniques are plotted in Figure 5.10, assuming a glider horizontal speed of $U_g = 0.35\text{m/s}$. However, according with the RU27 path, also shown in the figure, the glider was traveling at a lower speed of $U_g \approx 0.15\text{m/s}$, as a consequence of the biofouling. Although simulated with a higher speed, the DtG approach follows a path similar to RU27, since the strategy is equivalent to manual piloting (see Section 4.2). Note that with $U_g = 0.15\text{m/s}$ it was not possible to reach the target, as shown in Figure 5.12 (b); the same happened to the real glider, indeed. For this reason, we try to take the glider as close as possible to the target, and we even found a path that reached the target for certain situations at the end of the mission, as shown in the figure (a). The blind approach of DtG executes very fast, but the path found takes more time than our graph-based path planning methods. While the DtG strategy sets the bearing in the straight direction to the goal, the other two methods explore several paths starting from the current location. Later, with long-term scenarios we will quantify the difference among them using the evaluation metrics.

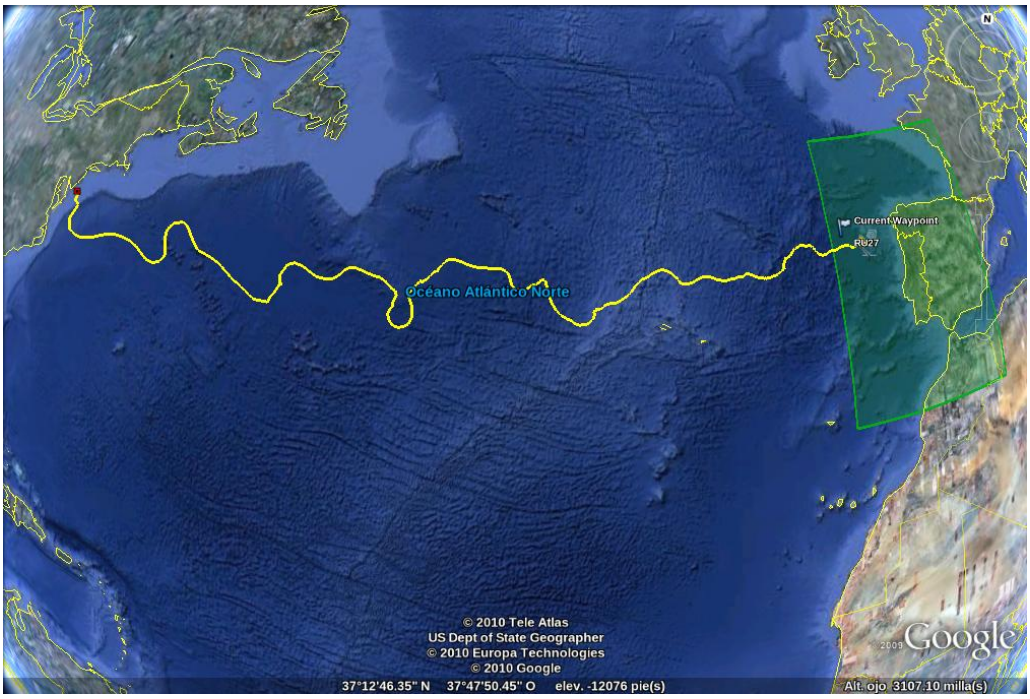
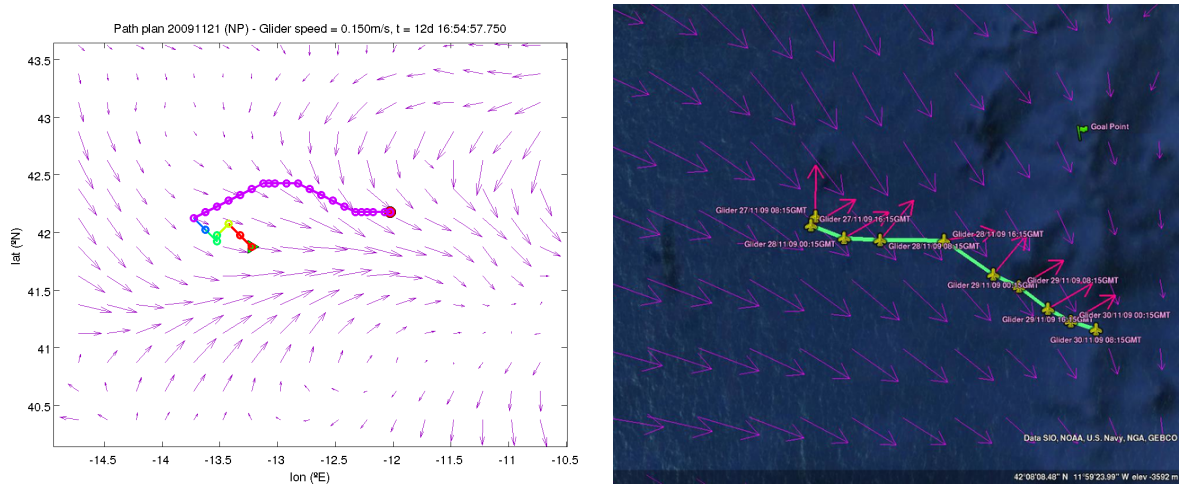


Figure 5.11: Map showing the RU27 path and the ESEOAT region, where we apply our path planning techniques to assist in the glider piloting, at the end of the trans-Atlantic mission.



(a) Plot output. (b) KML output.

Figure 5.12: *Pinzón* paths found for the RU27 trans-Atlantic mission: (a) plot, and (b) KML output; for two different situations at the end of the mission.

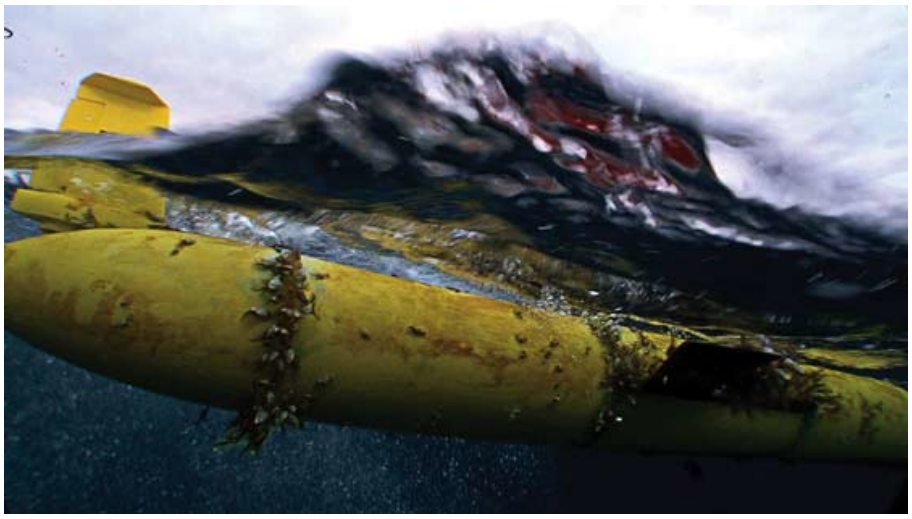


Figure 5.13: SLOCUM glider affected by biofouling after several months flying. Barnacles have attached to the vehicle's hull and have grown to the point that the dynamics are significantly affected. Image taken from the Atlantic Crossing blog (Glenn et al., 2009), which shows the conditions of RU27 by the end of the trans-Atlantic mission.

5.4.2 Initial Results

By now, we just want to highlight that the global view of the path planners helps in finding a better solution, close to the optimal one. And they search for that solution by adequately managing the dynamic ocean currents, the glider motion model, and the time-based cost function. The limitations of A* can be observed in Figure 5.10. A* represents each surfacing point with a node in a regular grid. As a consequence, the set of bearings evaluated for the next action is reduced to the ones that connect the current node with its neighbors. The surfacing point of the path found by A* shows that gridded pattern in the figure. Also, the distance traveled is constant, but not the time. Meanwhile, the path found by CTS-A* has constant time stints, which a travel distance which is clearly different among each of them in the figure. With CTS-A*, the underwater trajectory is also computed with the unconstrained glider motion model. This can be appreciated in the bend path of CTS-A* between surfacing points. On the contrary, A* uses a constrained motion model that —apart from some improvements made in our implementation— only takes a single mean velocity to represent the ocean current speed during a stint between adjacent nodes. Consequently, it does not compute the motion accurately, neither considers the variability of the ocean currents during the stint.

Although the path found by A* may have a similar cost —or even better, in some experiments— to CTS-A* one, the important point to bear in mind is the fact that A* does not model the glider motion adequately. Or, put differently, with CTS-A* the glider motion model is accurately computed, and the constant time surfacing behavior is guaranteed within the path planning algorithm. All in all, for the test run shown in Figure 5.10, both A* and CTS-A* find paths of lower cost than the DtG one, which costs approximately $8\mathrm{h}$ more. As it have been previously mentioned, the improvement in the path cost is greater when the speed ratio is high. For this reason, we have analyzed the influence of the ocean currents strength, their temporal variability and the size of ocean eddies, on the results of our path planning methods and the DtG approach.

The effect of the ocean current field strength actually relies on the speed ratio given by $\frac{U_{\mathrm{cmax}}}{U_g}$, used in the heuristic shown in Section 4.3. It determines whether the glider can beat them or not. From that speed ratio equation we have that an U_g increase is equivalent to navigate through weaker ocean currents; or, on the contrary, a U_g decrease is like moving in strong ones. For testing purposes we consider a given ocean

current field and different values for U_g. For slow speeds (or strong ocean currents) the glider will be more sensitive to the drift —almost following the ocean currents—, as observed in the more curved trajectory of the $U_g = 0.2\mathrm{m/s}$ case of Figure 5.14. On the contrary, the resulting paths tend to the straight line with the faster $U_g = 0.35\mathrm{m/s}$ speed (weaker ocean currents), as shown in the figure as well. Clearly, intelligent path planning is more important under strong ocean currents, or more specifically, when the speed ratio is high. In fact, for weak currents the DtG approach suffices to capture a nearly optimal path. Furthermore, the figure shows the influence of strong ocean currents on the glider trajectory while it glides underwater, as computed by our CTS-A*. It integrates the equations of the unconstrained motion model of Section 2.2, so such trajectories are accurately computed and used by the planner. In the case of A*, we do not have those trajectories, but the effect of strong ocean is more severe, because it prunes many path from the search space prematurely.

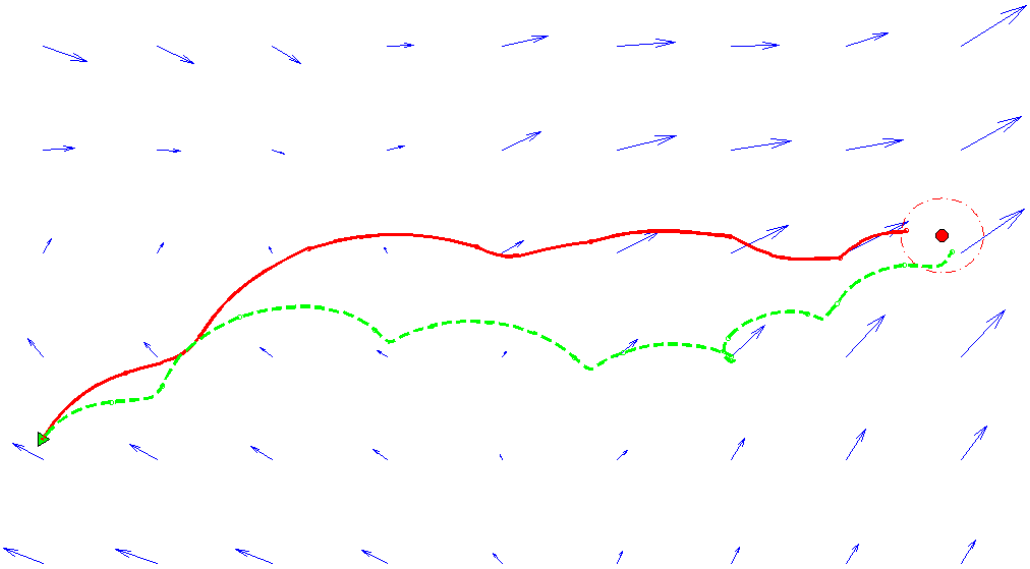


Figure 5.14: Paths planned for November 21, 2009, considering fast $U_g = 0.35\mathrm{m/s}$ (speed ratio of 167.02%) —— and slow $U_g = 0.2\mathrm{m/s}$ (speed ratio of 292.28%) - - - glider horizontal speeds, which is equivalent to path planning for weak and strong ocean currents, respectively. The paths were found with the CTS-A* algorithm using the ocean currents provided by ESEOAT in an area of $\approx 100\mathrm{km}$, close to the coast of Galicia (Spain).

Similarly, it is possible to test the path planner for different structures of the ocean currents. Eddies are common structures that may vary in size; from small and short life, to large and long life ones —like mesoscale eddies. When they are large, the resulting paths are more intricate, provided that the ocean currents are sufficiently strong. Therefore, path planning proves particularly useful in such cases, as we will see throughout the experiments of this chapter. Indeed, during the RU27 flight sub-mesoscale eddies were oscillating over time in the ESEOAT region.

5.4.3 ABS CTS-A*

Before we compare all the path planning techniques implemented in this thesis for the minimal time path problem, we are going to evaluate our CTS-A* algorithm in terms of computing time. Such evaluation includes the Adaptive Bearing Sampling (ABS) strategy explained in Section 4.4.3, which helps to reduce the CTS-A* computing time. Furthermore, we consider several numbers of samples M to evaluate up to which point it can be reduced before the path quality degenerates significantly. Since we are interested in establish such value

of M, according with the cost of the path found, we will also evaluate that cost, along with the computing time. For this purpose, we have built 21 test cases using ocean current maps of the ESEOAT region. For these experiments we have considered 2D surface ocean currents, which suffice to illustrate the ABS proposal.

We have run the next experiments on long-term scenarios, and thus we have static ocean currents. With this setup, the difference between the techniques is more evident and clear than with short-term, dynamic cases. Anyway, all the techniques are designed to accommodate dynamic ocean currents, and have been tested accordingly. The start and goal points are generated randomly and those test cases more representative and sufficiently complex have been selected. All tests have been run on an Intel® Core™-i7 CPU 2630 QM running at $2\mathrm{GHz}$ with $8\mathrm{GiB}$ of RAM.

Table 5.1: Minimal time path cost speedup of the paths found for CTS-A* with uniform sampling using $M = 18$ samples, and the ABS for different numbers of samples M, for the 21 test cases run. We observe that even with $M = 8$ samples, the cost of the paths found is not significantly affected, and even in some cases a small speedup is obtained, i.e. $> 100\%$.

	ABS CTS-A*		
Case #	$M = 18$	$M = 15$	$M = 8$
1	98.9%	98.8%	98.4%
2	100.1%	100.2%	100.2%
3	100.3%	99.9%	98.6%
4	100.4%	98.1%	98.7%
5	100.5%	100.7%	101.1%
6	99.8%	99.8%	99.8%
7	101.1%	100.3%	99.8%
8	103.1%	99.3%	100.5%
9	100.5%	100.6%	99.8%
10	100.2%	100.4%	100.8%
11	97.9%	104.2%	103.4%
12	101.0%	99.8%	101.1%
13	100.3%	100.0%	99.6%
14	99.7%	99.6%	100.1%
15	100.2%	100.2%	100.0%
16	99.2%	100.5%	98.9%
17	101.0%	101.8%	101.3%
18	101.0%	99.3%	98.8%
19	99.8%	99.3%	98.6%
20	100.4%	100.6%	99.9%
21	100.7%	100.8%	100.7%
Mean (std)	$100.3(1.0)\%$	$100.2(1.2)\%$	$100.0(1.2)\%$

We have evaluated two bearing sampling strategies. On one hand, we have the uniform sampling of the original CTS-A*. On the other, its extension with the ABS procedure, using a bearing distribution defined by $f_{\psi_d}(\psi_e) = \mathcal{N}(\psi_e;\ \mu, \sigma^2)$ with $\sigma^2 = 10^\circ \approx 0.1745\mathrm{rad}$ and $f_{\psi_c}(\psi_e)$. The simulations have been run with $U_g = 0.4\mathrm{m/s}$ and $t_s = 8\mathrm{h}$ between consecutive surfacing points. For the uniform sampling CTS-A* we have used $M = 18$ bearing samples (20° bearing resolution). A greater M would have given a higher resolution, but at a very high computational cost. On the other hand, the ABS permits the reduction of the number of samples M without compromising the exploration of the search space and, consequently, the path quality. Therefore, as we will see, we still can obtain good results in less computing time. However, we cannot reduce

Table 5.2: CPU (computing) time speedup to find the paths in the 21 test cases run, for CTS-A* with uniform sampling using $M = 18$ samples, and the ABS for different numbers of samples M.

	ABS CTS-A*		
Case #	$M = 18$	$M = 15$	$M = 8$
1	93.5%	161.6%	199.1%
2	93.4%	162.7%	202.6%
3	93.2%	163.8%	196.7%
4	93.6%	158.3%	194.7%
5	95.9%	168.4%	207.0%
6	94.1%	165.6%	204.6%
7	95.3%	164.9%	200.3%
8	111.4%	170.4%	237.6%
9	94.8%	165.3%	198.6%
10	92.7%	162.3%	202.0%
11	93.2%	153.6%	280.9%
12	97.2%	115.3%	206.9%
13	95.5%	117.0%	198.8%
14	94.6%	112.6%	198.2%
15	98.3%	118.7%	211.4%
16	92.7%	113.9%	203.8%
17	96.5%	118.6%	206.2%
18	115.9%	114.6%	203.2%
19	97.0%	117.0%	203.2%
20	96.2%	113.9%	192.9%
21	97.8%	114.9%	209.4%
Mean (std)	$96.8(5.9)\%$	$140.6(24.7)\%$	$207.5(19.1)\%$

M to the minimum because the sampling process degrades. The reason is twofold: first, with a small M the probabilistic sampling distribution of the ABS is badly modeled; secondly, a reduction of M also reduces the search space. As a consequence, the path planning algorithm loses its exploration capabilities —in the exploration vs. exploitation trade-off—, and the path found may be far from the optimal. In our experiments we have set $M = \{18, 15, 8\}$ to illustrate the influence on the computing time and path cost. As we will see in the sequel, these M values are an adequate selection because the penalty on the path cost is not very severe. In fact, the path cost allows us to check the impact on the path search, by simply comparing against the result of CTS-A* with uniform sampling, which is taken as a reference for comparison.

Since the ABS strategy is a probabilistic algorithm, we have run a Mann–Whitney–Wilcoxon (MWW) test (Fay et al., 2010) to assess whether the CPU time (computing time) and path cost are significantly different for the ABS approach (null hypothesis H_0) or not. In Table 5.1 and 5.2 we have the results for the test cases run to evaluate the ABS strategy, designed to improve the computing time of the CTS-A* algorithm. Respectively, we have the minimal time path cost and the CPU time for each test case in the rows of the tables. The results of CTS-A* with $M = 18$ samples are taken as a reference. We can observe the tendency of the path cost and the CPU time for the decreasing values of M tested in the ABS strategy. Clearly, the computing time is reduced with M for all the cases. Indeed, by reducing M we have been able to drop down the CPU time up to $\approx 50\%$ —i.e. a $\approx 200\%$ speedup— using ABS with $M = 8$, in comparison with the uniform sampling with $M = 18$. We can also see that ABS introduces an overhead of $\approx 3\%$, which is clearly appreciated for $M = 18$, which is the same number of samples used in the uniform sampling CTS-A*.

Table 5.3: Summary of minimal time path cost (days) of the 21 test cases run for CTS-A* with uniform sampling and the ABS strategy. We show the mean and the 95% CI of all the test cases.

Method	M	Median	95% CI
CTS-A*	18	20.62	$[13.49, 27.75]$
ABS CTS-A*	18	20.55	$[13.35, 27.75]$
ABS CTS-A*	15	20.63	$[13.43, 27.83]$
ABS CTS-A*	8	20.71	$[13.49, 27.93]$

The path cost requires special attention. The probabilistic nature of ABS CTS-A* produces results that must be evaluated globally for all the test cases, and for several runs on the same test case as well. Interestingly, we observe that for small values of M the path cost is not very different from the CTS-A* one. In Table 5.3 we have a summary of the path cost for all the test cases. In our tests we observe that ABS CTS-A* with $M = 18$ samples, as in the uniform sampling approach, produces even better results in average. With fewer samples M, the minimal time path cost increases, but not significantly. Indeed, with $M = 8$ the cost remains acceptable in average. The CPU time, on the other hand, is clearly lower, with a $\approx 140\%$ speedup with $M = 15$, and a $\approx 200\%$ speedup with $M = 8$, with respect to CTS-A*, according with the mean values in Table 5.2.

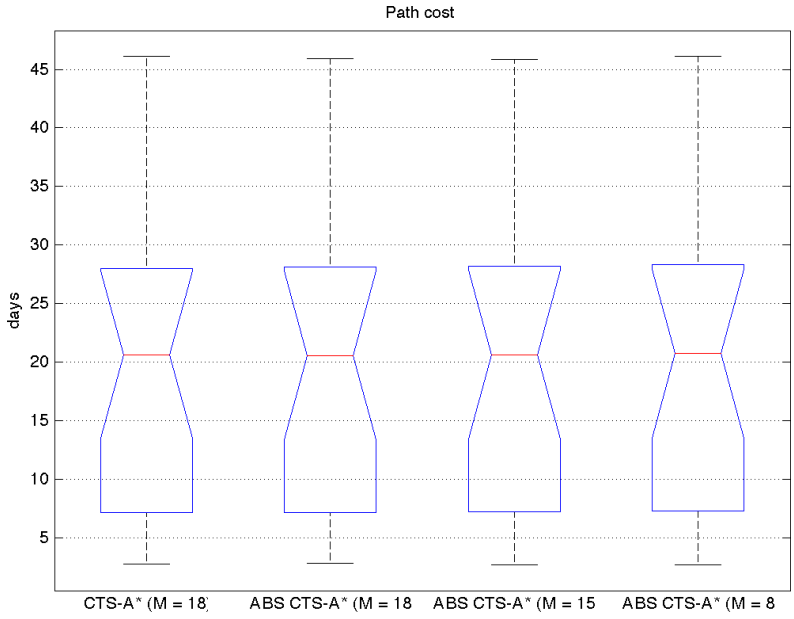


Figure 5.15: Path cost box and whisker plot for the 21 test cases run to evaluate the ABS strategy of CTS-A*.

We have gone a step further in the evaluation of the ABS strategy. We have run an hypothesis test to evaluate whether the ABS CTS-A* minimal time path cost is significantly different with respect to the uniform sampling CTS-A* or not. Before we apply the test, we can obtain some intuition of the difference between CTS-A* and the several configurations of ABS CTS-A* —for $M = \{18, 15, 8\}$ samples— using a box and whisker plot for all the test cases. In the box and whisker plots of Figure 5.15 we have more information about all the test cases than in the summary table commented thus far; it shows the median and 95% confidence intervals (CI) —discarding outliers— of the distribution of the results for all the test cases, for the path cost.

Although it seems clear that there is no significant difference in the path cost between CTS-A* and any of the ABS CTS-A* configurations, we are going to run a Mann–Whitney–Wilcoxon (MWW) test.

Table 5.4: Mann–Whitney–Wilcoxon (MWW) test for ABS CTS-A* with respect to (uniform sampling) CTS-A* with $M = 18$ samples —for the latter. The test is applied on the ABS with $M = \{18, 15, 8\}$ samples. The results for several runs on each of the 21 test cases are shown for a $\alpha = 0.05$ significance level hypothesis test. Note the the U value is the result for the equivalent Mann-Whitney U-test.

	Computing time			Path cost		
M	U	p-value	$H?$	U	p-value	$H?$
18	438.50	0.753	H_0	456.00	0.101	H_0
15	511.00	0.138	H_0	452.00	1.000	H_0
8	551.00	0.013	H_1	451.00	1.000	H_0

In Table 5.4 we have the results of the MWW test for both the CPU time and the minimal time path cost, with an $\alpha = 0.05$ significance level. The test shows that the CPU time is clearly different, and actually better, for the ABS CTS-A* with $M = 8$. Certainly, with $M = 15$ the algorithm is also faster than CTS-A* with uniform sampling and $M = 18$, but the difference for all the test case is still not significant. Regarding the path cost, we have that for any value of M tested, the ABS CTS-A* have found paths that are not significantly different from the ones found with the uniform sampling CTS-A*. This means that the ABS CTS-A* is valid, faster and still provides good results in terms of the path planning cost function. In fact, with $M = 8$ the reduction in computing time is justified, since the penalty on the path cost is not significant, according with the MWW test results for our test cases.

Figure 5.16 shows an example of the paths obtained using uniform sampling and the ABS, with $M = 18$ and $M = 15$ samples, respectively. This example corresponds with one of the test cases used for the previous evaluation of the ABS strategy. We observe a long-term scenario in the figure, in which ocean currents are static. The distance from the starting node to the target one is in the order of hundreds of kilometers and it takes several days to reach it. More specifically, the travel time goes from approximately 1 week to no more than 2 months, so the temporal dynamics of ocean currents within a day can be neglected, and we also avoid seasonal changes in the average ocean current field that we use. Furthermore, the path planning algorithm must explore a great number of alternative paths from the origin in order to find the optimal one. With this type of scenarios we are able to check the exploration and exploitation capabilities of the path planning algorithms, which are reflected in the path cost and computing time. In fact, for the particular case of the figure, we observe that the shape of both paths is very similar, and the cost is nearly the same. Consequently, the ABS CTS-A* can be used instead of CTS-A* —with uniform sampling— when we need a result faster, as long as we can deal with a small penalty on the path optimality.

In sum, the ABS strategy is a probabilistic technique that reduces the computing time by means of reducing the search space in a guided way (see Section 4.4.3), at the expense of degrading the optimality of the path found slightly. It is for this last reason that in the following comparison results with other techniques —for this and other problems— we use the deterministic version. That is, we use the uniform sampling, which is slower because it requires a higher number of samples M, but it produces the best possible results for the CTS-A* algorithm, in terms of path cost.

5.4.4 Comparison in static long-term scenarios

Now that we have validated our implementation of the A* adaptation to glider path planning, and our novel CTS-A* approach, we proceed to compare the techniques developed in this thesis. The methods compared are the Direct to Goal approach, A*, RRT, CTS-A*, ABS CTS-A* and Iterative Optimization, all explained

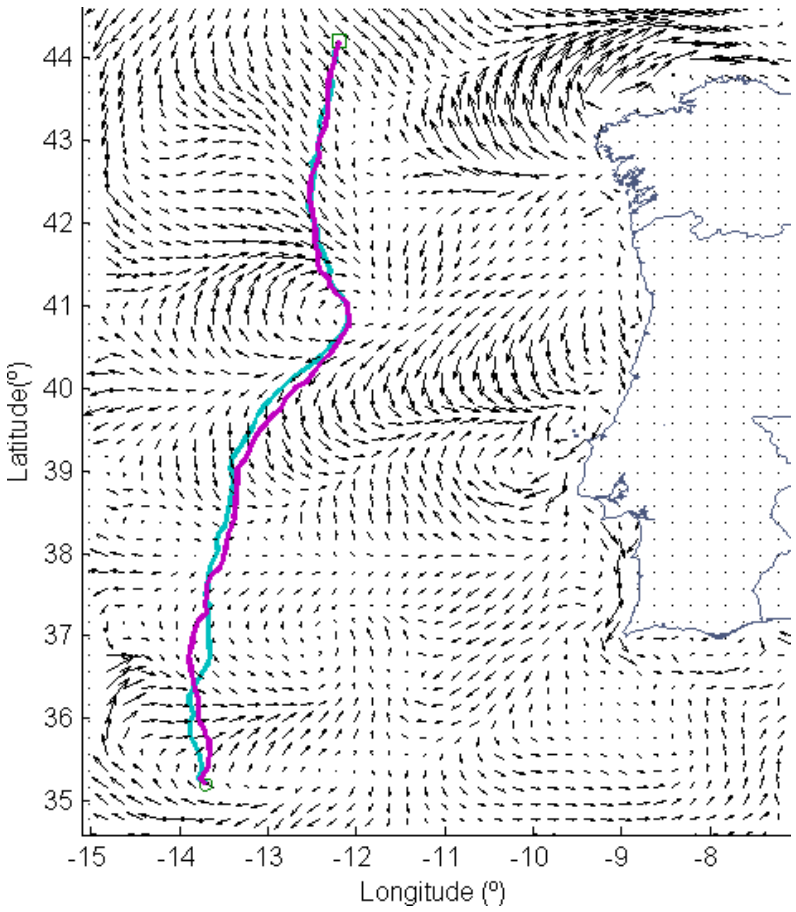


Figure 5.16: Path found by CTS-A* ($M = 18$) ——— and ABS CTS-A* ($M = 15$) ——— for the test case $\#2$ with start □ and goal ○. Strong ocean currents ($\geq v_g$, where $v_g = 0.4\mathrm{m/s}$) are highlighted ⟶. The ABS CTS-A* has a similar cost to CTS-A*, which uses an uniform sampling of the bearing configuration space. However, with ABS only $M = 15$ samples are required, rather than $M = 18$. Hence, the computing time is reduced, but we still have a good result, significantly equivalent to the CTS-A* one.

in Chapter 4. We have used the same equivalent discretization level for each method, when applicable. For instance, the search grid used for A* and CTS-A* has a cell size of $1/20^\circ$ in x and y axes. For the CTS-A* algorithm we have set a division of 20° in the bearing rose, i.e. $M = 18$ samples. We reduce the number of samples to $M = 15$ in the ABS CTS-A*. Similarly, the Iterative Optimization method uses the SQP algorithm in the optimization phase.

For this comparison we have picked out several test cases from the ESEOAT region. The test cases are defined by a start and target waypoints, and a static ocean current field. Being static ocean currents, as in the example test case of Figure 5.17, we can include the RRT algorithm in the comparison. The use of static ocean currents is also a consequence of the fact that the number of forecast maps provided by the ROM is small. Therefore, since we want representative, long test cases, we are compelled to run our experiments on static ocean currents. Anyway, all the techniques —except for RRT— are applicable to dynamic ocean currents, as we will show several experiments for other problems and applications throughout this chapter. It is also worth noting that A* results require special consideration. The constrained motion model (see

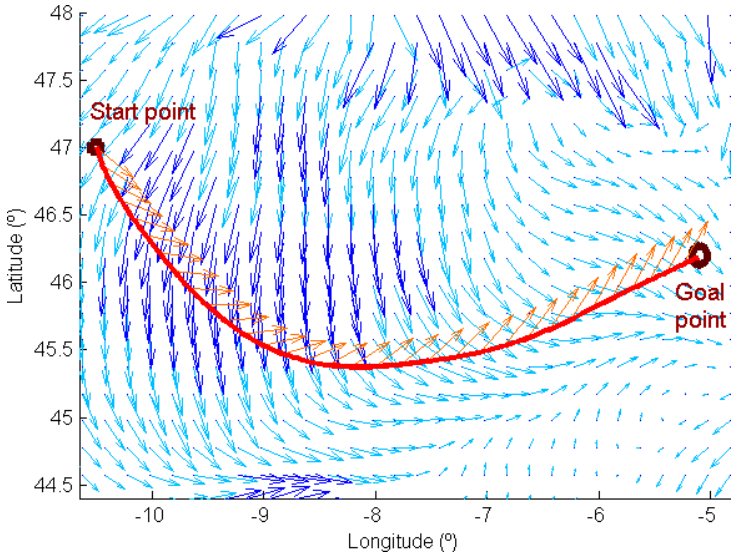


Figure 5.17: Minimal time path obtained with the Iterative Optimization technique for an example case. The glider speed was set to $U_g = 0.4\text{m/s}$. The distance traveled was $d = 423.22\text{km}$, while the travel time was $t_t = 10.5$ days. The computing time to obtain this result was $t_c = 24.9\text{s}$. The figure shows the path over the ocean currents → (ocean currents that exceed the glider speed U_g are highlighted →) from the start ● to the target waypoint ◎. Finally, the glider bearing to set at each surfacing point is shown with orange arrows →.

Table 5.5: Path cost (travel time) in days for the Minimal Time Path problem. The mean for the 20 test cases is shown for comparison, for a glider nominal speed of $U_g = 0.4\text{m/s}$. When a technique does not find a solution for some test cases, we include the percentage of NA (No Arrival) cases.

Method	All	Successful
DtG	NA in 60%	18.0
RRT	NA in 20%	17.7
A*	19.4	17.1
CTS-A*	19.2	16.9
ABS CTS-A*	19.2	16.9
Iterative Optimization	18.9	16.7

Section 2.2.2) that it uses to generate the trajectories produces unrealistic non-constant stint times between surfacing points. That is to say, the surfacing points in A* will not generally correspond with the actual surfacing points of the glider, as they are forced to be the nodes of the search graph.

The experiments described here were configured for a glider speed of $U_g = 0.4\text{m/s}$ and a period between surfacing points of $T_s = 8\text{h}$. A set of 20 test cases, with an average distance of $\approx 650\text{km}$ from the start to the target waypoint, has been analyzed. We apply the path cost and computing time evaluation metrics described in Section 5.2 to compare the results of the path planning techniques. Note that the computing time is quite an important factor, as sometimes it is necessary to obtain a path in a few minutes. Although this depends on the particular case and field missions constraints, a small computing time is useful.

Table 5.5 summarizes the path cost results of the 20 test cases by showing the mean of the travel time of the path found for each method. As some algorithms did not find a solution in some test cases —shown as

Table 5.6: Effective velocity in km/day for the Minimal Time Path problem. The mean and standard deviation for the 20 test cases is shown for comparison, for a glider nominal speed of $U_g = 0.4\mathrm{m}/\mathrm{s}$.

Method	Mean (std)
DtG	31.7 (5.0)
RRT	32.6 (4.5)
A*	32.9 (4.1)
CTS-A*	33.4 (4.5)
ABS CTS-A*	33.4 (4.5)
Iterative Optimization	34.3 (4.6)

Table 5.7: CPU (computing) time (in seconds) for the Minimal Time Path problem. The mean for the 20 test cases is shown for comparison, for a glider nominal speed of $U_g = 0.4\mathrm{m}/\mathrm{s}$. The ABS CTS-A* with the same number of samples M as the CTS-A* takes slightly more time, but it can run faster with fewer samples; however, the path cost would be affected, so we do not show any CPU time, because it depends on M.

Method	Computing time (s)
DtG	0.1
RRT	1.0
A*	75.0
CTS-A*	120.0
Iterative Optimization	74.0

the no arrival percentage under the mean of all cases column—, we have added a third column showing the mean for only those test cases where all the methods found a solution. The interpretation of these results is almost the same for both columns, i.e. for all the test cases or for only the successful ones for all the methods. In fact, the rows of the table show the methods in order from the worst to the best one. The DtG approach finds the paths which are less optimal, taking up to 18 days on average for the successful test cases. But more interesting is the fact that it does not find a path to reach the target waypoint in a 35% of the cases. It gets trapped in some areas between the start and target waypoint, because of its local blind search to the target. On the contrary, with the path planning algorithms we are able to find a path for all the cases; except for RRT, which fails in a 5% of the cases —much less than the DtG. Although RRT improves the results of the DtG approach, it is with graph-based methods and optimization when we have a greater improvement. CTS-A* performs slightly better than A* on average, but it is the Iterative Optimization approach the one that finds the path closest to the optimal.

An equivalent metric to the path cost is the effective speed, which normalizes the cost by the distance from the start to the target waypoint as explained in Section 5.2. In Table 5.6 we have the mean and standard deviation of the effective speed for each path planning method. The conclusion remains the same as with the path cost evaluation metric of Table 5.5. The effective speed is a good metric for evaluation in this case because our 20 test cases cover a wide range of scenarios, with different distances from the start to the target waypoint.

Table 5.7 shows the computing time for each method. The DtG approach takes less than $1\mathrm{s}$ on average, being an extremely fast method to obtain a rough path. With RRT we still can obtain a path in $\approx 1\mathrm{s}$. However, recall that RRT is only applicable to static ocean currents and it does not find optimal paths. Furthermore, both DtG and RRT do not find a path to the target waypoint for all test cases —specially the DtG approach. With the graph-based methods the computing time increases, but we always find a path. First, A* takes more than 1 minute on average. Our CTS-A* algorithm with $M = 18$ samples requires ≈ 2

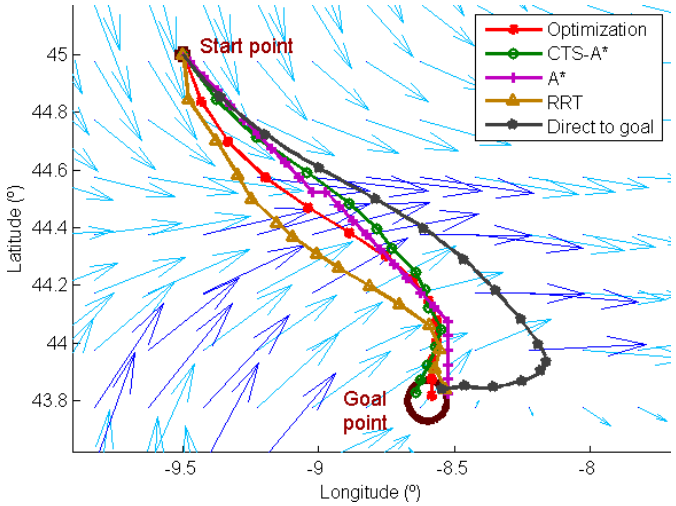


Figure 5.18: Comparison of the Minimal Time Path for the DtG, RRT, A*, CTS-A* with $M = 18$ samples, and Iterative Optimization, for the case $\# 1$. The glider speed was set to $U_g = 0.4\text{m/s}$. The distance traveled was $d = 151.4\text{km}$. The figure shows the path over the ocean currents $\longrightarrow$ (ocean currents that exceed the glider speed U_g are highlighted $\longrightarrow$) from the start ● to the target waypoint ○. The travel time t_t (in days) for each method was: DtG: 4.9; RRT: 4.6; A*: 4.5; CTS-A*: 4.3; Iterative Optimization: 4.0.

minutes. This increment in computing time is a direct consequence of the fact that with the unconstrained motion model more nodes are expanded than in A*. We could use the ABS strategy with less samples M to reduce the computing time. The Iterative Optimization method takes slightly less than the A*. But more interestingly, it yields the best path on average, and contrary to A* it also integrates the glider trajectory underwater with the unconstrained motion model.

In Figure 5.18 and 5.19 we have the comparison of the minimal time path found by each technique for 2 test cases of our 20. In the case $\# 2$ we observe that the DtG approach does not find any path, while the path planning techniques do. The results of these test cases agree with the summary tables discussed thus far. In conclusion, the experimental results demonstrate that the Iterative Optimization approach offers better results. However, the graph-based and optimization-based methods presented have a computational time of the same order of magnitude.

5.4.5 Discussion

Regarding the graph-based methods, if we had used a smaller cell size for the search grid of A*, or analogously, if we had increased the number of bearing samples M that CTS-A* explores, the path cost would have improved. However, the computing time would have increased notably. Consequently, from the results obtained with the 20 test cases, we can conclude that with the Iterative Optimization method we can find a close to optimal path in a reasonable time. Furthermore, this approach scales well to accommodate strong and dynamic ocean currents. For instance, in Figure 5.20 we include an example of the paths obtained for $U_g = 0.2\text{m/s}$. This is equivalent to navigating through strong ocean currents. In this type of adverse conditions, DtG tends to produce worse results or the path may not arrive to the target waypoint. The same may happen to the path planning techniques. Indeed, we observe that the difference in path cost is magnified, with respect to the average values of Table 5.5.

Finally, the test case shown in Figure 5.21 illustrates the problems of the DtG and RRT, which gets trapped in an area of strong ocean currents, while the path planning methods find a path to the goal that avoids such

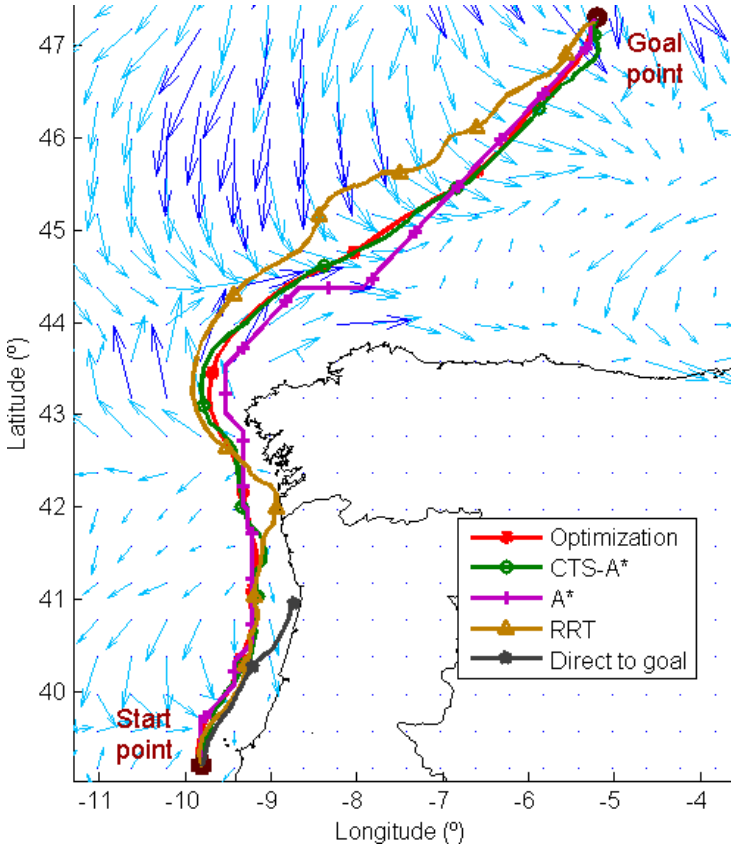


Figure 5.19: Comparison of the Minimal Time Path for the DtG, RRT, A*, CTS-A* with $M = 18$ samples, and Iterative Optimization, for the case $\# 2$. The glider speed was set to $U_g = 0.4\text{m/s}$. The distance traveled was $d = 974\text{km}$. The figure shows the path over the ocean currents $\longrightarrow$ (ocean currents that exceed the glider speed U_g are highlighted $\longrightarrow$) from the start ● to the target waypoint ○. The travel time t_t (in days, or NA for No Arrival) for each method was: DtG: NA; RRT: 31.3; A*: 27.3; CTS-A*: 27.2; Iterative Optimization: 26.3.

region. On the other hand, graph-based methods perform a more exhaustive search and find a path close to the optimal one. The Iterative Optimization method performs a less exhaustive search than graph-based methods, but it is able to successively refine the solution to one closer to the optimal.

With $U_g = 0.2\text{m/s}$ the Optimization method does not find a solution in some test cases because of the strong ocean currents with respect to the glider nominal speed U_g. Therefore, such ocean currents may become obstacles to the glider navigation and, consequently, the optimization-based methods gets trapped. This particular situation is later studied in Section 5.6, where we evaluate the algorithms for the obstacle avoidance problem.

5.5 Path of Minimal Distance to the Target

The second basic path planning problem consists on finding the path that minimizes the distance from the final point of the path to a target waypoint, for a given travel time. In other words, with such objective function we try to find the path that leaves the vehicle closer to the target after a known time. This problem

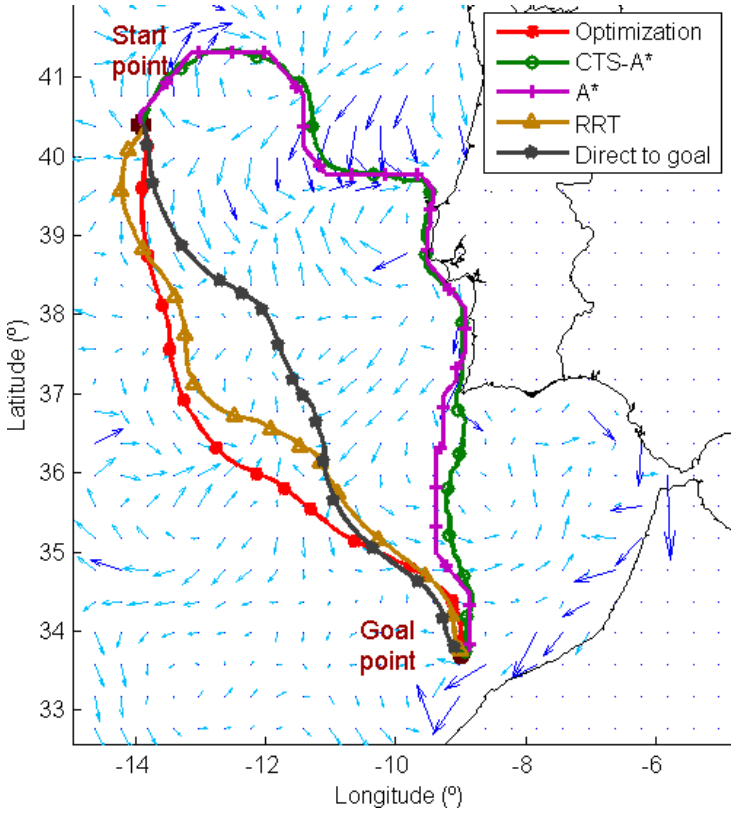


Figure 5.20: Comparison of the Minimal Time Path for the DtG, RRT, A*, CTS-A* with $M = 18$ samples, and Iterative Optimization, for a high speed ratio, i.e. for strong ocean currents, by setting the glider speed to $U_g = 0.2\text{m/s}$. The distance traveled was $d = 861.88\text{km}$. The figure shows the path over the ocean currents $\longrightarrow$ (ocean currents that exceed the glider speed U_g are highlighted $\longrightarrow$) from the start ● to the target waypoint ○. The travel time t_t (in days) for each method was: DtG: 53.8; RRT: 49.7; A*: 49.6; CTS-A*: 50.0; Iterative Optimization: 47.4.

is interesting because ROMs provide forecast only for a few days in advance. A key aspect of this problem is precisely that we must know the travel time on advance. As explained in Section 3.1.2, this is common in short-term path planning scenarios, as for glider recovery or when navigating with the ROM forecast maps. In fact, when RU27 was close to Baiona (Spain), at the end of the trans-Atlantic mission, we also tried this kind of path planning. The high temporal variability of the strong currents, as a consequence of the bad weather, forced us to take into account the hourly evolution of the ocean currents, in order to move forward to the target. At that time we used the forecast maps of the ESEOO project for the ESEOAT region. Similarly, for our experiments, we have used the forecast maps of the ESEOAT and ESEOCAN regions. They provide the hourly predictions of surface ocean currents up to $3 \times 24\text{h} = 72\text{h}$ in advance, as explained in Section 5.1.1.

For RU27 we set a target in the direction to the coast, and run our path planning algorithms for the minimal distance to the target problem, starting at the current glider location. That way we were able to move forward to the coastline despite of the strong ocean currents, for the forecast slot of time. For such sort of field experiments, the target was not defined as a waypoint, but rather as a meridian —to pass—, which seemed to be a more appropriate predicate for the objective function under those circumstances.

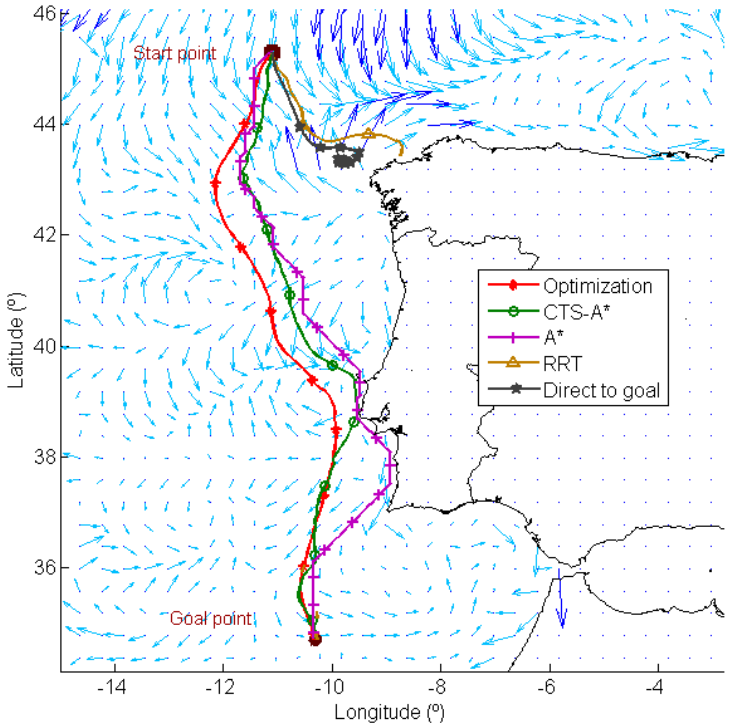


Figure 5.21: Comparison of the Minimal Time Path for the DtG, RRT, A*, CTS-A* with $M = 18$ samples, and Iterative Optimization, for the case $\# 3$, from the second set of test cases. The glider speed was set to $U_g = 0.4\text{m/s}$. The distance traveled was $d = 1178.9\text{km}$. The figure shows the path over the ocean currents $\longrightarrow$ (ocean currents that exceed the glider speed U_g are highlighted $\longrightarrow$) from the start ● to the target waypoint ◎. The travel time t_t (in days, or NA for No Arrival) for each method was: DtG: NA; RRT: NA; A*: 30.9; CTS-A*: 30.9; Iterative Optimization: 30.1.

5.5.1 Experiments Setup

In the experiments shown here, we have used target waypoints, for a total of 65 test cases generated over the ESEOAT and ESEOCAN regions. We use the 3 days of hourly forecast maps, so we run our path planning algorithms on dynamic ocean currents. We assume a constant glider horizontal speed of $U_g = \{0.2, 0.4\}\text{m/s}$ and a stint duration of $t_s = 8\text{h}$ between consecutive surfacing points. The travel time t_t required as an input for the minimal distance to the target problem is set to the forecast slot of time, i.e. $t_t = 3$ days. Consequently, since $t_s = 8\text{h}$, the number of surfacing points n is also constant and known *a priori*. In this case it yields $n = t_t/t_s = 72/8 = 9$, i.e. we only have to find 9 bearing angles to set at each surfacing point. This is what the path planning methods must find. In the particular case of the Optimization method, the iterative part is no longer required, because the number of bearings —the objective function variables— is already known. As explained in Section 4.6.1, the Optimization method becomes more simple and efficient for this problem hence.

Figure 5.22 illustrates the high dynamic ocean currents in the region used for the experiments. This figure shows several snapshots of the trajectory of the path found with the Optimization method. We can appreciate the curved trajectory of the stints, which is a consequence of the dynamic variability of the ocean currents. Therefore, apart from the comparison of the techniques for the minimal distance to the target problem, here we

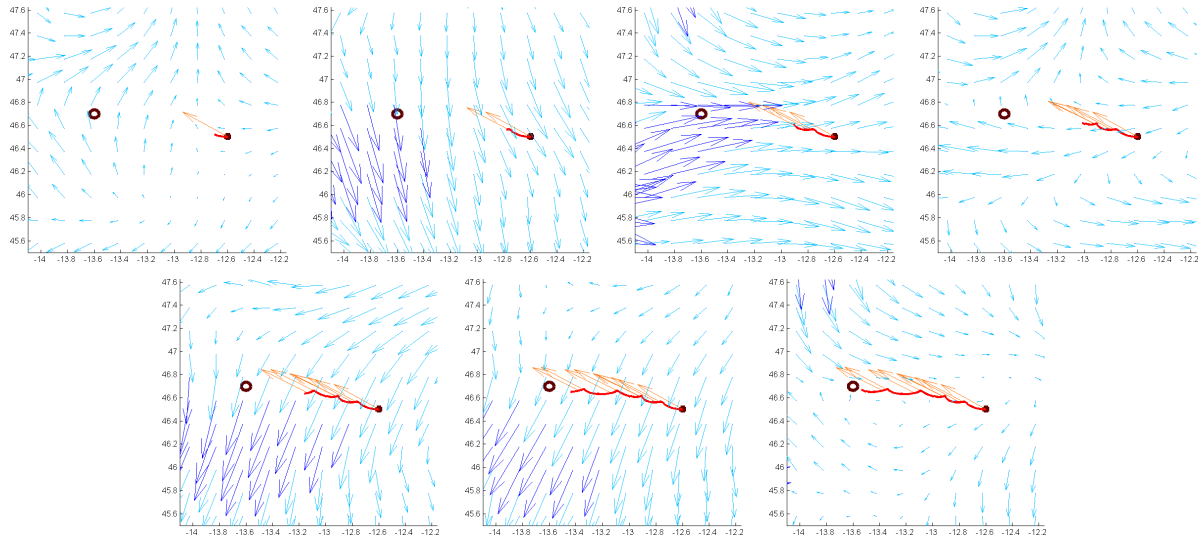


Figure 5.22: These 7 snapshots show the trajectory (red line) of the path found with the Optimization algorithm for a 3-day period of time. We are solving the minimal distance to the target problem using dynamic ocean currents, provided as hourly maps for the ESEOAT region. The start waypoint is marked with a square, while the target waypoint is drawn as a circle. The figure shows the path over the ocean currents → (ocean currents that exceed the glider speed U_g are highlighted →) from the start ● to the target waypoint ○. Finally, the glider bearing to set at each surfacing point is shown with orange arrows →.

show that our techniques are suitable for time varying ocean currents, and we evaluate them under such kind of environment. The comparison includes all the methods discussed in this thesis —DtG, A*, CTS-A*, and Optimization—, with the important exception of RRT. Recall that RRT algorithm is not suitable for dynamic ocean currents, and therefore is not included in the analysis. In brief, the RRT search can be expanded only from the starting point on a time-dependent ocean current field (see Section 4.5). Similarly, in the case of A* algorithm, the explored vertices in the search graph can no longer be discarded during the exploration, due to the temporal variation of ocean currents. As a consequence, the cost function, which depends on the ocean currents, varies as well, but the vertices are no re-explored anyway. Assuming this limitation, we have run A* in this scenario, but we cannot ensure the optimality of the result, since it does not reflect that fact. Additionally, if we had considered the time component in the graph, as another dimension, the search space would have grown significantly, and the exploration would have taken an unreasonable amount of time.

Regarding the configuration of the algorithms used in the comparison, we have used the same equivalent discretization level for each method, when applicable. In fact, the spatial grid for A* and CTS-A* has $1/20$ degrees of resolution. And CTS-A* uses a division of 20° in the bearing rose. For the Optimization method, no iterative scheme is required in this problem, since the number of bearings is known to be $n = 9$ in advanced. We use the default optimization kernel, as indicated in Section 4.6.

5.5.2 Results

Table 5.8 shows the average distance to the target difference with respect to the DtG, for the whole set of test cases. In all of them, the Optimization method gets better results, being able to find paths that end closer to the target waypoint. In order to obtain more insight on the results of the experiments, we have divided them into two subsets. According to the strength and direction —with respect to the straight direction from the start to the target waypoint— of ocean currents, we distinguish strong and weak ocean current cases. Hence,

Table 5.8: Average distance to the target difference (in km) with respect to the DtG approach; positive is better. Value for the 65 test cases run for the minimal distance to the target problem.

Method	All	Strong	Weak
A*	0.5	4.9	-6.0
CTS-A*	5.2	8.6	0.2
Optimization	8.5	12.4	2.7

the strong test cases are those in which ocean currents are against the direction to the target waypoint and are greater in magnitude than the glider speed U_g; meanwhile, weak cases are the rest —i.e. ocean currents are slower than U_g, or aligned with the direction to the target waypoint. As expected, the improvement on the path cost is always greater for the strong cases, for all the algorithms with respect to the DtG method. We can also conclude that, for the short-term test cases run, the improvement is high.

5.5.3 Discussion

According with the results summarized in Table 5.8 we also see that the best results are always given by the Optimization method, followed by the novel CTS-A*, with $\approx 25\%$ less of improvement. Finally, the lowest improvement is obtained with A*. But the results of A* required some clarification. Recall that the discretization of the search space and, above all, the discretization of the bearing angles —in multiples of $45°$ for the 8 neighborhood— justifies that in certain cases the optimum is not found, or even no path is found. This happens because of the discretization, so we can alleviate this problem by increasing the spatial resolution or by considering neighborhood radius $r > 1$ —that would yields a greater bearing resolution $< 45°$. However, that would also increase considerably the computing time. According to the results, the best method for this problem is the Optimization one.

In Figure 5.23 two illustrative examples are presented. Unfortunately, only a static ocean current map (the last one snapshot) can be showed in this printed document; the video sequence of the path simulation along with the ocean currents allows to identify clearly the correlation between the trajectory and the ocean currents direction. Being short-term test cases, most of the results obtained in our experiments yield a path cost only a few kilometers closer to the target waypoint, when compared to the DtG, or even among all the algorithms. Although, the Optimization is always the best, there are some test cases where we indeed have great differences between the final trajectories and path costs (see Figure 5.24). This is mainly due to the influence of strong currents and their temporal variability. Therefore, these techniques are more convenient in such environments. Those situations are common with bad weather, or close to the coastline.

5.6 Obstacle Avoidance

So far, we have evaluated the two basic problems for glider path planning. Now that we know that our techniques perform correctly on those problems, we proceed to analyze the influence of the presence of obstacles on the path cost. The previous experiments were done on almost obstacle free environments. Here, the test cases include obstacles close to the start or target waypoint, or even between them —in their line of sight trajectory. It is important to recall that in the ocean we identify two main types of obstacles, regarding glider path planning. On one hand, we have land regions —usually inflated—, and also zones of shallow depths, where the glider cannot navigate safety. On the other hand, strong ocean currents in the opposite direction to the target waypoint can be seen as obstacles as well, especially when they are considered as static ocean currents. Although most common ocean glider missions are generally done offshore, there is a

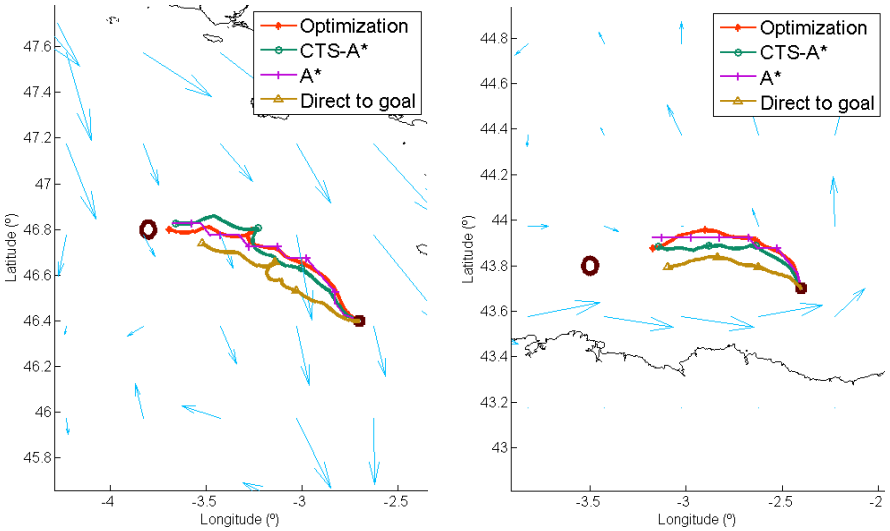


Figure 5.23: Snapshot of two different test cases run with $U_g = 0.4\text{m/s}$ using 3-day forecast maps in the ESEOAT region. In the left figure the distance from the start to the target waypoint is 95.3km. The result, which is the remaining distance to reach the target, is: Optimization $= 8.4\text{km}$; CTS-A* $= 11.2\text{km}$; A* $= 9.9\text{km}$; DtG $= 22.5\text{km}$. For the right figure the distance is 89.3km, and the results are: Optimization $= 27.7\text{km}$; CTS-A* $= 29.9\text{km}$; A* $= 29.6\text{km}$; DtG: $= 32.8\text{km}$.

good number of applications near the coast (see Section 3.2), where the path planning techniques must avoid obstacles efficiently.

Some path planning techniques already integrate the logic for obstacle avoidance, like graph-based method —i.e. A* and CTS-A* in this thesis. Others, on the contrary, required some sort of adaptation or special initialization. That is precisely the case of our Optimization approach —including its Iterative flavor. In Section 4.6.2 it is explained the initialization procedure for the Optimization method. In this section we include both the method with and without the initialization phase, in order to show that it is actually required —i.e. without the initialization phase, the method does not find any solution in many cases, when there are obstacles in the environment. Additionally, we can evaluate the impact of the initialization phase on the overall computing time of the algorithm. We extend the computing time comparison to the rest of methods as well.

Both the minimal time path and the minimal distance to the target problems are covered in this section. For each, we have run new test cases, that now include obstacles in the form of land areas or strong and opposite ocean currents. Depending on the problem we have used a different setup and environment. Although the details are explained in the next sections, we distinguish two types of scenarios. The first ones is the short-term path planning with static ocean currents, for the minimal time path problem. The second one is the long-term path planning with dynamic ocean currents, for the minimal distance to the target problem. Finally, we have also run the experiments for fast $U_g = 0.4\text{m/s}$ and slow $U_g = 0.2\text{m/s}$ glider speeds, which is equivalent to navigating weak and strong ocean currents, respectively.

5.6.1 Comparison in static long-term scenarios

Following the order of presentation of the previous problems, we will start by analyzing our techniques for the minimal time path problem with obstacles. We consider —as it is common in this type of problem— static ocean currents and long-term scenarios. In this kind of scenario, with static ocean currents, we can

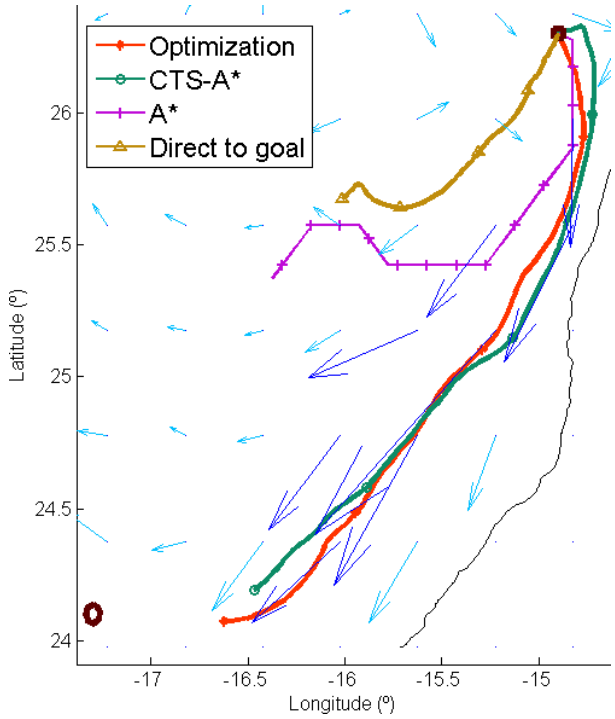


Figure 5.24: Snapshot for a test case of 4 days; the last forecast is repeated to fill unavailable ones. The figure shows the path over the ocean currents ⟶ (ocean currents that exceed the glider speed $U_g = 0.4\mathrm{m/s}$ are highlighted ⟶) from the start ● to the target waypoint ○. The distance from the start to the target waypoint is $344.6\mathrm{km}$. The remaining distance to reach the target is, for each algorithm: Optimization $= 68.9\mathrm{km}$; CTS-A* $= 85.1\mathrm{km}$; A* $= 169.4\mathrm{km}$; DtG $= 217.6\mathrm{km}$.

compare all the methods discussed in this thesis, i.e. DtG, RRT, A*, CTS-A*, and the Iterative Optimization approach. For the Iterative Optimization we also analyze the version with the Initialization phase, specifically designed to deal with obstacles, termed Iterative Optimization-CTS-A* (see Section 4.6.2).

5.6.1.1 Experiments Setup

We have run a set of test cases in the ESEOAT and ESEOCAN regions. For weak ocean currents, using $U_g = 0.4\mathrm{m/s}$ we have run 70 test cases. Meanwhile, for strong ocean currents, with $U_g = 0.2\mathrm{m/s}$ we have run 50 test cases. The stint duration is the typical $t_s = 8\mathrm{h}$. We have divided the test cases in two different situations, which have been analyzed separately. The first set of cases corresponds to coastal scenarios, while the second includes only offshore, obstacle free scenarios. For the actual definition of scenarios with obstacles, we simply consider those test cases in which the DtG approach does not find a solution, or it is very bad. Hence, in the results we will see almost a 100% rate of NA (Not Arrival) for the DtG with obstacles.

Regarding the algorithms' configuration used for the experiments, we have used the same equivalent discretization level for each method, when applicable. In fact, the spatial grid for A* and CTS-A* has $1/20$ degrees of resolution. Additionally, CTS-A* uses a division of $20°$ in the bearing rose (see Figure 5.25). The Iterative Optimization-CTS-A* approach considers 6 angles in each node expansion for the initialization phase, which are:

1: One keeping the bearing of the previous stint.

Table 5.9: Average travel time (in days d+hours h) for the minimal time problem in static long-term scenarios with and without obstacles. The table summarizes the 70 test cases (28 with obstacles and 42 obstacle free) run with a glider speed $U_g = 0.4\mathrm{m/s}$. When a technique does not find a solution for some test cases, we include the percentage of NA (No Arrival) cases.

Method	With obstacles	Obstacle free
DtG	NA in 96%	13d + 21h
RRT	NA in 71%	13d + 07h
A*	11d + 15h	12d + 18h
CTS-A*	11d + 10h	12d + 10h
Iterative Optimization	NA in 50%	12d + 05h
Iterative Optimization-CTS-A*	11d + 09h	12d + 00h

Table 5.10: Average travel time (in days d + hours h) for the minimal time problem in static long-term scenarios with and without obstacles. The table summarizes the 50 test cases (35 with obstacles and 15 obstacle free) run with a glider speed $U_g = 0.2\mathrm{m/s}$. When a technique does not find a solution for some test cases, we include the percentage of NA (No Arrival) cases.

Method	With obstacles	Obstacle free
DtG	NA in 94%	32d + 08h
RRT	NA in 69%	27d + 15h
A*	NA in 17%	23d + 17h
CTS-A*	28d + 06h	23d + 02h
Iterative Optimization	NA in 57%	22d + 14h
Iterative Optimization-CTS-A*	NA in 9%	21d + 21h

Table 5.11: Effective speed of the glider in km/day (standard deviation) for the minimal time problem in static long-term scenarios; all test cases —with and without obstacles— are considered together. The table summarizes all the test cases run with a glider speed $U_g = 0.4\mathrm{m/s}$ and $U_g = 0.2\mathrm{m/s}$.

Method	Weak, $U_g = 0.4\mathrm{m/s}$	Strong, $U_g = 0.2\mathrm{m/s}$
DtG	28.2 (10.0)	14.9 (9.0)
RRT	29.2 (10.0)	16.8 (10.0)
A*	29.7 (10.0)	16.3 (9.0)
CTS-A*	30.5 (10.0)	17.0 (9.0)
Iterative Optimization	31.5 (11.0)	18.0 (10.0)
Iterative Optimization-CTS-A*	31.6 (11.0)	18.6 (10.0)

2, 3: Deviations of $\pm 22.5°$,

4, 5: and $\pm 45°$, with respect to **1**.

6: And the straight direction to the target waypoint, from the current location.

Table 5.12: Average computing time (in seconds) for the minimal time problem in static long-term scenarios; all test cases —with and without obstacles— are considered together. The table summarizes all the test cases run with a glider speed $U_g = 0.4\text{m/s}$ and $U_g = 0.2\text{m/s}$.

Method	Weak, $U_g = 0.4\text{m/s}$	Strong, $U_g = 0.2\text{m/s}$
DtG	0.1	0.2
RRT	1.0	3.0
A*	20.0	75.0
CTS-A*	44.0	80.0
Iterative Optimization	18.0	489.0
Iterative Optimization-CTS-A*	38.0	167.0

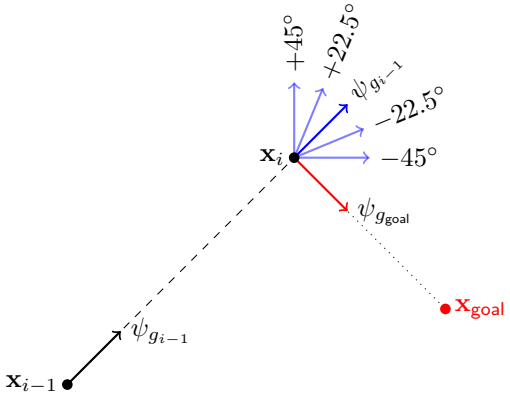


Figure 5.25: Diagram of the angles considered for expansion in the initialization phase of the Optimization-CTS-A* method, for the coarse version of the CTS-A* used.

5.6.1.2 Results

Table 5.9 shows the average travel time to reach the target waypoint. In some test cases, several algorithms did not find a solution. Therefore, we have included a last column that shows the average travel time of obstacle free cases, in the sense that the DtG approach finds a solution. This table summarizes the results using a glider speed of $U_g = 0.4\text{m/s}$. In this case, the average distance traveled by the glider has been 650km. With these results, under weak ocean currents, we observe some interesting conclusions. First, the DtG does not arrive to the target waypoint in any case with obstacles, in our experiments. Only the graph-based methods, A* and CTS-A*, and the Iterative Optimization-CTS-A* arrive always to the target. The RRT fails more than the 70% of the test cases, while the Iterative Optimization —without the initialization phase— fails on the 50%. Clearly, this shows that the initialization phase is really required in these scenarios, for our optimization-based method. Also, it seems to work well, since it has been able to reach the target waypoint for all the test cases, and also yields the best path —i.e. the lowest travel time—, as expected accordingly with the results of Section 5.4.

However, we want to evaluate the algorithms on more adverse conditions. Therefore, we have repeat the experiments but with a lower glider speed of $U_g = 0.2\text{m/s}$, which is equivalent to navigating stronger ocean currents. The results are summarized in Table 5.10). We see that some techniques that were able to reach the target with $U_g = 0.4\text{m/s}$, now do not. In particular, RRT and Iterative Optimization fail in more than a 60% of the cases. In sum, the Iterative Optimization-CTS-A* produces the best results for weak and

strong currents, so the experiments show it is a valid approach to deal with obstacles using our optimization-based method. Finally, to facilitate the comparison, Table 5.11 collects the average and standard deviation of the glider effective speed, which is obtained for the path found with each method. It reflects the same conclusions of the travel time, but in a probably more normalized way. In fact, we see that the improvement on the effective speed is higher with weaker ocean currents ($U_g = 0.4\mathrm{m/s}$), rather than with strong ones ($U_g = 0.2\mathrm{m/s}$).

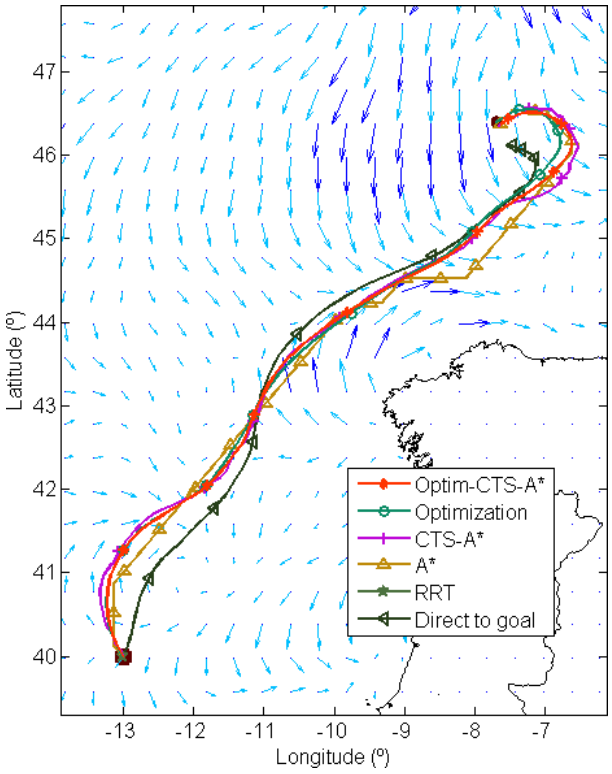


Figure 5.26: Comparison of the Minimal Time Path for the Obstacle Avoidance problem. Comparison of the methods for a test case run with a glider speed of $U_g = 0.4\mathrm{m/s}$ (weak ocean currents). The distance from the start to the target waypoint is $d = 831\mathrm{km}$. The figure shows the path over the ocean currents ⟶ (ocean currents that exceed the glider speed $U_g = 0.4\mathrm{m/s}$ are highlighted ⟶) from the start ● to the target waypoint ○. The travel time (in days d + hours h, or NA for No Arrival) for each technique is: Iterative Optimization-CTS-A*: $26\mathrm{d} + 21\mathrm{h}$; Iterative Optimization: $27\mathrm{d} + 10\mathrm{h}$; CTS-A*: $27\mathrm{d} + 18\mathrm{h}$; A*: $28\mathrm{d} + 19\mathrm{h}$; RRT: NA; DtG: NA.

Finally, Table 5.12 shows the computing time for each method, measured on a Intel® Core™-i7 CPU 2630 QM running at $2\mathrm{GHz}$ with $8\mathrm{GiB}$ of RAM. For $U_g = 0.4\mathrm{m/s}$, the results reflect that the Iterative Optimization-CTS-A* method requires more computing time, because of the intelligent initialization phase; but it is still fast to compute. Apart from the DtG and RRT approaches, the Iterative Optimization method takes less computing time for $U_g = 0.4\mathrm{m/s}$. A* requires almost the same computing time, but recall it does not integrate the unconstrained motion model. The CTS-A* takes more than 2 times to find a path, but it is still a reasonable time.

With strong ocean currents, i.e. for $U_g = 0.2\mathrm{m/s}$, we observe that the computing time of the Iterative Optimization method without the initialization phase degrades when it cannot find a solution. The larger computing time is associated with the fact that it has spent many time trying to find a path without success.

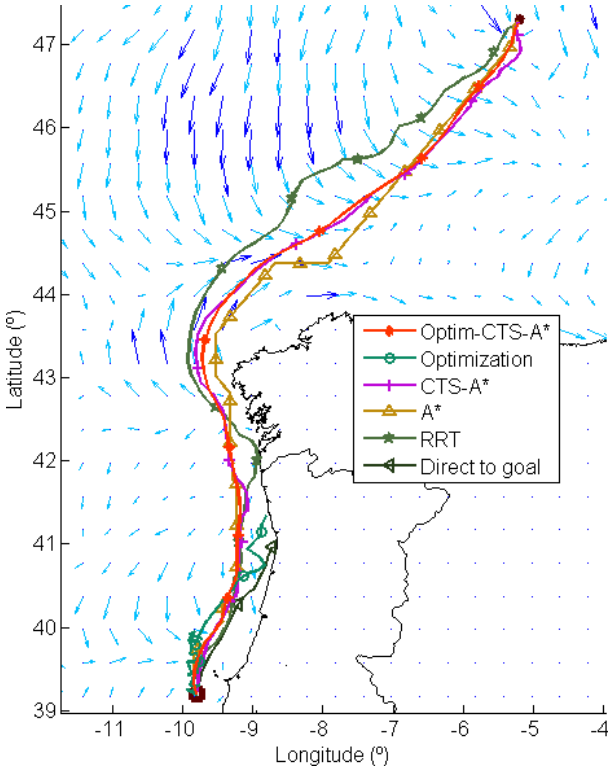


Figure 5.27: Comparison of the Minimal Time Path for the Obstacle Avoidance problem. Comparison of the methods for a test case near the coast, run with a glider speed of $U_g = 0.4\mathrm{m/s}$ (weak ocean currents). The distance from the start to the target waypoint is $d = 974\mathrm{km}$. The figure shows the path over the ocean currents $\longrightarrow$ (ocean currents that exceed the glider speed $U_g = 0.4\mathrm{m/s}$ are highlighted $\longrightarrow$) from the start ● to the target waypoint ◎ . The travel time (in days d + hours h, or NA for No Arrival) for each technique is: Iterative Optimization-CTS-A*: $26\mathrm{d} + 7\mathrm{h}$; Iterative Optimization: NA; CTS-A*: $27\mathrm{d} + 4\mathrm{h}$; A*: $27\mathrm{d} + 7\mathrm{h}$; RRT: $31\mathrm{d} + 7\mathrm{h}$; DtG: NA.

With the intelligent initialization phase the computing time is smaller. However, in this type of environments the graph-based methods take less computing time. A* and CTS-A* use the same resolution for the gridded search graph, with weak and strong ocean currents. Therefore, the growth in computing time only depends on the number of nodes explored, which increases moderately with strong ocean currents. On the contrary, the Iterative Optimization methods have to optimize more variables with strong ocean currents, because the number of bearings $|\mathcal{B}|$ required to reach the target waypoint $\mathbf{x}_{\text{goal}}$ is greater. Figure 5.26 and 5.27 show two of the cases included in the experiments. Both figures show experiments run with $U_g = 0.4\mathrm{m/s}$, i.e. weak ocean currents. However, the second one is closer to the coastline, so the path planning algorithm must deal adequately with the land obstacles. Indeed, in that case the Iterative Optimization method fails.

5.6.1.3 Discussion

In conclusion, the Iterative Optimization without the initialization phase gets trap in the obstacles for many test cases. On the contrary, the Iterative Optimization-CTS-A* version, which includes the initialization phase, is able to find a path to the target waypoint for almost all the test cases. A* and CTS-A* obtain worse results because of their discretization nature. In some cases they are not able advance in the exploration

search because the strong currents keep them from expanding new nodes. The solution to this consist on increasing the resolution of the search grid, at the expense of increasing the computing time of the algorithm.

Regarding the computing time, graph-based method scale better in the presence of obstacles, because they search in the same gridded graph. On the contrary, the Iterative Optimization methods requires more time with strong ocean currents, i.e. with $U_g = 0.2\mathrm{m/s}$, because the number of bearings $|\mathcal{B}|$ increases. Anyway, with the intelligent initialization phase, the computing time is still acceptable.

5.6.2 Comparison in dynamic short-term scenarios

Now that we have analyzed obstacle avoidance scenarios for the minimal time path problem, we proceed to do the same for the minimal distance to the target problem. In this case, we consider dynamic ocean currents instead of static ones. We use ROM forecast maps for the ESEOCAN region for short-term scenarios. The election of ESEOCAN region exclusively relies on the fact that it covers the Canary Islands, where the islands themselves constitute a good number of representative land obstacles for the test cases. Now, the algorithms evaluated are DtG, A*, CTS-A*, Optimization and Optimization-CTS-A*. Recall that RRT cannot be run in dynamic ocean currents (see Section 4.5). Regarding the optimization-based methods, note that for the minimal distance to the target problem, the travel time is set in advance. Consequently, the number of surfacing points n is known as well, for a given constant surfacing time. Therefore, we do not use the iterative version of the optimization-based approach.

Table 5.13: Difference of the remaining distance to reach the target waypoint with respect to the Optimization-CTS-A* method, for the minimal distance to the target problem. Average and standard deviation within brackets, both in km. The table summarize the 45 test cases run for a glider speed $U_g = 0.4\mathrm{m/s}$.

Method	Total	Coast	Offshore
DtG	42.4 (46.0)	67.4 (39.0)	13.6 (24.0)
A*	8.5 (18.0)	5.3 (7.0)	9.1 (6.0)
CTS-A*	5.2 (6.0)	5.8 (7.0)	6.5 (4.0)
Optimization	10.3 (21.0)	19.6 (26.0)	0.0 (0.0)

Table 5.14: Difference of the remaining distance to reach the target waypoint with respect to the Optimization-CTS-A* method, for the minimal distance to the target problem. Average and standard deviation within brackets, both in km. The table summarize the 45 test cases run for a glider speed $U_g = 0.2\mathrm{m/s}$.

Method	Total	Coast	Offshore
DtG	18.0 (29.0)	30.4 (29.0)	10.5 (27.0)
A*	13.5 (49.0)	10.5 (13.0)	15.3 (19.0)
CTS-A*	9.9 (10.0)	7.0 (7.0)	11.5 (10.0)
Optimization	6.2 (13.0)	16.3 (17.0)	0.2 (1.0)

5.6.2.1 Experiments Setup

Similar to the previous experiments, we have configured the glider speed to both $U_g = 0.2$ and $0.4\mathrm{m/s}$, and the stint duration to $t_s = 8\mathrm{h}$. The travel time have been set to 4 days using the ROM forecast maps. For this amount of time and t_s, the number of bearings to find is 12. In the case of optimization-based methods, that is the number of variables to optimize. A total of 45 test cases have been run, divided in two situations, which

again are coastal and offshore scenarios. The algorithms' parameters remain the same for the graph-based methods. Meanwhile, the Optimization-CTS-A* uses a division of 5° for the initialization phase, inserting a turning point candidate every 3 surfacing point —i.e. the equivalent to one day of navigation for our setup. Note that the initialization procedure is slightly different for the minimal distance to the target problem (see Section 4.6.2).

5.6.2.2 Results

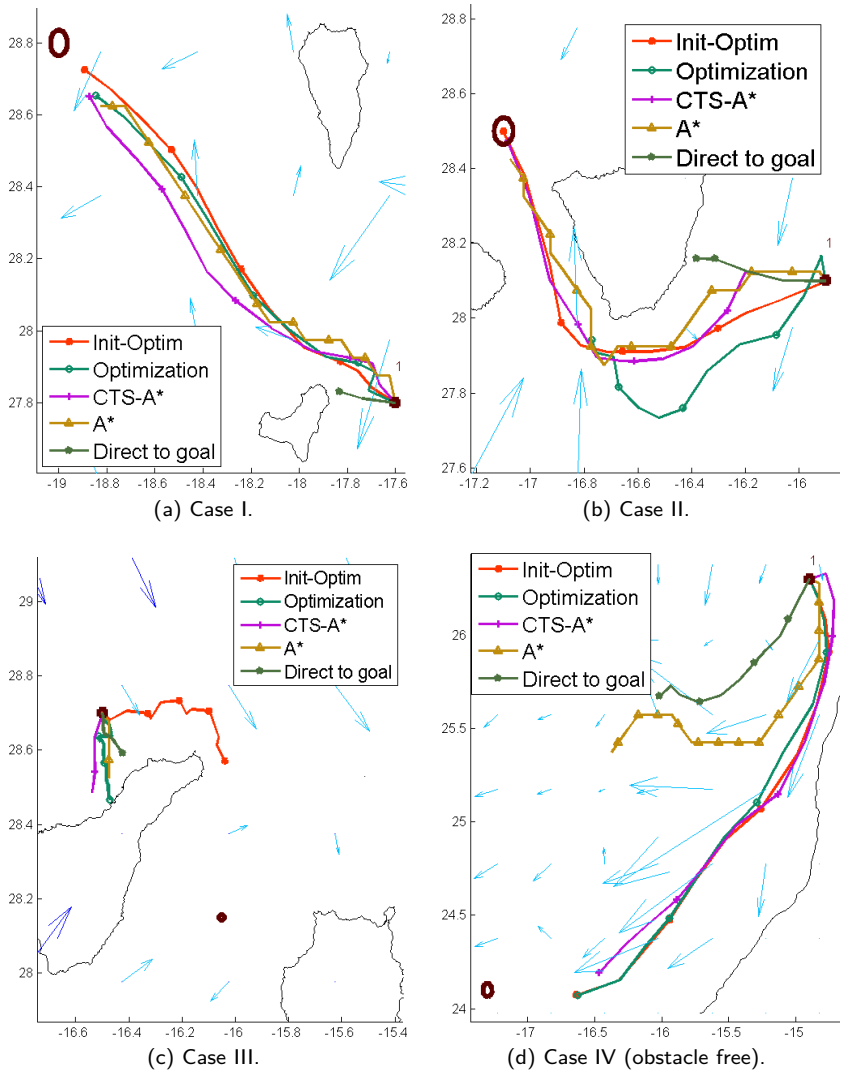
Table 5.13 shows the average and standard deviation of the difference of the remaining distance to reach the target waypoint with respect to the Optimization-CTS-A* approach, for each technique. The results are presented separately for the coast and offshore scenarios, being the average distance traveled by the glider of $120\mathrm{km}$. Note that a positive value means a worse path found, i.e. a path the finishes further than the one found by the Optimization-CTS-A* approach. A first point to analyze is the fact that in offshore scenarios, without obstacles, the path cost is the same for both optimization-based techniques, regardless of the initialization phase. This demonstrates that the initialization phase also works without obstacles correctly. It is for coastal environments where we observe that the Optimization method without the initialization phase degrades significantly. Regarding the graph-based method, both A* and CTS-A* behave similar. As we see in Section 5.5, they do not find a path as good as the Optimization-CTS-A* method, but they perform better than the Optimization method, when it does not use the initialization phase. Finally, DtG always finishes further from the target waypoint, than the other techniques, as expected.

We go a step forward by running the test cases with $U_g = 0.2\mathrm{m/s}$, which is equivalent to navigate stronger ocean currents. In Table 5.14 we have the results for these experiments, with an average distance travel of $60\mathrm{km}$, i.e. half of the other run, since U_g is also the half part. Now, almost the same conclusion as with $U_g = 0.4\mathrm{m/s}$ holds, but with some details. Now, the Optimization method is slightly worse than the Optimization-CTS-A* because with strong currents the initialization phase becomes more important. Consequently, the Optimization-CTS-A* approaches deals with obstacles under dynamic ocean currents better than other techniques, according with our experiments.

5.6.2.3 Discussion

Figure 5.28 shows 4 test cases of the experiments run so far. In (a)-(c) we have test cases near to coast zones —actually some islands. Meanwhile, in (d) we have and offshore —obstacle free— test case, for comparison purposes. The distance required to reach the target waypoint after the 4 days is shown for each method. It must be noted that the currents vary on time and only the last snapshot of them is shown in the figure. In the obstacle free case, we observe that the result of the Optimization method and Optimization-CTS-A* method are almost the same. This means that the initial guess find by the initialization phase is almost the same as the one use without initialization in this kind of scenarios. Therefore, it is valid for any kind of scenario: with or without obstacles. For the other 3 cases we observe how some methods get trapped because they cannot avoid the obstacles. This happens always for the DtG approach. It also happens with the Optimization method without initialization, as in the figures (b) and (c). Indeed, for (c) only the Optimization-CTS-A* is able to find a path to avoid the obstacle. In this case, the graph-based methods fail as a consequence of the search grid resolution, which should be increased in order to avoid the situation in which the current node cannot expand any neighbor, as it was also explained in the previous section.

Finally, as with static long-term scenarios, we have analyzed the influence of some algorithm parameters. As in the previous section, we have reduced the division of the bearing rose for the Optimization-CTS-A* methods, and duplicate the search grid of A*, obtaining similar results.



(a) Case I. (b) Case II.

(c) Case III. (d) Case IV (obstacle free).

Figure 5.28: Comparison of the Minimal Distance to the Target problem for the Obstacle Avoidance problem. Comparison of 4 test cases running at $U_g = 0.4\text{m/s}$. The remaining distance to the target waypoint is (SL if Stop in Land): (a) Optimization-CTS-A* $= 13.3\text{km}$, Optimization $= 22.1\text{km}$, CTS-A* $= 20.6\text{km}$, A* $= 25.9\text{km}$, DtG $= 157.1\text{km}$ (SL); (b) Optimization-CTS-A* $= 0.0\text{km}$, Optimization $= 69.7\text{km}$ (SL), CTS-A* $= 3.2\text{km}$, A* $= 8.7\text{km}$, DtG $= 80.0\text{km}$ (SL); (c) Optimization-CTS-A* $= 46.7\text{km}$, Optimization $= 54.2\text{km}$ (SL), CTS-A* $= 60.7\text{km}$ (SL), A* $= 58.9\text{km}$ (SL), DtG $= 61.3\text{km}$ (SL); (d) Optimization-CTS-A* $= 67.4\text{km}$, Optimization $= 68.8\text{km}$, CTS-A* $= 85.1\text{km}$, A* $= 169.4\text{km}$, DtG $= 217.6\text{km}$. Respectively, the total distance from the start to the target waypoint is: 176.5km, 125.8km, 75.2km, 343.4km.

5.7 Path Following

This and the next sections address problems that, in brief, impose constraints on the basic problems studied thus far. In fact, this first one here requires the glider to follow a predefined path. In other words, we impose

a constraint such that the surfacing points must lie as close as possible to the desired path. In the most simple case, such path is a simple line from the start to the target waypoint. So far, we have tried to find the minimal time path or the minimal distance to the target path. Now, for any of those problems —with or without obstacles—, we additionally require to minimize the distance from each surfacing point to that line. Clearly, this constitutes another objective function to be minimized. What we do is basically combine the objective function of any of the previous problems —e.g. the minimal time cost function— with this new conflicting objective function. Hence, the path following is a multiobjective optimization problem, as it was explained in Section 3.3. The path following problem can also be modeled as a constraint that forces the path to be within a certain bounds with respect to each side of the given path. As we will see in the sequel, we can configure that bounds in order to relax the path following constraint. As a consequence, we can have a set of solutions for this problems depending on a relaxation weight or factor. This constitutes a Pareto curve, that we will also analyze in this section.

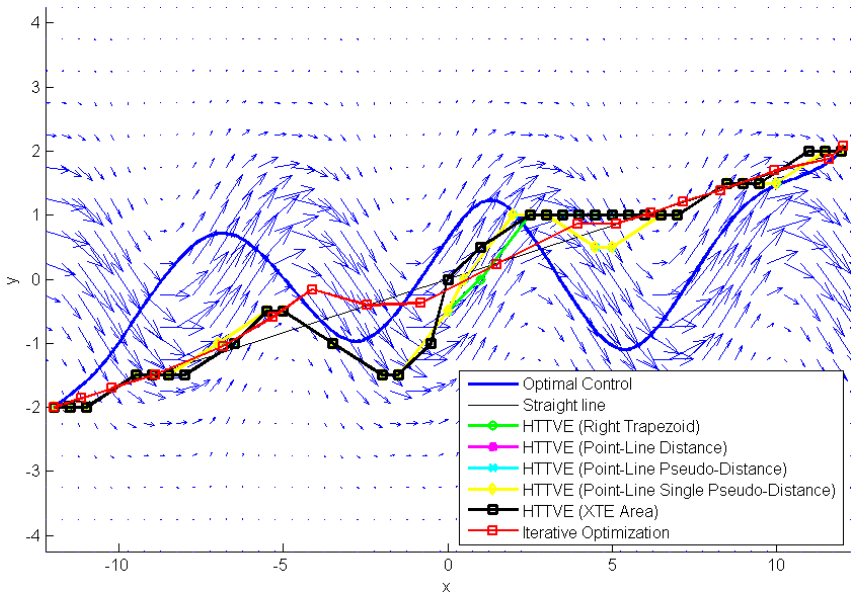


Figure 5.29: Comparison of the HTTVE and Iterative Optimization methods applied to the Path Following problem. For HTTVE, several approximations are used to compute the area between the desired and found path; these approximations are described in Section 3.3. The optimal control solution, using Zermelo equation, is also included. It shows the minimal time path, which is the objective function set for the problem, along with the path following constraint; which are conflicting objectives. An straight line from the start to the target waypoint represents the desired path to follow.

As it was explained in Section 3.3, we also refer to this problem as hold track, because it was originally based on the work of Eichhorn (2010). In that paper, the Hold Track Time-Varying Environment (HTTVE) algorithm is used to solve this problem. It computes a function based on the area between the desired path and the one found, which is applied during the search. This algorithm is based on the Time-Varying Environment (TVE) algorithm of the same author, proposed in a previous paper (Eichhorn, 2009). In general, it is equivalent to our A* implementation since it bases on a modification of Dijkstra's algorithm, which turns to be equivalent to A* without the heuristic function. Indeed, it is trivial to include the heuristic in the algorithm, as the author shows on (Eichhorn, 2010). Although we have the implementation of these variants, here we use the basic HTTVE algorithm for the sake of clarity, and because we are interested in the result, rather than on the computing time, which is what the variants are concerned of.

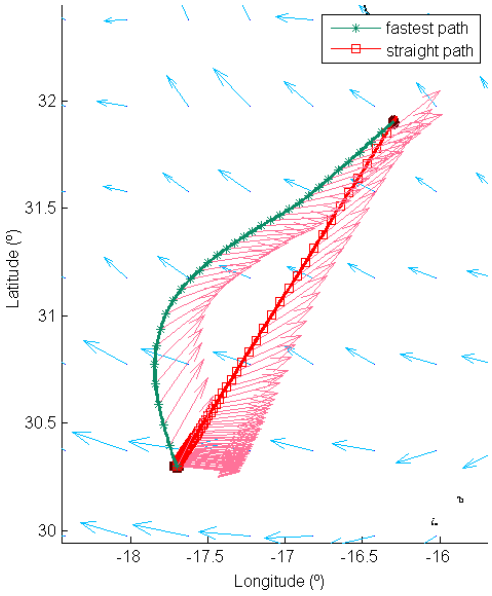


Figure 5.30: Path following problem solved using the Iterative Optimization, for a straight line. The path found is compared with the minimal time path, obtained with the Iterative Optimization method as well. Along with the path, the figure shows the glider bearings → at each surfacing point, the ocean currents →, and the start ■ and target waypoint ○. The distance traveled is $d = 222\mathrm{km}$.

5.7.1 Evaluation of Path Following Constraints

Regarding the computation of the area between the two paths, there are actually several ways to compute it. We can compute the area between them, as in (Eichhorn, 2010), using the XTE area, which is almost equivalent the area of a right trapezoid. Or we can take some approximation of the area, based on different point to line equations, as the ones that has already been shown in Section 3.3. Also note that the area is computed in a piecewise manner, for each sub-region created every time one path crosses the other. Figure 5.29 shows a comparison of several techniques to compute the area between the desired and found path, embedded in the HTTVE algorithm. It also includes the optimal control path, which gives the minimal time path using Zermelo equation. Finally, we compare the result of the graph-based method with the Iterative Optimization approach, which has been adapted to include a constraint such that the path found must be at most some distance apart from the desire one. This is some kind of corridor constraint defined over the path line given, for which we can setup the corridor width. Clearly, we have a trade-off between the path following constraint and the objective function, which in this case is the minimal time. In the figure we observe that the different techniques to compute the area within the HTTVE algorithm yield almost the same result —they overlap. Therefore, it is safe to use some line-to-point approximation if we need to reduce the computing time.

The path following problem does not impose a strong constraint on holding the vehicle to navigate on the desired path, which in most cases is actually impossible because of the ocean currents. On the contrary, we look for the minimum separation from such desired path. In fact, if we compare the results with the optimal control path, we see that the minimal time path does not follow the desired path at all. The ocean currents have been generated with a synthetic streamfunction (see Section 5.1.2), where the currents are favorable —so the optimal control can find the optimal path—, but with a sinusoidal oscillation that makes the best path to be out of the desired straight line path. According with the results, although all the variants produce the same path for the HTTVE, with the Iterative Optimization method we have an alternative path, but of equivalent cost, as the figure shows. For both techniques, we have been able to reduce the separation for

the desired path, and still reach the target waypoint. However, the travel time is increased, since the two objectives are in conflict. In our Iterative Optimization, we can adjust the bound that defines the corridor around the desired path, so we can generate a set of results of different travel time and separation from the desired path. Later we will see that these pairs of solutions define a Pareto curve.

5.7.2 Experiments Setup

We have run several tests in the ESEOCAN region for the path following problem using the Iterative Optimization method. We have started with simple cases, in which we navigate weak ocean currents and just try to follow a path defined as a line, as Figure 5.30 shows. In this example, the maximum deviation from the straight line is $170\mathrm{m}$; note that we are traveling several hundreds of kilometers, so we are almost navigating through the desired line. As it was explained at the initial chapters of this dissertation, there is special interest on data sampling in the ocean modeling field. In most data sampling or monitoring applications, we must follow a given path —a line or a concatenation of them—, rather than finding the minimal time path, in order to obtain samples along a certain transect the crosses some area of interest. For this reason, here we focus on a particular case, which consist on crossing mesoscale eddies.

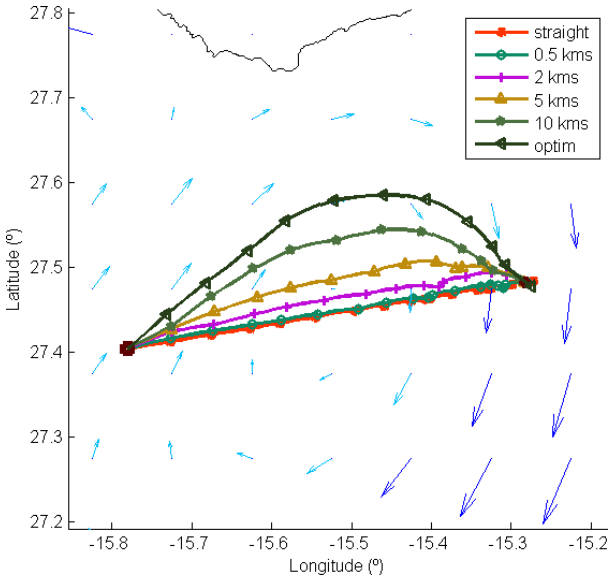


Figure 5.31: Comparison of paths found with a glider traveling at $U_g = 0.4\mathrm{m/s}$ to cross an eddy from one side to another, for the path following problem. The desired path is defined by a straight line. The Iterative Optimization method have been run with a path following constraint that establishes the bound —i.e. maximum distance/separation— around the desired path. Such bound has been set to 0 —i.e. strictly follow the path— and relaxed by increasingly different amounts from $0.2\mathrm{km}$ to $10\mathrm{km}$. The minimal time path is also included, in which no constraint is imposed, which actually corresponds for the case in which an infinite bound is allowed, i.e. the unbounded, unconstrained optimization. In the figure, the surfacing points are marked along the paths. The ocean currents $\longrightarrow$ (ocean currents that exceed the glider speed U_g are highlighted $\longrightarrow$) and the land coastline are shown as well.

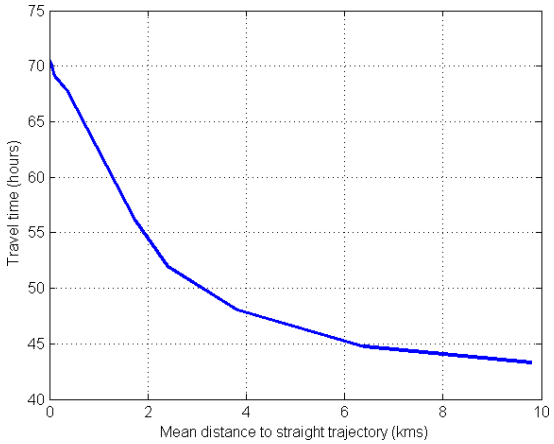


Figure 5.32: Pareto curve that shows the travel time vs. the average distance to the desired path, for the paths found in Figure 5.31. The Pareto curve plots the two conflicting objectives against each other, i.e. the travel time (y-axis) vs. average distance to the desired path (x-axis) —which is the straight line in this example. All paths try to keep the vehicle close to the straight line, but this constraint is relaxed in each simulation by means of increasing the size of a corridor around such line. As the figure shows, the tighter the corridor, the higher the travel time. Meanwhile, as the corridor size grows, we approach to the minimal time path of the unconstrained optimization.

5.7.3 Results and Discussion

We have run several experiments to study the behavior of the Iterative Optimization algorithm in a cross-eddy missions. Crossing an eddy consist basically in following a path from a starting waypoint that finishes at a target waypoint that lies in the opposite side of the eddy. It is important to highlight the fact that only with such type of linear path crossing the structure, we are able to characterize the eddy size, centroid and boundaries properly. Thus far, in Figure 5.30, the path following problem was solved by forcing the path to be the closest to the straight, desired line path. On the contrary, in Figure 5.31 we allow some separation from the desired path, given as a bound around that path, which defined some sort of corridor. The desired path is $57\mathrm{km}$ long, from the start to the target waypoint, passing across the centroid of the eddy. For each run, we have considered stints of $t_s = 4\mathrm{h}$ with a period of $15\mathrm{min}$ drifting at the surface, for communications and self-localization; the reason for taking into account the surfacing period is because these tests have been done to simulate some field experiments. In the first case, with a null bound, the travel time is $79\mathrm{h}$ hours and the largest distance to the straight line is $80\mathrm{m}$ —almost none. In the second case, in which the constraint was relaxed in order to allow paths inside a corridor of $1\mathrm{km}$ on each side of the straight line, the travel time was reduced by $11\mathrm{h}$. The next cases produce paths of better minimal time, but within wider corridors, until no bound is set for the classical minimal time path problem. If we plot the travel time vs. the maximum distance to the desired path, for each of the paths found in these experiments, we obtain the Pareto curve of Figure 5.32. This curve shows all the non-dominated solutions for the each corridor configuration set up in the algorithm, for the test case. These results are very interesting, because we can pick out a solution from the Pareto curve, according with some design criteria, depending on the trade-off between the minimal time path and path following problems. For our experiments, it is always better to set a small corridor, rather than the strict constraint to follow the path, because in such cases the travel time is excessively long. Therefore, it is better to accept a small deviation from the desired path.

In some test cases, the effect of strong currents —i.e. those that exceed the glider speed U_g— makes impossible to keep the desired path, as it can be observed in Figure 5.33. At the very beginning of the path

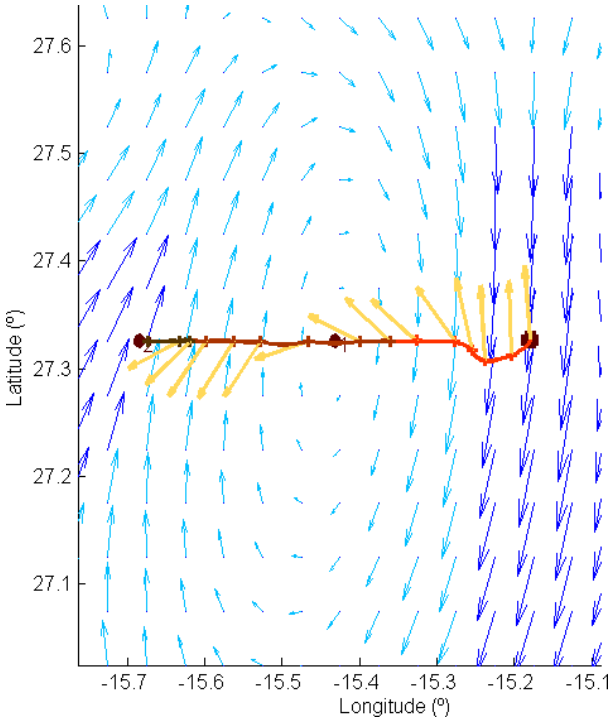


Figure 5.33: Path following problem applied to cross a mesoscale eddy under strong ocean currents. The desired path to cross the eddy goes from one side to another crossing its centroid, as described by an straight line. The glider speed is $U_g = 1\mathrm{km/h}$, and the strong ocean currents that exceed U_g are highlighted →. The bearings at each surfacing point → are shown, along with the ocean currents → from the start ■ to the target waypoint ○.

we observe a significant deviation from the straight line. However, the overall result is acceptable, since the Iterative Optimization method manage to find a path that in general follows the desired path closely. From this result for a single transect crossing the eddy we can build a composition of several desired lines to define any trajectory, of any length. For instance, in Figure 5.34 a journey across several predefined waypoints is shown. The whole trajectory is a piecewise composition of straight lines that cross the eddy multiple times from different angles. We obtain a windmill trajectory that will allow to characterize the eddy centroid, boundary and size adequately in a real field mission. Dynamic, evolving eddies will be addressed later, in an *ad hoc* manner. Similar applications can be found in Oceanography, when we need to collect data from a particular known trajectory, not only a certain point. The HTTVE, or our Iterative Optimization method, can be applied in such scenarios to solve the path following problem. Furthermore, we have shown that we can relax the path following constraint, so we have a Pareto curve with several paths/solutions. Each of them, with a different trade-off between the path following constraint and the minimal time path —or the minimal distance to the target, if we had chosen that other basic problem for optimization.

5.8 Multi-Glider Path Planning and Coordination

So far, we have tried to find optimal paths for a single glider using different objective functions —e.g. the minimal time path, the minimal distance to a target waypoint, etc. In many oceanographic tasks, however, a large region must be covered —e.g. sampling exhaustively such region for data assimilation. For such

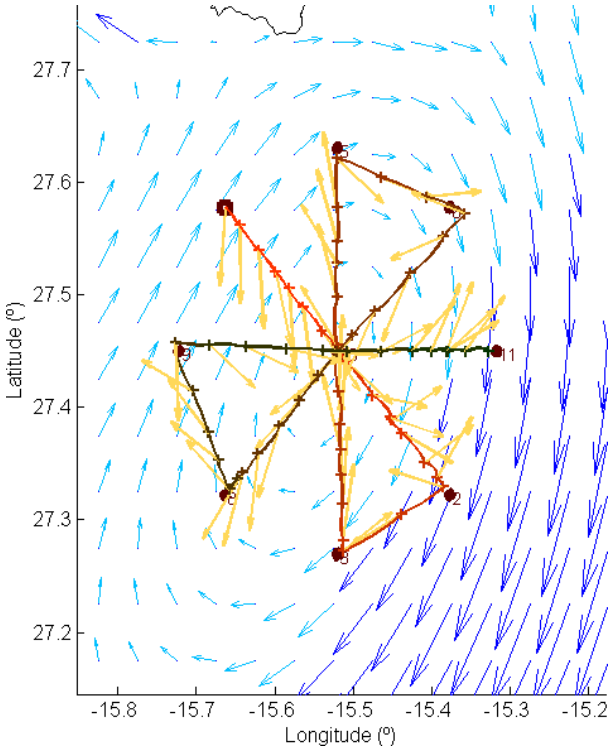


Figure 5.34: Assemble of path following problem to perform a mesoscale eddy sampling and characterization mission. The windmill trajectory is decomposed in straight line paths that cross the eddy multiple times from different directions, for which the path following problem is solved using the Iterative Optimization method. The glider bearings ⟶ at each surfacing point, along with the ocean currents ⟶ from the start ■ to the target waypoint ○ are shown.

problems, it is always better to employ multiple vehicles, rather than a single one. With a fleet of multiple gliders exploring the same region, arises the possibility to apply efficient methods to do so. For instance, we can try to optimize the time required to cover the whole area with a fleet of gliders, or similarly, maximize the area explored after a given period of time. Additionally, we can coordinate the paths of several gliders to navigate in formation.

In most cases, the exploration of a region is basically a succession of several path following problems, in which the vehicle performs a lawnmower trajectory obtained with a Boustrophedon Cell Decomposition (BCD) (Choset and Pignon, 1997). Once this coverage path planning route is obtained, we just have to follow the concatenation of line paths that form it. For that we can use the algorithms evaluated in Section 5.7 for the path following problem. It is reasonable to extend this approach to multiple vehicles just by imposing some constraint to make them navigate in formation, i.e. maintaining a certain distance among them. This is precisely the problem that we evaluate here. Given a fleet of N gliders, we try to find a path that optimizes certain objective function, while a constraint is satisfied at every point in the path.

5.8.1 Experiments Setup

For the objective function, we use the one of the minimal time path problem (see Section 3.3 and 5.7), using static ocean currents for long-term scenarios, so the evaluation is more representative. Regarding the

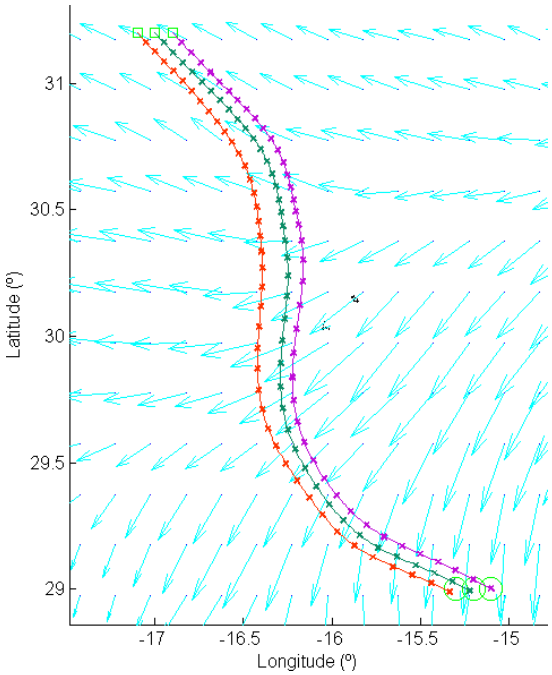


Figure 5.35: Multi-glider path planning, coordinating the navigation of a fleet of 3 gliders. The vehicles are constrained to travel within a distance range of $[5, 15]\mathrm{km}$ with respect to the neighbor gliders. A gap of $0.1°(\approx 11.1195\mathrm{km})$ is set among them at the start □ and target ○ waypoints, which for this test case are separated a distance $d = 299\mathrm{km}$.

multi-glider problem, we consider a simple but flexible problem. It consists on navigating a fleet of gliders in parallel. Other kinds of formation can be plug into the path planning algorithm easily. In fact, we only have to implement such constraint —the rest remains the same—, if we use the Iterative Optimization method, which is the one used for the experiments below. This constraint can be configured to establish the minimum and maximum separation among the fleet of gliders. As it was previously said, this solution can be applied to the coverage path planning problem when we have multiple vehicles, although we can also use others schemes. Indeed, later we will see another example of multi-glider coordination to solve the problem of efficient recovery. In brief, we want to reduce the time required to recover a fleet of gliders spread on the ocean, using a ship. Note that all these problems have already been explained in more detailed in Section 3.4.

5.8.2 Navigation in Parallel Formation

Regarding the coordination problem of navigating a fleet of vehicles, we have run several experiments on the ESEOAT region, for a fleet of 3 gliders (see Figure 5.35 and 5.36). During the optimization process, the distance range —$[5, 15]\mathrm{km}$ in both figures— is checked for each surfacing point, and also any intersection among trajectories are forbidden —even underwater. In sum, the Iterative Optimization method finds the minimal time path for each glider, subject to such constraint, but taking into account the whole fleet of vehicles altogether, for a single optimization run. Consequently, the figures illustrate that the paths not only satisfy the constraint, but also minimize the travel time.

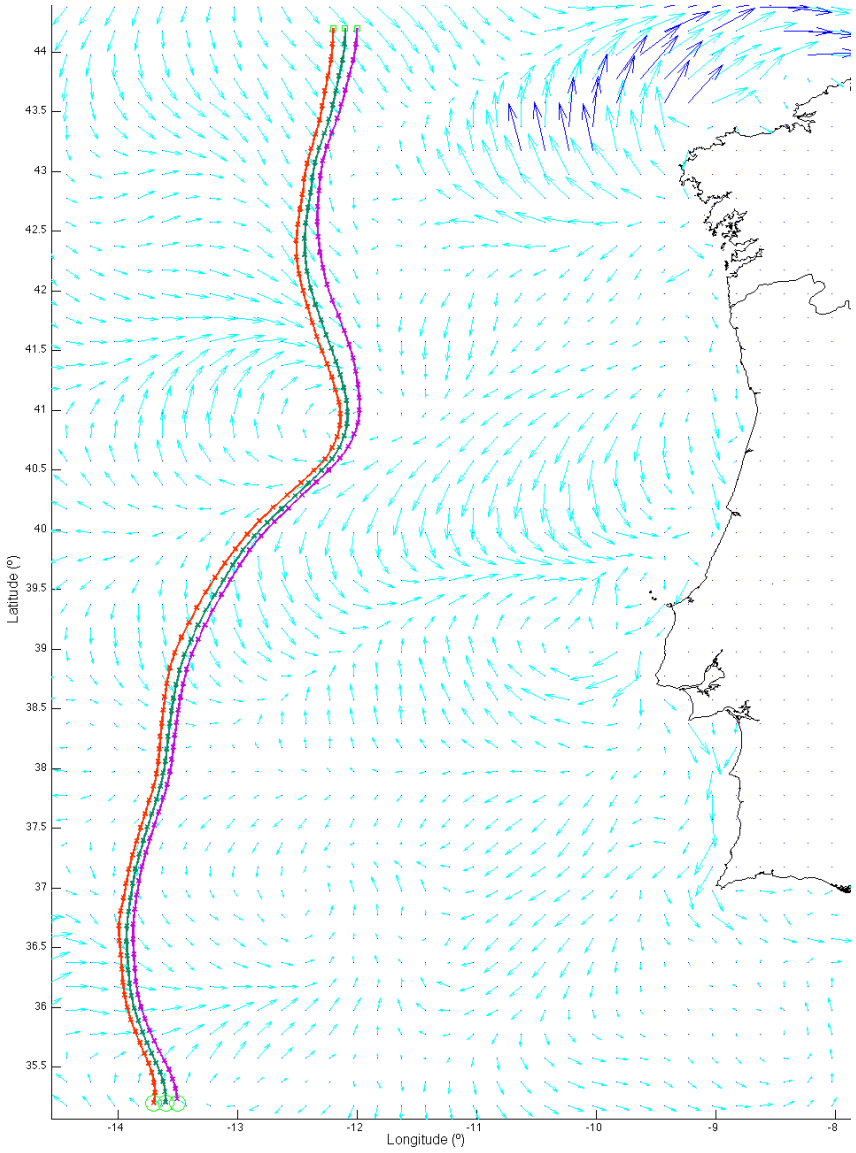
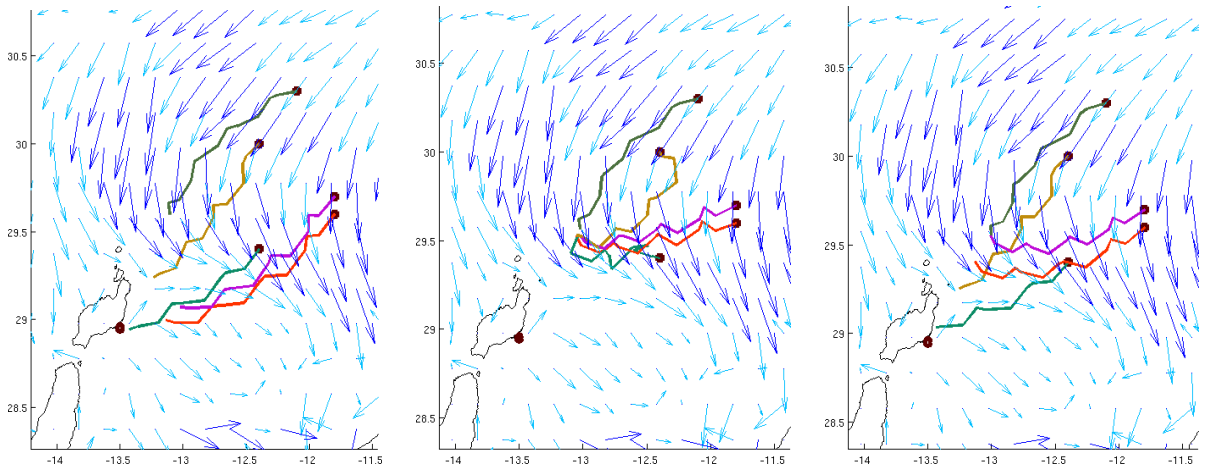


Figure 5.36: Multi-glider path planning, coordinating the navigation of a fleet of 3 gliders. The vehicles are constrained to travel within a distance range of $[5, 15]\mathrm{km}$ with respect to the neighbor gliders. A gap of $0.1°$ is set among them at the start □ and target ○ waypoints, which for this test case are separated a distance $d = 1008\mathrm{km}$.

5.8.3 Efficient Recovery

Another example of multi-glider path planning and coordination is the problem of efficient recovery. In brief, a ship must recover a fleet of N gliders in the minimal time. Initially, the gliders are spread in the ocean, and the ship is at a certain point —usually at the port. We try to find the path that leaves the gliders in the best location, so the ship can travel less time, and consequently we can save ship time and oil.

For this problem we have run several experiments in the ESEOCAN region, using dynamic ocean currents



(a) Individual. (b) Meet together. (c) Ship path.

Figure 5.37: Multi-glider path planning and coordination for a fleet of 5 gliders that have to be recovered from the sea by a ship in the minimum time. This efficient recovery problem is solved with the Optimization method, while it tries to solve the minimal distance to the target problem for a period of 4 days. The glider travels at $U_g = 1\text{km/h}$, with a surfacing time of $t_s = 8\text{h}$. In (a), we have the individual path planning solution. On the contrary, the other two figures show the results using multi-glider path planning. In (b), all the gliders try to meet at a point for the recovery. Finally, in (c), the ship path is included in the optimization process. The ship has to travel: (a) 192.0km; (b) 167.5km; (c) 165.4km, respectively.

for short-term missions. We use the Optmization method to solve the minimal distance to the target problem, where the target is usually the ship location or the port. However, with a fleet of gliders, we impose some constraints to leave the gliders at the best location, in terms of minimizing the recovery time. Note that the recovery time is assumed to be proportional to the recovery distance, since the ship navigates much faster than the ocean currents, and therefore these are neglected in the ship motion model (see Section 3.4.1 for more details on the efficient recovery problem).

In our experiments we have set the glider speed to $U_g = 1\text{km/h} \approx 0.2778\text{m/s}$, with a surfacing time of $t_s = 8\text{h}$. We consider dynamic ocean currents for short-term missions. This way, we have tested both static and dynamic, long-term and short-term, and the minimal time path and minimal distance to the target problems, with multiple gliders. The ROM forecast maps of ESEOCAN region are used to obtain the dynamic ocean currents, for a period of 4 days. The ship is assumed to go for the recovery after that time, so we are searching for the paths that leave all the gliders in the best location at the end of the mission, reducing the recovery time then.

5.8.3.1 Results and Discussion

In Figure 5.37 we have three different results for the same test case with a fleet of 5 gliders. First, in (a), we have the solution without multi-glider path planning. Each glider on its own tries to end at the closest point to the target waypoint, which is set at the ship/port location. The drawback of this approach is that some gliders are still far, and will take much to to go and recover it. With this approach, the cost of the recovery is dominated by the glider that finishes further from the ship/port. Therefore, the ship must travel 192.0km.

With a multi-glider path planning algorithm, we can proceed in two different ways, to alleviate such limitation. First, as shown in Figure 5.37 (b), all the gliders try to finish at a meeting point after the whole mission —4 days in the example. The meeting point does not have to be set. The Optimization method

simply imposes a constraint that forces all the gliders to finish close to each other. With this approach, the ship goes to the meeting point and there it can recovers all the gliders. This approach has several advantages over the individual approach. The ship only has to go to the meeting point, recover the gliders and come back to the port. Formerly, it had to decide in which order to go for each glider individually, which takes more time. Furthermore, now there is a recovery point —i.e. the meeting point—, which is closer to the ship/port than the finish location of the glider that would have end further from it, using the previous approach. In fact, in this case the ship only has to travel $167.5\mathrm{km}$, much less than before.

The second approach consists on including the recovery path that the ship will actually perform to recover the gliders from their finish locations. By doing so, we are actually minimizing the recovery time. In this case, the gliders do not finish at a meeting point (see Figure 5.37 (c) for instance), but even though, they are in the best location, in terms of recovery time. Now the ship only has to travel $165.4\mathrm{km}$, which is the minimum so far, of all the approaches. Anyway, the improvement with respect to the meeting point approach is not very big. Therefore, in conclusion, any of the two multi-glider path planning strategies is better than individual path planning to solve the efficient recovery problem, which is a real application for ocean glider missions.

5.9 Tracking Evolving Ocean Structures

Previously, we have mentioned the considerable importance of sampling ocean processes (see Section 1.1). Most of the phenomena in the oceans are dynamic and span throughout relatively large regions. The techniques discussed in this thesis manage dynamic ocean currents, so they allow us to move and sample at a desired location with great expectation. However, when we need to sample large evolving structures, we usually have to sample certain zones of such structure, not only a point.

We have already seen, in Section 5.7, that by solving the path following problem we can sample along a desired path. Indeed, we have shown a few examples where we applied this approach to characterize mesoscale eddies. However, in a more general case, it is not enough to sample this kind of evolving structures by simply crossing them with a set of desired paths defined as lines. Since the whole structure is moving, we actually have to track/follow it as a whole. Furthermore, during that tracking task, we must sample certain areas of the structure, and we can set out the sampling strategy in many different ways. For instance, one might track and sample the centroid of the structure exclusively; or, on the contrary, take samples from its boundary. For large ocean structures the problem is not trivial, precisely because of their size, when we must sample a given zone of the evolving structure.

5.9.1 Experiments Setup

Here, we focus on the most common evolving structure that appears on the ocean currents velocity field: the ocean eddies. The ocean is populated by eddies of different size and lifespan. Probably, mesoscale eddies are the most interesting ones in Oceanography, because of their tight relationship with many biological, physical and chemical processes in the sea (see Section 1.1). They have a size of several tens to hundreds of kilometers, and a lifespan that goes from weeks to a few months. For our experiments, we have designed the method explained in Section 4.8, which is based on a discretized version of the eddy model proposed by Jimenez et al. (2008). In brief, we divide the volume of the eddy structure into several sectors rotating at different velocities. Each sector is defined by its minimum and maximum limits in depth, the radius and angle dimensions. Note how this description of the structure considers the depth, contrary to the path following approach discussed thus far.

In Section 4.8, the three phases of the method used for this experiment are explained. At this point, it is important to recall that the second phase of the algorithm uses an objective function that is defined in terms of the eddy sectors that should be sampled and the desired mission duration. This scheme allows to define a great number of sampling strategies, such as tracking the centroid of the eddy, navigating its border, or

maximize the sampled volume, focus in a particular depth, and so on, as well as time-dependent formulations. A Genetic Algorithm has been used to obtain the path that meets those specifications.

5.9.2 Results and Discussion

We have run several test cases for the Canary Islands eddy system using the forecast maps of MyOcean-IBI. Our only intention is to validate the approach. In Figure 5.38, a 5 days trajectory optimized to track the eddy border is shown. Note that we are using the dynamic ocean currents of the 5 days forecast provided by the ROM. Since the figure only shows the snapshot at the end of the mission, we only see that at the end the location of the glider is effectively at the boundary of the eddy. The rest of the path is actually tracking that border, by considering the sectors of the eddy model, and the dynamic ocean currents. Therefore, the path found tries to follow the eddy boundary as much as possible at every instant of time during the whole mission. It seems clear, that using a path following strategy it would be difficult to define the path found by the Genetic Algorithm to solved this problem; it is not a simple straight line, and the resulting curve is only known after the optimization process, by considering the temporal evolution of the structure as a whole. Therefore, this technique is not only more flexible, but it is also required to follow certain regions of evolving ocean structures, like eddies. Finally, although the majority of evolving structures in the ocean are eddies, our approach is also applicable to other type of structures, with the same model or a similar one adapted to the particular structure to track.

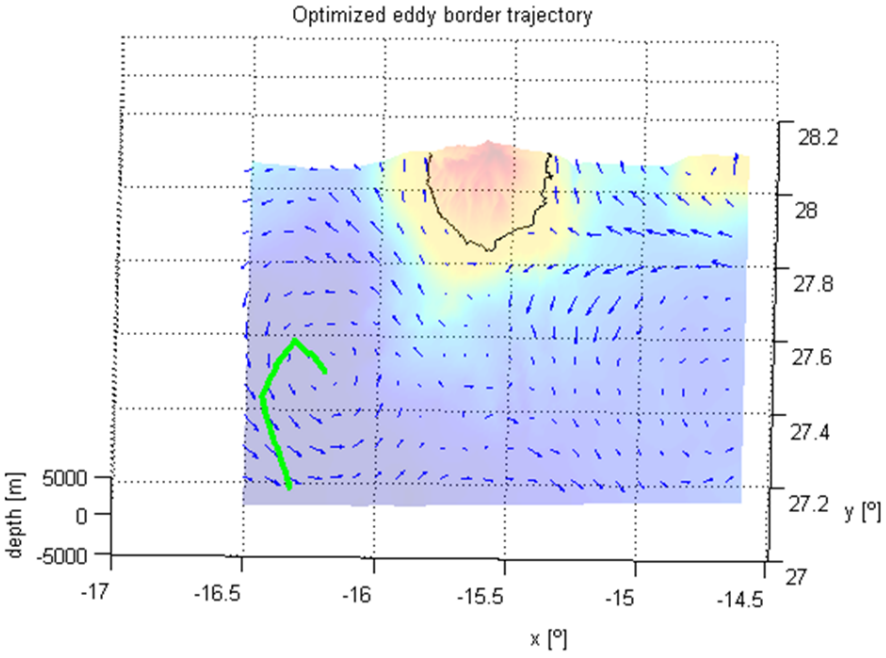


Figure 5.38: Example in which the boundary of a mesoscale eddy is tracked using our Genetic Algorithm and discretized eddy model approach. The figure shows a snapshot of the ocean currents and the optimized trajectory at the end of the mission; note that the eddy moves while it is being sampled.

In conclusion, the method evaluated here constitutes a valid alternative that can contribute to extend the glider operational capabilities to track evolving ocean structures of large size, such as mesoscale eddies. Obviously, the results of this method are dependent on the prediction accuracy of the ROM used. However,

in our opinion, is still a reasonable option when a try and error scheme is not admissible for a field mission. Indeed, the most suitable real applications are *one-shot type*, where a maximization of the success probabilities is highly desirable —e.g. due to budget/time restrictions. Furthermore, this approach is more flexible than aggregating several path following path, since we consider the whole structure in three dimensions, and model its shape, as in the case here for the mesoscale eddies.

Chapter 6

Conclusions

The present work has been conceived to validate the following hypothesis

> Novel Path Planning algorithms designed for the problem of Autonomous Underwater Gliders (AUGs) navigation bring the possibility of computing (sub-)optimal paths automatically, and hence improve, simplify and, in some cases, enable missions in zones with strong ocean currents that exhibit a complex distribution or a high temporal variability.

The main developing lines are organized in the analysis of the glider path planning context, the discussion of the motion models, the identification of the set of representative problems, and the proposal of path planning algorithms.

6.1 Topic Analysis

Path planning plays an important role in glider navigation (Davis et al., 2009) as a consequence of the special motion characteristics these vehicles exhibit. First of all, gliders have a slow nominal surge speed U_g, which is a consequence of the buoyancy-driven navigation system explained thus far (see Section 2.1). This confers gliders a large autonomy at the expense of traveling relatively slow with respect to ocean currents. In fact, they sometimes face ocean currents U_c stronger than U_g, which may head against them. A global path planner is required to provide a way to avoid these troublesome scenarios and reach the target waypoint in the minimal time or optimizing any other parameter of the mission.

A second fundamental feature of ocean gliders is inherent to almost all AUVs. They cannot localize themselves accurately underwater, unless very expensive systems are deployed. The GPS signal does not penetrate into the water, so there is no way to geo-localize the vehicle while it navigates underwater doing the yo-yo profiles. The dead reckoning estimates the vehicle pose underwater, but the drift produced by the ocean currents is not observable by the low power consuming navigation sensors on board. Path planning algorithms can reproduce such dead reckoning by integrating the appropriate motion model, which does include the ocean currents obtained with ROMs. This way we can predict the next surfacing point $\mathbf{x}_{i+1}$, and therefore command the vehicle to head in the most appropriate direction to reach the target waypoint $\mathbf{x}_{\text{goal}}$.

Ocean gliders surface at regular intervals of time t_s, which are typically around 2 to 8 hours. Only when they are on surface they can geo-localize and correct their bearing ψ_g in order to head to $\mathbf{x}_{\text{goal}}$. The fact that the vehicle is only commanded every surfacing time t_s, is a key feature in the design of glider path planners. Consequently, contrary to ground robots or UAVs, which can be commanded at high rates, ocean gliders set their bearings at very low rates —i.e every surfacing time t_s. This is a third key feature that drives the design of glider path planning techniques, since they must reflect this very low bearing commanding rate. In fact,

the novel approaches presented in this thesis include this aspect of the glider motion model inside the path planner.

Another particularity of glider path planning relies on the special characteristics of the ocean. Contrary to ground mobile robots, the glider motion model can be significantly affected by the ocean currents. Apart from this fact, it is worth noting that ocean currents form a dynamic and asymmetric vector field. On one hand, they change over time and define a variable cost map, in the path planning sense. On the other hand, the cost of traveling from one location $\mathbf{x}_i$ to another $\mathbf{x}_{i+1}$ is asymmetric, i.e. $c(\mathbf{x}_i, \mathbf{x}_{i+1}) \neq c(\mathbf{x}_{i+1}, \mathbf{x}_i)$. We have relaxed the first factor and considered static and dynamic ocean currents separately, in this thesis. This allows us to obtain a comprehensive analysis of the problem, and to evaluate the techniques in the simplified case of static ocean currents. The second factor cannot be simplified, but we have considered different glider vs. ocean currents speed ratios SR in order to analyze the behavior of the algorithms under weak and strong ocean currents. Indeed, it is with strong ocean currents where glider path planning becomes more useful, since that is the typical scenario where naive approaches like the DtG and manual piloting fail.

Another feature that is remarkable in glider path planning is the uncertainty, generated by from two main sources. One —that we have already mentioned— is the uncertainty in the glider localization underwater. The other is the uncertainty of the maps computed by the Ocean Models, which comes from measurement errors in data assimilation, or incorrect modeling of ocean dynamics and processes. This produces a tightly coupled uncertainty in the glider motion model, since its uncertainty may come from the ocean model. Within the path planning algorithm, the localization is solved by means of such glider motion model, so the result inherits the uncertainty of the ocean currents provided by the ROM. Furthermore, it is also very difficult to obtain precise values for the motion model parameters. As a consequence, the accuracy in the estimation of the next surfacing point degrades. Indeed, in Section 2.2.2.1 we have analyzed the effect of such uncertainty in the glider nominal surge speed U_g and the heading error $\hat{\psi}_e$, by comparing the output of our unconstrained glider motion model and the surfacing points of the RU27 glider during the trans-Atlantic mission. We have observed the U_g reduction over time due to the bio-fouling, and the unpredictable behavior of $\hat{\psi}_e$ from one stint to the next. Therefore, in real missions it is important to feedback the path planner with the new surfacing point $\mathbf{x}_i$ every time the glider wakes, in order to recompute the optimal path.

Finally, in this thesis we have discussed several motion models. First, the constrained and unconstrained point motion models, which are simple approximations that model the glider as a single, holonomic particle without mass. These models can be applied in 2 dimensions —using only the surface ocean currents, or depth averages— or in 3 dimensions, which requires the glider pitch θ as an additional parameter, apart from the nominal surge speed U_g and the bearing ψ_g. In our experiments we have used both approaches depending on the data provided by the ROMs, and we have also taken into account the time spent by the glider on surface —of $\approx 15\text{min}$—, while it moves like a drifter. Furthermore, we have tested a force balance motion model (see Section 2.2.3) that incorporates the mass and buoyancy of the vehicle, along with the drag and lift forces. It requires much more parameters than the point motion models, but if correctly configured it can model the glider yo-yo profiles more accurately. However, in general we do not need such accuracy in glider path planning, because the uncertainty in the ocean model introduces higher errors in the integration of the motion model for the long time t_s between surfacing points.

6.2 Problem Identification

The problem of AUG navigation is actually a series of different problems that gliders must solve in order to be used in real applications. In the present work we have identified the main problems and applications where gliders constitute a useful platform. More specifically, we contribute with novel approaches for glider path planning, on top of the navigation control system on the vehicle. From the path planning point of view, we consider two atomic problems that are the building elements of more elaborated problems and real applications. Regarding the applications, we have put the focus on the Ocean Research demands, but we may

extend the applicability of the glider path planning techniques to other domains.

The two atomic problems address the equivalent to the Shortest Path Problem (SPP) in the scope of ocean gliders. The direct equivalent to the SPP is the minimal time path problem discussed in Section 3.1.1. The cost of traveling from one location $\mathbf{x}_i$ to another $\mathbf{x}_{i+1}$ in the ocean is not simply the distance, but the travel time that depends on the effective speed, which combines the glider nominal surge speed U_g with the ocean currents U_c. The second problem looks for the minimal distance to the target waypoint for a given period of time T, which gives a fixed number of surfacing points $|\mathcal{B}|$ (see Section 3.1.2). In most cases, the minimal time path problem is associated with long-term missions, in which we use static ocean currents because there are no forecast maps available for long periods of time in advance. Meanwhile, the minimal distance to the target problem is usually solved for short-term missions, using the dynamic ocean currents of the ROMs forecast maps. All the glider path planning techniques discussed in this thesis support both of these atomic problems, expect for the RRT that only addresses the minimal time path problem with static ocean currents. In this way we contribute a number of solutions to the basic problems of glider path planning; among them two important novel approaches.

Apart from the atomic problems, we have addressed more elaborated ones, which are tightly connected with real glider applications in oceanography. Although not strictly a new problem, we ensure that our methods avoid obstacles, so they can be used in coastal environments, where we find obstacles in the form of land areas, shipping routes, shallow waters, and —as in offshore environments— strong ocean currents against the heading to the target waypoint. Obstacle avoidance is an important requirement to have practical algorithms for real field missions, and as we have shown in Section 5.6, graph-based algorithms and optimization-based methods with the intelligent initialization phase fulfill this demand.

Three other application problems are covered by this work. First, the path following problem that tries to obtain a path $\mathcal{P}$ as close as possible to a desired path $\mathcal{P}_d$, but still arriving at the target in the minimal travel time. This constitutes a multi-objective problem, which can also be solved as an optimization with constraints problem. Indeed, we have successfully applied this latter approach using the optimization-based methods. Furthermore, we can relax the path following constraint by defining a corridor rather than a path, so we obtain a Pareto curve of non-dominated paths for each objective (see Section 5.7). We also apply graph-based methods to solve this problem with a cost function that represents the area between $\mathcal{P}$ and $\mathcal{P}_d$ —or simply a point to line/curve distance. This problem is the building block to sample large areas by means of coverage path planning. We have illustrated this problem in the particular case of eddy sampling, using a windmill pattern —made of several linear transects— that crosses an eddy from one side of its boundary to the other passing through the centroid. This coverage path is more adequate to sample and characterize these ocean structures than a lawnmower one. This happens because it is important to estimate accurately the centroid location, specially for these evolving ocean structures.

A second problem is directly concerned with tracking and sampling evolving structures, as an extension of the path following problem. We have seen thus far, that we can sample eddies by means of a coverage path and solving the path following problem for each transect it defines. However, eddies move on the ocean over time, so we must track them and decide where to maximize the sampling strategy. A three-dimensional model of the eddy is proposed, and a Genetic Algorithm looks for the best path to sample the desired regions within the eddy —defined as 3D sectors (see Section 3.5).

Finally, we address a family of problems that comprises path planning for gliders fleets. We optimize certain objective functions for several vehicles simultaneously. We illustrate the ability of our novel optimization-based approach to accommodate multi-glider problems with two particular applications. On one hand, the parallel formation of k gliders, which is of interest to obtain simultaneous measurements of large regions in data assimilation. On the other hand is the efficient recovery problem, which can reduce the cost of the recovery operations at the end of a mission. It looks for the best locations for all the gliders, so a ship has to travel the minimal distance/time to recover the whole fleet. In Section 5.8 several experiments validate our implementation, and compare different approaches for the efficient recovery problem.

6.3 Algorithms

As the thesis hypothesis claims, we have designed several novel path planning algorithms for the aforementioned glider path planning problems. We started by adapting the A* algorithm to the minimal time path problem using the constrained motion model of Section 2.2.2.2, as it was the state-of-the-art approach (Garau et al., 2005, 2009; Soulignac, 2010). However, this approach suffers several drawbacks. Although A* guarantees to find the optimal path, it only does so within the search grid. Since the glider path planning problems are clearly defined in a continuous space, with A* we cannot find the real optimum hence. Furthermore, the discretization of the grid reduces the number of bearings tested from each surfacing point $\mathbf{x}_i$ to $k\pi/4$ where $k = 0, \ldots, 7$, in the usual 8-vicinity grid. However, the main problem relies on the constrained motion model. It forces the glider to reach a given location $\mathbf{x}_{i+1}$ —of a given neighbor. Due to the effect of the ocean currents, many headings cannot be followed and consequently the algorithm fails to search many paths. Indeed, with strong ocean currents, the range of feasible heading angles may be such narrow that we could not travel to any neighbor. In those cases, the search gets trapped and it may even fail to find a single path to the target waypoint.

The grid resolution or the neighborhood radius r can be increased to alleviate this problem, but we have run A* with up to $\times 8$ resolution per dimension with a large increase in the computational cost and no significant improvement. With an $r = 3$ neighborhood, new neighbors represent new heading angles and help to alleviate the problem, but the computing time is even larger. Additionally, the cost of traveling from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$ is variable. This cost is the travel time of a stint, which according to the glider navigation must be constant, i.e. the constant time surfacing t_s. Therefore, A* does not reflect this glider behavior, and it neither integrates the unconstrained motion model while it navigates underwater doing the yo-yo profiles. For this reason, even when A* finds a solution, it must be interpreted as a rough approximation, constrained to the search grid used, that may not fit well to the real scenario. Furthermore, A* only guarantees the optimality of the path found for constant costs, but the ocean currents vector field defines a time-dependent cost map.

In order to alleviate the A* limitations, we have developed a novel approach that reflects the constant time surfacing behavior of gliders. It is based on A* search and it basically modifies the generation of successors within this graph-based algorithm. The Constant Time Surfacing A* (CTS-A*), hence, uses the unconstrained motion model of Section 2.2.2.1 in order to integrate the glider trajectory from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$ for the constant surfacing time t_s. A grid search is used like in A* to save the resulting surfacing points $\mathbf{x}_{i+1}$, so the memory usage does not grow without bounds. Now we discretize the number of bearings M to test, but we do not force the vehicle to travel in any given heading, so all the bearings can generate a new path and the algorithm does not get trapped. However, the time complexity of the CTS-A* algorithm depends on the number of bearings M expanded from each surfacing location $\mathbf{x}_i$. For this reason, we have proposed an Adaptive Bearing Sampling (ABS) strategy (see Section 4.4.3) that replaces the uniform sampling of the basic CTS-A*, using a PDF distribution. This ABS takes more bearings in the most promising directions, according to the direction to the goal and the ocean currents direction nearby $\mathbf{x}_i$. Therefore, we can keep M small, while we still find near optimal paths, so the computing time is reduced. Although we have not performed a thorough analysis of the PDF distribution $f_\Theta(\psi_e)$ parameterization, we have observed that $\sigma^2 = 10^\circ$ for $f_{\psi_d}(\psi_e)$ and a direct composition with $f_{\psi_c}(\psi_e)$ gives good results.

We have tested optimization-based methods for the minimal distance to the target problem —where the number of bearings $|\mathcal{B}|$ is known in advance— with successful results. This approach optimizes an objective function defined on the configuration space of the bearings and integrates the unconstrained motion model, so we reflect the constant time surfacing t_s and simulate the glider navigation underwater properly. The method runs fast and also finds optimal paths. Indeed, it is possible to run it during the few minutes —usually $\approx 15\text{min}$— a glider stays on surface between stints, in case of re-planning —although we usually have more time. For the minimal time path problem, since $|\mathcal{B}|$ is unknown *a priori*, we simply adopt an iterative approach. Both the Optimization and Iterative Optimization methods require an initial guess. In offshore environments it suffices with the DtG solution or simply by setting all the bearings pointing to the target waypoint. However,

in the presence of obstacles this approach fails, as shown in the results of Section 5.6. We have developed an intelligent initialization phase that uses a coarse version of the CTS-A* algorithm (see Section 4.6.2), so it runs fast and allows the optimization-based methods to avoid obstacles in coastal environments, where the strength of ocean currents is usually higher and there is a potential risk of collision.

Other techniques like the RRT have also some drawbacks. In the case of RRT, the computing time is very fast, but this type of algorithms does not look for the optimal path, but rather for a solution, which is generally non-optimal. Also, it performs bad in the presence of obstacles, including strong ocean currents against the heading to the target waypoint. Furthermore, we have observed that we must run the RRT with more than a single tree to obtain reasonable results for the glider path planning problems. For this reason it cannot be applied in the case of dynamic ocean currents, since we do not know the arrival time to the tree rooted at the target waypoint (see Section 4.5). Similarly, the manual piloting modeled with the DtG approach in this thesis (see Section 4.2) always performs worse than any of the path planning methods because it usually gets trapped within obstacles, showing the importance of them in the field of glider navigation and piloting. The DtG approach runs very fast, but it is a blind search that gets easily trapped in obstacles or relatively strong ocean currents. It also fails on obtaining the optimal path, and it is only applicable for the atomic problems. On the contrary, the CTS-A* and the (Iterative) Optimization method find the optimal path —improving the manual piloting—, and in the case of the latter the computing time is relatively fast. Furthermore, the (Iterative) Optimization method is easily adaptable for other problems, like path following, multi-glider path planning and tracking evolving features. Therefore, it simplifies many missions that rely on those path planning problems.

A complete set of experiments have been designed and conducted in order to support the thesis conclusions and validate the proposals originated from this work.

6.4 Contributions

Here we will enumerate the contributions of this thesis, and we will discuss the questions we formulated at the beginning of this dissertation to define the scope and extension of the present work (see Section 1.6). The insight obtained on such questions is indeed tightly related with the contributions, so they are discussed together in the sequel.

- The present work compiles a thorough study of the glider path planning literature, as well as path planning problems with variable costs and asymmetries. Although we have seen few contributions in this area, we establish which are the most similar problems: the Canadian Traveler Problem (CTP), dynamic traffic routing, and costmaps or path planning with costs. We include an analysis of state-of-the-art methods, their limitations and the reasoning that justifies why some of them are not applicable to glider path planning.

- Rather than a theoretical analysis of the problem and the path planning algorithms, in this work we have adopted a pragmatic study of methods applicable to glider path planning. We have adapted classical graph-based methods like A*, and probabilistic sampling techniques like RRT, following previous works from other authors (Rao and Williams, 2009).

- The minimal time path problem is unusual in field ground robotics, where the most common path planning problem is the aforementioned Shortest Path Problem (SPP). We have identified that the main difference is that the cost of traveling from one point $\mathbf{x}_i$ to another $\mathbf{x}_{i+1}$ not only depends on the distance between them $\|\mathbf{x}_{i+1} - \mathbf{x}_i\|$, but also on the vehicle speed, according with the glider motion model (see Section 2.2) and, eventually, the level of biofouling it suffers.

- A* is widely used to address the SPP, so it seems reasonable to analyze its applicability to ocean glider path planning. We have adapted A* to the minimal time path problem by incorporating the constrained

motion model of Section 2.2.2.2. This way, we find the optimal path within the gridded search graph of the algorithm.

- We have shown that although the A* algorithm can be adapted to solve the minimal time path problem, it does not reflect the glider navigation behavior. In particular, it only finds the optimal path within the search grid; the grid discretization imposes a constrained motion model that in the presence of ocean currents can lead to dead paths, because no node can be expanded from such regions. Additionally, the travel time from one location to its neighbors does not produce constant time surfacings, and the glider navigation equations are not integrated while it flights underwater. The analysis of these limitations is an important contribution of this thesis work.

- The particularities of glider path planning, that make it a hard problem, derive from the fact that the ocean currents conform a dynamic and asymmetric cost map, in the most general case. This prevents the use of many path planning techniques, because they cannot manage variable costs and they usually assume a metric cost function. This thesis highlights the importance of modeling the glider navigation faithfully by incorporating the simulation of its unconstrained motion model, including the constant time surfacings and the 3D yo-yo profiles underwater.

- We have identified several limitations on the A* and RRT algorithms, for the glider path planning problems. Therefore, we have developed several novel approaches to alleviate such drawbacks. In brief, we apply the glider unconstrained motion model and include the constant time surfacing behavior, as part of the path planning algorithm. We do this for both the CTS-A* and the different variants of the Optimization methods described in this thesis. This way, the paths found —the bearing commands— can also be used/sent in the field, in real missions.

- We have run several experiments to evaluate the feasibility, optimality and computing time of the different algorithm proposals. We have observed that almost for all test cases, the best paths are found with the Optimization method, which is also the faster in computing time. The CTS-A* also finds nearly optimal results, and the ABS algorithm allows to reduce its computing time. Finally, a rough version of the CTS-A* algorithm is used in the initialization phase of the Optimization methods in order to avoid obstacles in coastal environments.

- The results in Chapter 5 also show that our Optimization methods are easy to adapt to a good number of glider path planning problems. This comprises path following, multiple glider path planning including the navigation in parallel and the efficient recovery problem, and tracking evolving ocean structures like mesoscale eddies. Similarly, we have distinguished between long-term and short-term missions in our experiments, since they are solved with dynamic and static ocean currents, respectively. This allows for a detailed study of the algorithms under different conditions. Analogously, we separate weak and strong ocean currents, and also offshore and coastal environments, to analyze the obstacle avoidance problem.

- We have seen that the benefits of the Optimization approach proposed in this thesis rely on the particularities of gliders. It optimizes the path in the space of bearings, rather than in the space of locations. As a consequence of the constant time stints of $t_s \approx 8$ hours between surfacing points, the dimensionality of the optimization problems is reduced dramatically, which helps to keep the computing time small. For other vehicles with a higher control rate —i.e. with a smaller t_s—, the number of bearings $|\mathcal{B}|$ will increase and we will lose this advantage.

- We have used point glider motion models intensively within our path planning methods. It is a reasonable election, given the relatively large missions performed by gliders, in comparison with the accuracy that other motion models can achieve. We have also evaluated a kinematic force balance motion model (see Section 2.2.3) and compared its results with the surfacing points of the RU27 glider in the trans-Atlantic mission. As already mentioned, the higher accuracy of these models is not required for glider

path planning, they would increase the computing time of the path planner with no tangible benefit, and they require a great number of parameters —many of them very difficult to quantify.

6.5 Conference Papers and Publications

Here we enumerate and describe briefly the list of conference papers and publications derived from the development of this PhD Thesis. First, we show the list of papers about glider path planning.

OCEANS 2010: In (Fernández Perdomo et al., 2010b) we describe the novel Constant-Time Surfacing A* (CTS-A*) algorithm and compare it with our A* adaptation (see Section 4.4 and 4.3, respectively). The unconstrained and constrained point motion models of Section 2.2.2 —which are used by CTS-A* and A* respectively—, are discussed as well. Furthermore, these techniques were used to assist on the piloting of the RU27 glider at the end of the trans-Atlantic mission, using the ocean currents provided by the ROM of the ESEOO project for the ESEOAT region.

EuroCAST 2011 (I): We introduced our Iterative Optimization (see Section 4.6.1) method for the minimal time path problem in (Isern-González et al., 2011b). This work includes a comparison with the DtG approach and the graph-based methods A* and CTS-A*.

EuroCAST 2011 (II): Further comparison tests of DtG, A* and CTS-A* were presented in (Fernández-Perdomo et al., 2011b).

LNCS 2011: An extended version of the **EuroCAST 2011 (I)** paper was selected for the Lecture Notes in Computer Science (LNCS) (Isern-González et al., 2011c), in which more results are included.

ICRA 2011 (I): Our Adaptive Bearing Sampling (ABS) strategy for the CTS-A* algorithm (see Section 4.4.3) was presented in (Fernández-Perdomo et al., 2011a). The ABS strategy is explained thoroughly and compared with the uniform bearing sampling of CTS-A*, for different numbers of samples M.

ICRA 2011 (II): We address the minimal time path and minimal distance to the target problems in (Isern-González et al., 2011e). This work includes a comparison of all the techniques discussed in this thesis for such problems, being the DtG, RRT, A*, CTS-A* and Iterative Optimization.

OCEANS 2011 Santander: We address multi-glider path planning problems in another conference work (Isern-González et al., 2011d), using Iterative Optimization. The parallel navigation and efficient recovery problems are modeled with constraints and explained in the paper, as in Section 3.4. Additionally, further comparison tests of DtG, RRT, A*, CTS-A* and Iterative Optimization methods are included, for the ESEOAT and ESEOCAN regions.

OCEANS 2011 Kona: The path following —or Hold Track— problem is addressed in (Isern-González et al., 2011f). We incorporate the HTTVE algorithm of Eichhorn (2010) and the analysis of different ways to model the deviation from the desired path $\mathcal{P}_d$, using several point to line distances and the right trapezoid area (see Section 3.3). We adapt our Iterative Optimization method to this problem and show how to combine several paths in sequence in order to sample eddies with a windmill pattern. Also, we relax the path following constraint and define a corridor. As a result, we obtain a Pareto curve of non-dominated solutions for the path following and minimal time objective functions.

WAF 2011: For the minimal distance to the target problem, we improved the initialization phase of our Optimization method in order to avoid obstacles (see Section 4.6.2). Several comparison tests are run for the DtG, A*, CTS-A* and Optimization methods in this work (Isern-González et al., 2011a).

JoPha 2012: The **WAF 2011** paper was selected for publication in the JoPha journal, so more experiments were included to complement the previous results with more complex scenarios. In fact, in (Isern-González et al., 2012) we consider the ESEOCAN region, which comprises the Canary Islands. We also address the minimal time problem, with the Iterative Optimization method.

EuroCAST 2013 (I): We address the problem of tracking evolving structures in (Hernández et al., 2013a), for the particular case of mesoscale eddies (see Section 3.5.1). Here, our objective function is concerned with the optimal sampling strategy of the eddies in the ESEOCAN region.

LNCS 2013: An extended version of the **EuroCAST 2013 (I)** paper was selected for the Lecture Notes in Computer Science (LNCS) (Hernández et al., 2013b), in which more results are included.

6.5.1 Communication Papers

Also, some communication papers have been presented in the following meetings in the area:

NRC 2010: The contributions to the glider path planning and piloting of RU27 during the trans-Atlantic flight were covered in (Cabrera Gámez et al., 2010). This work comprises the graph-based methods developed in this thesis, being the A* adaptation and our novel CTS-A*, with and without the ABS strategy.

EGO 2011 (I): We contributed our results on glider path planning with optimization-based methods to the European Glider Observatory (EGO) meeting (Isern-González et al., 2011g).

EGO 2011 (II): Our contributions to the assistance in the RU27 piloting during the trans-Atlantic flight were also presented at the EGO meeting (Ramos et al., 2011a). We presented our glider path planning tools, termed *Pinzón*, and several experiments on the ESEOAT region.

EGO 2011 (III): Also part of *Pinzón*, we described our contributions to the *Cook Crossing* glider mission —being Cook a thermal SLOCUM glider— in the EGO meeting as well. These tools provide assistance to evaluate transoceanic glider missions (Ramos et al., 2011b).

6.5.2 Additional Related Research and Publications

Finally, additional research and more publications have been done in the field of Underwater Robotics, including the participation in the 2012 edition of the Student Autonomous Underwater vehicle Challenge - Europe (SAUC-E).

WAF 2009: As a result of the Degree Thesis, several contributions were compiled in (Fernández Perdomo et al., 2009) for the mission specification of underwater robots, using the CoolBOT framework of Domínguez-Brito et al. (2007) for the components of the control architecture.

JoPha 2010: The **WAF 2009** paper was selected for publication in the JoPha journal, providing more details and results (Fernández Perdomo et al., 2010a).

SAUCE 2012: As part of the participation in the SAUC-E 2012 with the AVORA team, we elaborated a paper describing the mechanics, electronics and software of the AUV developed from scratch (Mahtani et al., 2012). Also, the algorithms developed to passed the 6 tasks are explained.

EuroCAST 2013 (II): Having received the first prize in one of the categories of the SUAC-E 2012 competition, we presented our winning entry and the awarded underwater telepresence system in (Mahtani et al., 2013a).

LNCS 2013: An extended version of the **EuroCAST 2013 (II)** paper was selected for the Lecture Notes in Computer Science (LNCS) (Mahtani et al., 2013b). It includes a summary of the AUV electronics, mechanics and software system, the navigation algorithms, the missions and the techniques developed to solve them, and the telepresence system details.

It is also worth mentioning that during this period, more papers have been published in Robotics conferences, as well as internal reports and the Master Thesis entitled *Test and Evaluation of the FastSLAM Algorithm in a Mobile Robot* (Perdomo, 2009). All this work covers topics like Control Architectures, Robotic Frameworks, Underwater Robotics and Vision, SLAM, Embedded Systems, and Inertial Navigation Systems (INS).

6.6 Future Works

Further works might address three main topics: the parallelization of the techniques discussed in this thesis; a thorough evaluation of the techniques in field tests; and to explore applications in the field of data assimilation by extending the first step taken in tracking and sampling mesoscale eddies.

Regarding the algorithms parallelization, it is worth mentioning that some approaches discussed here can benefit significantly from multiple computing nodes and distributed computing technologies. Indeed, it is possible to parallelize the A* algorithm, as well as our novel CTS-A*, which will improve their computing time. Furthermore, the integration of the unconstrained motion model is susceptible of massive parallelization for each successor node generated from a discovered node. Additionally, we might explore the development of some sort of bootstrap phase to speed up the A* search phase. The extension to a 3 or 4 dimensional path planning —by means of incorporating the depth limits $[z_{\min}, z_{\max}]$, the pitch angle θ, etc., or the time dimension— would benefit from such parallelization. Also, by incorporating a three-dimensional dimensional search, we can look for the optimal path considering the physical phenomena that happen beneath the surface; for instance, the Ekman-like spiral described by the ocean currents when the depth increases (Davis, 1981), suggests to modify the depth limits or the pitch in accordance, e.g. in order to optimize the travel time. Some details can be further improved and analyzed as well, in order to reduce the computing time. In the case of the ABS strategy defined for the CTS-A* in Section 4.4.3, we could evaluate PDF distributions built in a different manner than the one proposed here, to analyze whether it is possible to reduce the number M of bearings sampled even more or not.

Most of the results presented in this work have been run in simulation. Some of them have been used to assist in the glider piloting of real missions, like the RU27 trans-Atlantic crossing. The next step should consist on running our algorithms for the surfacing points $\mathbf{x}_i$ of a real glider navigating on the ocean, and command it with the bearing angles $\mathcal{B}$ of the path found by them. This way we will be able to evaluate the accuracy of the glider motion model run by the path planner and the ocean currents provided by the ROM; recall the coupled uncertainty of both the motion and ocean models, in the estimation of the vehicle pose. More interestingly, we could make more tests for more elaborated problems, like path following and multi-glider missions. In the particular case of path following, we propose to start by characterizing the eddies that generate in the Canary Islands region. In fact, the windmill pattern shown in Section 3.5 will do the coverage path planning required to characterize the eddy adequately, by means of a succession of path following problems.

In connection with sampling mesoscale eddies, further research may explore applications in the field of data assimilation by extending the first steps taken in tracking and sampling these evolving structures. In fact, in this thesis we have proposed an approach to optimize the sampling strategy of mesoscale eddies (see Section 4.8) which constitutes a good starting point. This approach is interesting when the budget or time restrictions are such that we must guarantee a good sampling strategy, since it may not be possible to repeat the experiments in a regular basis. Obviously, the results of this method depend on the prediction accuracy of the ocean model used. However, in our opinion, it is still a reasonable option when a *try and error* scheme is not admissible. So far, we have tested the algorithm in simulation for the Canary Islands eddy system. Our

intention is to additionally test the method using maps from other eddy scenarios, such as the Balearic Sea and the southern California System. This will also help to start using gliders to monitor ocean currents, and therefore validate the output of the ocean models. Withing the research group, we are planning to execute also some real field trials as a continuation of this thesis, which may lead to further research on the topic.

Appendices

Appendix A

Ocean Models

In this thesis we have employed the output of several Ocean Models to run our glider path planning algorithms. We have generally used the ocean currents, but we can also incorporate other ocean parameters provided by these models, like the temperature, salinity and SSHA. Here we will enumerate the Ocean Models used during the development of the present work, indicating which ones are ROMs. Similarly, we describe them briefly: their spatial and temporal resolution, the number of forecast provided on advance, the products they compute, and the region they model.

Then we also discuss here how to compute distances and azimuth angles, since they are required to deal with the spherical coordinates that the Ocean Models use to represent their products. We indicate the exact metrics, and describe a series of approximations which are compared. These approximations are meant to speed up the computation within our glider path planning algorithms, and the glider motion model integration.

A.1 Numerical Ocean Models

An Ocean Model is basically a numerical model that represents the ocean behavior by means of the primitive equations of the ocean. These equations are far from the scope of this thesis. Indeed, here we use the output of these models to find optimal paths. The output is in the form of several products, which include the ocean currents velocity field that we use for path planning. From this point of view, only the features of such products are important in our problem. Therefore, some key features are: the spatial resolution, the dimensionality —bi-dimensional surface data, or three-dimensional depth data—, and the temporal resolution.

Most global ocean models have poor spatial resolution, and they provide surface daily means. On the contrary, Regional Ocean Models (ROMs) include some products that model the depth, are solved hourly, and employ a high resolution spatial grid. ROMs focus on modeling a relatively small region of the ocean, and incorporate accurate measurements to improve the model by means of data assimilation. For this reason, we use ROMs in our experiments, specially for short-term missions that required a high temporal resolution, in the order of hours.

Below we enumerate and describe briefly some Ocean Models, all of them used for our tests and assistance in glider path planning during the last years.

ESEOO Project: The ESEOO project uses a POLCOMS ocean model for the ESEOAT, ESEOCAN and ESEOMED regions. These regions comprises the North Atlantic sea, the Canary Islands and the Mediterranean sea, respectively, covering all the Spanish coasts. The model provides $72\mathrm{h}$ forecast maps, with hourly surface ocean currents or three-dimensional daily means using 32 σ-levels; the σ-levels are non evenly spaced depth levels, which are adapted to the bathymetry. This project is currently finished, but it is continued by MyOcean.

MyOcean: It is an European network for the ocean monitoring and forecasting. It provides products for several regions. We have used the Iberian Biscay Irish (IBI) region in our experiments. It uses NEMO, a 3D baroclinic model that solves the Navier Stokes equations in a grid of $1/36^\circ$ and 50 depth levels.

HYCOM: The HYbrid Coordinate Ocean Model is adapted to operate in both the open, stratified ocean, and in shallow coastal regions. It covers different regions, both regional and global, with $1/12^\circ$ and $1/25^\circ$, respectively. It provides daily means for -5 days hindcast —for evaluation from data assimilation processes— and $+5$ day forecast maps.

GeoEye: The NCOM and NLOM models provide surface ocean currents or just the SSHA with $1/8^\circ$ and $1/32^\circ$ spatial resolution, respectively. They are global models that actually integrate data from different oceanographic instruments on board satellites.

Additionally, we have used HF RADAR data from the coast of California, which provides high resolution ocean currents up to some kilometers from the coastline. These are surface ocean currents of a great accuracy, and high spatial and temporal resolution. Similarly, we have taken the winds velocity field of the HIRLAM atmospheric model.

A.1.1 Ocean Model Products

Apart from the ocean currents, extensively used in this thesis, Ocean Models also compute other physical parameters of the water. This comprises the temperature, salinity, and the Sea Surface Height Anomaly (SSHA) —or just the topography—, among others. From the temperature and salinity, if they are given at depth, we can compute the density of the water using the TEOS-10 equations (IOC et al., 2010). In fact, we have computed the water density to integrate the force balance motion model described in Section 2.2.3. Figure A.1 shows an example of (a) temperature and (b) SSHA maps, respectively. The outputs correspond to the region of the Canary Islands within the IBI region of MyOcean model.

Furthermore, there are chemical and biological parameters that can also form part of the product suite of Ocean Models, or satellite imagery. A representative example is the concentration of CHLorophyll (CHL). This parameters can be used to define forbidden regions, that the path planning algorithms must avoid. In the case of the CHL this was used in real missions to reduce the risk of biofouling.

In the case of the SSHA, we have used the NLOM maps to compute the geostrophic ocean currents. Given a SSHA map $\mathbf{s}(x, y, t)$ defined for every location $(x, y) \in \mathbb{R}^2$ and time t, the geostrophic ocean currents velocity field $\mathbf{v}(x, y, t)$ is computed for each component $u(x, y, t)$ and $v(x, y, t)$ as

$$u(x, y, t) = -\frac{g}{f}\frac{\partial \mathbf{s}(x, y, t)}{\partial y} \tag{A.1}$$

$$v(x, y, t) = \frac{g}{f}\frac{\partial \mathbf{s}(x, y, t)}{\partial x} \tag{A.2}$$

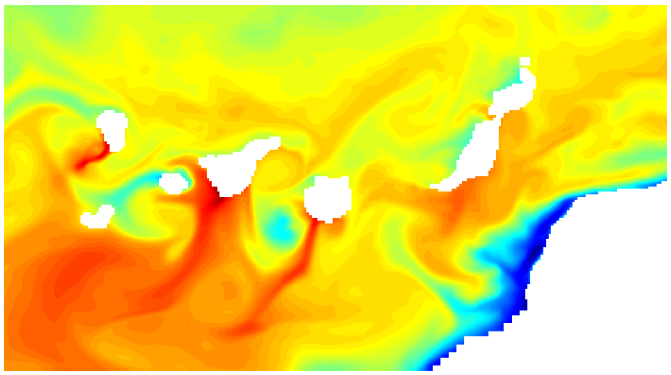
where g is the gravity force, f is the Coriolis parameter

$$f = 2\Omega \sin\phi \tag{A.3}$$

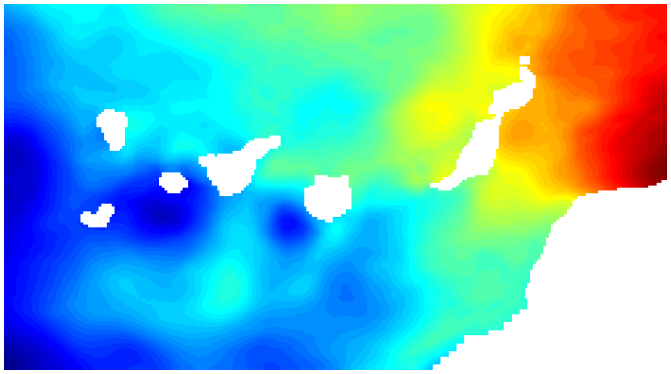
which depends on the latitude ϕ, with $2\Omega = 1.458423 \times 10^{-4}\mathrm{rad/s}$, and the partial derivatives define the gradient of the SSHA field. That is,

$$\nabla \mathbf{s}(x, y, t) = \begin{pmatrix} \dfrac{\partial \mathbf{s}(x, y, t)}{\partial x} \\ \dfrac{\partial \mathbf{s}(x, y, t)}{\partial y} \end{pmatrix} \tag{A.4}$$

which is solved numerically from the SSHA product data.



(a) Temperature.



(b) SSHA.

Figure A.1: Example of the output of MyOcean model for the IBI region: (a) temperature and (b) SSHA.

A.2 Distance Metrics

Here we describe different ways to compute the distance of two points on the Earth surface. Although the great circle distance gives the exact solution, we will see that there are some approximations that suffice for most glider missions, and run in less computing time. Below we give the notation used later to describe several metrics, which are finally compared on some tests.

A.2.1 Notation

Given two locations $P_0 = (\lambda_0, \phi_0)$ and $P_1 = (\lambda_1, \phi_1)$ in geographical coordinates longitude λ and latitude ϕ, we define the differences in longitude and latitude as follow,

$$\Delta\lambda = \lambda_1 - \lambda_0 \tag{A.5}$$

$$\Delta\phi = \phi_1 - \phi_0 \tag{A.6}$$

In some distance metrics it is common to use the mean latitude

$$\phi_m = \frac{\phi_0 + \phi_1}{2} \tag{A.7}$$

Finally, the colatitude θ is expressed in terms of the latitude ϕ as

$$\theta = \frac{\pi}{2} - \phi \tag{A.8}$$

For the radius R of the Earth we use the standard value $R = 6371\text{km}$, although it is possible to use more precise values, as the one for the WGS84 datum ellipsoid.

The distance metrics bellow compute the distance d in meters for the two points P_0 and P_1 on the Earth. Most of them assume a spherical or ellipsoidal shape for the Earth, and they are presented in order of accuracy and complexity of the equations managed. At the end, in Section A.2.9, we will compare all the distance metrics against the great circle distance, used as ground truth. The comparison it meant to show the error ε of the distance metrics and their computational time, in order to observe the speed up with respect to the great circle distance computed with the `distance` of Matlab®.

A.2.2 Pythagorean Formula with Parallel Meridians

This basic distance metric simply uses the Pythagorean theorem, which yields the straight line distance as the shortest one. Therefore, it computes the distance d as in the classical Euclidean space, although the result is multiplied by R. That is,

$$d = R\sqrt{\Delta\lambda^2 + \Delta\phi^2} \tag{A.9}$$

There exists an alternative formulation that uses an approximation of the Earth radius R. It separates R into two constants R_λ and R_ϕ that scale the differences $\Delta\lambda$ and $\Delta\phi$ respectively. Hence,

$$d = \sqrt{(R_\phi \Delta\phi)^2 + (R_\lambda \Delta\lambda)^2} \tag{A.10}$$

where

$$R_\lambda = 53\text{sm} \tag{A.11}$$
$$R_\phi = 69.1\text{sm} \tag{A.12}$$

This distance metric is only meant to compute distances in the plane, so it should only be used for very short distances and close to the equator.

A.2.3 Pythagorean Formula with Converging Meridians

For the equator, with greater latitudes, the longitude difference $\Delta\lambda$ is distorted, becuase the meridians converge at the poles. Such distortion can be corrected using some coefficient based on the mean latitude ϕ_m. There are two formulations, depending on the Earth shape assumed: spherical or ellipsoidal.

Both of the two distance metrics below are still meant for short distances, but they can be used outside the equator. However, the error grows as we approach to the poles, i.e. with large latitudes ϕ.

A.2.3.1 Spherical Earth projected to a plane

This distance follows the Pythagorean theorem, but the longitude is scaled by ϕ_m as follows,

$$d = R\sqrt{\Delta\lambda^2 + \cos(\phi_m)\,\Delta\phi^2} \tag{A.13}$$

Similarly to Section A.2.2, there is an alternative formulation that uses an approximation of the Earth radius R. In this cases, it only uses R_ϕ —which actually approximates R—, for the expression,

$$d = R_\phi\sqrt{\Delta\lambda^2 + \cos(\phi_m)\,\Delta\phi^2} \tag{A.14}$$

This approximation is very fast and produces fairly accurate results for small distances.

A.2.3.2 Ellipsoidal Earth projected to a plane

If we instead assume an Ellipsoidal Earth, rather than a sphere, we have

$$d = 1000\sqrt{(K_0\Delta\lambda)^2 + (K_1\Delta\phi)^2} \tag{A.15}$$

where

$$K_0 = 111.41513\cos(\phi_m) - 0.09455\cos(3\phi_m) + 0.00012\cos(5\phi_m) \tag{A.16}$$

$$K_1 = 111.13209 - 0.56605\cos(2\phi_m) + 0.00120\cos(4\phi_m) \tag{A.17}$$

This formula is prescribed for distances not exceeding 475km. It is worth mentioning the special meaning of the constants K_0 and K_1,

$$K_0 = \cos(\phi_m)\, N\frac{\pi}{180} \tag{A.18}$$

$$K_1 = M\frac{\pi}{180} \tag{A.19}$$

which are the kilometers per degree for the longitude and latitude differences, respectively. The values M and N are the Meridional and its Normal radii of curvature.

A.2.4 Polar Coordinate Flat-Earth formula

Using polar coordinates and still assuming a flat Earth, we have

$$d = R\sqrt{\theta_0^2 + \theta_1^2 - 2\theta_0\theta_1\cos(\Delta\lambda)} \tag{A.20}$$

where θ_0 and θ_1 are the colatitudes of the latitudes of the points P_0 and P_1, respectively, computed with (A.8).

A.2.5 Spherical Law of Cosines

Contrary to the previous distance metrics, here we use an spherical-surface formula, rather than the flat-Earth one. This is the simplest formula of this kind, defined as

$$d = R\arccos(\sin\phi_0\sin\phi_1 + \cos\phi_0\cos\phi_1\cos\Delta\lambda)\,; \tag{A.21}$$

With this expressions we can obtain $\varepsilon \approx 0.5\%$ of error, since it uses the sphere that best approximates the surface of the Earth. As we did for Section A.2.2 and A.2.3.1, we can use the R_ϕ approximation of R.

A.2.6 Haversine formula

The Haversine formula gives great circle distances between two points on a sphere. It is computed as follows,

$$d = 2R\arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_0\cos\phi_1\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \tag{A.22}$$

The Spherical law of cosines is actually a similar formula written using cosines instead of haversines. However, if the two points are close together —e.g. a kilometer apart— we will have numerical problems. The haversine formula, using sines, avoids that problem.

As with the law of cosines, this formula assumes a spherical Earth. However, the radius of curvature of a North-South meridian on the Earth's surface is 1% greater at the poles than at the equator. Therefore, we have a $\varepsilon \approx 0.5\%$ error.

A.2.7 Vincenty's formula

The Vincenty's formula consider the Earth's ellipticity, so it breaks with the spherical Earth assumption of the two previous methods. Therefore, it can reduce the error on the distance computation. The Vincenty's formula consists on an iterative approximation scheme. Here we use a simplified explicit computation that uses the expression

$$d = R\sigma \tag{A.23}$$

where σ is the arc length in the auxiliary sphere, iteratively computed as

$$\sigma = \arctan \frac{\sin \sigma}{\cos \sigma} \tag{A.24}$$

until σ converges to a value. The expression is computed at each step using

$$\sin \sigma = \sqrt{(\cos \phi_1 \sin \Delta\lambda)^2 + (\cos \phi_0 \sin \phi_1 - \sin \phi_0 \cos \phi_1 \cos \Delta\lambda)^2} \tag{A.25}$$

$$\cos \sigma = \sin \phi_0 \sin \phi_1 + \cos \phi_0 \cos \phi_1 \cos \Delta\lambda \tag{A.26}$$

Indeed, this is the special case of the Vincenty's formula computed for a sphere. Meanwhile, we must solve the method of Vincenty's formula for ellipsoids.

A.2.8 Great Circle

Finally, we compute the great circle distance on the WGS84 datum ellipsoid that approximates the Earth using the `distance` function of Matlab®. We could have used the Vincenty's formula method, but the `distance` provides more flexibility to set the desired datum. In Figure A.2 we observe the so called great circle that gives the shortest distance between P_0 and P_1 in a sphere or ellipsoid, in the most general case.

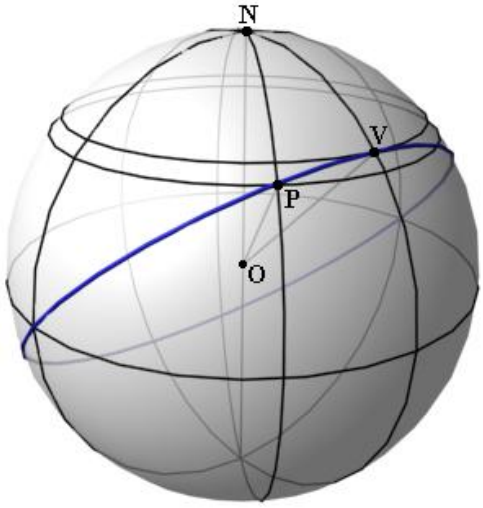
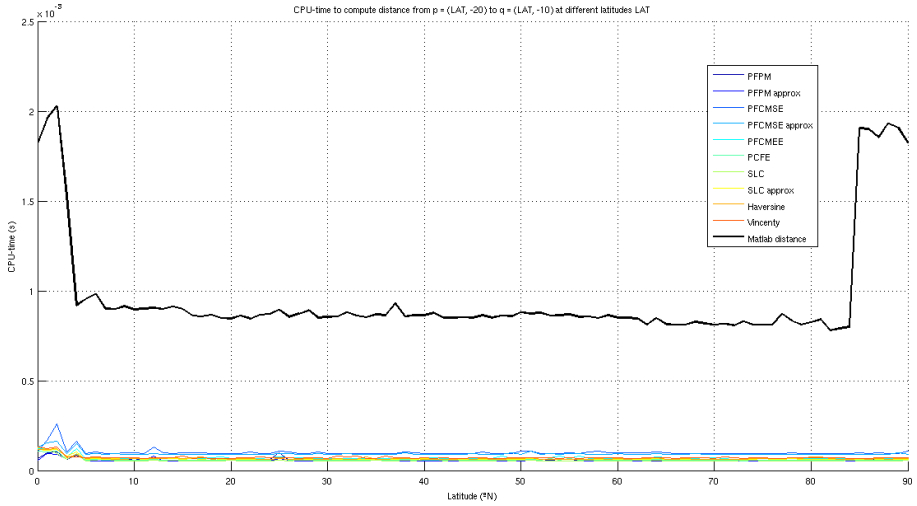


Figure A.2: Great Circle in a sphere shown in blue. It is the unique great circle that crosses the two points P and V. The arc of the great circle that goes from P to V gives the shortest distance between two points in a sphere or ellipsoid —which is a good approximation of the Earth shape.

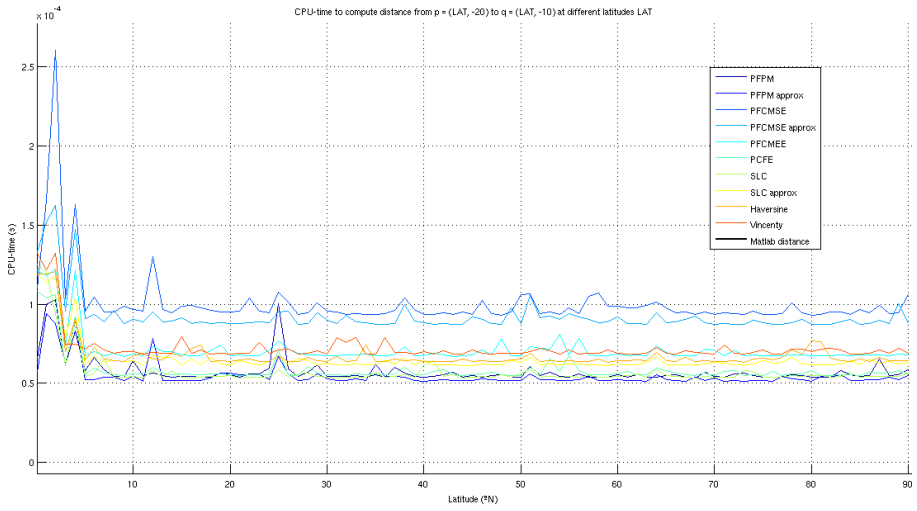
This distance metric is used as ground truth in the next section, since it uses an ellipsoid shape Earth approximation that gives the most accurate distance so far, for all the distance metrics discussed here.

A.2.9 Comparison

Here we compare the computing time, the distance and the error with respect to the ground truth given by the `distance` function. For the comparison we have run a series of distance computations for two points $P_0 = (\phi, -20°)$ and $P_1 = (\phi, -10°)$, where the latitude $\phi \in [0, \pi/2]$ goes from the equator to the North pole. This way we evaluate the distance metrics at different latitudes, which is the basic difficulty on computing distances on the Earth shape —being spherical or ellipsoidal.



(a) All distance metrics.



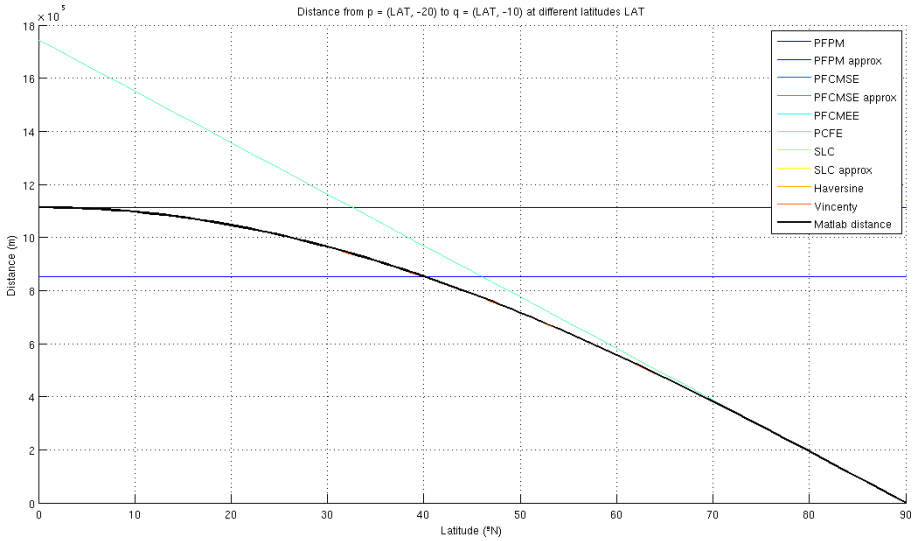
(b) Zoom in without the slowest.

Figure A.3: CPU time (s) for (a) all the distance metrics described here, and (b) zoom in discarding the great circle `distance` of Matlab®, which is the slowest approach.

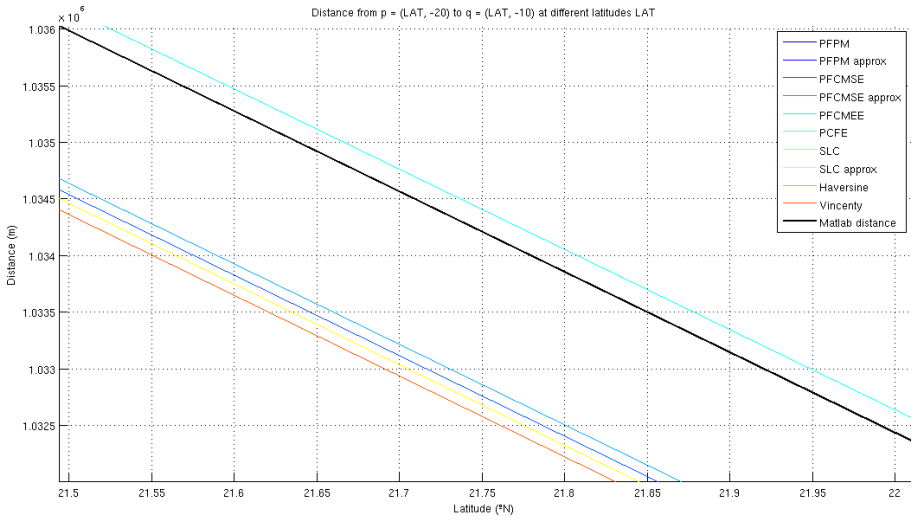
For the sake of clarity, we use the following acronyms for the distance metrics compared here.

PFPM: Pythagorean Formula with Converging Meridians.

PFCMSE: Pythagorean Formula with Converging Meridians, using Spherical Earth projected to a plane.



(a) All distance metrics.



(b) Zoom in without the worst.

Figure A.4: Distance (m) for (a) all the distance metrics described here, and (b) zoom in discarding the approaches with higher error, being the PFPM and its approximation variant.

PFCMEE: Pythagorean Formula with Converging Meridians, using Ellipsoidal Earth projected to a plane.

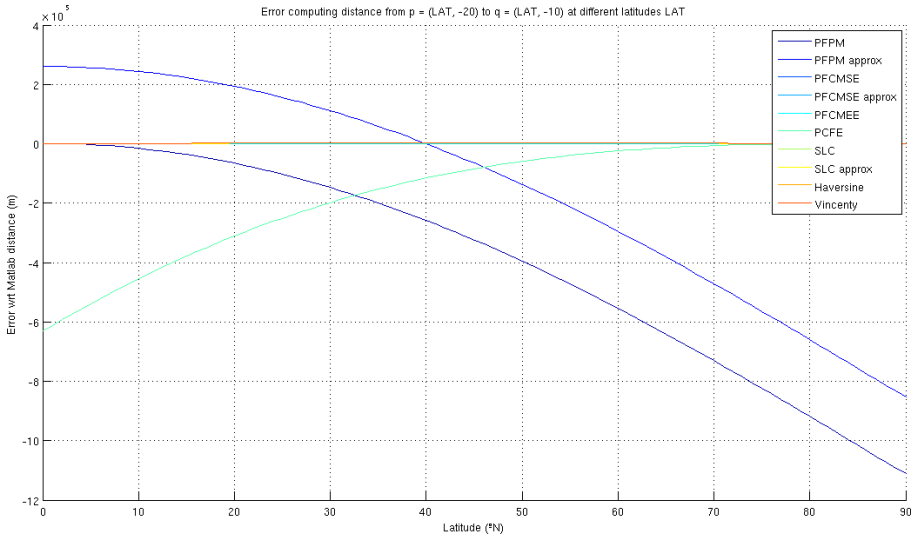
PCFE: Polar Coordinate Flat-Earth formula.

SLC: Spherical Law of Cosines.

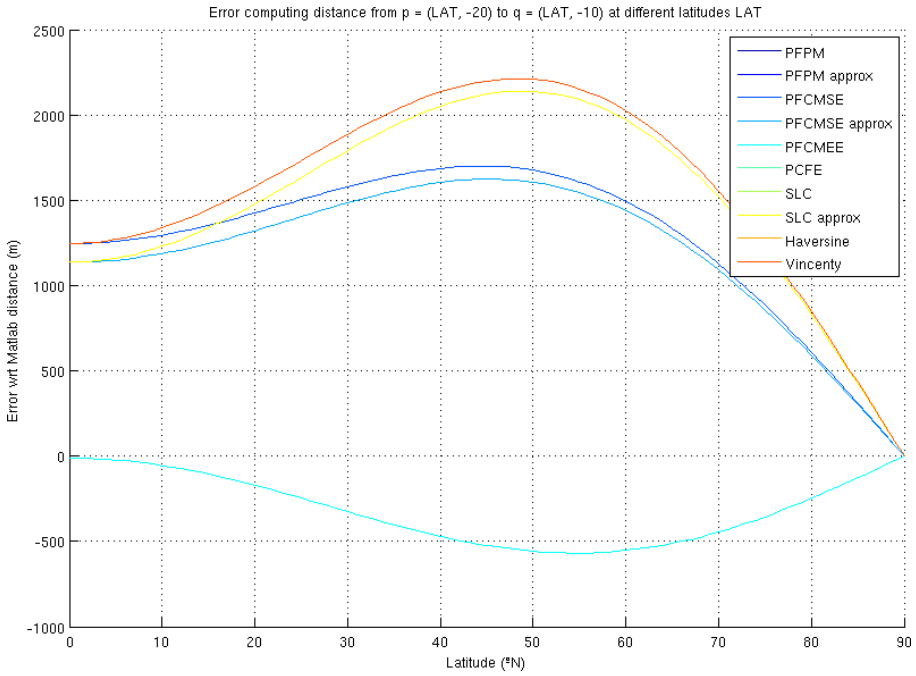
Haversine: Haversine formula.

Vincenty: Vincenty's formula.

The items with the suffix *approx* refer to the approximation variant of the given distance metric.



(a) All distance metrics.



(b) Without the worst.

Figure A.5: Error (m) with respect to the great circle `distance` metric for (a) all the distance metrics described here, and (b) discarding the approaches with higher error, being the PFPM and its approximation variant.

We start by comparing the CPU time, to validate our first hypothesis, i.e. the `distance` function is slower than the approximations presented here. In fact, the figure Figure A.3 (a) shows that it is one order of magnitude slower than all the distance metrics described here. In (b) we zoom in to inspect the CPU time

of the rest of the formulas. The PFCMSE and its approximation is the slowest of them. Then, as expected, we have the Haversine and the Vincenty's formulas, which are also the more accurate formulas presented here. The fastest CPU time is obtained with the PFPM formula and its approximation, since they use fewer trigonometric functions than the others.

Now that we know that it is worth using an approximation in place of `distance`, we compare the distance and error, shown in Figure A.4 and A.5, respectively. In Figure A.4 (a) we observe how at the poles the distance vanishes, and it grows as we approach to the equator. Except for the simplest approaches, the rest of the formulas reproduce the same curve as the one obtained with `distance`. The PFPM, and its approximation variant, generate a constant distance, since it does not depend on the latitude ϕ. The PCFE does not produce a curve, but a line with a slope tangent to the curve at the pole $\phi = 90°$. If we zoom in, we observe in Figure A.4 (b) a small error between the rest of the formulas.

The error with respect to `distance` is shown Figure A.5 (a) for all the distance metrics. We remove the worst ones in order to observe the error in more detail. Figure A.5 (b) shows how the error varies with the latitude ϕ. At first sight, we observe that at the pole all the formulas have no error, because the distance is $d = 0$ there. Curiously, only the PFCMEE has almost no error at the equator. The reason is that it approximates the Earth with an ellipsoid, as `distance`. The rest use an sphere shape approximation, which produce a clear offset from the correct distance. We can change the Earth radius R, to reduce the error at the equator, but it will grow at other latitudes.

A.3 Azimuth Angle Metrics

Similar to the distance metrics, we can also compute the azimuth angle ψ between the two points P_0 and P_1. Indeed, the approaches to compute ψ are twin formulas of the equations discussed thus far for the distance metrics. However, only a few provide the azimuth ψ. Below we describe each, to later compare them against the great circle `azimuth` function of Matlab®, considered as the ground truth.

A.3.1 Pythagorean Formula with Parallel Meridians

As for the distance computation, the Pythagorean theorem is also applied here to compute the angle. We only have to compute the $\arctan$ of the differences $\Delta\lambda$ and $\Delta\phi$. That is,

$$\psi = \frac{\pi}{2} - \arctan\left(\frac{\Delta\phi}{\Delta\lambda}\right) \tag{A.27}$$

Being an azimuth angle, we subtract the $\arctan$ value from $\pi/2$, to obtain the correct angle, with $0°$ at the North and growing counter-clockwise. This basic equation it only valid for close points, and also near the equator.

A.3.2 Pythagorean Formula with Converging Meridians

A.3.2.1 Spherical Earth projected to a plane

If we scaled the longitude λ according with ϕ_m, we obtain the twin formula of the Spherical Earth using the Pythagorean formula with converging meridians. Hence,

$$\psi = \frac{\pi}{2} - \arctan\left(\frac{\Delta\phi}{\cos(\phi_m)\,\Delta\lambda}\right) \tag{A.28}$$

This formula will allow to compute the azimuth at any latitude, not only close to the equator. However, the error increases when we approach to the poles.

A.3.3 Great Circle

For the great circle formula we distinguishes between the spherical and ellipsoid Earth approximations.

A.3.3.1 Spherical Earth

Assuming a sphere shape Earth, we have

$$\psi = \arctan\left(\frac{\cos\phi_1 \sin\Delta\lambda}{\cos\phi_0 \sin\phi_1 - \sin\phi_0 \cos\phi_1 \cos\Delta\lambda}\right) \tag{A.29}$$

This is the particular case of an ellipsoid degenerated into a sphere. The error with this approach is $\varepsilon \approx 0.5\%$, since the sphere assumption is less accurate for large latitudes ϕ, close to the poles.

A.3.3.2 Ellipsoidal Earth

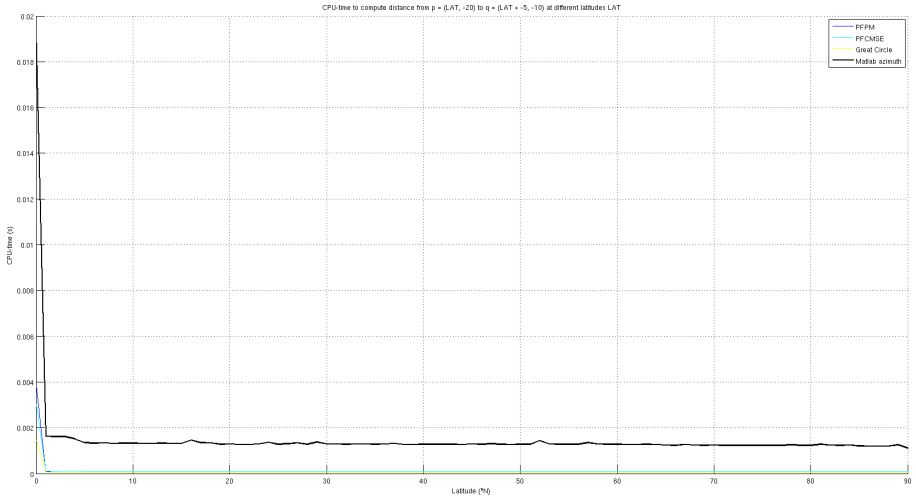
Finally, we use the `azimuth` function of Matlab®, using the WGS84 datum ellipsoid to approximate the Earth surface. This approach gives the best estimation of the azimuth, since it does not assume a spherical Earth, but ellipsoidal. Therefore, this angle metric is considered as the ground truth for the next comparison.

A.3.4 Comparison

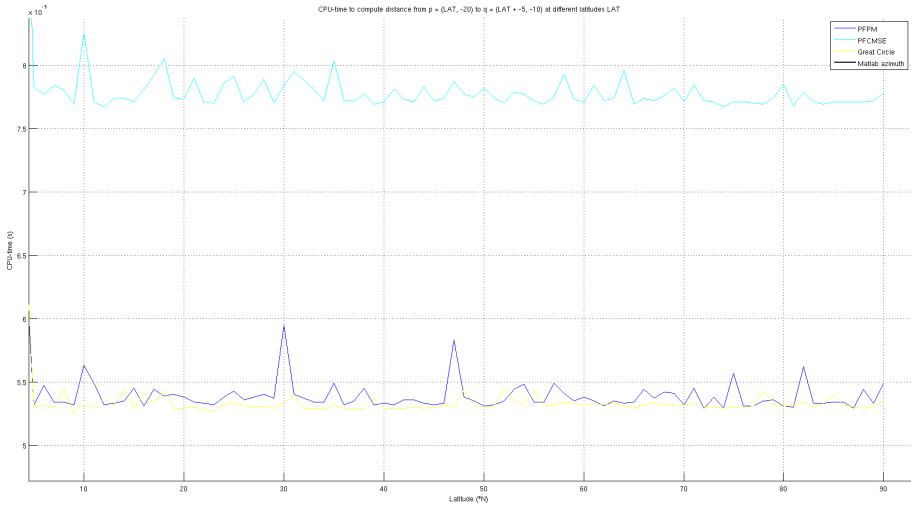
Here we compare the computing time, the distance and the error with respect to the ground truth given by the `azimuth` function. For the comparison we have run a series of distance computations for two points $P_0 = (\phi, -20°)$ and $P_1 = (\phi - 5, -10°)$, where the latitude $\phi \in [0, \pi/2]$ goes from the equator to the North pole; we use $\phi - 5$ in P_1 in order to have a more representative example, and not simply a horizontal $\psi = 0°$ azimuth angle. This way we evaluate the angle metrics at different latitudes, which is the basic difficulty on computing them on the Earth shape —being spherical or ellipsoidal.

In Figure A.6 (a) we observe that the `azimuth` function takes more time to compute the azimuth angle than the approximations described here. Again, the speed up is of one order of magnitude or even more. If we zoom in for the approximation formulas in (b), we observe that the PFCMSE is the slowest. As expected, the PFPM is very fast, but surprisingly the spherical Earth great circle —shown as Great Circle in the figure— is even slightly faster; note that this approach is also more accurate.

In fact, in Figure A.7 (a) we see that the spherical Earth great circle yields almost the same azimuth as the ellipsoidal version provided by `azimuth`. The PFCMSE shows a similar curve, but with some offset error. Meanwhile, the PFPM is constant, since it does not depends on the latitude ϕ. Therefore, in Figure A.7 (b) the error with respect to `azimuth` is almost null for the spherical Earth great circle.

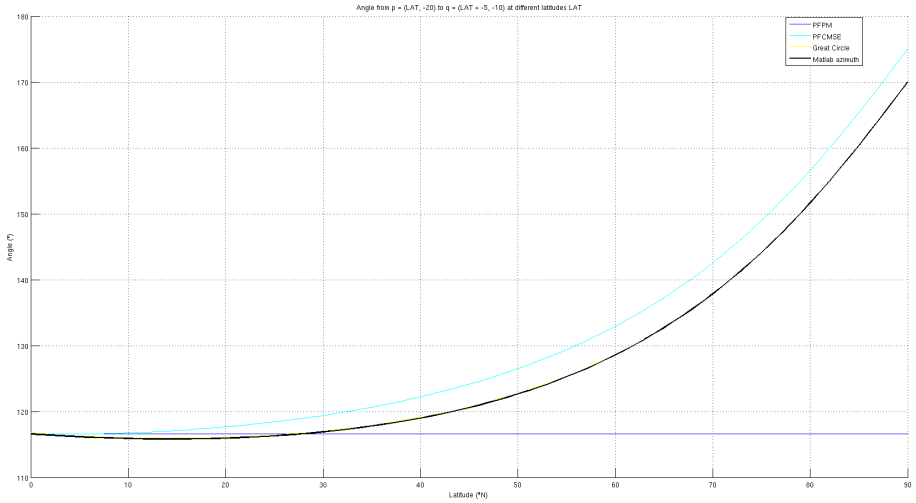


(a) All azimuth angle metrics.

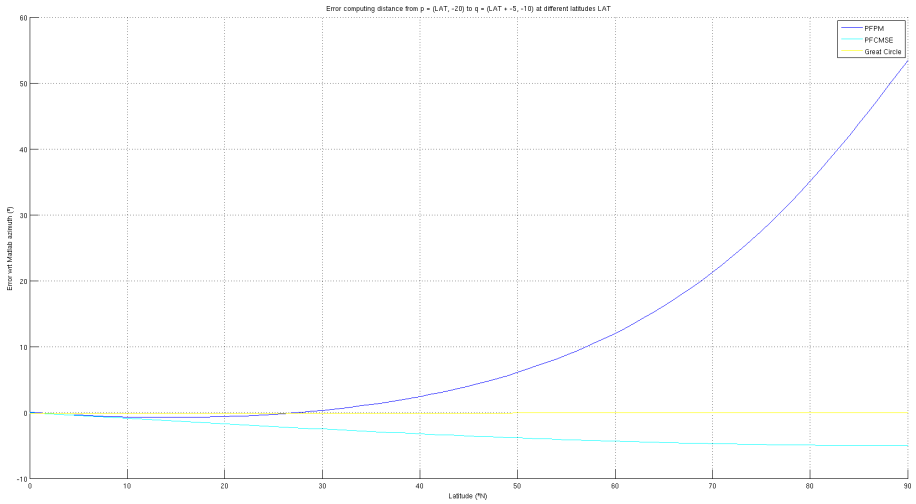


(b) Zoom in without the slowest.

Figure A.6: CPU time (s) for (a) all the azimuth angle metrics described here, and (b) zoom in discarding the great circle `azimuth` of Matlab®, which is the slowest approach.



(a) Azimuth angle metrics.



(b) Error.

Figure A.7: Azimuth angle (°) for (a) all the azimuth angle metrics described here, and (b) error with respect to the `azimuth` function of Matlab®.

Appendix B

Motion Models

Here we cover some simplified or reduced motion models that derivate from the glider motion models discussed in Section 2.2. This comprises drifters and profiler floats, which can be modeled with the extrema cases of the glider motion models, at the singular values of some parameters like the pitch θ or the nominal surge speed U_g.

B.1 Point Motion Models

The glider point motion models shown in Section 2.2.2 can be adapted to operate as a drifter with a few minor changes. This models are usually particularizations of the unconstrained point motion model of Section 2.2.2.1.

B.1.1 Drifter

A drifter is a vehicle that does not generate any speed at all —i.e. $\mathbf{U} = (0, 0, 0)\mathrm{m/s}$. It simply moves as a consequence of the effect of the ocean currents that carry it from one location to another. The vehicle can be modeled including its mass and forces, and apply a force balance motion model as in Section B.2.2, but drifters are usually modeled as a point that travels according to an unconstrained point motion model defined by the differential equations

$$\dot{x} = u \tag{B.1}$$
$$\dot{y} = v \tag{B.2}$$
$$\dot{z} = w \tag{B.3}$$

where u, v and w are the x, y and z components of the ocean current speed at a given location $\mathbf{x} = (x, y, z)$ and time t.

This drifter motion model is used to integrate the glider trajectory when it drifts on the surface, affected by the surface ocean currents (u, v) —at surface $w = 0$— for a time of $\approx 15\mathrm{min}$, which is the typical time the vehicle stays at surface before it dives again.

B.2 Force Balance Motion Models

The glider force balance motion model shown in Section 2.2.3 can be adapted to operate as a profiling float or a drifter with a few minor changes. For your convenience, we recall the force balance equations for a glider

below.

$$F_B - \cos\gamma F_L - \sin\gamma F_D - F_g = 0 \tag{B.4}$$
$$-\cos\gamma F_D + \sin\gamma F_L = 0 \tag{B.5}$$

where F_B is the (net) buoyancy force, F_g is the force due to gravity, F_L is the lift force, F_D is the drag force and γ is the glide angle, defined as the sum of the pitch angle θ and the angle of attack α.

B.2.1 Profiling Float

The glider force motion model discussed above might be applied for a profiling float. We only have to consider a glide angle of $\gamma = \frac{\pi}{2}$. Therefore, the pitch angle is $\omega = \gamma = \frac{\pi}{2}$ and the angle of attack is $\alpha = 0$. As a result, the motion model is simplified significantly. The lift force —perpendicular to the vehicle displacement— has no effect, while the drag force is produced exclusively by a constant parasite drag C_{D_0}.

For a profiling float we only have a vertical force balance, and consequently only vertical speed. Taking the same gravity and buoyancy forces of the glider force motion model, (2.30) and (2.31), and the simplified drag force

$$F_D = \frac{1}{2}\rho C_{D_0} S U^2 \tag{B.6}$$

where S might be thought as the float's stabilizers surface, the resulting force balance is

$$F_g - F_B - F_D = 0 \tag{B.7}$$
$$F_g - F_B - \frac{1}{2}\rho C_{D_0} S U^2 = 0 \tag{B.8}$$

Then, the speed U can be evaluated with

$$U = \sqrt{\frac{2(F_B - F_g)}{\rho S C_{D_0}}} \tag{B.9}$$

In this case, there is only a vertical speed component. For this reason, we actually have $U = w$, since $\sin\gamma = \sin\frac{\pi}{2} = 1$. However, it is important to treat the speed sign properly. This sign determines the direction taken by the buoy, which is positive $+$ for climbing and negative $-$ for diving. The sign is given by the difference between the gravity and buoyancy forces, so we can consider the absolute value inside the square root and apply the sign to the final result,

$$w = U = \operatorname{sign}(F_B - F_g)\sqrt{\frac{2\,|F_B - F_g|}{\rho S C_{D_0}}} \tag{B.10}$$

B.2.2 Drifter

A drifter performs four steps during its operation cycle:

1. Starting at surface, the first task consists on diving to approximately $1000\mathrm{m}$ depth.
2. Then, the buoy drifts for $8 \sim 10$ days.
3. After drifting at a fix depth, the buoy dives to 1500 or $2000\mathrm{m}$.
4. Finally, a profiling step is performed. It consists on climbing from the current depth to the surface, retrieving data of the water column.

The profiling step is covered by the previous section. Meanwhile, in order to drift at a fix depth, the vertical speed must be null $w = U = 0$. In other words, the gravity and buoyancy forces must be equal,

$$\begin{aligned} F_g - F_B - \frac{1}{2}\rho C_{D_0} S U^2 = 0 & \\ F_g = F_B & \\ g m_g = g\rho(V_g(1 - \varepsilon P + \alpha_T(T - T_0)) + \Delta V_{bp}) & \end{aligned}$$

The buoy must control the volume displaced by the buoyancy engine to compensate the mass m_g of the vehicle. This volume ΔV_{bp} can be evaluated with

$$\Delta V_{bp} = \frac{m_g}{\rho} - V_g(1 - \varepsilon P + \alpha_T(T - T_0)) \tag{B.11}$$

Here, the density ρ, the pressure P and the temperature T depend on the current position of the buoy. We assume that ΔV_{bp} is computed once the fix depth is reached and is kept in the sequel. This may lead to slightly changes of depth, due to the variability of temperature and density basically.

Bibliography

Alvarez, A., Caiti, A., and Onken, R. (2004). Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *IEEE Journal of Oceanic Engineering*, 29(2):418–429.

Barker, W. P. (2012). An Analysis of Undersea Glider Architectures and an Assessment of Undersea Glider Integration into Undersea Applications. Master's thesis, Naval Postgraduate School, Monterey, California.

Barrick, D. E., Evans, M. W., and Weber, B. L. (1977). Ocean Surface Currents Mapped by Radar. *Science*, 198(4313):138–144.

Bellman, R. E. (2003). *Dynamic programming*. Dover Books on Computer Science Series. Dover Publications, Incorporated.

Bhatta, P., Fiorelli, E., Lekien, F., Leonard, N. E., Paley, D. A., Zhang, F., Bachmayer, R., Davis, R. E., Fratantoni, D. M., and Sepulchre, R. (2005). Coordination of an underwater glider fleet for adaptive sampling. In *Proceedings of the International Workshop on Underwater Robotics*, pages 61–69.

Biggs, M. C. (1975). Constrained minimization using recursive quadratic programming. *Towards Global Optimization*, pages 341–349.

Bindoff, N. L., Willebrand, J., Artale, V., Cazenave, A., Gregory, J., Gulev, S., Hanawa, K., Quéré, C. L., Levitus, S., Nojiri, Y., Shum, C. K., Talley, L. D., and Unnikrishnan, A. (2007). Observations: Oceanic Climate Change and Sea Level. In Cambridge University Press, Cambridge, U. K. and New York, NY, U., editors, *Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*.

Bishop, C. M. (2008). Sensor Dynamics of Autonomous Underwater Gliders. Master's thesis, Memorial University of Newfoundland.

Brock, O., Trinkle, J. C., and Ramos, F. (2009). *Robotics: science and systems IV*. MIT Press.

Broyden, C. G. (1970). The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90.

Bryan, K. and Cox, M. (2011). A numerical investigation of the oceanic general circulation. *Tellus A*, 19(1).

Byrd, R. H., Y., M. E. H., and Z., J. N. (1999). An interior point algorithm for large scale nonlinear programming. *SIAM Journal on Optimization*.

Cabrera Gámez, J., Hernández Sosa, D., Isern González, J., Domínguez Brito, A., Fernández Perdomo, E., Prieto Marañón, V., and Ramos, A. (2010). Path Planning for Gliders. In *Proceedings of the National Research Council (NRC) Canada*.

Carroll, K., McClaran, S., Nelson, E., Barnett, D., Friesen, D., and William, G. (1992). AUV path planning: an A* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones. In *Proceedings of the 1992 Symposium on Autonomous Underwater Vehicle Technology*, pages 79–84.

Cencini, M., Lacorata, G., Vulpiani, A., and Zambianchi, E. (1998). Mixing in a Meandering Jet: a Markovian Approximation. In *Journal of Physical Oceanography*, volume 29, pages 2578–2594.

Choset, H. and Pignon, P. (1997). Coverage Path Planning: The Boustrophedon Cellular Decomposition. In *International Conference on Field and Service Robotics*.

Coleman, T. F. and Li, Y. (1996). An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds. *SIAM Journal of Optimization*, 6:418–445.

Daniel, K., Nash, A., Koenig, S., and Felner, A. (2010). Theta*: Any-Angle Path Planning on Grids. *Journal of Artificial Intelligence Research*, 39:533–579.

Davis, R. (1981). Variability in the upper ocean during MILE. Part II: Modeling the mixed layer response. *Deep Sea Research Part I: Oceanographic Research*, 28:1453–1475.

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*, 56:173–187.

Dechter, R. and Pearl, J. (1985). Generalized best-first search strategies and the optimality of A*. *J. ACM*, 32(3):505–536.

Denardo, E. V. and Fox, B. L. (1979). Shortest Route Methods: Reaching, Pruning and Buckets. *Operations Research*, 27:161–186.

Dijkstra, E. W. (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, pages 269–271.

Dobricic, S., Pinardi, N., Testor, P., and Send, U. (2010). Impact of data assimilation of glider observations in the Ionian Sea (Eastern Mediterranean). *Dynamics of Atmospheres and Oceans*, 50(1):78–92.

Domínguez-Brito, A. C., Hernández-Sosa, D., Isern-González, J., and Cabrera-Gámez, J. (2007). CoolBOT: A Component Model and Software Infrastructure for Robotics. In Brugali, D., editor, *Software Engineering for Experimental Robotics*, volume 30 of *Springer Tracts in Advanced Robotics*, pages 143–168. Springer Berlin Heidelberg.

Donaldson, P. (2007). Tracing a eureka moment. *Unmanned Vehicles*, page 17. Technically Speaking. Comment.

Dormand, J. R. and Prince, P. J. (1980). A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26.

Dorst, L. and Trovato, K. (1989). Optimal Path Planning By Cost Wave Propagation In Metric Configuration Space. In *SPIE 1007, Mobile Robots III*, pages 186–197.

Doucet, A. and Johansen, A. M. (2008). *A Tutorial on Particle Filtering and Smoothing: Fifteen years later*.

Earle, S. A. (2012). *Un mundo azul. El rumbo de los océanos, el futuro de la Tierra*. RBA Libros, S.A., 1 edition. Translation by Efrén del Valle Peñamil, The world is blue : how our fate and the ocean's are one, 2009.

Eichhorn, M. (2009). A new concept for an obstacle avoidance system for the AUV "SLOCUM glider" operation under ice. In *Proceedings of the OCEANS 2009*, pages 1–8.

Eichhorn, M. (2010). Solutions for practice-oriented requirements for optimal path planning for the AUV "SLOCUM Glider". In *Proceedings of the OCEANS 2010*, pages 1–10.

Fay, M. P., Proschan, M. A., et al. (2010). Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics Surveys*, 4:1–39.

Fernández Perdomo, E., Cabrera Gámez, J., Domínguez Brito, A., and Hernández Sosa, D. (2009). Mission Specification in Underwater Robotics. In *Proceedings of the X Workshop de Agentes Físicos 2009 (WAF'2009)*, Cáceres, Spain.

Fernández Perdomo, E., Cabrera Gámez, J., Domínguez Brito, A., and Hernández Sosa, D. (2010a). Mission Specification in Underwater Robotics. *Journal of Physical Agents*, 4(1):25–34.

Fernández-Perdomo, E., Cabrera-Gámez, J., Hernández-Sosa, D., Isern-González, J., Domínguez-Brito, A., Prieto-Marañón, V., and Ramos, A. (2011a). Adaptive Bearing Sampling for a Constant-Time Surfacing A* path planning algorithm for gliders. In *Proceedings of the IEEE 2011 International Conference on Robotics and Automation (ICRA 2011)*, Shanghai, China.

Fernández-Perdomo, E., Cabrera-Gámez, J., Hernández-Sosa, D., Isern-González, J., Domínguez-Brito, A., Prieto-Marañón, V., and Ramos, A. (2011b). Path Planning for Underwater Gliders with the CTS-A* Algorithm. In *Proceedings of the Thirteenth International Conference on Computer Aided Systems Theory (Eurocast 2011)*, Las Palmas de Gran Canaria, Spain.

Fernández Perdomo, E., Cabrera Gámez, J., Hernández Sosa, D., Isern González, J., Domínguez Brito, A., Redondo, A., Coca, J., Ramos, A. G., Álvarez Fanjul, E., and García, M. (2010b). Path Planning for gliders using Regional Ocean Models: Application of *Pinzón* path planner with the ESEOAT model and the RU27 trans-Atlantic flight data. In *Proceedings of the OCEANS 2010 IEEE Sydney Conference and Exhibition*.

Fishman, G. S. (1996). *Monte Carlo. Concepts, Algorithms, and Applications*. Springer Series in Operations Research. Springer, United States of America, 1 edition. 6^{th} printing.

Fletcher, R. and Powell, M. J. D. (1963). A Rapidly Convergent Descent Method for Minimization. *The Computer Journal*, 6(2):163–168.

Fossen, T. I. (2002). *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics.

Fredman, M. L. and Tarjan, R. E. (1984). Fibonacci Heaps And Their Uses In Improved Network Optimization Algorithms. In *25th Annual Symposium on Foundations of Computer Science (FOCS 1984)*, pages 338–346.

Fredman, M. L. and Tarjan, R. E. (1987). Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615.

Fu, L.-L. and Holt, B. (1982). Seasat Views Oceans and Sea Ice With Synthetic-Aperture Radar. Technical report, Jet Propulsion Laboratory (JPL), National Aeronautics and Space Administration (NASA), California Institute of Technology, Pasadena, California.

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.

Garau, B., Bonet, M., Alvarez, A., Ruiz, S., and Pascual, A. (2009). Path Planning for Autonomous Underwater Vehicles in Realistic Oceanic Current Fields: Application to Gliders in the Western Mediterranean Sea. *Journal of Maritime Research*, 6(2):5–22.

Gill, P. E., Murray, W., and Wright, M. H. (1981). *Practical optimization*. Academic Press.

Glenn, S., Schofield, O., and Kohut, J. (2009). The Scarlet Knight's Trans-Atlantic Challenge. Blog.

Graver, J. G. (2005). *Underwater Gliders: Dynamics, Control and Design*. PhD thesis, Princeton University.

Hammersley, J. M. and Handscomb, D. C. (1964). *Monte Carlo Methods*. Methuen's Monographs on Applied Probability and Statistics. Methuen & Co. Ltd., Fletcher & Son Ltd., Norwich, London. Catalogue No. (Methuen) 12/5234/64.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

Hass, H. (1994). *Del pez al hombre*. Biblioteca científica Salvat. Salvat.

He, M., Williams, C. D., and Bachmayer, R. (2009). Simulations of an Iterative Mission Planning Procedure for Flying Gliders into Strong Ocean Currents. In *Simposium on Unmanned Untethered Submersible Technology*.

Hernández, D., Smith, R., Fernández, E., Isern, J., Cabrera, J., Domínguez, A., and Prieto, V. (2013a). Glider Path-Planning for Optimal Sampling of Mesoscale Eddies. In *Proceedings of the Fourteenth International Conference on Computer Aided Systems Theory (Eurocast 2013)*, Las Palmas de Gran Canaria, Spain.

Hernández, D., Smith, R., Fernández, E., Isern, J., Cabrera, J., Domínguez, A., and Prieto, V. (2013b). Glider Path-Planning for Optimal Sampling of Mesoscale Eddies. In Moreno-Díaz, R., Pichler, F., and Quesada-Arencibia, A., editors, *Computer Aided Systems Theory — EUROCAST 2013*, Lecture Notes in Computer Science. Springer Berlin Heidelberg.

Huang, Y. and Liao, C.-S. (2012). The Canadian Traveller Problem Revisited. In Chao, K.-M., sheng Hsu, T., and Lee, D.-T., editors, *Algorithms and Computation*, volume 7676 of *Lecture Notes in Computer Science*, pages 352–361. Springer Berlin Heidelberg.

Hátún, H., Eriksen, C. C., and Rhines, P. B. (2007). Buoyant Eddies Entering the Labrador Sea Observed with Gliders and Altimetry. *Journal of Physical Oceanography*, 37:2838–2854.

Ide, K., Courtier, P., Ghil, M., and Lorenc, A. C. (1997). Unified Notation for Data Assimilation: Operational, Sequential and Variational. *Journal of Meteorological Society of Japan*, 75(1):181–189.

Inanc, T., Shadden, S. C., and Marsden, J. E. (2005). Optimal trajectory generation in ocean flows. In *Proceedings of the 2005 American Control Conference*, pages 674–679.

IOC, SCOR, and IAPSO (2010). *The international thermodynamic equation of seawater — 2010: Calculation and use of thermodynamic properties*. Intergovernmental Oceanographic Commission, UNESCO, manuals and guides no. 56 edition.

Isern-González, J., Hernández-Sosa, D., Fernández-Perdomo, E., Cabrera-Gámez, J., Domínguez-Brito, A., and Prieto-Marañon, V. (2011a). Avoiding obstacles in Underwater Glider Path Planning. In *Proceedings of the XII Workshop de Agentes Físicos 2011 (WAF'2011)*, Albacete, Spain.

Isern-González, J., Hernández-Sosa, D., Fernández-Perdomo, E., Cabrera-Gámez, J., Domínguez-Brito, A., and Prieto-Marañón, V. (2011b). Application of Iterative Optimization Algorithms to Trajectory Planning for Underwater Gliders. In *Proceedings of the Thirteenth International Conference on Computer Aided Systems Theory (Eurocast 2011)*, Las Palmas de Gran Canaria, Spain.

Isern-González, J., Hernández-Sosa, D., Fernández-Perdomo, E., Cabrera-Gámez, J., Domínguez-Brito, A., and Prieto-Marañón, V. (2011c). Application of Optimization Algorithms to Trajectory Planning for Underwater Gliders. In Moreno-Díaz, R., Pichler, F., and Quesada-Arencibia, A., editors, *Computer Aided Systems Theory — EUROCAST 2011*, volume 6928 of *Lecture Notes in Computer Science*, pages 433–440. Springer Berlin Heidelberg.

Isern-González, J., Hernández-Sosa, D., Fernández-Perdomo, E., Cabrera-Gámez, J., Domínguez-Brito, A., and Prieto-Marañón, V. (2011d). Iterative Optimization-Based Path Planning with Variable Costs for Underwater Gliders. In *Proceedings of the OCEANS 2011 IEEE Santander Conference*.

Isern-González, J., Hernández-Sosa, D., Fernández-Perdomo, E., Cabrera-Gámez, J., Domínguez-Brito, A., and Prieto-Marañón, V. (2011e). Path planning for underwater gliders using iterative optimization. In *Proceedings of the IEEE 2011 International Conference on Robotics and Automation (ICRA 2011)*, Shanghai, China.

Isern-González, J., Hernández-Sosa, D., Fernández-Perdomo, E., Cabrera-Gámez, J., Domínguez-Brito, A., and Prieto-Marañón, V. (2012). Obstacle Avoidance in Underwater Glider Path Planning. *Journal of Physical Agents*, 6(1).

Isern-González, J., Hernández-Sosa, D., Fernández-Perdomo, E., Cabrera-Gámez, J., Domínguez-Brito, A., Prieto-Marañón, V., and Eichhorn, M. (2011f). Hold Track Problem in Glider Piloting. In *Proceedings of the OCEANS 2011 MTS/IEEE Kona Conference*.

Isern-González, J., Hernández-Sosa, D., Fernández-Perdomo, E., Cabrera-Gámez, J., Domínguez-Brito, A., Prieto-Marañón, V., Ramos, A., Coca, J., and Redondo, A. (2011g). An Optimization-Based Path Planner for Underwater Gliders. In *Proceedings of the Fifth EGO (European Glider Observatory) Meeting and Glider School*, Las Palmas de Gran Canaria, Spain.

Jimenez, B., Sangra, P., and Mason, E. (2008). A numerical study of the relative importance of wind and topographic forcing on oceanic eddy shedding by tall, deep water islands. *Ocean Modelling*, 22(3):146–157.

Karger, D. and Nikolova, E. (2008). Exact Algorithms for the Canadian Traveller Problem on Paths and Trees.

Kasten, J., Hotz, I., and Hege, H.-C. (2012). On the Elusive Concept of Lagrangian Coherent Structures. In Peikert, R., Hauser, H., Carr, H., and Fuchs, R., editors, *Topological Methods in Data Analysis and Visualization II*, Mathematics and Visualization, pages 207–220. Springer Berlin Heidelberg.

Keay, W. (1995). *Land Navigation: Routefinding with Map & Compass*. Duke of Edinburgh's Award.

Kerfoot, J. and Aragon, D. (2010). SLOCUM Glider Mission Writing. Technical report, Institute of Marine and Coastal Sciences, Rutgers University.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680.

Koenig, S. and Likhachev, M. (2002). D* Lite. In *Proceedings of the AAAI Conference of Artificial Intelligence (AAAI)*, pages 476–483.

Konsberg (2013). Autonomous underwater vehicle – HUGIN AUV. Konsberg Maritime.

Kruger, D., Stolkin, R., Blum, A., and Briganti, J. (2007). Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2007.*, pages 4265–4270.

Lagarias, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E. (1998). Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. *SIAM Journal of Optimization*, 9:112–147.

Lavalle, S. M. (1998). Rapidly-Exploring Random Trees: A New Tool for Path Planning. Technical report, Computer Science Dept, Iowa State University.

LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press, Cambridge, U.K. Available at `http://planning.cs.uiuc.edu`.

LaValle, S. M. and Jr., J. J. K. (1999). Randomized kinodynamic planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 473–479.

Lee Dowd Jr., G. (1927). The First Plane to Germany. *Popular Science*, 111(2):121.

Lekien, F. and Marsden, J. (2005). Tricubic Interpolation in Three Dimensions. *Journal of Numerical Methods and Engineering*, 63:455–471.

Lekien, F., Shadden, S. C., and Marsden, J. E. (2007). Lagrangian coherent structures in n-dimensional systems. *Journal of Mathematical Physics*, 48(6):19.

Leonard, N. E., Paley, D. A., Davis, R. E., Fratantoni, D. M., Lekien, F., and Zhang, F. (2010). Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay. *Journal of Field Robotics*, 27(6):718–740.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168.

Lolla, T., Ueckermann, M. P., Yigit, K., Haley, P. J., and Lermusiaux, P. F. J. (2012). Path planning in time dependent flow fields using level set methods. In *International Conference on Robotics and Automation (ICRA)*, pages 166–173.

Lopez, C. B., Dortch, Q., Jewett, E. B., and Garrison, D. (2008). Scientific Assessment of Marine Harmful Algal Blooms. Technical report, Interagency Working Group on Harmful Algal Blooms, Hypoxia, and Human Health of the Joint Subcommittee on Ocean Science and Technology, Washington, D. C.

Madow, W. G. (1949). On the Theory of Systematic Sampling, II. *The Annals of Mathematical Statistics*, 20(3):333–354.

Madow, W. G. (1953). On the Theory of Systematic Sampling, III. Comparison of Centered and Random Start Systematic Sampling. *The Annals of Mathematical Statistics*, 24(1):101–106.

Madow, W. G. and Madow, L. H. (1944). On the Theory of Systematic Sampling, I. *The Annals of Mathematical Statistics*, 15(1):1–24.

Mahmoudian, N. and Woolsey, C. (2008). Underwater Glider Motion Control. In *Proceedings of the* 47^{th} *IEEE Conference on Decision and Control*, pages 552–557.

Mahtani, A., Sánchez, L., Martínez, A., García, D., Morales, D., Fernández, E., Maniscalco, F., and Cabrera, J. (2012). AVORA I: The SAUC-E'12 Challenge. In *Proceedings of the Student Autonomous Underwater vehicle Challenge - Europe (SAUC-E 2012)*, La Spezia, Italy.

Mahtani, A., Sánchez, L., Martínez, A., García, D., Morales, D., Fernández, E., Maniscalco, F., and Cabrera, J. (2013a). AVORA I successful participation in SAUC-E'12. In *Proceedings of the Fourteenth International Conference on Computer Aided Systems Theory (Eurocast 2013)*, Las Palmas de Gran Canaria, Spain.

Mahtani, A., Sánchez, L., Martínez, A., García, D., Morales, D., Fernández, E., Maniscalco, F., and Cabrera, J. (2013b). AVORA I successful participation in SAUC-E'12. In Moreno-Díaz, R., Pichler, F., and Quesada-Arencibia, A., editors, *Computer Aided Systems Theory — EUROCAST 2013*, Lecture Notes in Computer Science. Springer Berlin Heidelberg.

Marsden, J. E. and Hoffman, M. J. (1993). *Elementary Classical Analysis*. W. H. Freeman.

Martin, J. P., Lee, C. M., Eriksen, C. C., Ladd, C., and Kachel, N. B. (2009). Glider observations of kinematics in a Gulf of Alaska eddy. *Journal of Geophysical Research: Oceans*, 114(C12).

Merckelbach, L., Smeed, D., and Griffiths, G. (2010). Vertical Water Velocities from Underwater Gliders. *Journal of Atmospheric and Oceanic Technology*, 27(3):547–563. doi: 10.1175/2009JTECHO710.1.

Milam, M. B., Mushambi, K., and Murray, R. M. (2000). New Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems. In *Conference on Decision and Control*.

Minguez, J., Member, A., and Montano, L. (2004). Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios. *IEEE Transactions on Robotics and Automation*, 20.

Moré, J. (1978). The Levenberg-Marquardt algorithm: Implementation and theory. In Watson, G. A., editor, *Numerical Analysis*, volume 630 of *Lecture Notes in Mathematics*, chapter 10, pages 105–116. Springer Berlin / Heidelberg.

Nash, A., Daniel, K., Koenig, S., and Felner, A. (2007). Theta*: Any-Angle Path Planning on Grids. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1177–1183.

Nash, A., Koenig, S., and Likhachev, M. (2009). Incremental Phi*: Incremental Any-Angle Path Planning on Grids. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1824–1830.

Nencioli, F., Dong, C., Dickey, T., Washburn, L., and McWilliams, J. C. (2010). A Vector Geometry-Based Eddy Detection Algorithm and Its Application to a High-Resolution Numerical Model Product and High-Frequency Radar Surface Velocities in the Southern California Bight. *Journal of Atmospheric and Oceanic Technology*, 27(1):564–579.

Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer series in operations research and financial engineering. Springer Science+Business Media, LLC.

Orr, J. C., Fabry, V. J., Aumont, O., Bopp, L., Doney, S. C., Feely, R. A., Gnanadesikan, A., Gruber, N., Ishida, A., Joos, F., Key, R. M., Lindsay, K., Maier-Reimer, E., Matear, R., Monfray, P., Mouchet, A., Najjar, R. G., Plattner, G.-K., Rodgers, K. B., Sabine, C. L., Sarmiento, J. L., Schlitzer, R., Slater, R. D., Totterdell, I. J., Weirig, M.-F., Yamanaka, Y., and Yool, A. (2005). Anthropogenic ocean acidification over the twenty-first century and its impact on calcifying organisms. *Nature*, 437(7059):681–686.

Papadimitriou, C. (1991). Shortest paths without a map. *Theoretical Computer Science*, 84(1):127–150.

Pearl, J. (1984). *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. The Addison-Wesley Series in Artificial Intelligence. Addison-Wesley.

Perdomo, E. F. (2009). Test and Evaluation of the FastSLAM Algorithm in a Mobile Robot. Master's thesis, Universidad de Las Palmas de Gran Canaria, Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI). Evaluación y prueba del Algoritmo FastSLAM sobre un robot Móvil.

Pohl, I. (1973). The avoidance of (relative) catastrophe, heuristic competence, genuine dynamic weighting and computational issues in heuristic problem solving. In *Proceedings of the 3rd international joint conference on Artificial intelligence*, IJCAI'73, pages 12–17, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Pohl, I., of Edinburgh. Dept. of Machine Intelligence, U., and Perception (1969). *First Results on the Effect of Error in Heuristic Search*. MIP-R-. Edinburgh University, Department of Machine Intelligence and Perception.

Pêtrès, C. (2007). *Trajectory Planning for Autonomous Underwater Vehicles*. PhD thesis, School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh, Scotland.

Pêtrès, 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.

Pêtrès, C., Pailhas, Y., Petillot, Y., and Lane, D. (2005). Underwater Path Planing using Fast Marching Algorithms. In *Oceans 2005 — Europe*, volume 2, pages 814–819.

Ramos, A., Cabrera, J., Fernández, E., Hernández, D., Coca, J., Redondo, A., Isern, J., Domínguez, A., Sotillo, M., Fanjul, E., Glenn, S., Schofield, O., and Kohut, J. (2011a). Pinzón path planning for the transoceanic glider Scarlet Knight RU-27 from the ESEOO Regional Ocean Model forecast of the NE Atlantic. In *Proceedings of the Fifth EGO (European Glider Observatory) Meeting and Glider School*, Las Palmas de Gran Canaria, Spain.

Ramos, A., Glenn, S., Schofield, O., Kohut, J., Cabrera, J., Kaminsky, D., Harrison, S., Fernández, E., Coca, J., Hernández, D., Barrera, C., Redondo, A., Domínguez, A., Isern, J., and Álvaro Lorenzo (2011b). Cook Crossing 2010: An international platform of research, cooperation and formation for future transoceanic glider missions. In *Proceedings of the Fifth EGO (European Glider Observatory) Meeting and Glider School*, Las Palmas de Gran Canaria, Spain.

Ramsay, A. and Richtmyer, R. (1995). *Introduction to Hyperbolic Geometry*. Ecological Studies. Springer.

Rao, D. and Williams, S. B. (2009). Large-scale path planning for Underwater Gliders in ocean currents. In *Australasian Conference on Robotics and Automation (ACRA)*, Sydney, Australia.

Regier, L. and Stommel, H. (1978). Memo: Some adavantages of SOFAR floats that move under their own power. Memo, Woods Hole Oceanographic Institution (WHOI).

Rhodes, M. (2006). Re-evaluating Wireless Capabilities. *Underwater Electromagnetic Propagation*, 10(10).

Ribas, D., Palomeras, N., Ridao, P., Carreras, M., and Mallios, A. (2012). Girona 500 AUV: From Survey to Intervention. *IEEE/ASME Transactions on Mechatronics*, 17(1):46–53.

Robel, A. (2010). Identification of Lagrangian Coherent Structures in the Florida Current Using Finite Time Lyapunov Exponents. Undergraduate thesis, Department of Physics, Duke University.

Rudnick, D. L., Davis, R. E., C. C. Eriksen, D. M. F., and Perry, M. J. (2004). Underwater Gliders for Ocean Research. *Marine Technology Society Journal*, 38(1):48–59.

Russell, S. and Norvig, P. (2010). *Artificial intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence. Pearson Education/Prentice Hall.

Sathyaraj, B. M., Jain, L. C., Finn, A., , and Drake, S. (2008). Multiple UAVs path planning algorithms: a comparative study. *Fuzzy Optimization Decision Making*, 7:257–267.

Shadden, S. C., Lekien, F., and Marsden, J. E. (2005). Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena*, 212(3–4):271–304.

Shampine, L. F., Gladwell, I., and Thompson, S. (2003). *Solving ODEs with MATLAB*. Cambridge University Press.

Simmons, R. and Urmson, C. (2003). Approaches for heuristically biasing RRT growth. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1178–1183.

Simonetti, P. (1992). SLOCUM GLIDER: Design and 1991 Field Trials. Technical report, Webb Research Corp., Woods Hole Oceanographic Institution, Office of Naval Technology.

Smith, R., Chao, Y., Li, P., Caron, D., Jones, B., and Sukhatme, G. (2010). Planning and implementing trajectories for Autonomous Underwater Vehicles to track evolving ocean processes based on predictions from a Regional Ocean Model. *International Journal of Robotics Research*, 29(12).

Sniedovich, M. (2006). Dijkstra's algorithm revisited: the dynamic programming connexion. *Control and cybernetics*, 35(3):599–620.

Sotillo, M. G., Jordi, A., Ferrer, M. I., Conde, J., Tintoré, J., and Álvarez Fanjul, E. (2007). The ESEOO Regional Ocean Forecasting System. In *Proceedings of the* 17^{th} *International Offshore Ocean and Polar Engineering Conference (ISOPE-2007)*.

Sotillo, M. G., Álvarez Fanjul, E., Castanedo, S., Abascal, A. J., Menendez, J., Emelianov, M., Olivella, R., García-Ladona, E., Ruiz-Villarreal, M., Conde, J., Gómez, M., Conde, P., Gutierrez, A. D., and Medina, R. (2008). Towards an operational system for oil spill forecast in Spanish coastal waters: initial developments and implementation test. *Marine pollution Bulletin*, 1(56):686–703. DOI: 10.1016/j.marpolbul.2007.12.021.

Soulignac, M. (2010). Feasible and Optimal Path Planning in Strong Current Fields. *IEEE Transactions on Robotics*, 27(1):89–98.

Soulignac, M., Taillibert, P., and Rueher, M. (2008). Adapting the wavefront expansion in presence of strong currents. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, pages 1352–1358.

Soulignac, M., Taillibert, P., and Rueher, M. (2009). Time-minimal Path Planning in Dynamic Current Fields. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*.

Stachniss, C. (2009). *Robotic Mapping and Exploration*. Springer Tracts in Advanced Robotics. Springer.

Stentz, A. and Mellon, I. C. (1993). Optimal and Efficient Path Planning for Unknown and Dynamic Environments. *International Journal of Robotics and Automation*, 10:89–100.

Tan, C. S., Sutton, R., and Chudley, J. (2004). An incremental stochastic motion planning technique for autonomous underwater vehicles. In *Proceedings of IFAC Control Applications in Marine Systems Conference*, pages 483–488.

Techy, L., Woolsey, C., and Morgansen, K. (2010). Planar Path Planning for Flight Vehicles in Wind with Turn Rate and Acceleration Bounds. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*.

Thompson, D. R., Chien, S., Arrott, M., Balasuriya, A., Chao, Y., Li, P., Meisinger, M., Petillo, S., and Schofield, O. (2009). Mission Planning in a Dynamic Ocean Sensorweb. In 19^{th} *International Conference on Automated Planning and Scheduling (ICAPS) — Scheduling and Planning Applications woRKshop (SPARK)*.

Thompson, D. R., Chien, S., Chao, Y., Li, P., Cahill, B., Levin, J., Schofield, O., Balasuriya, A., Petillo, S., Arrott, M., and Meisinger, M. (2010). Spatiotemporal Path Planning in Strong, Dynamic, Uncertain Currents. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4778–4783.

Tsou, M.-C. and Hsueh, C.-K. (2010). The Study of Ship Collision Avoidance Route Planning by Ant Colony Algorithm. *Journal of Marine Science and Technology*, 18(2):746–756.

Walsh, K. and Banerjee, B. (2010). Fast A* with Iterative Resolution for Navigation. *International Journal on Artificial Intelligence Tools (IJAIT)*, 19(1):101–119.

Wang, D. (2007). *Autonomous Underwater Vehicle (AUV) Path Planning and Adaptive On-board Routing for Adaptive Rapid Environmental Assessment*. Ph.d. dissertation, Massachusetts Institute of Technology.

Widditsch, H. R. (1973). SPURV — The First Decade. Technical report, University of Washington, Applied Physics Laboratory.

Witt, J. and Dunbabin, M. (2008). Go With the Flow: Optimal AUV Path Planning in Coastal Environments. In *Proceedings of the Australian Conference on Robotics and Automation*.

Yilmaz, N. K., Evangelinos, C., Lermusiaux, P. F. J., and Patrikalakis, N. M. (2008). Path Planning of Autonomous Underwater Vehicles for Adaptive Sampling Using Mixed Integer Linear Programming. *IEEE Journal of Oceanic Engineering*, 33(4):522–537.

Zhang, F., Fratantoni, D. M., Paley, D. A., Lund, J. M., and Leonard, N. E. (2007). Control of coordinated patterns for ocean sampling. *International Journal of Control*, 80(7):1186–1199.

Zhang, W., Inane, T., Ober-Blobaum, S., and Marsden, J. E. (2008). Optimal trajectory generation for a glider in time-varying 2D ocean flows B-spline model. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2008.*, pages 1083–1088.

Part I

Resumen Extendido en Español

Capítulo 1

Introducción

La oceanografía es la disciplina que se encarga del estudio de los océanos. Se trata de un ámbito complejo en el que convergen múltiples áreas de interés como son el estudio de las corrientes marinas, el oleaje o la dinámica de fluidos; tectónica de placas y geología del fondo marino; organismos marinos y dinámica de ecosistemas; difusión de sustancias químicas y propiedades físicas del océano.

El estudio de los diferentes procesos presentes en el medio marino requiere la definición y ajuste de modelos que permitan caracterizar su dinámica y ayudar a predecir su evolución. Un aspecto clave, del que depende en gran medida el éxito de esta tarea, es la necesidad de disponer de sistemas de muestreo adecuados que suministren datos reales con los que contrastar la bondad de dichos modelos. Esta tesis se centra en los aspectos ligados al uso de dispositivos robóticos en la monitorización del medio marino.

1.1 Técnicas de muestreo

Tradicionalmente se han empleado en la observación del océano diferentes sistemas de captura de datos como son la tele-detección basada en satélite, radares, boyas o los buques oceanográficos. En las últimas décadas han surgido nuevas técnicas de muestreo que intentan suplir diferentes limitaciones identificadas en los sistemas más tradicionales. Se trata de los vehículos autónomos submarinos.

Podemos distinguir dos tipos principales de vehículos submarinos: los operados remotamente o ROVs, y los autónomos o AUVs. Dentro de estos últimos, existe un tipo especial, que son los planeadores submarinos o gliders. El presente trabajo está enfocado a proporcionar herramientas destinadas a este tipo de vehículos.

1.2 Planeadores Submarinos Autónomos

Los Planeadores Submarinos Autónomos, también denominados comúnmente planeadores o *gliders* —atendiendo al término en inglés—, son un tipo de vehículo autónomo submarino que se desplaza usando un mecanismo de propulsión único basado en modificar su flotabilidad. Estos vehículos poseen un sistema que emula el funcionamiento de la vejiga natatoria de los peces para provocar movimientos de ascenso y descenso cíclicos. Dichos desplazamientos verticales se traducen en una lenta pero efectiva velocidad de avance gracias a la interacción de los planos de control y sustentación con el medio acuático. Se generan de esta manera perfiles de movimiento en "V" característicos, denominados yo-yos.

Figura 1.1: Spray de Bluefin Robotics (delante), SLOCUM de Teledyne Webb Research (medio) y Seaglider de iRobot (detrás), en el Laboratorio de Gliders en las instalaciones de la Plataforma Oceánica de Canarias (PLOCAN). Cortesía de PLOCAN. Nótese que las alas son extraíbles para facilitar el transporte y mantenimiento, y por esa razón han sido extraídas del casco en los vehículos aquí mostrados.

Un glider eléctrico, como el modelo SLOCUM de la Figura 1.1, utiliza una bomba hidráulica para trasvasar un cierto volumen de aceite entre el interior y el exterior del casco del vehículo. De esta forma, el consumo de energía necesario para la propulsión se produce únicamente en los puntos de inflexión en los que el robot activa la bomba, permitiendo unos niveles de autonomía difíciles de igualar para otros vehículos.

Los gliders operan en base a la definición de unos puntos de paso o waypoints que el robot trata de alcanzar

sucesivamente. Una vez fijado el rumbo hacia el próximo objetivo el vehículo ejecuta una serie predefinida de ciclos de ascenso/descenso, denominados transecto o stint; y que están delimitados entre dos valores de profundidad mínima y máxima, tras los cuales emerge de nuevo. Una vez en superficie, el glider se localiza por medio del GPS —o por cualquier otro GNSS—, transmite datos básicos a su centro de control y, en su caso, recibe nuevas órdenes. Transcurrido un corto periodo de tiempo, en torno a $15 - 20$ minutos, el glider inicia un nuevo stint (tramo sumergido), cuya duración suele oscilar entre 6 y 8 horas.

1.3 Planificación de rutas

El principal inconveniente que tiene el uso de los gliders en la monitorización del medio marino es su dependencia de las condiciones del medio. La lenta velocidad de avance que los caracteriza (aproximadamente un kilómetro por hora) puede llegar a ser igualada e incluso superada por las corrientes marinas. Por lo tanto, es necesario anticipar su efecto en la trayectoria del vehículo si se desea conseguir alcanzar un punto determinado o seguir una ruta deseada. La fuente de información básica con la que se trabaja son los modelos de predicción oceánicos, que permiten tener en cuenta con varios días de antelación el estado más probable de la zona en la que se encuentre el glider.

Dependiendo del tipo de misión y la región que se esté considerando, existirán diferentes modelos disponibles. En general, los mejores resultados son los aquellos que se obtienen con modelos de predicción regionales, especialmente ajustados para capturar las condiciones particulares de una determinada zona.

En un esquema de pilotaje de gliders tradicional, un operador humano realiza un seguimiento del estado presente y futuro del mar y de la evolución del vehículo, decidiendo cuándo es necesario modificar los puntos de destino. En una misión real, las propias medidas del glider se convierten en un indicio de si las predicciones de los modelos oceánicos se están verificando o no. La Figura 1.2 ilustra la trayectoria final del glider RU27 tras finalizar la misión trans-Atlántica. Ésta se llevo a cabo mediante un pilotaje manual o tradicional. No obstante, durante los últimos stints cerca de las costas españolas, este pilotaje fue asistido por las herramientas de planificación de caminos para gliders desarrolladas en este trabajo de tesis.

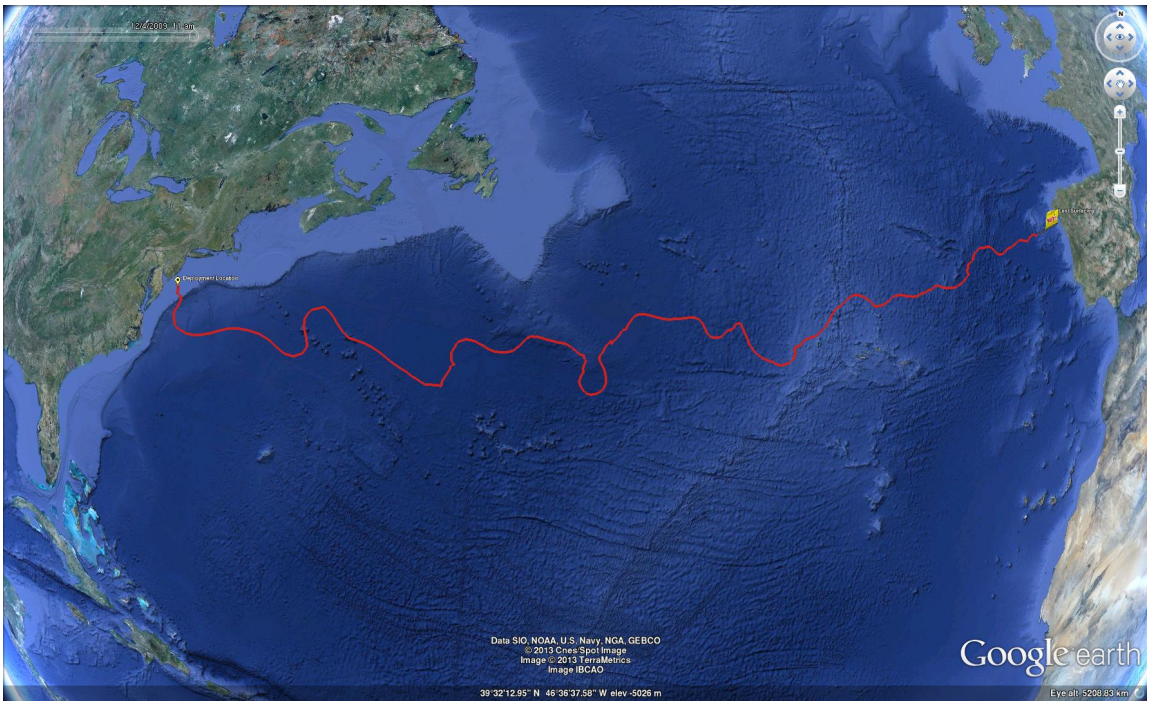


Figura 1.2: Camino seguido por el glider RU27 *Scarlet Knight* (SLOCUM) en la misión trans-Atlántica, realizada en 2010. La missión fue dirigida por la Universidad de Rutgers, con la colaboración de la División de Robótica y Oceanografía Computacional (DROC), SIANI.

1.4 Hipótesis de la tesis

En este trabajo se plantea como hipótesis el que pueden definirse mecanismos automáticos para la planificación de rutas en gliders que aporten ventajas significativas en el uso de estos dispositivos como instrumentos de muestreo del océano.

1.5 Objetivos

Los objetivos de esta tesis se resumen en las siguientes cuestiones que trata de resolver. Asimismo, definen el ámbito y extensión del presente trabajo.

- La literatura sobre planificaciones de caminos recoge un importante número de algoritmos usados para obtener la secuencia (cuasi-)óptima de configuraciones de estado que permita a un vehículo moverse desde una posición a otra. Es por tanto de gran interés estudiar si estos algoritmos son aplicable en el campo de la planificación de rutas para gliders. De hecho, en esta tesis analizamos si el método A* es

aplicable. Si puede ser adapta al problema con gliders, o si hay alguna particularidad que lo impide, o hace que resulte ineficiente.

- Veremos como las técnicas clásicas no son directamente aplicables a nuestro problema, a menos que sea adaptadas. En este sentido trataremos de analizar cuáles son las particularidades de la planificación de caminos con gliders que justifican la necesidad de esta adaptación. Así, obtendremos conocimiento sobre las siguientes cuestiones:
  - Cuáles son las diferencias respecto a otros problemas que se solucionan empleando planificadores de caminos clásicos? Nos referimos a problemas como la planificación de caminos en laberintos, mapas de coste, etc.
  - Qué razones explican por qué las técnicas clásicas no son aplicables o útiles en este caso? O que impide que puedan manejar eficientemente el problema?
- El estudio de la bibliografía y el estado del arte nos permitirá categorizar la tipología del problema que intentamos resolver. Es decir, en qué categoría taxonómica ubican el problema otros autores, o al menos establecer el tipo de problema más similar.
- Desde un punto de vista más práctico, pretendemos investigar y desarrollar métodos que resuelvan el problema. En este trabajo implementamos varios técnicas novedosas. Éstas son evaluadas conforme a dos criterios. Primero, evaluamos la optimalidad de la solución encontrada o lo lejos que queda del óptimo. Segundo, el tiempo de cómputo y uso de memoria.
- De esta forma, podremos comparar nuestros resultados con otras aproximaciones de la literatura sobre el tema. Así podremos establecer cómo atacan el problema otros autores, así como las simplificaciones y los supuestos que éstos asumen. Consecuentemente, podremos valorar si éstos y nuestros algoritmos son escalables y generalizables para resolver otro tipo de problemas de planificación de rutas.
- Respecto a la dimensionalidad del problema, veremos que los modelos oceánicos ofrecen 4 dimensiones: espacio tridimensional y tiempo. Los gliders navegan por este espacio tridimensional que cambia significativamente en el tiempo. Asimismo, es posible considerar o despreciar ciertas características de la dinámica del vehículo y su navegación, como el ángulo de planeo, las profundidades mínima y máxima de los perfiles yo-yo, la velocidad nominal, etc. La forma en que manejar toda esta complejidad

y la alta dimensionalidad del problema eficientemente, es también un aspecto fundamental que se trata en el presente trabajo. Igualmente, queremos ver si es posible garantizar la optimalidad de la solución encontrada, o simplemente hasta cierto punto y bajo ciertas suposiciones.

- A partir de la implementación y evaluación de deferentes técnicas obtendremos cierta intuición respecto a cuáles son más apropiadas para determinadas condiciones y entornos. Por ello, también se realiza una categorización de escenarios: áreas con obstáculos, zonas de alta variabilidad temporal, corrientes fuertes, etc.

Las cuestiones anteriores definen el ámbito y los objetivos de la tesis, puesto que ésta trata de darles respuesta. Adicionalmente, para poder tratar con la complejidad del problema, se han asumido ciertas simplificaciones y marcado algunas limitaciones.

- La alta dimensionalidad del problema se ha reducido hasta cierto punto. Asumimos un modelo de movimiento del glider simplificado. En algunos casos la planificación de rutas se hace considerando sólo las corrientes bidimensionales en superficie, con una resolución temporal de horas; mientras que en otros casos tomamos corrientes tridimensionales, aunque como medias diarias. Esta reducción en la dimensionalidad del problema es habitualmente aceptable, al mismo tiempo que reduce la complejidad sustancialmente.

- Algunos detalles de las misiones y el pilotaje de gliders son despreciados. En muchos casos no es posible observar fenómenos como el *biofouling*, que afecta a la velocidad y equilibrado del glider. Esto produce un cambio en el ángulo de planeo y puede producir una desviación sistemática en el rumbo. Del mismo modo, varía el centro de masas y la flotabilidad, reduciéndose la velocidad del vehículo. En otros casos, sólo podemos estimar ciertos parámetros con un determinado error, como el tiempo que permanece el vehículo en superficie. Estos valores no se suelen modelar porque en realidad nos interesan otros aspectos del sistemas y podemos ignorar sus efectos en la mayoría de casos.

- El sistema embebido y el software en los gliders modernos tiene capacidades muy reducidas. Por ello en la práctica el pilotaje automático o autónomo no es trivial. Aunque no se trata estrictamente de una limitación, hemos recogido este hecho en nuestra simulaciones. Puesto que la calidad del *dead-reckoning* es mala, las comunicaciones no son posibles, y la deriva producida por las corrientes no es observable bajo el agua, se requiere una planificación de rutas con comandos discretos espaciados significativamente

en el tiempo. Incluso si fuera posible variar el rumbo del vehículo con mayor frecuencia, el vehículo aún no podría disponer de los mapas de corriente abordo para actualizar el rumbo convenientemente.

En la tesis se aborda en primer lugar el planteamiento y metodología, describiendo el vehículo y el modelado de su movimiento. Posteriormente se analizan diferentes problemas y aplicaciones de interés. Se pasa entonces a la exposición de los algoritmos de planificación actualmente existentes y, a continuación, las propuestas desarrolladas en el marco de esta tesis. Seguidamente se describen los experimentos realizados con vistas a la validación de las propuestas presentadas, para concluir con los resultados y conclusiones.

Capítulo 2

Planteamiento y Metodología

El planteamiento y metodología del problema comienza aquí con el estudio de los planeadores oceánicos y sus modelos de movimiento. Sin embargo, hay que entender que se extiende a los capítulos siguientes, con el estudio de los problemas de planificación de caminos para gliders y los algoritmos propuestos en el presente trabajo de tesis.

2.1 Planeadores Oceánicos

Disponer de un modelo adecuado del comportamiento de un glider es un requisito indispensable para poder realizar simulaciones fiables. De esta forma, las trayectorias planificadas automáticamente tendrán más validez al presentar una buena correspondencia con las realizadas por el robot en misiones reales.

La Figura 2.1 y 2.2 muestran el tipo de navegación que realizan los gliders, así como las distintas fases de las que ésta se compone. La primera figura ilustra el perfil yo-yo y el patrón en forma de diente de sierre que sigue el glider mientras navega sumergida, usando la vejiga de aceite y la bomba de flotabilidad. La figura recoge los valores más comunes para el intervalo de profundidades durante un tramo sumergido y el tiempo que permanece en superficie. La segunda figura muestra las fases de un tramo de navegación del glider. Esto incluye las siguientes fases:

1. La emersión y localización con el GPS.

2. La deriva producida por las corrientes oceánicas y el viento en superficie durante $t_s \approx 15\mathrm{min}$, mientras

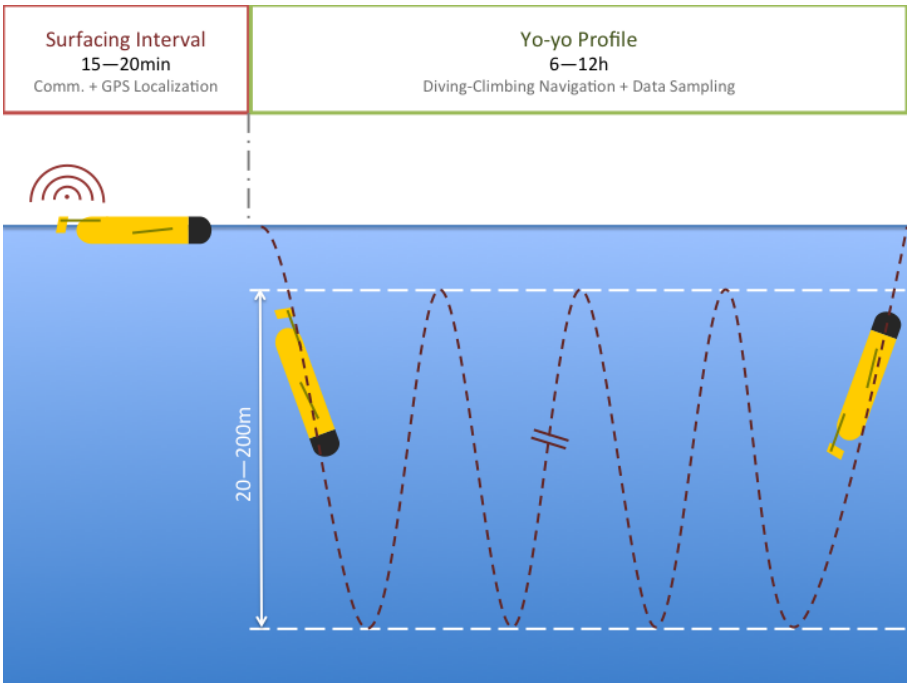


Figura 2.1: Diagrama del perfil yo-yo y el patrón de diente de sierra seguido por un glider mientras navega sumergido.

también se comunica con la estación base.

3. La inmersión.

4. El perfil yo-yo durante el periodo de navegación sumergido, con puntos de inflexión en el intervalo de profundidad $[z_{\text{mín}}, z_{\text{máx}}]$ dado.

En superficie, la localización del glider se resuelve usando el GPS, pero después del punto de inmersión es desconocida. Las elipses representan la incertidumbre de la estimación de la posición bajo el agua. La principal fuente de incertidumbre es la deriva causada por las corrientes oceánicas. En cada punto de emersión dicha incertidumbre colapsa con el primer *fix* GPS.

2.2 Modelos de Movimiento

Se analizan tres modelos: modelo puntual, equilibrio de fuerzas y dinámico.

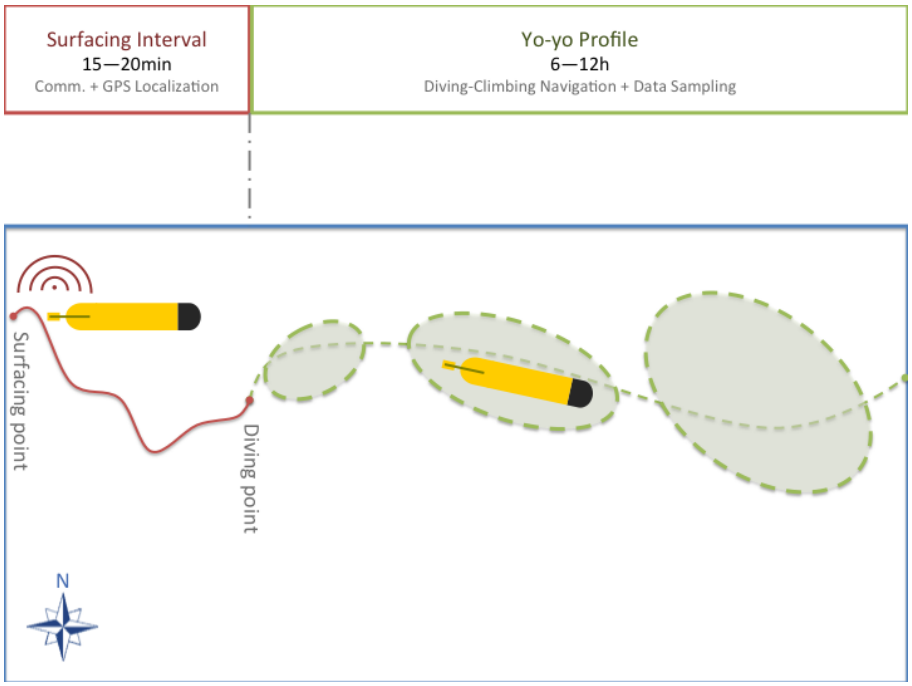


Figura 2.2: Diagrama de un tramo de navegación de un glider.

2.2.1 Modelo de Movimiento puntual

El modelo puntual asume el esquema más simple, donde el glider se modela como un punto sin masa. La trayectoria se integra a partir de la combinación del rumbo del glider con las corrientes de la zona en que navega, proporcionadas por un modelo numérico de predicción. Se trata de un modelo simple que contiene un único parámetro de configuración: la velocidad nominal del glider.

Se plantean dos posibles variantes para este modelo, dependiendo de las restricciones impuestas a la simulación de la navegación. En la variante no restringida, la trayectoria se integra a partir de una composición vectorial directa del rumbo del glider y su velocidad con el campo de corrientes. En la variante restringida, el rumbo del glider no viene dado, sino que se precalcula a partir de la definición de un punto inicial y un punto objetivo en la trayectoria. Ambas variantes del modelo de movimiento puntual se describen en las siguientes secciones.

2.2.1.1 Modelo no restringido

La velocidad efectiva $\mathbf{U}_e$ se computa usando

$$\mathbf{U}_e = \mathbf{U}_g + \mathbf{U}_c \tag{2.1}$$

donde $\mathbf{U}_g$ es la velocidad nominal del glider y $\mathbf{U}_c$ es la velocidad de la corriente a lo largo del camino $\mathcal{P}$ descrito por el glider.

Puesto que $\mathbf{U}_c$ es diferente en cada punto del entorno, debemos integrar la velocidad a lo largo de $\mathcal{P}$. Por tanto, tenemos

$$\mathbf{U}_e = \mathbf{U}_g + \int_{\mathbf{x}_{i-1}}^{\mathbf{x}_i} \mathbf{U}_c(\mathbf{x})\, d\mathbf{x} \tag{2.2}$$

para un camino $\mathcal{P}$ que empiece en $\mathbf{x}_{i-1}$ y termine en $\mathbf{x}_i$, que corresponderían con puntos de emersión.

Algorithm 2.1 Integración de la trayectoria

Require: Tiempo entre emersiones t_s y paso de integración dt. Velocidad nominal del glider $\mathbf{U}_g$, campo de velocidad de las corrientes oceánicas $\mathbf{U}_c(\mathbf{x})$ para todas las localizaciones $\mathbf{x}$, y localización inicial $\mathbf{x}_0$.
Ensure: Localización final $\mathbf{x}_{t_s}$ después de integrar la trayectoria para el tiempo t_s.
Algorithm: trajectory(t_s, dt, $\mathbf{U}_g$, $\mathbf{U}_c(\mathbf{x})$, $\mathbf{x}_0$) **return** $\mathbf{x}_{t_s}$

```
x = x_0                                   ▷ Localización inicial
for all t = 0 to t_s with step dt do
    U_e = U_g + U_c(x)
    x = x + U_e · dt
end for
return x
```

La integral a lo largo del camino $\mathcal{P}$ se discretiza e integra numéricamente durante un tiempo fijo t_s entre emersiones. Dicha integración permite obtener la trayectoria del glider bajo las condiciones del campo de corrientes oceánicas, como ilustra la Figura 2.3. Por tanto, implementamos la siguiente recursión

$$\mathbf{U}_e(t) = \mathbf{U}_g + \mathbf{U}_c(\mathbf{x}(t), t) \tag{2.3}$$

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \mathbf{U}_e(t) \cdot \Delta t \tag{2.4}$$

donde Δt es el paso de integración y $\mathbf{x}(0)$ se inicializa con la localización inicial del vehículo. El algoritmo de integración se resume en el Algoritmo 2.1. Resolvemos esta Ecuación Diferencial Ordinaria mediante las fórmulas de Runge-Kutta de orden 4 y 5 (Dormand and Prince, 1980) de manera numérica, sin necesidad de

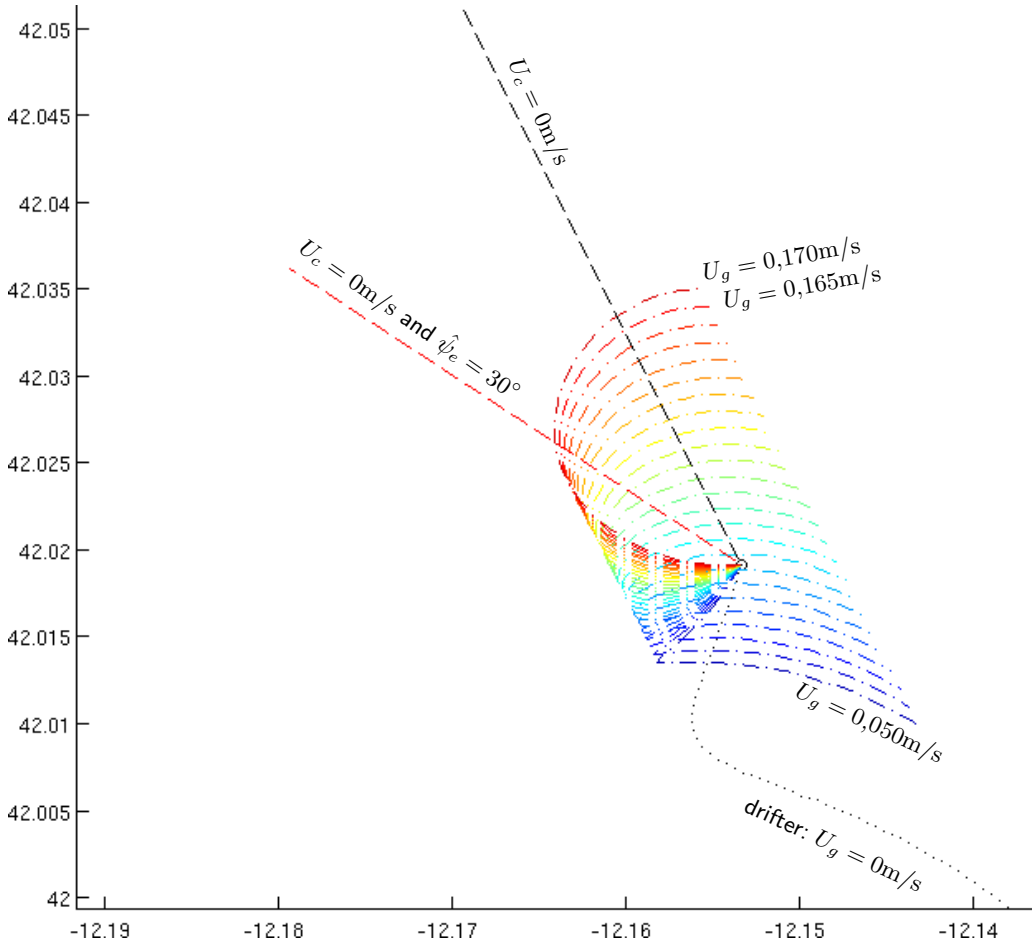


Figura 2.3: Ejecución del modelo de movimiento puntual no restringido para diferentes velocidades del glider U_g, para un error de rumbo (*heading*) constante de $\hat{\psi}_e = 30°$N. Observamos la deriva en las trayectorias, efecto de las corrientes oceánicas, desde $U_g = 0{,}050\text{m/s}$ a $U_g = 0{,}170\text{m/s}$ ▮. A medida que U_g se reduce, tienden a la trayectoria de un derivador (*drifter*) con $U_g = 0\text{m/s}$ ·····. Al contrario, la trayectoria sin corrientes $U_c = 0\text{m/s}$ sería recta, como se muestra sin error de rumbo - - - y $\hat{\psi}_e = 30°$ - - -.

indicar Δt. En este proceso, interpolamos las corrientes $\mathbf{U}_c(\mathbf{x}(t), t)$ en cada localización $\mathbf{x}(t)$ de la trayectoria y tiempo t.

2.2.1.2 Modelo restringido

El modelo restringido funciona de forma radicalmente diferente. Ahora básicamente resolvemos el problema de navegación de ir de una localización inicial $\mathbf{x}_0$ a una final $\mathbf{x}_1$. El ángulo entre $\mathbf{x}_0$ y $\mathbf{x}_1$ define un rumbo ψ_e que queremos que el vehículo siga. Así, dado el vector de corrientes $\mathbf{U}_c$ y la velocidad nominal horizontal del

glider U_g, el problema de navegación consiste en encontrar la dirección (*bearing*) ψ_g necesaria para movernos en ψ_e, si es posible. De hecho, cuando las corrientes son más fuertes que U_g, existe un rango de rumbos ψ_e que no son factibles, i.e. no existe ninguna dirección ψ_g tal que permita al vehículo alcanzar $\mathbf{x}_1$ poniendo rumbo ψ_e.

Dada la velocidad nominal del glider U_g, el rumbo deseado ψ_e, y la velocidad U_c y dirección ψ_c de la corriente, podemos computar la corrección de la deriva que el glider debe aplicar para mantener el rumbo. Así, obtenemos la dirección ψ_g con

$$\psi_g = \begin{cases} \psi_e + \arcsin s & \text{if } |s| \leq 1 \\ \nexists & \text{en otro caso} \end{cases} \tag{2.5}$$

y la velocidad resultante U_e con

$$U_e = \begin{cases} U_g\sqrt{1-s^2} + U_c \cos\psi_{ec} & \text{if } |s| \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

donde

$$s = \frac{U_c}{U_g} \sin\psi_{ec} \tag{2.7}$$

y $\psi_{ec} = \psi_e - \psi_c$. Al margen de la diferencia de ángulos ψ_{ec}, podemos definir el ratio de velocidad SR como

$$SR = \frac{U_c}{U_g} \tag{2.8}$$

que permite categorizar las corrientes en fuertes o débiles, dependiendo de si $SR > 1$ o $SR < 1$, respectivamente.

Consecuentemente, si $U_g > U_c$ el conjunto de rumbos factibles ψ_e no tiene límites. De lo contrario, está acotado por el conocido *cono de accesibilidad* con un arco de ángulo ψ_r dado por

$$\psi_r = 2\arcsin\frac{U_g}{U_c} \tag{2.9}$$

Para el caso particular de $U_g = U_c$ tenemos $\psi_r = \frac{\pi}{2}$. Luego, el rumbo ψ_e es físicamente factible si y sólo si

cae en $\psi_e \in [-\frac{\psi_r}{2}, \frac{\psi_r}{2}]$.

Nótese que e el caso particular de $U_g = U_c$ la condición $|s| \leq 1$ es siempre verdadera, puesto que $s = \sin\psi_{ec}$. Operando, (2.5) y (2.6) se reducen a

$$\psi_g = \begin{cases} 2\psi_e - \psi_c & \text{if } |\psi_{ec}| \leq \dfrac{\pi}{2} \\ \nexists & \text{otherwise} \end{cases} \tag{2.10}$$

y

$$U_e = \begin{cases} 2U_g \cos\psi_{ec} & \text{if } |\psi_{ec}| \leq \dfrac{\pi}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

respectivamente, cumpliendo $U_e \geq 0$.

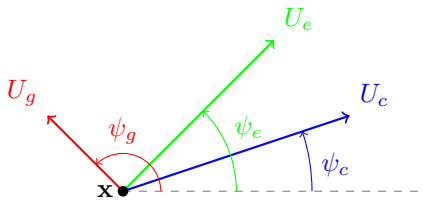


Figura 2.4: Composición de vectores para la corrección de la deriva. Una dirección (*bearing*) ψ_g debe comandarse para obtener el rumbo (*heading*) ψ_e tal que compensa una corriente en la dirección ψ_c. La velocidad efectiva resultante U_e depende de la velocidad del glider U_g y de la corriente U_c.

En la Figura 2.4 se muestra la composición de vectores del proceso de corrección de la deriva. Nótese que todas estas velocidades $U\cdot$ no son vectores, sino magnitudes $\|\mathbf{U}\cdot\|$. Asumimos $U_e = 0$ cuando no es posible mantener un rumbo ψ_e. Esto es normalmente consecuencia de una velocidad del glider relativamente baja U_g respecto a la velocidad de la corriente U_c, que fuerza al glider a navegar siguiendo su dirección ψ_c.

2.2.2 Modelo de Movimiento de Equilibrio de Fuerzas

En el modelo de equilibrio de fuerzas se parte de una caracterización de los cambios de flotabilidad del glider para estimar la velocidad de ascenso/descenso del vehículo. A partir de ahí se obtiene la velocidad de avance que es posible integrar para generar la trayectoria simulada. Se trata de un modelo más elaborado, en el que existen multitud de parámetros que es preciso identificar adecuadamente a fin de obtener una simulación fiable.

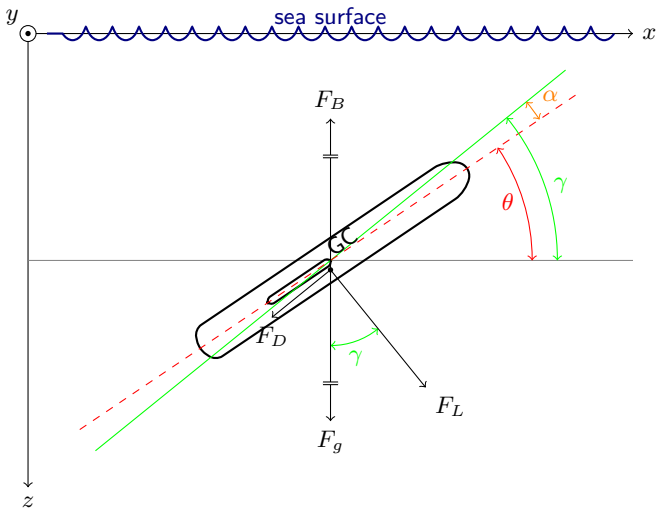


Figura 2.5: Diagrama de equilibrio de fuerzas en un glider que representa el movimiento de un glider en el plano vertical xz junto con las fuerzas que actúan y la definición de ángulos.

Las fuerzas que intervienen en un glider son la flotabilidad F_B, que va hacia arriba a la superficie; la gravedad F_g, que va hacia abajo en dirección opuesta a F_B; la sustentación (*lift*) F_L, que va hacia delante y abajo o arriba, cuando el glider emerge o se sumerge, respectivamente; y el arrastre (*drag*) F_D, que va hacia atrás, en contra del movimiento. La sustentación es responsable de generar el movimiento de avance del vehículo, mientras sea suficiente para compensar la fuerza de arrastre F_D. El ángulo de cabeceo θ representa el ángulo del eje horizontal del vehículo con respecto al plano horizontal. El ángulo de planeo γ representa el ángulo del plano de ascenso/descenso con respecto al plano horizontal, e incluye el ángulo de ataque α, que es producido por la dinámica del diseño de las alas y el casco. Por tanto, la diferencia entre el ángulo de planeo γ y el cabeceo θ es el ángulo de ataque α.

De acuerdo con el diagrama de la Figura 2.5, los equilibrios de fuerzas verticales y horizontales son

$$F_B - \cos\gamma F_L - \sin\gamma F_D - F_g = 0 \tag{2.12}$$

$$-\cos\gamma F_D + \sin\gamma F_L = 0 \tag{2.13}$$

respectivamente, donde F_B es la fuerza de flotabilidad (neta), F_g es la fuerza debida a la gravedad, F_L es la fuerza de sustentación, F_D es la fuerza de arrastre y γ es el ángulo de planeo, que se define como la suma del ángulo de cabeceo θ y el de ataque α, i.e. $\gamma = \theta + \alpha$.

Tras operar, la velocidad U se puede evaluar como

$$U = \sqrt{\frac{2\sin\gamma(F_B - F_g)}{\rho S(C_{D_0} + C_{D_1}\alpha^2)}} \tag{2.14}$$

Dado el rumbo ψ_e del glider, como teníamos para el modelo de movimiento puntual, las componentes horizontal y vertical del vector de velocidad son

$$u_g = U\cos\gamma\cos\psi_e \tag{2.15}$$

$$v_g = U\cos\gamma\sin\psi_e \tag{2.16}$$

$$w_g = U\sin\gamma \tag{2.17}$$

respectivamente, donde $\gamma = \theta + \alpha$. La velocidad u_g corresponde al eje x o longitud, y v_g se mapea al eje y o latitud. Nótese además que la velocidad de avance del glider U_g es simplemente la composición de las velocidades horizontales u_g y v_g, que da

$$U_g = U\cos\gamma \tag{2.18}$$

Con esta expresión de la velocidad del glider U, la mayor velocidad de avance U_g se obtiene para aproximadamente $\theta = 35°$. De hecho, es posible crear una gráfica *spider* para el modelo de movimiento de equilibrio de fuerzas, como muestra la Figura 2.6 para un perfil yo-yo de descenso. Este gráfico representa la velocidad de avance horizontal U_g frente a la velocidad vertical w_g para varios cambios de flotabilidad ΔV_{bp} de la vejiga. Junto con las curvas para diferentes ΔV_{bp} obtenemos iso-líneas de mismo ángulo de ataque α. Además, podemos obtener el cabeceo θ usando el hecho de que $\lambda = \theta + \alpha$. La figura muestra u_g y v_g para $\Delta V_{bp} \in [50, 300]$cc —que es negativo en los descensos. Como era esperable, la gráfica muestra un claro incremento de la velocidad del glider al aumentar los cambios de flotabilidad ΔV_{bp}, puesto que producen una mayor fuerza de flotabilidad F_B. De manera similar, pequeños ángulos de ataque α incrementan la velocidad vertical w_g, pero reducen la de avance nominal U_g.

2.2.3 Modelo de Movimiento Dinámico

El modelo dinámico constituye la aproximación más elaborada, al tener en cuenta la combinación de todas las fuerzas implicadas en el sistema físico. De esta forma, se considera la fuerza de la gravedad, la flotabilidad,

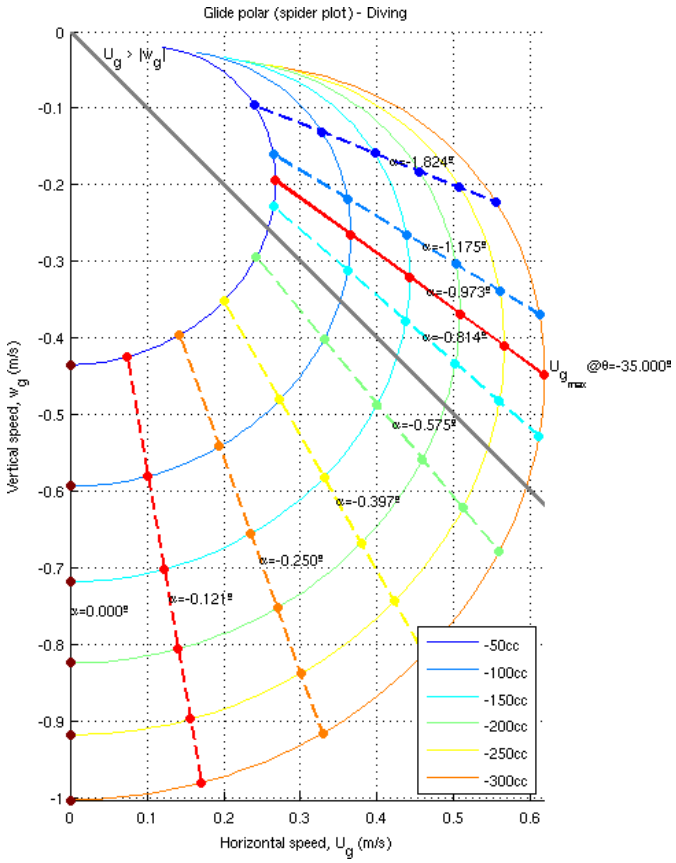


Figura 2.6: Gráfica *spider* para el modelo de movimiento de equilibrio de fuerzas durante una inmersión. Muestra las velocidad de avance del glider U_g frente a la velocidad vertical w_g para diferentes cambios de flotabilidad ΔV_{bp} —comúnmente la capacidad de la vejiga o la bomba de flotabilidad. Los valores por encima de la línea $U_g > |w_g|$ se corresponden con las velocidades de avance U_g mayores, y los valoers más comunes de cabeceo θ se toman de ese semiplano. De hecho, la mayor U_g se obtiene con $\theta \approx -35°$, que da $U_g \approx 0{,}5\mathrm{m/s}$ para $\Delta V_{bp} = -200\mathrm{cc}$, que concuerda con la velocidad de planeo típica de los gliders SLOCUM, para dicho cabeceo y configuración de la bomba de flotabilidad.

la hidrodinámica, etc.

2.3 Análisis

Los diferentes modelos representan un compromiso entre exactitud de la simulación y dificultad de calibración. En general, para el tipo de aplicaciones abordadas en esta tesis, el modelo puntual es la elección más adecuada, puesto que permite obtener resultados razonablemente aproximados sin requerir un esfuerzo de identificación de parámetros excesivo.

Capítulo 3

Problemas y Aplicaciones

A lo largo de esta tesis se han ido identificando un conjunto de problemas y aplicaciones vinculadas a los gliders que se han considerado relevantes por las contribuciones que su resolución aporta a estos vehículos. En un primer nivel de problemas básicos está la optimización de rutas, que puede verse como un recurso capacitante para abordar otras aplicaciones más complejas. En un segundo nivel se consideran la evitación de obstáculos, el seguimiento de trayectorias, la planificación multi-glider o el muestreo de estructuras oceánicas móviles.

3.1 Optimización de Rutas

El problema básico que se debe resolver es la optimización de la ruta que debe seguir un glider para alcanzar un determinado destino. Se plantean dos dos problemas de optimización diferentes, en los que se busca: el camino de mínimo tiempo y el de mínima distancia restante al destino.

3.1.1 Camino de Mínimo Tiempo

El problema del camino de mínimo tiempo consiste en la minimización del tiempo requerido por un glider para llegar a un destino determinado. La principal dificultad de este planteamiento reside en tratar de abordar su resolución reflejando de la forma más fiel posible el funcionamiento de un glider real.

La Figura 3.1 ilustra este problema, en el que se busca el camino óptimo desde el waypoint inicial $\mathbf{x}_0$ al de destino $\mathbf{x}_{\text{goal}}$. En este problema, utilizando un esquema de optimización no es posible resolverlo de manera

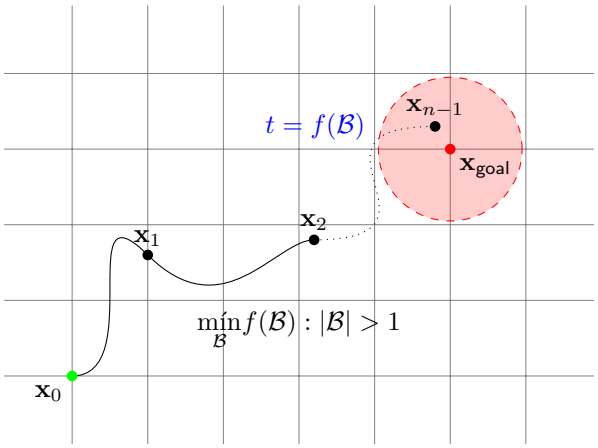


Figura 3.1: Diagrama del problema del camino de mínimo tiempo. Buscamos el camino $\mathcal{P}$ —formado por un conjunto de direcciones (*bearings*) $\mathcal{B}$— que minimice el tiempo viajado $t_t = f(\mathcal{B})$ para ir desde la localización inicial $\mathbf{x}_0$ a la de destino $\mathbf{x}_{\text{goal}}$. El número de direcciones $|\mathcal{B}|$ no es fijo, puesto que éste realmente determina el tiempo viajado. El destino $\mathbf{x}_{\text{goal}}$ se considera alcanzado con el punto final $\mathbf{x}_{n-1}$ cae en un radio alrededor suyo.

directa ya que se desconoce a priori el número de variables implicadas, i.e. el número de direcciones $|\mathcal{B}|$ que forma el camino. La resolución pasa, por tanto, por implementar un esquema iterativo.

3.1.2 Camino de Mínima Distancia Restante al Destino

El problema del camino de mínima distancia restante al destino consiste en tratar de acercarse lo más posible a un punto de destino en un tiempo determinado. El planteamiento es bastante directo cuando se aborda con técnicas de optimización, puesto que se pueden considerar los rumbos a comandar como las variables del proceso de optimización. Además, la resolución es directa, puesto que el número de direcciones $|\mathcal{B}|$ es conocido de antemano.

La Figura 3.2 ilustra este problema, donde partimos de $\mathbf{x}_0$ y llegamos hasta la localización final $\mathbf{x}_{n-1}$ tras comandar $|\mathcal{B}| = n$ direcciones, que forman el camino a buscar, para una duración de tiempo fija y dependiente de n según el tiempo entre emersiones t_s. La optimización minimiza la distancia restante d, mostrada en la figura.

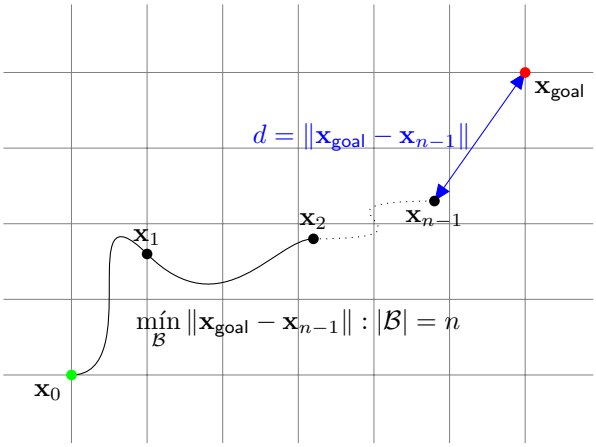


Figura 3.2: Diagrama del problema de mínima distanca al destino. Buscamos el camino $\mathcal{P}$ defino por un conjunto de n direcciones $\mathcal{B}$ que minimicen la distancia $d = \|\mathbf{x}_{\text{goal}} - \mathbf{x}_{n-1}\|$ desde el punto final $\mathbf{x}_{n-1}$ hasta el destino $\mathbf{x}_{\text{goal}}$. En este problema el número de direcciones $|\mathcal{B}| = n$ es fijo y conocido por anticipado.

3.2 Evitación de Obstáculos

Una de las principales limitaciones de los gliders es su reducida velocidad y maniobrabilidad. Esto hace que determinadas trayectorias deban de ser cuidadosamente planteadas si se quiere llegar a una solución válida. Ejemplos de esos escenarios son la presencia de obstáculos en las inmediaciones de la ruta a planificar. Dichos obstáculos pueden materializarse de diferente forma, incluyendo obstáculos fijos como islas (véase la Figura 3.3), costa, aguas someras, puertos, etc.; móviles como barcos; o restricciones del tipo corrientes de intensidad similar o superior a la velocidad de avance del glider.

Los procesos de optimización en estos supuestos deben ser complementados para evitar problemas de convergencia en la solución o trayectorias no válidas.

3.3 Seguimiento de Trayectorias

Otro conjunto de problemas de interés son aquellos en los que se requiere que el vehículo siga una ruta predefinida de la forma más ajustada posible. Los procesos de optimización en este caso requieren una verificación más estricta, puesto que debe tratar de garantizarse un comportamiento adecuado tanto en los puntos de emersión como a lo largo de los stints.

Para determinar la separación o desviación respecto a la trayectoria deseada se pueden aplicar diversas fórmulas para su cálculo. En principio, se trata del cómputo del área de un trapezoide recta entre el camino

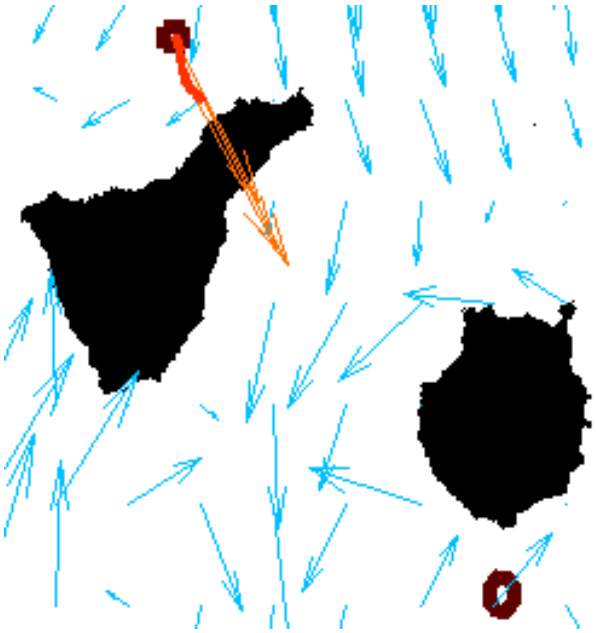


Figura 3.3: Evitación de obstáculos en problemas de optimización de planificación de caminos para gliders. Los obstáculos aparecen en el océano en diferentes formas: corrientes oceánicas fuertes en contra, zonas de tierra —como las islas de la figura—, rutas de tráfico marítimo, aguas someras, y básicamente cualquier área dónde el glider no puede navegar realizando su característico perfil yo-yo bajo el agua.

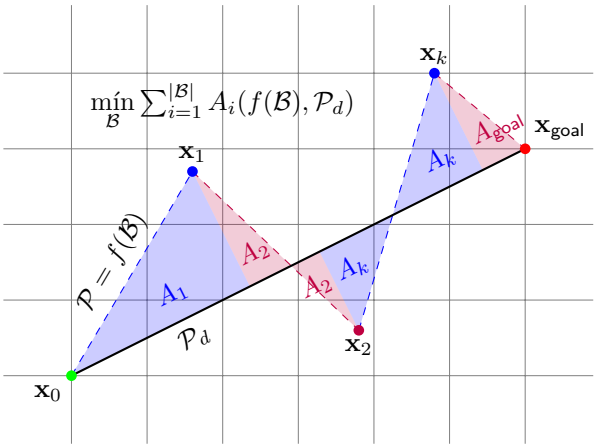


Figura 3.4: Diagrama del problema de seguimiento de trayectorias. Buscamos el camino $\mathcal{P}$ defino por un conjunto de direcciones $\mathcal{B}$ que minimice la desviación respecto al camino deseado $\mathcal{P}_d$. Tal desviación puede medirse como una distancia d o una función del área encerrada A. Este problema de seguimiento de trayectorias se resuelve como una restricción sobre el problema del camino de mínimo tiempo, que permite alcanzar el waypoint de destino $\mathbf{x}_{\text{goal}}$.

deseado y el resultante, computada para cada pareja de puntos de emersión consecutivos, tal y como muestra la Figura 3.4.

3.3.1 Relajación de la restricción de seguimiento

En la mayoría de estas aplicaciones, la falta de flexibilidad en la trayectoria se traduce en un pero aprovechamiento de las corrientes por parte del glider. Una ligera relajación de las condiciones, por ejemplo, se traduce en una ruta más adecuada en términos del tiempo invertido.

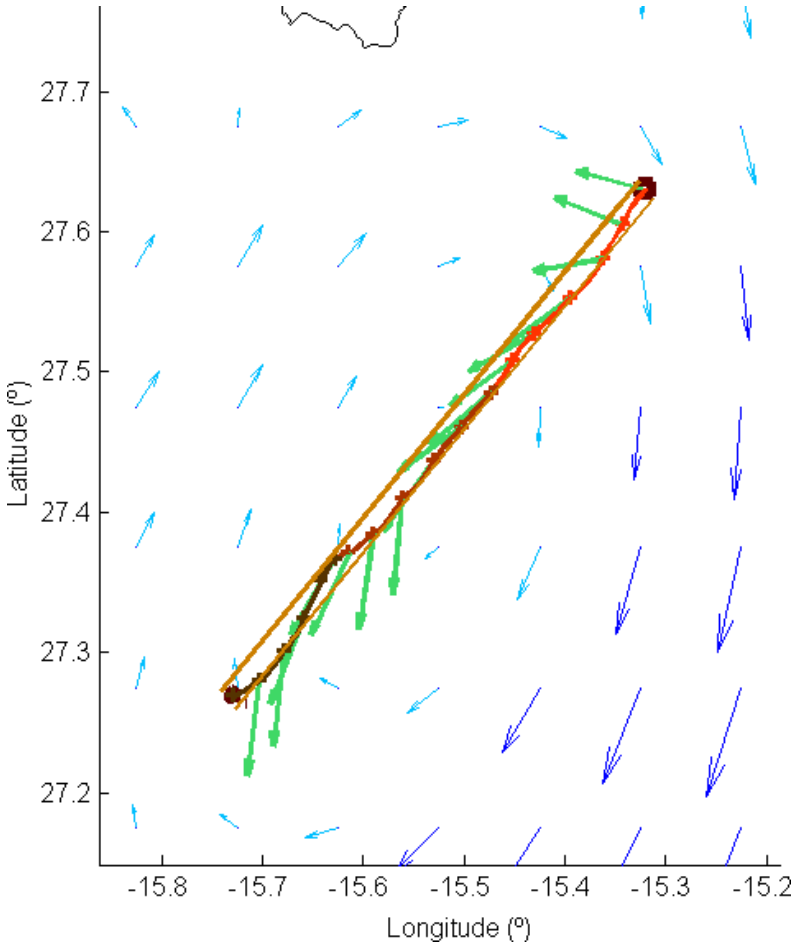


Figura 3.5: Ejemplo de seguimiento de trayectorias para gliders. La figura muestra el camino más corto $\mathcal{P}$ encontrado para seguir el camino deseado $\mathcal{P}_d$ desde el waypoint inicial $\mathbf{x}_0$ al de destino $\mathbf{x}_{\text{goal}}$, dentro de un pasillo alrededor de $\mathcal{P}_d$ —i.e. el problema de planificación de ruta se relaja. Las direcciones del glider $\mathcal{B}$ → para el camino $\mathcal{P}$ en cada punto de emersión se muestran junto con las corrientes oceánicas → (las corrientes que exceden la velocidad del glider $v_g = 0{,}4\text{m/s}$ están resaltadas →) desde el waypoint inicial ■ al de destino ○.

Esto se ilustra en la Figura 3.5, donde se define un pasillo alrededor del camino deseado. De esta forma es posible relajar a medida la restricción de seguimiento de la trayectoria. De hecho, las soluciones que se obtienen para que pasillo de mayor tamaño definen una curva de Pareto de soluciones no dominadas, para

el objetivo de seguimiento de trayectorias y del camino de mínimo tiempo entre los extremos del camino deseado.

3.4 Planificación Multi-glider

Con frecuencia nos encontramos escenarios en los que varios robots deben cooperar para resolver una determinada tarea. Surgen entonces problemas que no pueden tratarse de forma independiente para cada vehículo. Ejemplos de este tipo de aplicaciones son el muestreo en formación o la coordinación para facilitar las tareas de recuperación de los gliders tras finalizar una misión conjunta.

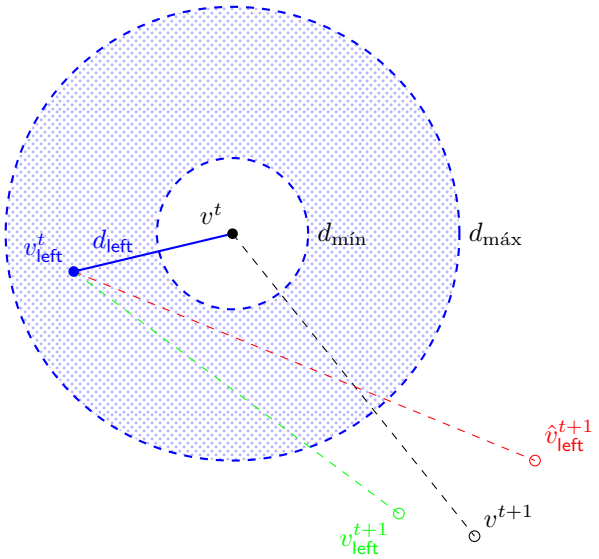


Figura 3.6: Diagrama de restricción de coordinación multi-glider, en el cual cada gliders fija su trayectoria de acuerdo con una restricción. En este caso, en un instante de tiempo dado t, para un glider o vehículo v^t concreto en la float, imponemos que la distancia d_{left} con respecto al glider izquierdo v^t_{left} debe caer dentro del rango $[d_{\text{mín}}, d_{\text{máx}}]$. Además, nótese que para un instante dado de tiempo t chequeamos que los tramos no se intersectan —i.e. no se cruzan entre sí. Por este motivo, el tramo desde v^t_{left} a $\hat{v}^{t+1}_{\text{left}}$ se descarta. Procederemos de manera similar con el glider de la derecha v^t_{right}, si está presente.

Precisamente la Figura 3.6 ilustra el caso de la navegación formando en paralelo. Una flota de gliders debe navegar garantizando que la restricción impuesta por la condición de vecindad ilustrada en la figura se cumple en todo momento, desde el waypoint inicial al de destino. Mientras, se resuelve otro problema, como puede ser el del camino de mínimo tiempo. Éste se resolverá simultáneamente para todos los vehículos de la flota, al mismo tiempo que la optimización está sujeta a la restricción de formación en paralelo, en este caso.

3.4.1 Recogida Eficiente

Otra problema multi-glider, con una aplicación clara en misiones con múltiples vehículos, es el de la recogida eficiente de los mismos. Tal y como muestra la Figura 3.7, un barco debe recoger la flota formada por k vehículos en el menor tiempo posible, puesto que con ello se reduce el coste económico de la recogida —al reducir el gasto de combustible.

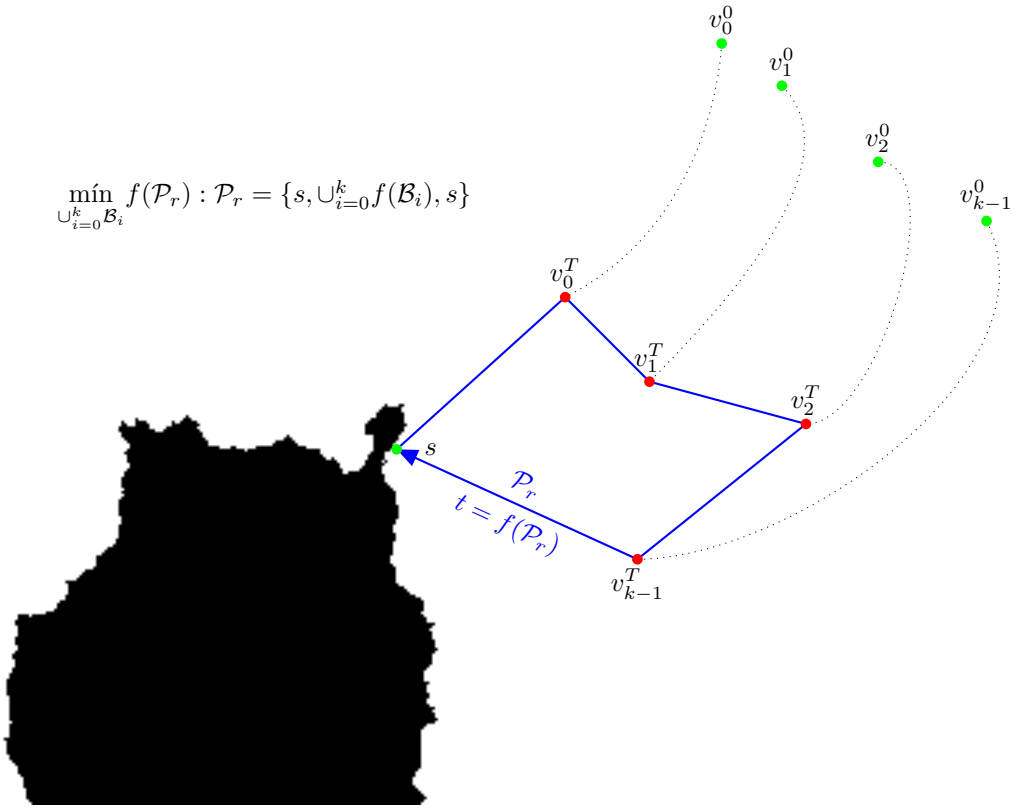


Figura 3.7: Diagrama del problema de la recogida eficiente. Una flota de k vehículos se mueve a la localización más apropiada v_i^T durante un tiempo dado T, para cada vehículo $i = 0, \ldots, k-1$, de tal modo que el camino $\mathcal{P}_r$ para recogerlos con un barco s requiera el mínimo tiempo $t_t = f(\mathcal{P}_r)$.

3.5 Muestreo de Estructuras

Un conjunto interesante de aplicaciones son aquellas en las que se necesita muestrear una estructura que evoluciona en el tiempo. La planificación en estos casos debe tener en cuenta de forma simultánea la previsión

de las condiciones marinas y la evolución del proceso a muestrear, de forma que se maximicen las posibilidades de obtener una buena caracterización.

Dentro de esta categoría se recogen aplicaciones destinadas a la detección y seguimiento de proliferaciones de algas nocivas, filamentos, frentes o giros oceánicos.

3.5.1 Estructuras Coherentes Lagrangianas

Igualmente, es posible obtener Estructuras Coherentes Lagrangianas, que pueden ser estáticas o dinámicas a su vez. Éstas se obtienen a partir del análisis de estabilidad de la corrientes oceánicas, utilizando los Exponentes de Lyapunov en Tiempo Finito (FTLE).

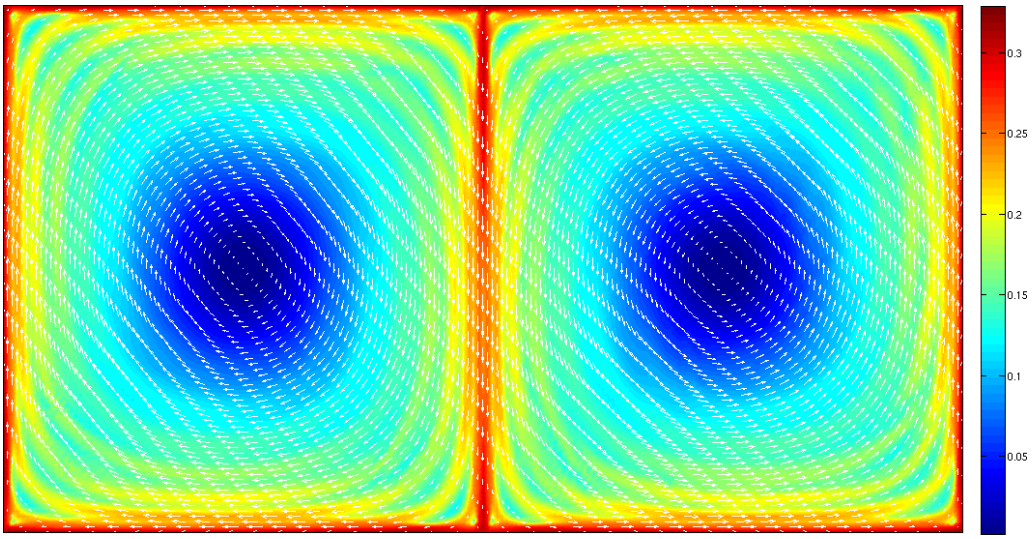


Figura 3.8: Exponentes de Lyapunov en Tiempo Finito (FTLE) computados sobre un campo de corrientes oceánicas sintético que forma un doble giro —superpuesto con flechas blancas. Los valores de FTLE representan la estabilidad de cada región en el campo, tal que las regiones inestables —como los bordes de los giros— aparecen en rojo y las estables en azul —como el centroide de los giros.

La Figura 3.8 muestra el campo de FTLE con las corrientes de un doble giro generado sintéticamente. Claramente observamos como las zonas del borde de los giros son altamente inestables. Esto es debido a que si el vehículo estuviera en estas zonas, un mínimo error en su localización podría significar que acabaría en posiciones muy diferentes pasado un tiempo T, asumiendo el modelo de movimiento puntual de un derivador. Dada la incertidumbre en la localización de los gliders bajo el agua, en la medida de lo posible es interesante evitar estas zonas duran la planificación de rutas.

3.5.2 Giros Mesoescalares

Los giros son un tipo de estructura que se encuentra a distintas escalas espaciales y temporales en el océano. Por ello son de alto interés en los estudios del comportamiento del océano. Para poder caracterizarlos es necesario muestrearlos siguiendo una trayectoria que se focalice en los características claves del giro, para un correcta caracterización. La Figura 3.9 muestra patrón de muestreo en forma de molino, donde se realizan transectos en forma de líneas que cruzan el centroide del giro desde un extremo a otro de su borde. Estas líneas se seguirán resolviendo el problema de seguimiento de trayectorias, discutido con anterioridad. Este patrón de muestreo resulta conveniente para estas estructuras, frente a otros patrones comunes en exploración, como el de *lawnmower*.

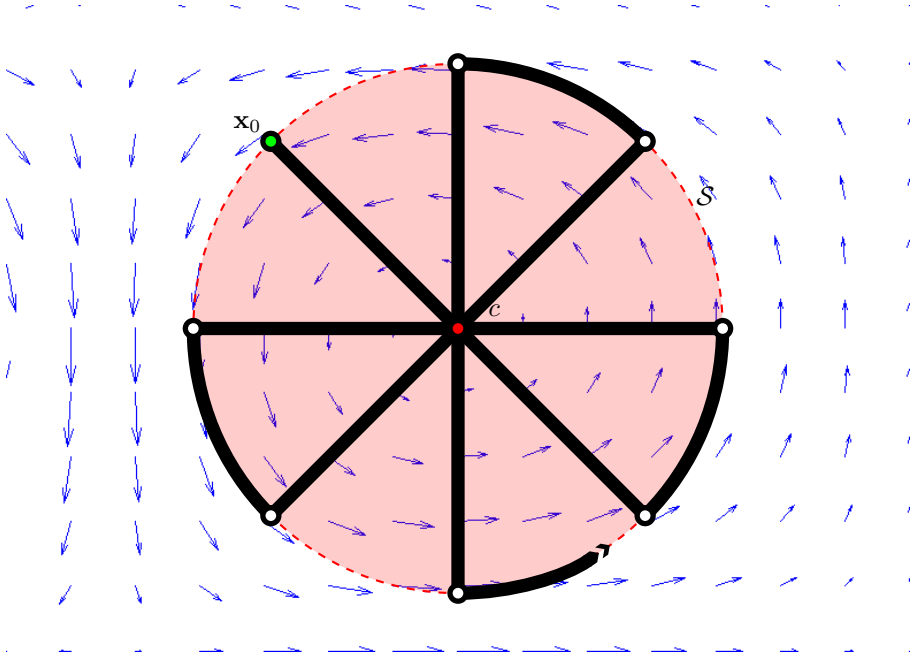


Figura 3.9: Patrón de muestreo de giros en forma de molino. Para un giro circular, el patrón de muestreo en forma de molino define un camino que cruza el centro c del giro sucesivamente con una serie de transectos empezando en $\mathbf{x}_0$. Cada transectos es un camino lineal que va desde un lado del giro al opuesto. Ahí, viaja cierta distance a lo largo del borde del giro antes de volver a cruzarlo.

Capítulo 4

Planificación de Rutas

El objetivo central de esta tesis es proponer algoritmos que permitan la optimización de rutas para gliders oceánicos aprovechando las condiciones de las corrientes marinas en el entorno.

4.1 Revisión

En la bibliografía se han planteado diferentes soluciones al problema de la planificación de trayectorias para vehículos marinos no tripulados, en general, y gliders en particular. Como aproximaciones más significativas podemos encontrar los métodos basados en grafos (A* y variantes), el muestreo probabilístico (RRT) y las técnicas basadas en optimización (bioinspiradas, teoría de sistemas, campos de potencial).

La Figura 4.1 muestra un diagrama resumen con las técnicas aplicadas a la planificación de rutas para gliders por otros autores. Estas técnicas se representan en líneas temporales organizadas por categorías, que representan el tipo de aproximación adoptado. Igualmente, aquellas técnicas inspiradas en variaciones aproximaciones, se señalan con líneas más finas que interconectan varias categorías.

4.2 Pilotaje Manual

El pilotaje manual de un glider, explicado con anterioridad, se utilizará como elemento base para comparar los algoritmos de planificación de caminos desarrollados en este trabajo. Aquí modelamos el pilotaje manual con una aproximación que fija las direcciones ψ_e del glider directamente hacia el destino $\mathbf{x}_{\text{goal}}$ en cada punto

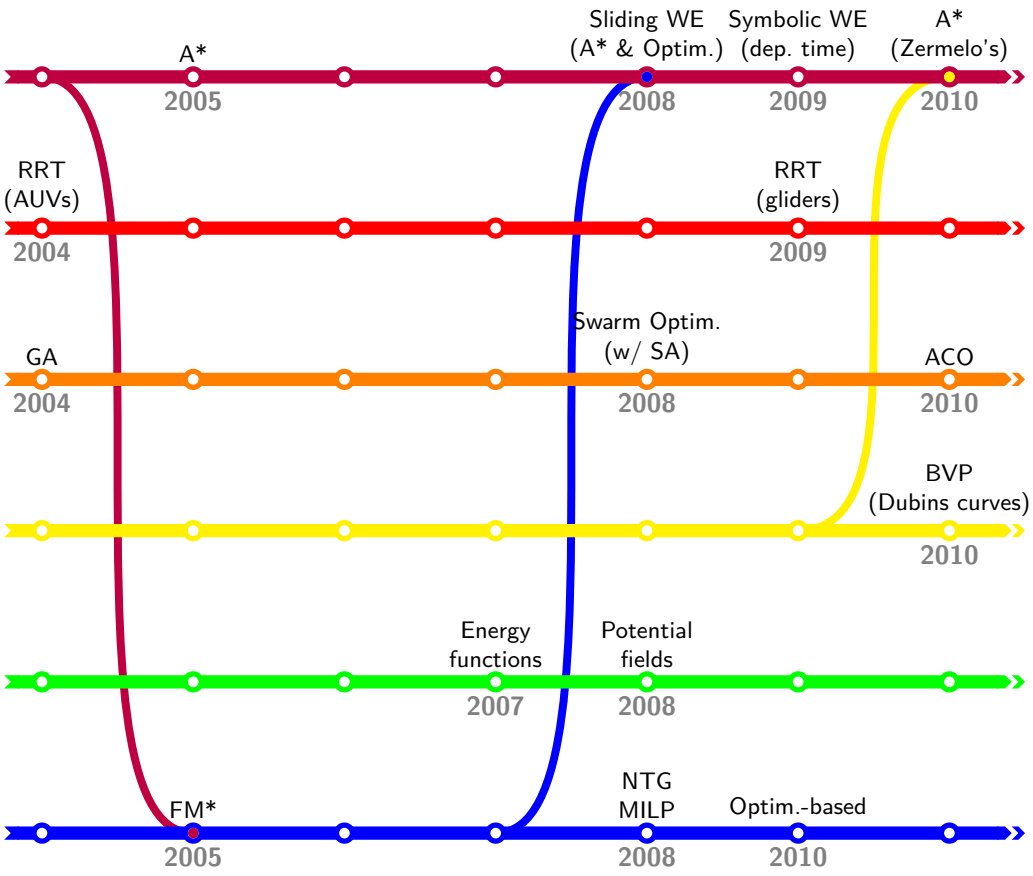


Figura 4.1: Diagrama temporal con algunas de las aproximaciones de planificación de caminos para gliders más relevantes dentro del campo. El año de publicación y la técnica se muestran en el diagrama, el cual está dividido por categorías de acuerdo con el tipo de técnica de planificación de caminos usado (de arriba a abajo): Métodos basados en grafos ; métodos basados en muestreo probabilísticos ; métodos de optimización bio-inspirados ; aproximaciones de Teoría de Sistemas ; funciones de energía ; y métodos de optimización numérica . Las líneas delgadas que conectan diferentes categorías representan algoritmos que combinan ideas de diferentes tipos de aproximaciones.

de emersión. A esta aproximación la denominamos DtG (*Direct to Goal*).

El diagrama de la Figura 4.2 ilustra el funcionamiento del DtG. Desde la localización $\mathbf{x}_i$ de un punto de emersión, se fija la dirección ψ_g igual al rumbo ψ_d directo al destino $\mathbf{x}_{\text{goal}}$. El problema de esta aproximación es que es ciega, es decir, no tiene en cuenta el efecto de las corrientes.

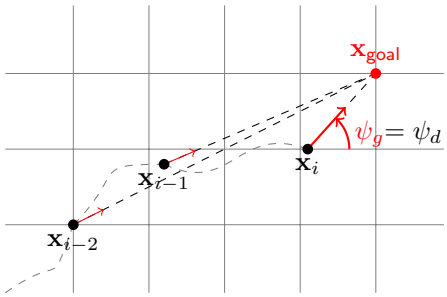


Figura 4.2: Diagrama de la aproximación Direct to Goal, la cual fija la dirección ψ_g igual al rumbo ψ_d hacia el waypoint de destino $\mathbf{x}_{\text{goal}}$ desde la localización actual $\mathbf{x}_i$.

4.3 Algoritmos propuestos

A lo largo de este trabajo de tesis se han desarrollado un conjunto de algoritmos de planificación especialmente adaptados a las características de los gliders. A continuación se incluye una descripción de sus principales características.

4.3.1 Adaptación del A*

Se ha adaptado el algoritmo clásico para operar en base a la simulación de las trayectorias posibles del glider sobre el mapa de corrientes. El A* se utiliza comúnmente para resolver el problema del camino más corto, pero utilizando el modelo de movimiento puntual restringido de la Sección 2.2.1.2 es posible resolver el problema del camino de mínimo tiempo.

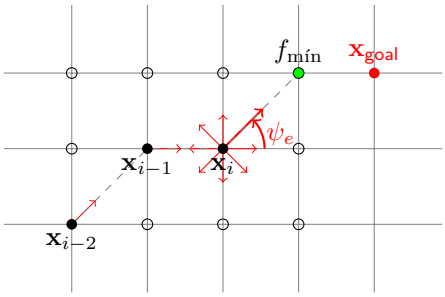


Figura 4.3: Algoritmo A* clásico sobre una rejilla uniforme que define el dominio del espacio de búsqueda Ω. Los ángulos de rumbo ψ_e están restringidos por la rejilla, produciendo trayectorias en línea recta entre nodos y tiempos de emersión no constantes para los problemas de planificación de caminos para gliders.

El funcionamiento del algoritmo A* se ilustra en el diagrama de la Figura 4.3. Podemos ver como la dirección ψ_g que se fija para el glider debe ser tal que permita seguir el rumbo ψ_e, tal que el vehículo pueda

ir del nodo actual a otro nodo de la rejilla del grafo de búsqueda. Es por eso que se requiere del modelo de movimiento restringido. El problema fundamental de esta aproximación es que en zonas de corrientes fuertes, el cono de accesibilidad se puede estrechar hasta el punto de que no sea posible llegar a ningún nodo vecino desde el nodo actual.

Algorithm 4.1 Pseudo-código del algoritmo A* para búsquedas en grafo implícitos.

Require: Nodos inicial n_{start} y destino n_{goal}. Función heurística $h(n)$ que estima el coste desde el nodo n a n_{goal}. La heurística $h(n)$ se asume consistente, de modo que se usa un lista cerrada $\mathcal{C}$ para hacer la búsqueda más eficiente. Función **parents**(n_i) que recupera el camino $\mathcal{P}$ extrayendo los padres de $n_i = n_{\text{goal}}$ a partir del mapa parent($\cdot$) map recursivamente, hasta que alcanza n_{start}.
Ensure: Camino óptimo $\mathcal{P}$ encontrado desde n_{start} a n_{goal}.
Algorithm: A*(n_{start}, n_{goal}) **return** $\mathcal{P}$

1: $g(n_{\text{start}}) = 0$ ▷ asumimos $g(n_i) = \infty$ para el resto
2: parent$(n_{\text{start}}) = \varnothing$ ▷ asumimos parent$(n_i) = \varnothing$ para el resto también
3: $\mathcal{O} = \mathcal{C} = \varnothing$ ▷ listas abierta y cerrada vacías
4: $\mathcal{O}$**.insert**$(n_{\text{start}}, g(n_{\text{start}}) + h(n_{\text{start}}))$
5: **while** $\mathcal{O} \neq \varnothing$ **do**
6: $n_i = \mathcal{O}$**.pop** ▷ extraer nodo n_i con el mínimo coste
7: **if** $n_i = n_{\text{goal}}$ **then** ▷ destino n_{goal} alcanzado
8: **return parents**(n_i) ▷ camino encontrado (extraído de parent($\cdot$))
9: **end if**
10: $\mathcal{C} = \mathcal{C} \cup \{n_i\}$ ▷ cerrar nodo n_i
11: **for all** $n_{i+1} \in$ **successors**(n_i) **do** ▷ para cada sucesor n_{i+1}
12: $g' = g(n_i) + c(n_i, n_{i+1})$ ▷ valor g candidato
13: **if** $(n_{i+1} \notin \mathcal{C} \cup \mathcal{O}) \vee (g' < g(n_{i+1}))$ **then** ▷ no descubierto o mejor g'
14: $g(n_{i+1}) = g'$ ▷ actualizar
15: parent$(n_{i+1}) = n_i$
16: $\mathcal{O}$**.insert**$(n_{i+1}, g(n_{i+1}) + h(n_{i+1}))$ ▷ decrementar clave (actualizar, o insertar si no descubierto)
17: **end if**
18: **end for**
19: **end while**
20: **return** $\varnothing$ ▷ ningún camino encontrado

En el Algoritmo 4.1 tenemos el pseudo-código del algoritmo A* utilizado. Éste emplea un grafo implícito, que se construye evaluando el modelo de movimiento restringido en la función que genera los nodos sucesores —o vecinos— del nodo actual. Como se mencionó anteriormente, en algunos casos el número de vecinos puede ser muy bajo o, incluso nulo, debido al efecto de las corrientes oceánicas.

El algoritmo A* emplea una heurística para acelerar la búsqueda del camino óptimo, siendo así más rápido que el algoritmo de Dijkstra. Para computar la heurística $h(n)$, estimamos la máxima velocidad $U_{c_{\text{máx}}} = \max_{\mathbf{x} \in \mathcal{P}}(\|\mathbf{U}_c(\mathbf{x}, t)\|)$ de las corrientes oceánicas a lo largo del camino $\mathcal{P}$ seguido por el glider. Puesto que $\mathcal{P}$

no se conoce de antemano, estimamos $U_{c_{\text{máx}}}$ usando un dominio Ω centrado espacial y temporalmente en la localización actual y tiempo. Dada $U_{c_{\text{máx}}}$ y la velocidad nominal del glider U_g, computamos una subestimación temporal del coste de alcanzar el destino desde al localización actual $\mathbf{x}$, del nodo n, usando la distancia $d = \|\mathbf{x} - \mathbf{x}_{\text{goal}}\|$ al destino,

$$h(n) = \frac{d}{U_g + U_{c_{\text{máx}}}} \tag{4.1}$$

que es una función heurística conservativa pero admisible. Además, mientras usemos la misma estimación de $U_{c_{\text{máx}}}$ para cada nodo n en el grafo, esta heurística también es consistente.

4.3.2 CTS-A*

El patrón de movimiento del glider es recogido con precisión en este algoritmo, de manera que los resultados tienen un mayor plausibilidad. A diferencia del A*, el CTS-A* genera tiempos constantes entre emersiones, al igual que los gliders. Para ello, en la generación de sucesores del grafo implícito de búsqueda, emplea el modelo de movimiento puntual no restringido explicado en Sección 2.2.1.1.

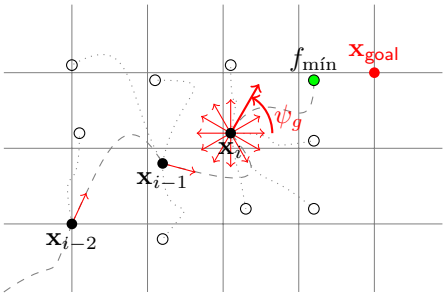


Figura 4.4: Diagrama de funcionamiento del algoritmo Constant-Time Surfacing A*. En cada localización de emersión $\mathbf{x}_i$ diferentes direcciones ψ_g son consideradas, integrando la trayectoria del glider durante el tiempo entre emersiones t_s, o múltiplos $s \cdot t_s$.

El diagrama de la Figura 4.4 ilustra como ahora se considera un conjunto de direcciones ψ_g y para cada una de ellas se integra el modelo de movimiento no restringido. La posición resultante de la emersión tras el tiempo t_s se lleva a una rejilla sólo con la finalidad de mantener constante el uso de memoria del algoritmo. Así, mantenemos los tiempos entre emersiones constantes y simulamos la verdadera trayectoria del glider realizando los perfiles yo-yo mientras navega sumergido entre emersiones.

En el Algoritmo 4.2 se muestra el procedimiento de generación de sucesores, en el que se considera un muestreo uniforme de direcciones ψ_g —con una determinada discretización— y para cada una se integra el

Algorithm 4.2 Generación de sucesores en el CTS-A*.

Require: Nodo actual n, el número k de direcciones (*bearings*) a muestrear y el tiempo entre emersiones t_s para integrar la trayectoria. La integración de la trayectoria require el paso de integración dt, la velocidad nominal del glider $\mathbf{v}_g(\psi_g)$ para una dirección ψ_g, y el campo de velocidades de las corrientes oceánicas $\mathbf{v}_c(\mathbf{x})$. Si un nodo vecino m tiene el mismo coste g' que la nueva trayectoria $\mathbf{x}_{\psi_g}$, se usa una función $d(\mathbf{x}, n)$. Si $d(\mathbf{x}, n) = h(n)$, podemos usar $f(n)$ en lugar de $g(n)$, con una sola condición.

Ensure: Lista de sucesors $\mathcal{S}$ y actualización del etiquetado de nodos considerando el coste $g(\mathbf{x})$ y la distancia $d(\mathbf{x}, m)$ de la localización $\mathbf{x}$ emparejada con el nodo m.

Algorithm: successors(n, k, t_s) **return** $\mathcal{S}$

$\mathcal{S} = \varnothing$
for all 1 to k **do**
  $\psi_g \sim [0, 2\pi)$ ▷ muestrear *bearing*
  $s = 0$
  repeat
    $\mathbf{x}_{\psi_g} =$ **trajectory**$(t_s, dt, \mathbf{v}_g(\psi_g), \mathbf{v}_c(\mathbf{x}), \mathbf{x}_n)$
    $m =$ **NN**$(\mathbf{x}_{\psi_g})$ ▷ nodo m más cercano a $\mathbf{x}_{\psi_g}$
    $s = s + 1$
  until $m \neq n$
  $g' = g(n) + s \cdot t_s$
  if $m \notin \mathcal{C} \cup \mathcal{O}$ **then** ▷ m no descubierto hasta ahora
    $m_{\mathbf{x}} = \mathbf{x}_{\psi_g}$
  else
    if $g' < g(m)$ **then** ▷ menor coste
      $m_{\mathbf{x}} = \mathbf{x}_{\psi_g}$
    else if $g' = g(m)$ **then** ▷ igual coste, pero
      if $d(\mathbf{x}_{\psi_g}, m) < d(m_{\mathbf{x}}, m)$ **then** ▷ mejor localización
        $m_{\mathbf{x}} = \mathbf{x}_{\psi_g}$
      end if
    end if
  end if
  $\mathcal{S} = \mathcal{N} \cup \{m\}$ ▷ añadir m a la lista de sucesors $\mathcal{S}$
end for
return $\mathcal{S}$

modelo de movimiento no restringido. Este algoritmo también controla el etiquetado de los nodos, que se encarga de guardar las localizaciones de las emersiones en los nodos. Se guardan en base al coste acumulado y una función que puede ser la distancia más cercano al nodo o la propia heurística.

4.3.3 ABS

Para aliviar el coste computacional del CTS-A*, se emplea un muestreo en base a una función de densidad de probabilidad (PDF) que tiene en cuenta tanto la dirección predominante de la corriente como el rumbo fijado para el glider. Esto mejora el muestreo uniforme de las direcciones ψ_g adoptado en la versión básica del

algoritmo CTS-A*.

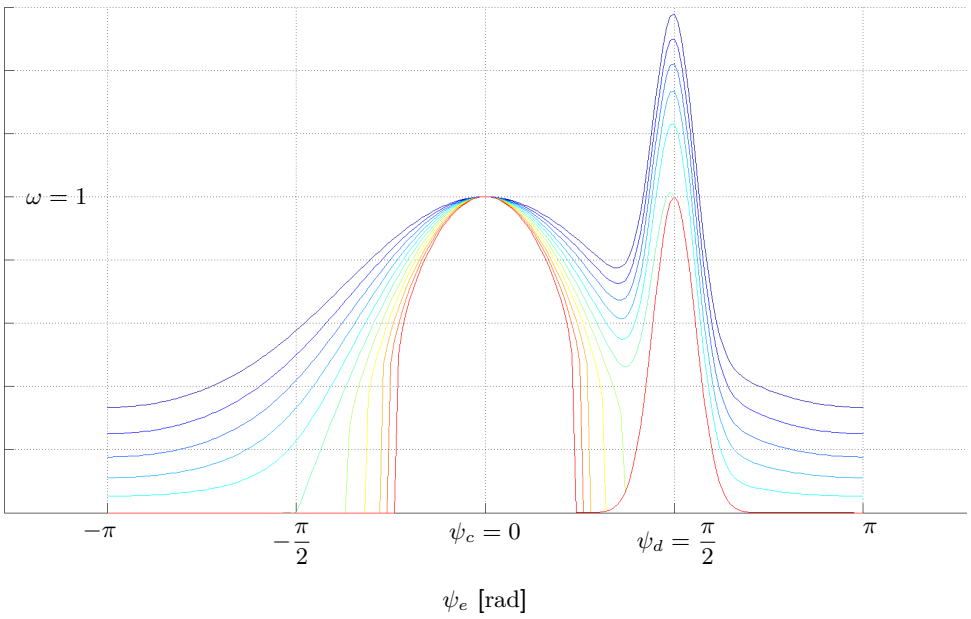


Figura 4.5: Combinación de distribuciones PDF $f_{\psi_c}(\psi_e)$ y una distribución PDF $f_{\psi_d}(\psi_e)$ que considera el rumbo $\psi_d = \frac{\pi}{2}$ directo al destino, modelada con (4.5). $SR \in [50, 150]\,\%$.

En la estrategia de muestreo adaptativo de direcciones (ABS) es posible modelar la PDF considerando la velocidad efectiva U_e y el rumbo ψ_d al destino. Primeramente, U_e se modela con el modelo de movimiento restringido de la Sección 2.2.1.2. Dados U_g, U_c, ψ_c y el rumbo deseado ψ_e, se computa la dirección ψ_g necesaria para compensar la deriva producida por las corrientes como

$$\psi_g = \begin{cases} \psi_e + \arcsin s & \text{if } |s| \leq 1 \\ \nexists & \text{otherwise} \end{cases} \tag{4.2}$$

que da la velocidad efectiva U_e

$$U_e = \begin{cases} U_g\sqrt{1-s^2} + U_c \cos\psi_{ec} & \text{if } |s| \leq 1 \\ 0 & \text{en otro caso} \end{cases} \tag{4.3}$$

donde

$$s = \frac{U_c}{U_g} \sin\psi_{ec} \tag{4.4}$$

y $\psi_{ec} = \psi_e - \psi_c$.

Se construye otra PDF $f_{\psi_d}(\psi_e)$ considerando la diferencia entre el rumbo ψ_e y ψ_d. Proponemos la siguientes distribución Normal para modelarla,

$$\mathcal{N}(\psi_e;\ \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(\psi_e - \mu)^2}{2\sigma^2}\right\} \quad \psi_e \in [0, 2\pi] \tag{4.5}$$

con media $\mu = \psi_d$ in rad y varianza σ^2.

Componiendo ambas PDFs, obtenemos la PDF propuesta tal y como se muestra en la Figura 4.5 para varios ratios de velocidad SR. A partir de esta nueva PDF, una vez normalizada se muestrean los rumbos ψ_e, y a partir de ellos se tienen las direcciones ψ_g que se usarán para integrar el modelo de movimiento no restringido tal y como se indica en el algoritmo CTS-A*. Así, podemos emplear un menor número de direcciones, reduciendo el tiempo de cómputo, pero manteniendo o mejorando la calidad de la solución, puesto que exploramos en las direcciones más prometedoras.

4.3.4 Adaptación del RRT

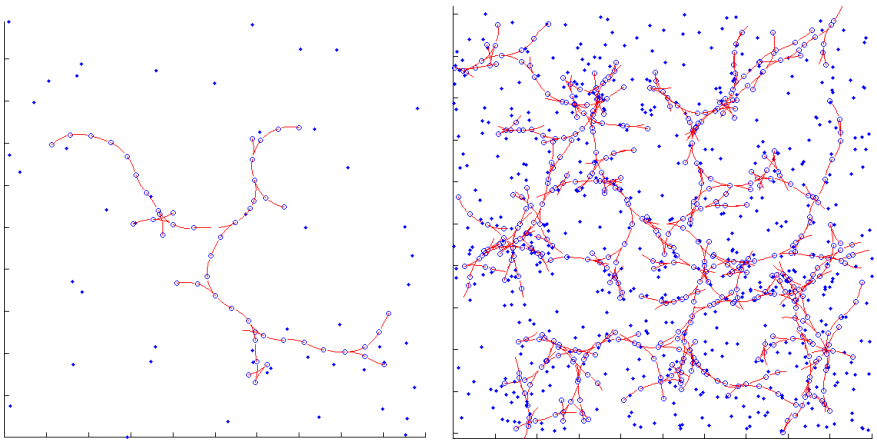
En este algoritmo se generan de forma rápida las trayectorias más probables que puede seguir el glider en una exploración dirigida de su entorno. El algoritmo RRT se muestra en el Algoritmo 4.3, en su versión genérica, la cual es aplicable en la planificación de caminos para gliders. Es importante indicar que este algoritmo no busca la solución óptima, sino una solución factible en poco tiempo de cómputo en problemas de gran dimensionalidad.

Algorithm 4.3 Pseuo-código del algoritmo RRT.

Require: Localización inicial $\mathbf{x}_0$ y destino $\mathbf{x}_{\text{goal}}$ y número de iteraciones K.
Ensure: Árbol o grafo $\mathcal{G}$, que conectará $\mathbf{x}_0$ con $\mathbf{x}_{\text{goal}}$.
Algorithm: RRT($\mathbf{x}_0$, K) **return** $\mathcal{G}$

```
1: G.init(x_0)                                   ▷ añadir localización inicial
2: for all 1 to K do
3:     x_rand = rand                             ▷ nueva localización aleatoria
4:     x_near = NN(S(G), x_rand)                 ▷ localización más cercana en el swath S(G)
5:     x_new = new(x_near, Δx)                   ▷ nueva localización
6:     G .add_vertex(x_new)
7:     G .add_edge(x_near, x_new)
8: end for
9: return G
```

En la práctica, hemos observado que en el problema de planificación de rutas para gliders, al menos se



(a) 50 iteraciones. (b) 500 iteraciones.

Figura 4.6: Ejemplo de un árbol RRT después de (a) 50 y (b) 500 iteraciones.

requieren dos árboles de búsqueda, una lanzado desde la localización inicial $\mathbf{x}_0$ y otro desde el destino $\mathbf{x}_{\text{goal}}$ que irá en sentido contrario —i.e. utilizando los valores de corrientes invertidos. Esta aproximación sólo es válida con campos de corrientes estáticas, ya que de ser dinámicas no sería posible determinar en que instante de tiempo se lanza el árbol desde el destino, pues para ello hay que resolver el problema.

4.3.5 Optimización

Este algoritmo permite afrontar planificaciones complejas a partir de un sistema de inicialización adecuado al problema seguido por un esquema de optimización iterativo.

Para resolver el problema de mínima distancia restante al destino, definimos una función objetivo basada en el punto final $\mathbf{x}_{n-1}$ de integración de la trayectoria del glider para todas las direcciones $\mathcal{B}$ comandadas desde el waypoint inicial $\mathbf{x}_0$,

$$\mathbf{x}_{n-1} = f(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G}) \tag{4.6}$$

donde $\mathcal{C}$ es el mapa de corrientes oceánicas y $\mathcal{G}$ es la configuración del glider —i.e. la velocidad nominal U_g y otros parámetros usados en el modelo de movimiento no restringido de la Sección 2.2.1.1. Puesto que queremos minimizar la distancia restante al waypoint destino $\mathbf{x}_{\text{goal}}$, construimos nuestra función objetivo como la distancia restante entre $\mathbf{x}_{n-1}$, dado por (4.6), y $\mathbf{x}_{\text{goal}}$. Es decir, la función objetivo f' es

$$f'(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G}) = \|\mathbf{x}_{\text{goal}} - f(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G})\| \tag{4.7}$$

donde $\| \cdot \|$ es una distancia métrica, que en nuestro caso es la longitud del arco del *great circle* que pasa por $\mathbf{x}_{n-1}$ y $\mathbf{x}_{\text{goal}}$, ya que navegamos una geometría esférica que aproxima a La Tierra; para misiiones cortas, con distancias pequeñas, podemos usar otra aproximación o incluso la distancia Euclídea (véase la Sección A.2).

4.3.5.1 Optimización Iterativa

En el caso del problema del camino de mínima distancia, no se conoce el tiempo que se tardará en llegar al destino, pues es lo que buscamos resolver. Por tanto, tampoco conocemos el número de direcciones $\psi_g \in \mathcal{B}$, que forman parte de la función objetivo. Para solucionar este problema planteamos un método de optimización iterativo, que se muestra en el Algoritmo 4.4.

Algorithm 4.4 Pseudo-código del algoritmo de Optimización Iterativa.

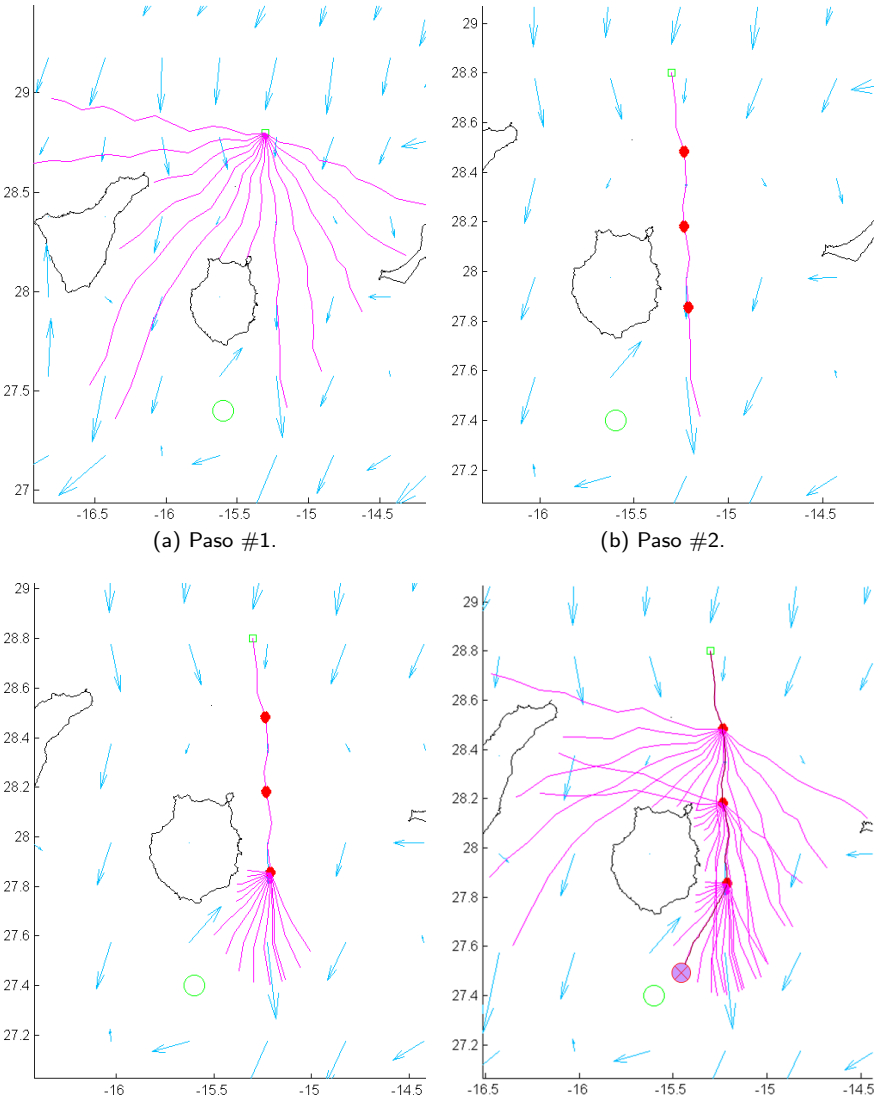
Require: Puntos inicial $\mathbf{x}_0$ y destino $\mathbf{x}_{\text{goal}}$, el mapa de corrientes oceánicas $\mathcal{C}$, la configuración del glider $\mathcal{G}$ y el radio temporal de precisión del destino $t_{\text{máx}}$.
Ensure: Lista de direcciones (*bearings*) $\mathcal{B}$ para el camino de mínimo coste encontrado.
Algorithm: iterative_optimization($\mathbf{x}_0$, $\mathbf{x}_{\text{goal}}$, $\mathcal{C}$, $\mathcal{G}$, $t_{\text{máx}}$) **return** $\mathcal{B}$

$\mathbf{x} = \mathbf{x}_0$ ▷ último punto de emersión hasta ahora
$\mathcal{B} = \varnothing$ ▷ lista de direcciones que representa el camino
repeat
  $\left|\hat{\mathcal{B}}\right| = \textbf{length}(\mathbf{x}, \mathbf{x}_{\text{goal}}, \mathcal{C}, \mathcal{G})$ ▷ # direcciones necesarias para alcanzar $\mathbf{x}_{\text{goal}}$ desde $\mathbf{x}$
  $\hat{\mathcal{B}} = \textbf{init}(\mathbf{x}, \mathbf{x}_{\text{goal}}, \left|\hat{\mathcal{B}}\right|)$ ▷ initializar las nuevas direcciones $\left|\hat{\mathcal{B}}\right|$
  $\mathcal{B} = \mathcal{B} \cup \hat{\mathcal{B}}$
  $\langle \mathcal{B}, \mathbf{x} \rangle = \textbf{optimize}(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G})$
  $t = \dfrac{\|\mathbf{x}_{\text{goal}} - \mathbf{x}\|}{U_{c_{\text{máx}}}}$ ▷ tiempo desde el último punto de emersión al destino
until $t < t_{\text{máx}}$
return $\mathcal{B}$

4.3.5.2 Inicialización Inteligente. Evitación de Obstáculos

El principal problema de las técnicas de optimización es que requieren una buena solución inicial para no verse atrapadas en mínimos locales. En el caso de entornos con obstáculos se requiere de un procedimiento de inicialización que encuentra una solución suficientemente buena, para luego ser optimizada.

En general, para entornos libres de obstáculos se puede utilizar la aproximación del DtG como solución inicial. Sin embargo, en presencia de obstáculos proponemos un algoritmo para la fase de inicialización que emplea una versión *gruesa* del algoritmo CTS-A*. Con ella conseguimos evitar obstáculos y así proporcionar una solución inicial al algoritmo de optimización. La Figura 4.7 ilustra los cuatro pasos de este procedimiento



(a) Paso #1. (b) Paso #2.

(c) Paso #3. (d) Paso #4.

Figura 4.7: La inicialización inteligentes para la Evitación de Obstáculos comprende los siguientes pasos: (a) Primer paso del proceso de inicialización: Vectores radiales emanando desde el punto inicial. (b) Segundo paso del proceso de inicialización: Selección de nodos candidatos. (c) Tercer paso del proceso de inicialización: Vectores radiales emanando desde cada nodo candidato. (d) Cuarto paso del proceso de inicialización: Selección de la mejor trayectoria.

de inicialización:

1. Se lanzan vectores en dirección radial desde el punto inicial $\mathbf{x}_0$.

2. Se seleccionan nodos candidatos.

3. Se vuelven a lanzar más vectores radiales desde cada nodo candidato.

4. Finalmente se selecciona la mejor trayectoria.

4.4 Planificación de Rutas sobre Estructures Oceánicas

A parte de las técnicas comentadas hasta ahora, es posible planificar rutas teniendo en cuenta estructuras oceánicas presentes en el océano, en lugar de todas las corrientes. Para ello podemos aplicar técnicas de identificación y seguimiento de dichas estructuras, como veremos a continuación.

4.4.1 Análisis de Estabilidad

El análisis de estabilidad del campo vectorial de corrientes oceánicas permite determinar qué zonas son estables y cuáles inestables, utilizando tanto corrientes estáticas como dinámicas. Para ello computamos el campo FTLE con la siguiente expresión,

$$\sigma_{t_0}^{T}(\mathbf{x}) = \frac{1}{|T|} \ln \sqrt{\lambda_{\text{máx}}(\Delta)} \tag{4.8}$$

que representa el FTLE más grande para un tiempo de integración finito T, asociado al punto $\mathbf{x}$ y el instante de tiempo t_0.

Algorithm 4.5 Pseudo-código para computar el campo FTLE.

Require: Localización inicial $\mathbf{x}_0$ de un conjunto de partículas $\mathcal{X}_0$ en el instante de tiempo t_0, para su advección durante el tiempo de integración T.
Ensure: Campo FTLE $\sigma_{t_0}^{T}(\mathbf{x})$.
Algorithm: FTLE($\mathcal{X}_0$, t_0, T) **return** $\sigma_{t_0}^{T}(\mathbf{x})$
1: **for all** $\mathbf{x}_0 \in \mathcal{X}_0$ **do**
2: $\mathbf{x}_T = \phi_{t_0}^{t_0+T}(\mathbf{x}_0)$: $\mathbf{x}_T \in \mathcal{X}_T$ ▷ advección de la partícula $\mathbf{x}_0$ durante el tiempo T
3: **end for**
4: $J = \nabla \mathcal{X}_T$ ▷ gradiente del mapa de flujo $\mathcal{X}_T$
5: $\Delta = J^* J$
6: $\sigma_{t_0}^{T}(\mathbf{x}) = \frac{1}{|T|} \ln \sqrt{\lambda_{\text{máx}}(\Delta)}$ ▷ aplicar (4.8)
7: **return** $\sigma_{t_0}^{T}(\mathbf{x})$

En el Algoritmo 4.5 se muestran los pasos para obtener el FTLE a partir del campo de corrientes. El resultado puede verse en la Figura 4.8 para un vórtice Stuart, generado sintéticamente. Podemos observar

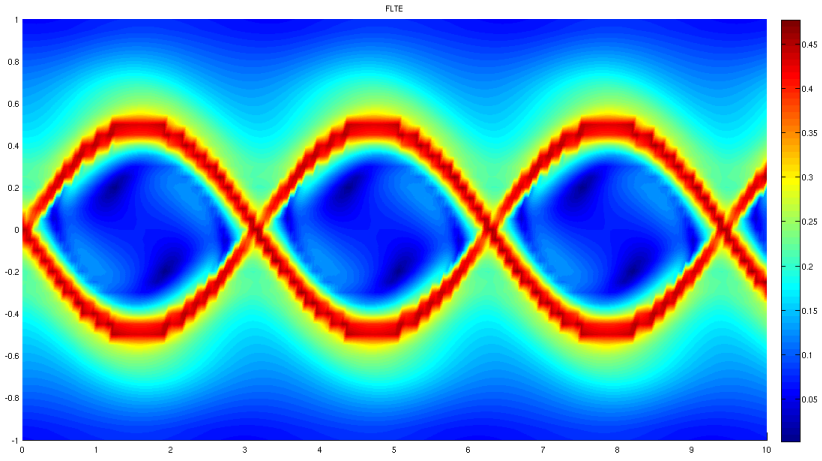


Figura 4.8: Análisis de Estabilidad computando el campo FTLE de un vórtice Stuart. Las regiones inestables se corresponden con LCSs, las cuales aparecen en tonos rojos en los bordes de los giros/vórtices.

claramente como las zonas del borde de los giros son altamente inestables. Este información no es fácil de apreciar directamente sobre el campo de corrientes, especialmente si es dinámico, como en este caso. De ahí surge la utilidad de computar el campo FTLE. A partir de este campo, es posible segmentar las zonas inestables y así obtener Estructuras Coherentes Lagrangianas (LCSs). Así, es posible planificar rutas evitando estas zonas, puesto que su inestabilidad indica que el siguiente punto de emersión puede variar bastante del esperado, es decir, aumentar su incertidumbre.

4.4.2 Giros Oceánicos

Otro tipo de estructuras muy comunes en el océanos son los giros. Nos encontramos giros de distintas escalas espaciales y temporales. Así, los giros mesoescalares son interesantes para el estudio de múltiples procesos oceánicos.

La Figura 4.9 muestra un modelado tridimensional de un giro mesoescalar en sectores. Este modelo se puede incorporar a un Algoritmo Genético para realizar el seguimiento de la estructura, tras su identificación. Se ha desarrollado una técnica para dar los primeros pasos en el muestreo de estas estructuras móviles, optimizando la estrategia de muestreo de una determinada zona del giro —v.g. centroide, borde, etc.—, así como su caracterización, lo cual tiene especial utilidad en la asimilación de datos por parte de modelos oceánicos.

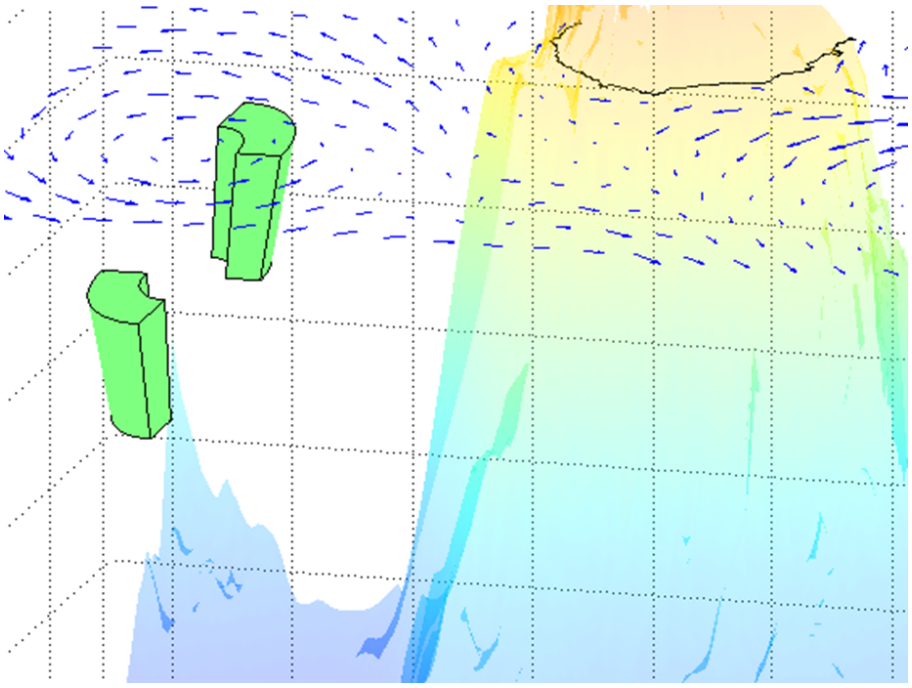


Figura 4.9: Modelo de giro mesoescalar usado para el algoritmo de planificación de caminos para detectar, caracterizar y seguir este tipo de estructuras oceánicas móviles. La figura muestra la segmentación del volumen del giro.

4.5 Librerías y Herramientas

Finalmente, no sólo se han contribuidos algoritmos novedosos a la planificación de rutas con gliders, sino también una serie de librerías y herramientas para la misma y la visualización de las salidas de los modelos oceánicos y los resultados de la planificación.

Bajo el nombre de *Pinzón*, nuestro software permite visualizar los caminos encontrados —junto con sus direcciones ψ_g en cada punto de emersión— sobre Google™ Earth, tal y como muestra la Figura 4.10. Asimismo, se ha automatizado la descarga de productos de múltiples modelos oceánicos y la visualización de sus productos, como la temperatura, salinidad, SSHA, las corrientes, etc. (véase el Apéndice A).

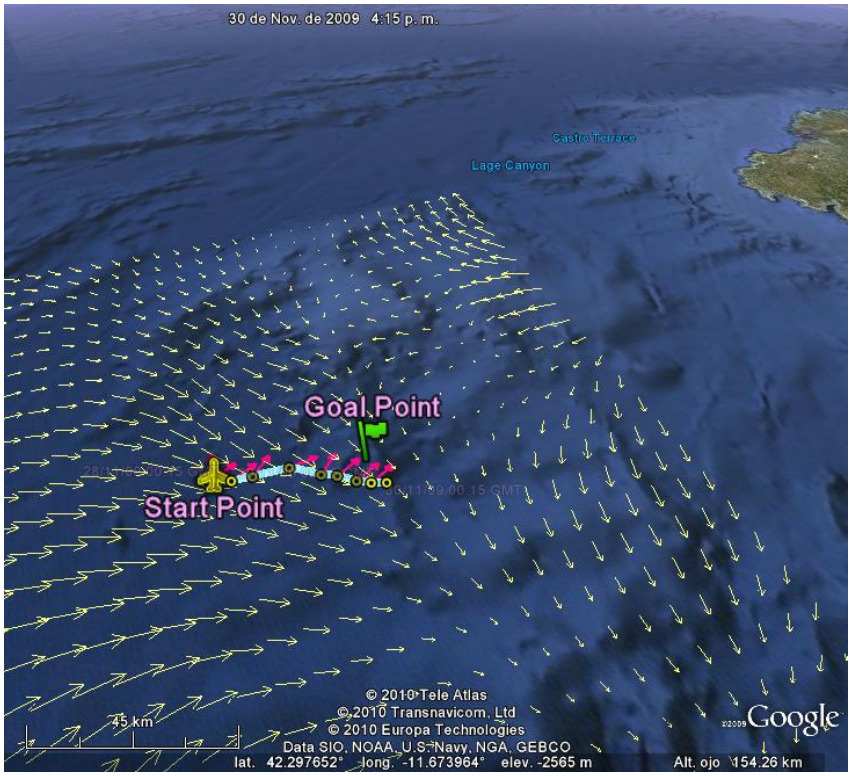


Figura 4.10: Ejemplo de camino planificado con *Pinzón* en la región ESEOAT del ROM del proyecto ESEOO usando los puntos de emersión de RU27, para alcanzar el objetivo especificado como cruzar un meridiano. La salida es una animación KML que puede ejecutarse en Google™ Earth; en la figura vemos una captura.

Capítulo 5

Resultados Experimentales

Los algoritmos desarrollados en este trabajo de tesis han sido probados para evaluar su validez y comparados entre sí sobre diferentes escenarios para determinar los más adecuados a cada problema. En estos experimentos se han utilizado, por un lado, mapas de modelos regionales de predicción, y por otro, mapas sintéticos generados para reproducir condiciones particulares de test.

Los principales problemas que se han analizado son los siguientes: camino de mínimo tiempo, camino de mínima distancia restante al destino, evitación de obstáculos, seguimiento de trayectorias, planificación multi-glider y, a modo preliminar, seguimiento de estructuras móviles.

En los problemas básicos se ha analizado el rendimiento de los diferentes algoritmos sobre una batería de casos de prueba representativos, incluyendo diferentes rangos de distancia y condiciones de corrientes. Se ha realizado una comparativa de las distintas aproximaciones aplicando las correspondientes métricas de evaluación.

5.1 Camino de Mínimo Tiempo

Para el camino de mínimo tiempo hemos probado los algoritmos DtG, RRT, A*, CTS-A* con $M = 18$ muestras, y Optimización Iterativa. Los resultados se resumen en el Cuadro 5.1.

Se han lanzado 20 casos de test que muestran como el DtG y RRT no son capaces de encontrar ninguna solución en algunos casos. Por contra, el A*, CTS-A* y el método de Optimización Iterativa, siempre encuentra solución, siendo la de éste último ligeramente mejor que los métodos basados en grafos.

Cuadro 5.1: Coste del camino (tiempo viajado) en días para el problema del Camino de Mínimo Tiempo. La media de 20 casos de test se muestra en la comparativa.

Method	All	Successful
DtG	NA in 60 %	18,0
RRT	NA in 20 %	17,7
A*	19,4	17,1
CTS-A*	19,2	16,9
ABS CTS-A*	19,2	16,9
Iterative Optimization	18,9	16,7

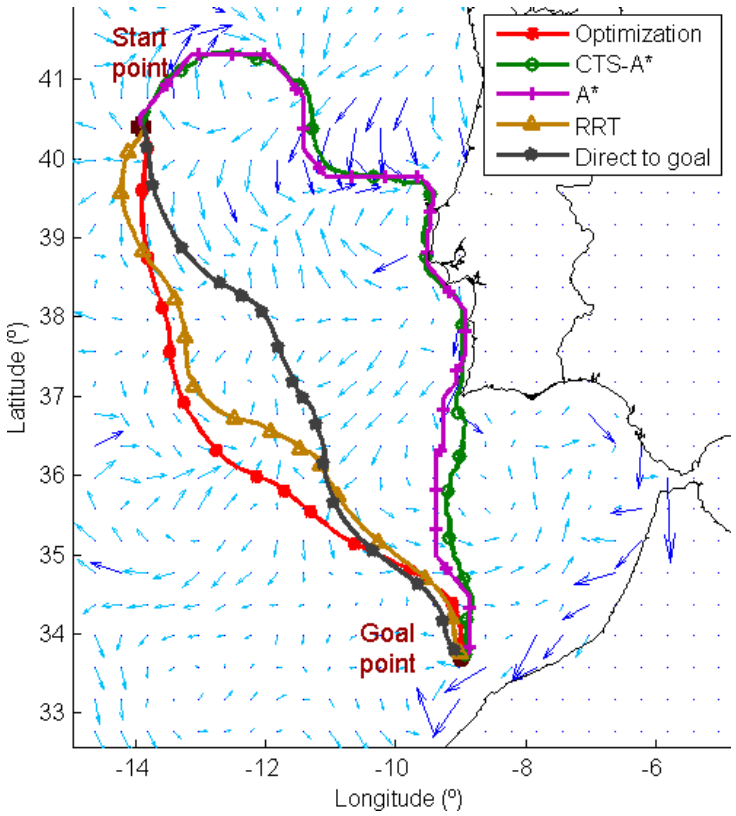


Figura 5.1: Comparativa del Camino de Mínimo Tiempo para el DtG, RRT, A*, CTS-A* con $M = 18$ muestras, y Optimización Iterativa, para un ratio de velocidad alto, i.e. para corrientes oceánicas fuertes, fijando la velocidad del glider a $U_g = 0{,}2\mathrm{m/s}$. La distancia viajada fue $d = 861{,}88\mathrm{km}$. La figura muestra el camino sobre las corrientes oceánicas ⟶ (las corrientes que exceden la velocidad del glider U_g están resaltadas ⟶) desde el waypoint inicial ● al de destino ◎. El tiempo viajado t_t (en días) para cada método fue: DtG $= 53{,}8$; RRT $= 49{,}7$; A* $= 49{,}6$; CTS-A* $= 50{,}0$; Optimización Iterativa: $47{,}4$.

La Figura 5.1 muestra un ejemplo de ejecución para el problema del camino de mínimo tiempo, utilizando una velocidad nominal del glider de $U_g = 0{,}2\mathrm{m/s}$, lo cual equivale a navegar con corrientes fuertes en este

ejemplo. Se observa claramente como la ruta del DtG y RRT son similares y mas largas que las de los métodos basados en grafos, A* y CTS-A*, y el de Optimización Iterativa, que encuentran caminos similares entre sí.

5.2 Camino de Mínima Distancia Restante al Destino

Para el camino de mínima distancia restante al destino hemos probado los algoritmos DtG, A*, CTS-A* con $M = 18$ muestras, y Optimización; no se prueba el algoritmo RRT porque manejamos corrientes dinámicas. Los resultados se resumen en el Cuadro 5.2.

Cuadro 5.2: Diferencia promedio de la distancia al destino (en km) con respecto a la aproximación DtG; positivo es mejor. Valores para los 65 casos de test ejecutado para el problema de mínima distancia restante al destino.

Method	All	Strong	Weak
A*	$0{,}5$	$4{,}9$	$-6{,}0$
CTS-A*	$5{,}2$	$8{,}6$	$0{,}2$
Optimization	$8{,}5$	$12{,}4$	$2{,}7$

Se han lanzado 65 casos de test, divididos en corrientes débiles y fuertes. La mejora respecto al DtG, que se muestra en el Cuadro 5.2, es claramente mayor con corrientes fuertes. Asimismo, el algoritmo de Optimización es el que deja al vehículo más cerca del destino, después del tiempo de misión T dado.

En la Figura 5.2 se muestra un caso de test, en el que tras para el tiempo de misión T vemos la posición final $\mathbf{x}_{n-1}$ del camino encontrado por cada algoritmo. El DtG y el A* se quedan bastante más lejos que el CTS-A* y el de Optimización.

5.3 Evitación de Obstáculos

Para la evitación de obstáculos se han seleccionado escenarios extraídos del mar de Canarias, donde la presencia de las islas permite definir configuraciones complejas de resolver para la planificación.

Se han analizado los algoritmos tanto para misiones largas con corrientes estáticas, como misiones cortas con corrientes dinámicas. Esto es equivalente a decir que se ha probado la evitación de obstáculos para el problema del camino de mínimo tiempo y el de mínima distancia restante al destino, respectivamente. A continuación se muestran los resultados para este segundo problema, resumidos en el Cuadro 5.3.

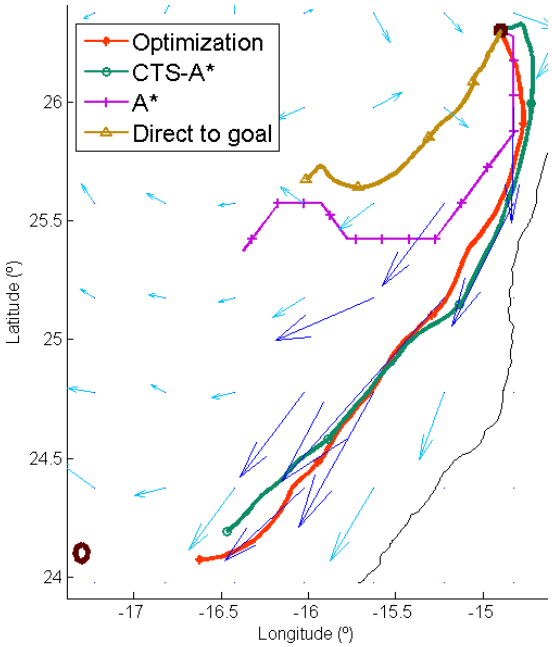
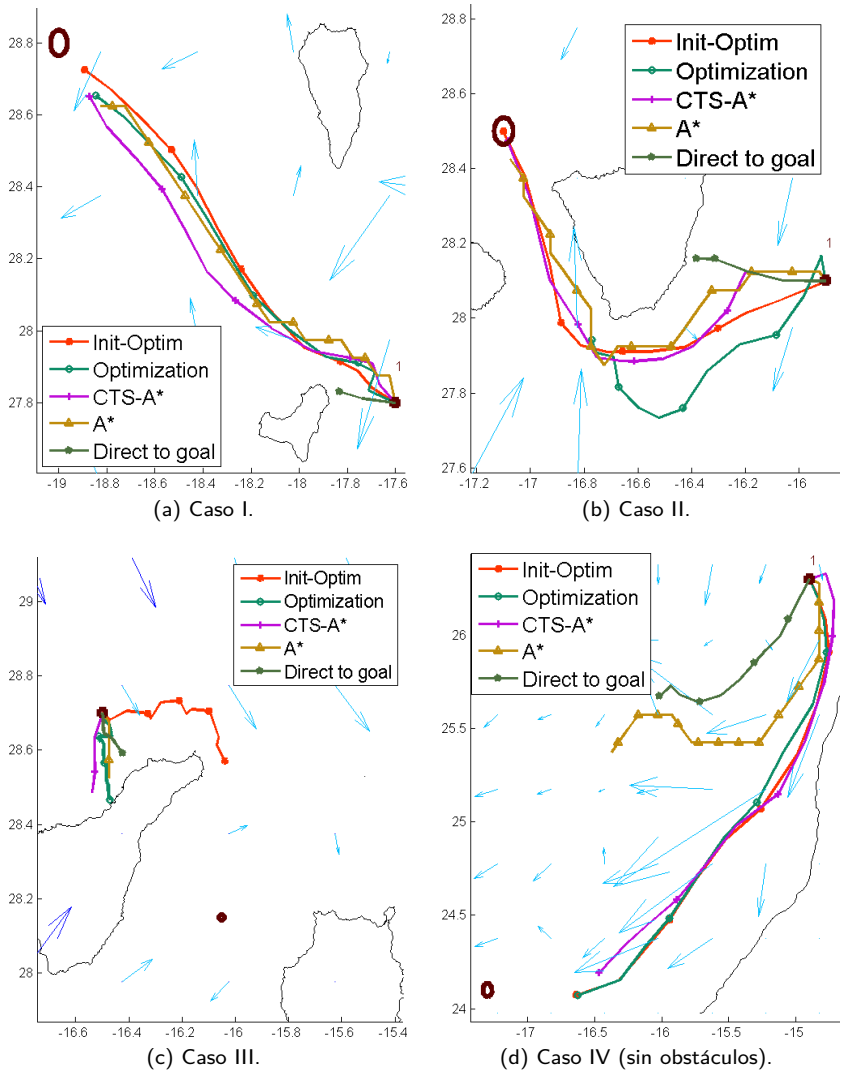


Figura 5.2: Captura de un caso de test para 4 días; la última predicción se repite para rellenar las no disponibles. La figura muestra el camino sobre las corrientes oceánicas $\longrightarrow$ (las corrientes que exceden la velocidad del glider $U_g = 0{,}4\mathrm{m/s}$ están resaltadas $\longrightarrow$) desde el waypoint inicial ● al de destino ◎. La distancia desde el waypoint inicial al de destino es $344{,}6\mathrm{km}$. La distancia restante para alcanzar el destino es, para cada algoritmo: Optimización $= 68{,}9\mathrm{km}$; CTS-A* $= 85{,}1\mathrm{km}$; A* $= 169{,}4\mathrm{km}$; DtG $= 217{,}6\mathrm{km}$.

Cuadro 5.3: Diferencia de la distancia restante para alcanzar el waypoint de destino con respecto al método Optimización-CTS-A*, para el problema de mínima distancia restante al destino. Promedio y desviación típica entre paréntesis, ambas en km. La tabla resume los 45 casos de test realizados para un velocidad del glider de $U_g = 0{,}4\mathrm{m/s}$.

Method	Total	Coast	Offshore
DtG	42,4 (46,0)	67,4 (39,0)	13,6 (24,0)
A*	8,5 (18,0)	5,3 (7,0)	9,1 (6,0)
CTS-A*	5,2 (6,0)	5,8 (7,0)	6,5 (4,0)
Optimization	10,3 (21,0)	19,6 (26,0)	0,0 (0,0)

Los resultados se han separado para zonas de costa y *offshore*, comparando cada técnica con la del método Optimización-CTS-A*, que aplica el método de inicialización inteligente utilizando una versión *gruesa* del algoritmo CTS-A*, tal y como se explica en la Sección 4.3.5.2. Observamos como lejos de costa el resultado sin esta fase de inicialización es equivalente a la usada en el método de Optimización, lo cual indica su validez en todos los escenarios. En zonas costeras, esta inicialización mejora cerca de $20\mathrm{km}$ la solución del método de Optimización. Además, se consiguen resultados en torno a $\approx 5\mathrm{km}$ mejores que con las técnicas basadas



(a) Caso I. (b) Caso II.

(c) Caso III. (d) Caso IV (sin obstáculos).

Figura 5.3: Comparativa del problema de Mínima Distancia Restante al Destino para el problema de Evitación de Obstáculos. Comparativa de 4 casos de test realizados $U_g = 0{,}4\text{m/s}$. La distancia restante al waypoint de destino is (SL si Stop in Land): (a) Optimización-CTS-A* $= 13{,}3\text{km}$, Optimización $= 22{,}1\text{km}$, CTS-A* $= 20{,}6\text{km}$, A* $= 25{,}9\text{km}$, DtG $= 157{,}1\text{km}$ (SL); (b) Optimización-CTS-A* $= 0{,}0\text{km}$, Optimización $= 69{,}7\text{km}$ (SL), CTS-A* $= 3{,}2\text{km}$, A* $= 8{,}7\text{km}$, DtG $= 80{,}0\text{km}$ (SL); (c) Optimización-CTS-A* $= 46{,}7\text{km}$, Optimización $= 54{,}2\text{km}$ (SL), CTS-A* $= 60{,}7\text{km}$ (SL), A* $= 58{,}9\text{km}$ (SL), DtG $= 61{,}3\text{km}$ (SL); (d) Optimización-CTS-A* $= 67{,}4\text{km}$, Optimización $= 68{,}8\text{km}$, CTS-A* $= 85{,}1\text{km}$, A* $= 169{,}4\text{km}$, DtG $= 217{,}6\text{km}$. Respectivamente, la distancia total desde el waypoint inicial al de destino es: $176{,}5\text{km}$, $125{,}8\text{km}$, $75{,}2\text{km}$, $343{,}4\text{km}$.

en grafos. Más lejos aún queda la aproximación DtG.

En la Figura 5.3 se muestran 4 casos de evitación de obstáculos cerca de islas. En (d) se incluye el caso

sin obstáculos visto previamente, a efectos comparativos. Podemos observar como en algunos casos de test la mayoría de técnicas quedan atrapadas; nótese que las técnicas basadas en grafos pueden verse atrapadas debido las corrientes en torno a determinadas zonas.

5.4 Seguimiento de Trayectorias

En el caso del seguimiento de trayectorias se ha comprobado la capacidad de las soluciones propuestas para mantener la ruta del glider sobre la ruta deseada. También se ha comprobado la mejora en el rendimiento que se consigue relajando ligeramente las condiciones del seguimiento.

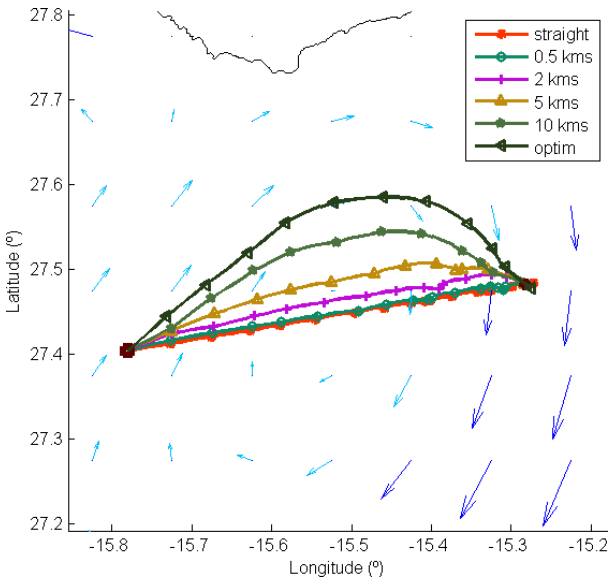


Figura 5.4: Comparativa de caminos encontrados con un glider viajando a $U_g = 0{,}4\mathrm{m/s}$ para cruzar un giro desde un lado a otro, para el problema del Seguimiento de Trayectorias. El Camino de Mínimo Tiempo también se incluye. Los puntos de emersión se marcan a lo largo del camino. Las corrientes oceánicas $\longrightarrow$ (las corrientes que exceden la velocidad del glider U_g están resaltadas $\longrightarrow$) y la línea de costa también se muestran.

Precisamente, la Figura 5.4 muestra los caminos relajando la condición de seguimiento de la trayectoria deseada, para diversos anchos desde $0{,}2\mathrm{km}$ a $10\mathrm{km}$. El camino deseado se definió como una línea recta, y el método de Optimización Iterativa se ejecutó con una restricción de seguimiento de trayectoria, alrededor del camino deseado, según el rango de relajación anterior. Igualmente, se calcula la trayectoria que sigue estrictamente el camino deseado.

Con la relajación de la condición de seguimiento de la trayectoria obtenemos varias soluciones, que definen

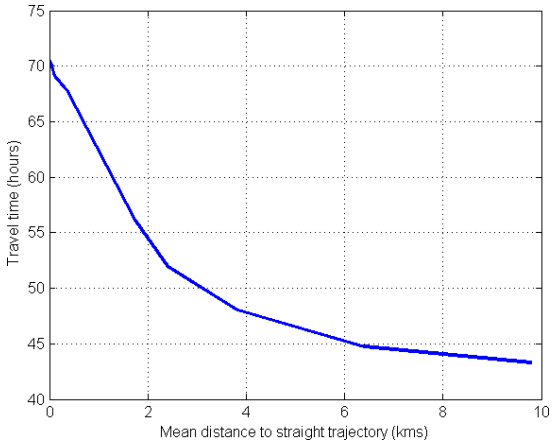


Figura 5.5: Curva de Pareto que muestra el tiempo viajado frente al promedio de la distancia al camino deseado, para los caminos encontrados en la Figura 5.4.

la curva de Pareto mostrada en la Figura 5.5. La curva de Pareto muestra los dos objetivos contrapuestos uno frente al otro, i.e. el tiempo viajado (eje y) frente al promedio de la distancia al camino deseado (eje x) —que es la línea recta en este ejemplo. Todos los caminos intentan mantener el vehículo cerca de la línea recta, pero esta restricción se relaja en cada simulación incrementando el tamaño del pasillo alrededor de la línea. Como muestra la figura, mientras más estrecho es el pasillo, mayor es el tiempo viajado. Mientras tanto, al agrandarse el pasillo, nos aproximamos al camino de mínimo tiempo de la optimización sin restricciones.

Finalmente, es posible concatenar múltiples problemas de seguimiento de trayectorias, como muestra el patrón en forma de molino de la Figura 5.6. Éste está compuesto por múltiples trayectorias en forma de líneas rectas que deben ser seguidas. Así, con la solución a este problema se pueden realizar aplicaciones de muestreo de giros siguiendo, por ejemplo, este patrón de molino.

5.5 Planificación Multi-glider

Para la planificación multi-glider se han seleccionado escenarios en los que los gliders debían mantener una posición relativa fija con respecto a un vehículo líder. También se han planteado problemas de optimización del tiempo de recuperación de múltiples unidades desplegadas.

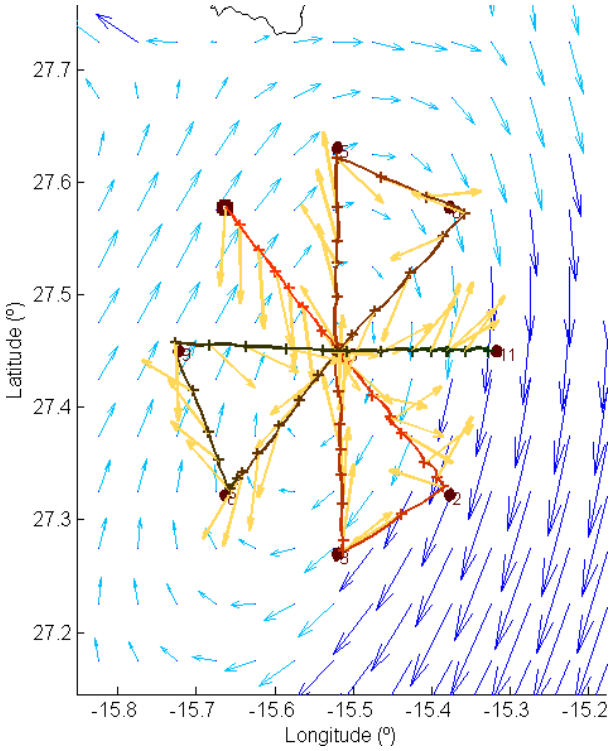
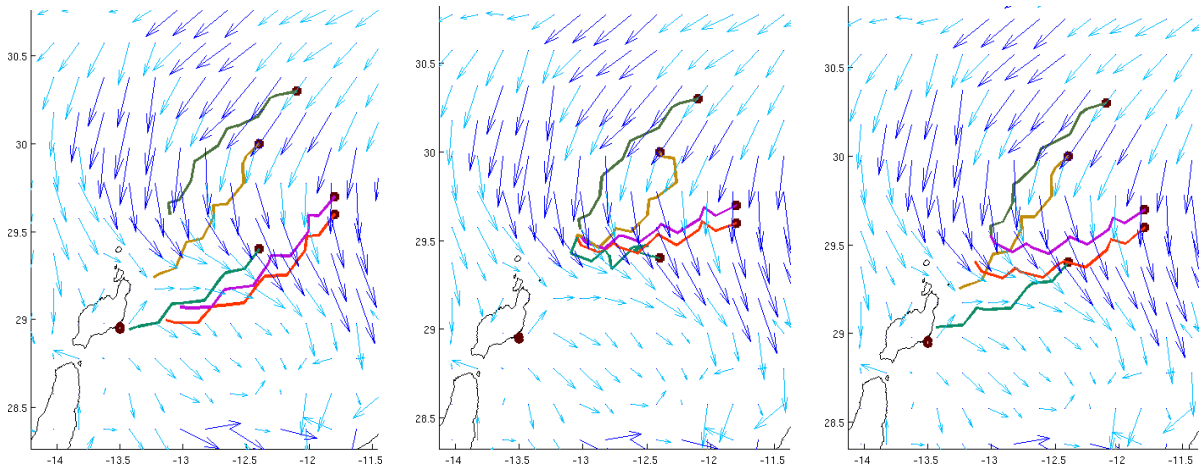


Figura 5.6: Ensamblaje de problema de seguimiento de trayectorias para realizar una misión de muestreo y caracterización de un giro mesoescalar. La trayectoria de molino se descompone en líneas rectas que cruzan el giro múltiples veces desde diferentes direcciones, para las cuales se resuelve el problema de seguimiento de trayectorias usando el método de Optimización Iterativa. Se muestran las direcciones del glider ⟶ en cada punto de emersión, junto con las corrientes oceánicas ⟶ desde el waypoint inicial ■ al de destino ○.

5.5.1 Recogida Eficiente

Para la recuperación de múltiples gliders tras una misión, planteamos el problema de su recogida eficiente con un barco. Este problema requiere la planificación coordinada de los caminos de todos los gliders de la flota, para así reducir el tiempo de recogida y, por consiguiente, el gasto económico de la mismo, atendiendo al combustible consumido por el barco.

En la Figura 5.7 se muestran varias soluciones para el problema de recogida eficiente utilizando el método de Optimización. Se busca la recogida de mínimo tiempo, a la vez que se intenta resolver el problema de mínima distancia restante al destino para un período de 4 días. Los gliders viajan a $U_g = 1\mathrm{km/h}$, con un tiempo entre emersiones de $t_s = 8\mathrm{h}$. En (a), tenemos las solución con planificación de caminos individual. Por el contrario, las otras dos figuras muestran los resultados usando planificación de caminos multi-glider. En (b), todos los gliders intentan encontrarse en un punto para la recogida. Finalmente, en (c), el camino del



(a) Individual. (b) Encontrarse juntos. (c) Camino del barco.

Figura 5.7: Planificación de caminos multi-glider y coordinación para una flota de 5 gliders que tienen que ser recuperados del mar por un barco en el mínimo tiempo. El barco tiene que viajar: (a) $192{,}0\mathrm{km}$; (b) $167{,}5\mathrm{km}$; (c) $165{,}4\mathrm{km}$, respectivamente.

barco se incluye en el proceso de optimización. Esta última aproximación es la que da mejores resultados.

5.6 Seguimiento y Muestreo de Estructuras Oceánicas Móviles

Finalmente, hemos realizado algunos simulaciones preliminares de la técnica comentada en la Sección 4.4.2, para realizar el seguimiento y muestreo de estructuras oceánicas móviles. Concretamente, hemos modelado giros mesoescalares e intentamos seguirlos al mismo tiempo que se muestran determinadas zonas de interés de los mismos.

La Figura 5.8 muestra una captura de las corrientes oceánicas y la trayectoria optimizada al final de la misión; nótese que el giro mueve mientras se muestrea. En este ejemplo se intenta muestrear de manera óptima un región situada en el borde del giro.

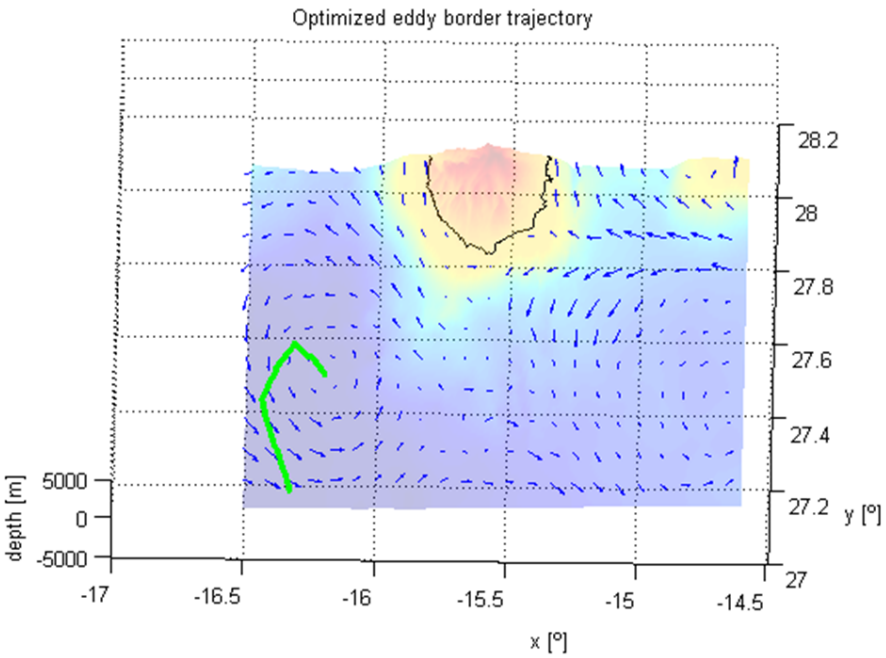


Figura 5.8: Ejemplo en el que el contorno de un giro mesoescalar se sigue usando nuestra aproximación con un Algoritmo Genético y el modelo discreto del giro.

Capítulo 6

Conclusiones

En esta tesis se ha abordado el estudio del problema de la planificación automática de rutas para gliders a partir de modelos de predicción numérica. Se ha demostrado que es posible mejorar las capacidades operativas de estos vehículos introduciendo herramientas de asistencia al pilotaje.

6.1 Aportaciones Originales

Como principales aportaciones conviene destacar el conjunto de algoritmos de planificación desarrollados. Las herramientas complementarias generadas constituyen asimismo aportaciones válidas que facilitan el trabajo en este ámbito de manera importante.

Además de lo anterior, en la tesis se han realizado diferentes análisis cuyas conclusiones pueden considerarse contribuciones en sí mismas, como es el caso de la revisión bibliográfica comentada, los estudios sobre las alternativas de modelado o el análisis de aplicabilidad de las técnicas de optimización.

De hecho, tras el análisis bibliográfico hemos visto que el problema del camino de mínimo tiempo no es común en la robótica de vehículos móviles terrestres, donde el problema de planificación de caminos más común es el del camino más corto. La principal diferencia radica en el hecho de que en el camino de mínima tiempo no sólo afecta la distancia entre origen y destino, sino también la velocidad del vehículo, que en el caso de la planificación de rutas para gliders se ve afectada por las corrientes U_c (véase la Sección 2.2).

El algoritmo A* se usa ampliamente para el problema del camino más corto. Varios autores lo han adaptado al problema del camino de mínimo tiempo para gliders (Garau et al., 2005, 2009; Soulignac, 2010).

Básicamente, se incorpora el modelo de movimiento restringido de la Sección 2.2.1.2, para encontrar el camino óptimo dentro de la rejilla del grafo de búsqueda del algoritmo. La discretización de la rejilla obliga a usar el modelo de movimiento restringido, el cual en presencia de corrientes puede llevar a caminos donde no se puede expandir ningún nodo. Además, el tiempo entre un nodo y su vecino no es constante ni igual al tiempo entre emersiones t_s, no reflejando así la forma en que navega un glider. Igualmente, tampoco se integra la trayectoria del perfil yo-yo. El análisis de estas limitaciones es una primera contribución de este trabajo de tesis.

El estudio de la literatura de planificación de caminos para gliders, así como la de problemas de planificación de caminos con costes variables y asimétricos, muestra pocas contribuciones en este área. Los problemas más similares —i.e. que manejan costes asimétricos variables— son: el Problema del Viajante Canadiense (CTP), rutas de tráficos dinámicas, y mapas de coste o planificación con costes.

En teoría de grafos, el CTP es una generalización del problema del camino más corte a grafos parcialmente observables, i.e. el grafo se obtiene mientras es explorado (Papadimitriou, 1991). El concepto de observabilidad parcial cubre los costes variables del problema del camino de mínimo tiempo para gliders. De manera similar, en los problemas de rutas de tráfico dinámicas, la densidad de vehículos produce costes variables para viajar dentro de la red de rutas. Esto es equivalente a los mapas de coste, que habitualmente se asocian a la planificación de caminos basada en relieve. De hecho, podríamos construir un mapa de coste a partir del campo vectorial de corrientes oceánicas, pero esto sería equivalente a aplicar el modelo de movimiento restringido usado por el A*. Aunque, en nuestra humilde opinión, esta es la categoría en que se ubican los problemas de planificación de caminos, en realidad no representan la navegación del glider ni ciertas características de su funcionamiento con facilidad. Más concretamente, estos problemas no incluyen la restricción de emersiones de tiempo constante.

Antes que un análisis teórico del problema y los algoritmos de planificación de caminos, este trabajo adopta un enfoque pragmático con el estudio de métodos aplicables a la planificación de rutas para gliders. Hemos adaptado métodos clásicos basados en grafos como el A*, y técnicas de muestreo probabilístico como el RRT, siguiendo trabajos previos de Rao and Williams (2009). Debido a las limitaciones del A* y el RRT, hemos desarrollado varias técnicas novedosas para eliviar sus defectos. Aplicamos el modelo de movimiento no restringido y reflejamos la restricción de emersiones separadas un tiempo constante. Hacemos estos tanto para el algoritmo CTS-A* como para las diferentes variantes de los métodos de optimización descritos en la tesis.

Hemos realizados varios experimentos para evaluar la validez, optimalidad y tiempo de cómputo. Para casi

todos los casos de test, los mejores caminos son encontrados con los métodos de optimización, que además es el más rápido. El CTS-A* también encuentra caminos muy cerca del óptimo, aunque con algo más de tiempo de cómputo. Con la estrategia de muestra ABS se reduce su tiempo de cómputo, puesto que permite reducir el número de muestras para las direcciones que se integran desde cada nodo. Además, un versión *gruesa* del CTS-A* se usa en la fase de inicialización de los métodos de optimización para evitar obstáculos en entornos costeros.

Los resultados experimentales también muestran que la técnica de optimización es fácil de adaptar a un buen número de problemas de planificación de caminos para gliders. Esto incluye el seguimiento de trayectorias, las planificación multi-glider y el seguimiento de estructuras oceánicas móviles. Sin embargo, las ventajas de nuestro esquema de optimización residen en las particularidades de los gliders. Por ello, tales beneficios se perderían si se aplica a la planificación de rutas con otro tipo de vehículo. Es en las emersiones separadas un tiempo constante donde encontramos la razón de esta falta de generalización. La optimización se realiza en el espacio de direcciones, en lugar de en el espacio de localizaciones. En consecuencia, gracias los tramos entre emersión de $t_s \approx 8\mathrm{h}$, la dimensionalidad del problema de optimización se reduce drásticamente, lo cual ayuda a obtener tiempos de cómputo bajos. Claramente, si usásemos vehículos con una tasa de control —i.e. con un tiempo t_s menos—, el número de direcciones $|\mathcal{B}|$ aumentaría, perdiendo esta ventaja.

En el caso de aproximaciones como los métodos basados en grafos, éstos tienen que manejar la gran dimensionalidad del problema, definido en 4 dimensiones en el caso más general —i.e. 3 dimensiones espaciales y tiempo. Además, varias direcciones ψ_g deben considerarse al generar los sucesores cuando se expande un nuevo nodo en la rejilla de búsqueda. Como resultado, el tiempo de cómputo es grande en estas aproximaciones.

Hemos usado modelos de movimiento puntuales en nuestros algoritmos de planificación de caminos, puesto que resulta una elección razonable, dada la incertidumbre de los modelos oceánicos y la localización de los gliders. Además, nos consideramos cambios de cabeceo θ o límites de profundidad $[z_{\mathrm{mín}}, z_{\mathrm{máx}}]$, puesto que generalmente no se modifican en la práctica. En algunos casos simulamos la deriva del glider mientras espera en superficie, pero normalmente sólo cuando el resultado se usa posteriormente para asistir en el pilotaje de gliders en misiones reales. En cualquier caso, también hemos evaluado un modelo cinemático de equilibrio de fuerzas (véase la Sección 2.2.2) y comparado sus resultados con los puntos de emersión del glider RU27 en la misión trans-Atlántica. Como ya se ha mencionado, la precisión de estos modelos no se requiere para la planificación de rutas con gliders, aumentarían el tiempo de cómputo del planificador sin ningún beneficio tangible, y requieren un mayor número de parámetros —muchos de ellos difíciles de cuantificar.

Finalmente, hemos avanzado en los primeros pasos en el muestreo de estructuras oceánicas móviles. En particular, hemos desarrollado una técnica para seguir giros mesoescalares y mejorar la estrategia de muestreo (véase la Sección 4.4.2). Este problema tiene enorme interés en la asimilación de datos, y su uso para mejorar el modelado de estas estructuras en los modelos oceánicos.

Part II

Resumen Extendido en Español

Capítulo 1

Introducción

La oceanografía es la disciplina que se encarga del estudio de los océanos. Se trata de un ámbito complejo en el que convergen múltiples áreas de interés como son el estudio de las corrientes marinas, el oleaje o la dinámica de fluidos; tectónica de placas y geología del fondo marino; organismos marinos y dinámica de ecosistemas; difusión de sustancias químicas y propiedades físicas del océano.

El estudio de los diferentes procesos presentes en el medio marino requiere la definición y ajuste de modelos que permitan caracterizar su dinámica y ayudar a predecir su evolución. Un aspecto clave, del que depende en gran medida el éxito de esta tarea, es la necesidad de disponer de sistemas de muestreo adecuados que suministren datos reales con los que contrastar la bondad de dichos modelos. Esta tesis se centra en los aspectos ligados al uso de dispositivos robóticos en la monitorización del medio marino.

1.1 Técnicas de muestreo

Tradicionalmente se han empleado en la observación del océano diferentes sistemas de captura de datos como son la tele-detección basada en satélite, radares, boyas o los buques oceanográficos. En las últimas décadas han surgido nuevas técnicas de muestreo que intentan suplir diferentes limitaciones identificadas en los sistemas más tradicionales. Se trata de los vehículos autónomos submarinos.

Podemos distinguir dos tipos principales de vehículos submarinos: los operados remotamente o ROVs, y los autónomos o AUVs. Dentro de estos últimos, existe un tipo especial, que son los planeadores submarinos o gliders. El presente trabajo está enfocado a proporcionar herramientas destinadas a este tipo de vehículos.

1.2 Planeadores Submarinos Autónomos

Los Planeadores Submarinos Autónomos, también denominados comúnmente planeadores o *gliders* —atendiendo al término en inglés—, son un tipo de vehículo autónomo submarino que se desplaza usando un mecanismo de propulsión único basado en modificar su flotabilidad. Estos vehículos poseen un sistema que emula el funcionamiento de la vejiga natatoria de los peces para provocar movimientos de ascenso y descenso cíclicos. Dichos desplazamientos verticales se traducen en una lenta pero efectiva velocidad de avance gracias a la interacción de los planos de control y sustentación con el medio acuático. Se generan de esta manera perfiles de movimiento en "V" característicos, denominados yo-yos.

Un glider eléctrico, como el modelo SLOCUM de la Figura 1.1, utiliza una bomba hidráulica para trasvasar un cierto volumen de aceite entre el interior y el exterior del casco del vehículo. De esta forma, el consumo de energía necesario para la propulsión se produce únicamente en los puntos de inflexión en los que el robot activa la bomba, permitiendo unos niveles de autonomía difíciles de igualar para otros vehículos.

Figura 1.1: Spray de Bluefin Robotics (delante), SLOCUM de Teledyne Webb Research (medio) y Seaglider de iRobot (detrás), en el Laboratorio de Gliders en las instalaciones de la Plataforma Oceánica de Canarias (PLOCAN). Cortesía de PLOCAN. Nótese que las alas son extraíbles para facilitar el transporte y mantenimiento, y por esa razón han sido extraídas del casco en los vehículos aquí mostrados.

Los gliders operan en base a la definición de unos puntos de paso o waypoints que el robot trata de alcanzar sucesivamente. Una vez fijado el rumbo hacia el próximo objetivo el vehículo ejecuta una serie predefinida de ciclos de ascenso/descenso, denominados transecto o stint; y que están delimitados entre dos valores de profundidad mínima y máxima, tras los cuales emerge de nuevo. Una vez en superficie, el glider se localiza por medio del GPS —o por cualquier otro GNSS—, transmite datos básicos a su centro de control y, en su caso, recibe nuevas órdenes. Transcurrido un corto periodo de tiempo, en torno a $15 - 20$ minutos, el glider inicia un nuevo stint (tramo sumergido), cuya duración suele oscilar entre 6 y 8 horas.

1.3 Planificación de rutas

El principal inconveniente que tiene el uso de los gliders en la monitorización del medio marino es su dependencia de las condiciones del medio. La lenta velocidad de avance que los caracteriza (aproximadamente un kilómetro por hora) puede llegar a ser igualada e incluso superada por las corrientes marinas. Por lo tanto, es necesario anticipar su efecto en la trayectoria del vehículo si se desea conseguir alcanzar un punto determinado o seguir una ruta deseada. La fuente de información básica con la que se trabaja son los modelos de predicción oceánicos, que permiten tener en cuenta con varios días de antelación el estado más probable de la zona en la que se encuentre el glider.

Dependiendo del tipo de misión y la región que se esté considerando, existirán diferentes modelos disponibles. En general, los mejores resultados son los aquellos que se obtienen con modelos de predicción regionales, especialmente ajustados para capturar las condiciones particulares de una determinada zona.

En un esquema de pilotaje de gliders tradicional, un operador humano realiza un seguimiento del estado presente y futuro del mar y de la evolución del vehículo, decidiendo cuándo es necesario modificar los puntos de destino. En una misión real, las propias medidas del glider se convierten en un indicio de si las predicciones de los modelos oceánicos se están verificando o no. La Figura 1.2 ilustra la trayectoria final del glider RU27 tras finalizar la misión trans-Atlántica. Ésta se llevo a cabo mediante un pilotaje manual o tradicional. No obstante, durante los últimos stints cerca de las costas españolas, este pilotaje fue asistido por las herramientas de planificación de caminos para gliders desarrolladas en este trabajo de tesis.

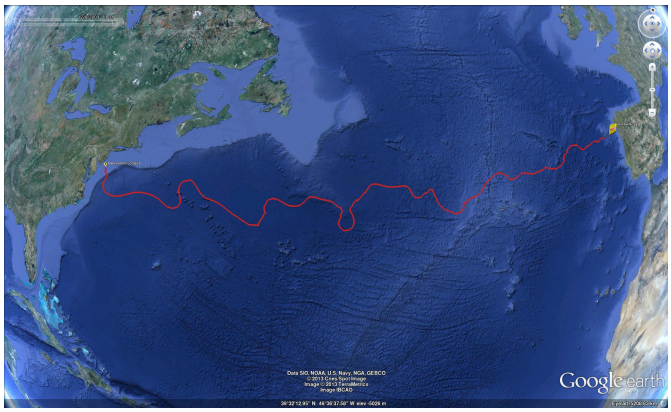


Figura 1.2: Camino seguido por el glider RU27 *Scarlet Knight* (SLOCUM) en la misión trans-Atlántica, realizada en 2010. La misión fue dirigida por la Universidad de Rutgers, con la colaboración de la División de Robótica y Oceanografía Computacional (DROC), SIANI.

1.4 Hipótesis de la tesis

En este trabajo se plantea como hipótesis el que pueden definirse mecanismos automáticos para la planificación de rutas en gliders que aporten ventajas significativas en el uso de estos dispositivos como instrumentos de muestreo del océano.

1.5 Objetivos

Los objetivos de esta tesis se resumen en las siguientes cuestiones que trata de resolver. Asimismo, definen el ámbito y extensión del presente trabajo.

- La literatura sobre planificaciones de caminos recoge un importante número de algoritmos usados para obtener la secuencia (cuasi-)óptima de configuraciones de estado que permita a un vehículo moverse desde una posición a otra. Es por tanto de gran interés estudiar si estos algoritmos son aplicable en el campo de la planificación de rutas para gliders. De hecho, en esta tesis analizamos si el método A* es aplicable. Si puede ser adapta al problema con gliders, o si hay alguna particularidad que lo impide, o hace que resulte ineficiente.
- Veremos como las técnicas clásicas no son directamente aplicables a nuestro problema, a menos que sea adaptadas. En este sentido trataremos de analizar cuáles son las particularidades de la planificación de caminos con gliders que justifican la necesidad de esta adaptación. Así, obtendremos conocimiento sobre las siguientes cuestiones:
  - Cuáles son las diferencias respecto a otros problemas que se solucionan empleando planificadores de caminos clásicos? Nos referimos a problemas como la planificación de caminos en laberintos, mapas de coste, etc.
  - Qué razones explican por qué las técnicas clásicas no son aplicables o útiles en este caso? O que impide que puedan manejar eficientemente el problema?
- El estudio de la bibliografía y el estado del arte nos permitirá categorizar la tipología del problema que intentamos resolver. Es decir, en qué categoría taxonómica ubican el problema otros autores, o al menos establecer el tipo de problema más similar.

- Desde un punto de vista más práctico, pretendemos investigar y desarrollar métodos que resuelvan el problema. En este trabajo implementamos varios técnicas novedosas. Éstas son evaluadas conforme a dos criterios. Primero, evaluamos la optimalidad de la solución encontrada o lo lejos que queda del óptimo. Segundo, el tiempo de cómputo y uso de memoria.

- De esta forma, podremos comparar nuestros resultados con otras aproximaciones de la literatura sobre el tema. Así podremos establecer cómo atacan el problema otros autores, así como las simplificaciones y los supuestos que éstos asumen. Consecuentemente, podremos valorar si éstos y nuestros algoritmos son escalables y generalizables para resolver otro tipo de problemas de planificación de rutas.

- Respecto a la dimensionalidad del problema, veremos que los modelos oceánicos ofrecen 4 dimensiones: espacio tridimensional y tiempo. Los gliders navegan por este espacio tridimensional que cambia significativamente en el tiempo. Asimismo, es posible considerar o despreciar ciertas características de la dinámica del vehículo y su navegación, como el ángulo de planeo, las profundidades mínima y máxima de los perfiles yo-yo, la velocidad nominal, etc. La forma en que manejar toda esta complejidad y la alta dimensionalidad del problema eficientemente, es también un aspecto fundamental que se trata en el presente trabajo. Igualmente, queremos ver si es posible garantizar la optimalidad de la solución encontrada, o simplemente hasta cierto punto y bajo ciertas suposiciones.

- A partir de la implementación y evaluación de deferentes técnicas obtendremos cierta intuición respecto a cuáles son más apropiadas para determinadas condiciones y entornos. Por ello, también se realiza una categorización de escenarios: áreas con obstáculos, zonas de alta variabilidad temporal, corrientes fuertes, etc.

Las cuestiones anteriores definen el ámbito y los objetivos de la tesis, puesto que ésta trata de darles respuesta. Adicionalmente, para poder tratar con la complejidad del problema, se han asumido ciertas simplificaciones y marcado algunas limitaciones.

- La alta dimensionalidad del problema se ha reducido hasta cierto punto. Asumimos un modelo de movimiento del glider simplificado. En algunos casos la planificación de rutas se hace considerando sólo las corrientes bidimensionales en superficie, con una resolución temporal de horas; mientras que en otros casos tomamos corrientes tridimensionales, aunque como medias diarias. Esta reducción en la dimensionalidad del problema es habitualmente aceptable, al mismo tiempo que reduce la complejidad sustancialmente.

- Algunos detalles de las misiones y el pilotaje de gliders son despreciados. En muchos casos no es posible observar fenómenos como el *biofouling*, que afecta a la velocidad y equilibrado del glider. Esto produce un cambio en el ángulo de planeo y puede producir una desviación sistemática en el rumbo. Del mismo modo, varía el centro de masas y la flotabilidad, reduciéndose la velocidad del vehículo. En otros casos, sólo podemos estimar ciertos parámetros con un determinado error, como el tiempo que permanece el vehículo en superficie. Estos valores no se suelen modelar porque en realidad nos interesan otros aspectos del sistemas y podemos ignorar sus efectos en la mayoría de casos.

- El sistema embebido y el software en los gliders modernos tiene capacidades muy reducidas. Por ello en la práctica el pilotaje automático o autónomo no es trivial. Aunque no se trata estrictamente de una limitación, hemos recogido este hecho en nuestra simulaciones. Puesto que la calidad del *dead-reckoning* es mala, las comunicaciones no son posibles, y la deriva producida por las corrientes no es observable bajo el agua, se requiere una planificación de rutas con comandos discretos espaciados significativamente en el tiempo. Incluso si fuera posible variar el rumbo del vehículo con mayor frecuencia, el vehículo aún no podría disponer de los mapas de corriente abordo para actualizar el rumbo convenientemente.

Capítulo 2

Problemas y Aplicaciones

En un primer nivel de problemas básicos está la optimización de rutas, que puede verse como un recurso capacitante para abordar otras aplicaciones más complejas. En un segundo nivel se consideran la evitación de obstáculos, el seguimiento de trayectorias, la planificación multi-glider o el muestreo de estructuras oceánicas móviles.

2.1 Optimización de Rutas

2.1.1 Camino de Mínimo Tiempo

El problema del camino de mínimo tiempo consiste en la minimización del tiempo requerido por un glider para llegar a un destino determinado. La principal dificultad de este planteamiento reside en tratar de abordar su resolución reflejando de la forma más fiel posible el funcionamiento de un glider real.

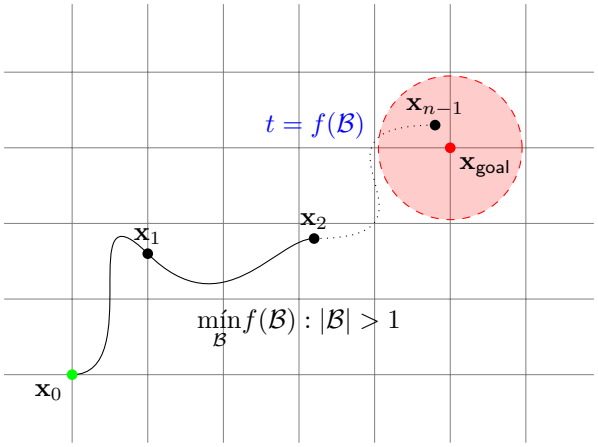


Figura 2.1: Diagrama del problema del camino de mínimo tiempo. Buscamos el camino $\mathcal{P}$ —formado por un conjunto de direcciones (*bearings*) $\mathcal{B}$— que minimice el tiempo viajado $t_t = f(\mathcal{B})$ para ir desde la localización inicial $\mathbf{x}_0$ a la de destino $\mathbf{x}_{\text{goal}}$. El número de direcciones $|\mathcal{B}|$ no es fijo, puesto que éste realmente determina el tiempo viajado. El destino $\mathbf{x}_{\text{goal}}$ se considera alcanzado con el punto final $\mathbf{x}_{n-1}$ cae en un radio alrededor suyo.

La Figura 2.1 ilustra este problema, en el que se busca el camino óptimo desde el waypoint inicial $\mathbf{x}_0$ al

de destino $\mathbf{x}_{\text{goal}}$. En este problema, utilizando un esquema de optimización no es posible resolverlo de manera directa ya que se desconoce a priori el número de variables implicadas, i.e. el número de direcciones $|\mathcal{B}|$ que forma el camino. La resolución pasa, por tanto, por implementar un esquema iterativo.

2.1.2 Camino de Mínima Distancia Restante al Destino

El problema del camino de mínima distancia restante al destino consiste en tratar de acercarse lo más posible a un punto de destino en un tiempo determinado. El planteamiento es bastante directo cuando se aborda con técnicas de optimización, puesto que se pueden considerar los rumbos a comandar como las variables del proceso de optimización. Además, la resolución es directa, puesto que el número de direcciones $|\mathcal{B}|$ es conocido de antemano.

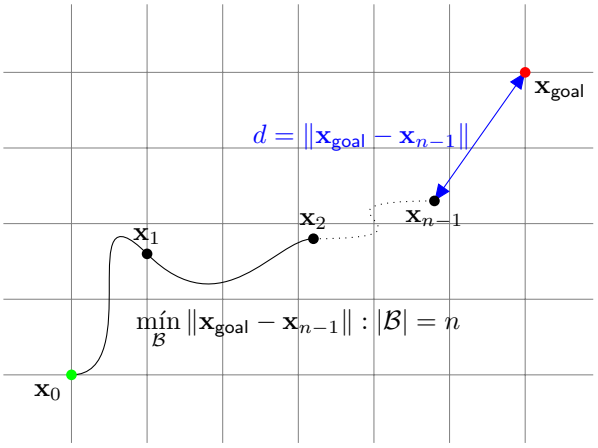


Figura 2.2: Diagrama del problema de mínima distanca al destino. Buscamos el camino $\mathcal{P}$ defino por un conjunto de n direcciones $\mathcal{B}$ que minimicen la distancia $d = \|\mathbf{x}_{\text{goal}} - \mathbf{x}_{n-1}\|$ desde el punto final $\mathbf{x}_{n-1}$ hasta el destino $\mathbf{x}_{\text{goal}}$. En este problema el número de direcciones $|\mathcal{B}| = n$ es fijo y conocido por anticipado.

La Figura 2.2 ilustra este problema, donde partimos de $\mathbf{x}_0$ y llegamos hasta la localización final $\mathbf{x}_{n-1}$ tras comandar $|\mathcal{B}| = n$ direcciones, que forman el camino a buscar, para una duración de tiempo fija y dependiente de n según el tiempo entre emersiones t_s. La optimización minimiza la distancia restante d.

2.2 Evitación de Obstáculos

Una de las principales limitaciones de los gliders es su reducida velocidad y maniobrabilidad. Esto hace que determinadas trayectorias deban de ser cuidadosamente planteadas si se quiere llegar a una solución válida. Ejemplos de esos escenarios son la presencia de obstáculos en las inmediaciones de la ruta a planificar. Dichos obstáculos pueden materializarse de diferente forma, incluyendo obstáculos fijos como islas, costa, aguas someras, puertos, etc.; móviles como barcos; o restricciones del tipo corrientes de intensidad similar o superior a la velocidad de avance del glider.

2.3 Seguimiento de Trayectorias

Otro conjunto de problemas de interés son aquellos en los que se requiere que el vehículo siga una ruta predefinida de la forma más ajustada posible. Los procesos de optimización en este caso requieren una

verificación más estricta, puesto que debe tratar de garantizarse un comportamiento adecuado tanto en los puntos de emersión como a lo largo de los stints.

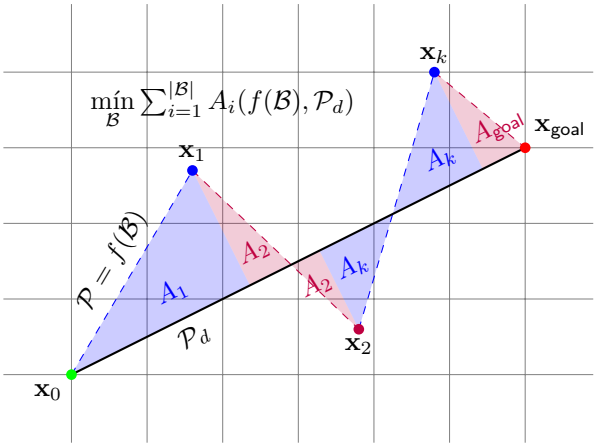


Figura 2.3: Diagrama del problema de seguimiento de trayectorias. Buscamos el camino $\mathcal{P}$ defino por un conjunto de direcciones $\mathcal{B}$ que minimice la desviación respecto al camino deseado $\mathcal{P}_d$. Tal desviación puede medirse como una distancia d o una función del área encerrada A. Este problema de seguimiento de trayectorias se resuelve como una restricción sobre el problema del camino de mínimo tiempo, que permite alcanzar el waypoint de destino $\mathbf{x}_{\text{goal}}$.

Para determinar la separación o desviación respecto a la trayectoria deseada se pueden aplicar diversas fórmulas para su cálculo. En principio, se trata del cómputo del área de un trapezoide recta entre el camino deseado y el resultante, computada para cada pareja de puntos de emersión consecutivos, tal y como muestra la Figura 2.3.

2.3.1 Relajación de la restricción de seguimiento

En la mayoría de estas aplicaciones, la falta de flexibilidad en la trayectoria se traduce en un pero aprovechamiento de las corrientes por parte del glider. Una ligera relajación de las condiciones, por ejemplo, se traduce en una ruta más adecuada en términos del tiempo invertido.

Se puede definir un pasillo alrededor del camino deseado. De esta forma es posible relajar a medida la restricción de seguimiento de la trayectoria. De hecho, las soluciones que se obtienen para que pasillo de mayor tamaño definen una curva de Pareto de soluciones no dominadas, para el objetivo de seguimiento de trayectorias y del camino de mínimo tiempo entre los extremos del camino deseado.

2.4 Planificación Multi-glider

Con frecuencia nos encontramos escenarios en los que varios robots deben cooperar para resolver una determinada tarea. Surgen entonces problemas que no pueden tratarse de forma independiente para cada vehículo. Ejemplos de este tipo de aplicaciones son el muestreo en formación o la coordinación para facilitar las tareas de recuperación de los gliders tras finalizar una misión conjunta.

Otra problema multi-glider, con una aplicación clara en misiones con múltiples vehículos, es el de la recogida eficiente de los mismos. Un barco debe recoger la flota formada por k vehículos en el menor tiempo posible, puesto que con ello se reduce el coste económico de la recogida —al reducir el gasto de combustible.

2.5 Muestreo de Estructuras

Un conjunto interesante de aplicaciones son aquellas en las que se necesita muestrear una estructura que evoluciona en el tiempo. La planificación en estos casos debe tener en cuenta de forma simultánea la previsión de las condiciones marinas y la evolución del proceso a muestrear, de forma que se maximicen las posibilidades de obtener una buena caracterización.

2.5.1 Estructuras Coherentes Lagrangianas

Igualmente, es posible obtener Estructuras Coherentes Lagrangianas, que pueden ser estáticas o dinámicas a su vez. Éstas se obtienen a partir del análisis de estabilidad de la corrientes oceánicas, utilizando los Exponentes de Lyapunov en Tiempo Finito (FTLE).

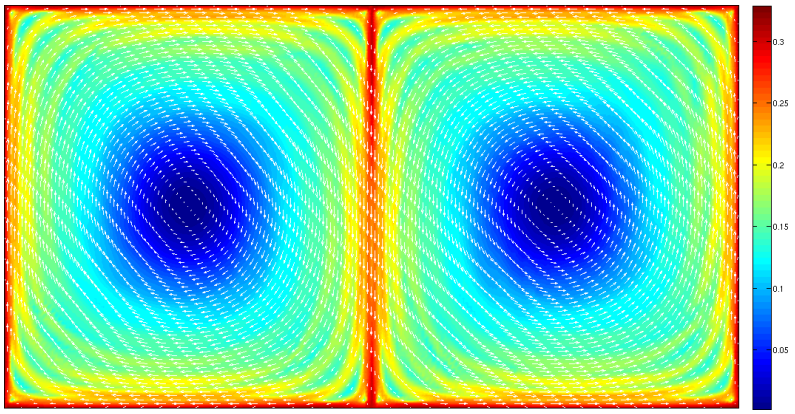


Figura 2.4: Exponentes de Lyapunov en Tiempo Finito (FTLE) computados sobre un campo de corrientes oceánicas sintético que forma un doble giro —superpuesto con flechas blancas. Los valores de FTLE representan la estabilidad de cada región en el campo, tal que las regiones inestables —como los bordes de los giros— aparecen en rojo y las estables en azul —como el centroide de los giros.

La Figura 2.4 muestra el campo de FTLE con las corrientes de un doble giro generado sintéticamente. Claramente observamos como las zonas del borde de los giros son altamente inestables. Esto es debido a que si el vehículo estuviera en estas zonas, un mínimo error en su localización podría significar que acabaría en posiciones muy diferentes pasado un tiempo T, asumiendo el modelo de movimiento puntual de un derivador. Dada la incertidumbre en la localización de los gliders bajo el agua, en la medida de lo posible es interesante evitar estas zonas duran la planificación de rutas.

2.5.2 Giros Mesoescalares

Los giros son un tipo de estructura que se encuentra a distintas escalas espaciales y temporales en el océano. Por ello son de alto interés en los estudios del comportamiento del océano. Para poder caracterizarlos es necesario muestrearlos siguiendo una trayectoria que se focalice en los características claves del giro, para un correcta caracterización. Este es el caso de un patrón de muestreo en forma de molino, donde se realizan transectos en forma de líneas que cruzan el centroide del giro desde un extremo a otro de su borde. Estas líneas se seguirán resolviendo el problema de seguimiento de trayectorias, discutido con anterioridad. Este patrón de muestreo resulta conveniente para estas estructuras, frente a otros patrones comunes en exploración, como el de *lawnmower*.

Capítulo 3

Planificación de Rutas

El objetivo central de esta tesis es proponer algoritmos que permitan la optimización de rutas para gliders oceánicos aprovechando las condiciones de las corrientes marinas en el entorno.

3.1 Revisión

En la bibliografía se han planteado diferentes soluciones al problema de la planificación de trayectorias para vehículos marinos no tripulados, en general, y gliders en particular. Como aproximaciones más significativas podemos encontrar los métodos basados en grafos (A* y variantes), el muestreo probabilístico (RRT) y las técnicas basadas en optimización (bioinspiradas, teoría de sistemas, campos de potencial).

El pilotaje manual de un glider, explicado con anterioridad, se utilizará como elemento base para comparar los algoritmos de planificación de caminos desarrollados en este trabajo. Aquí modelamos el pilotaje manual con una aproximación que fija las direcciones ψ_e del glider directamente hacia el destino $\mathbf{x}_{\text{goal}}$ en cada punto de emersión. A esta aproximación la denominamos DtG (*Direct to Goal*). Desde la localización $\mathbf{x}_i$ de un punto de emersión, se fija la dirección ψ_g igual al rumbo ψ_d directo al destino $\mathbf{x}_{\text{goal}}$. El problema de esta aproximación es que es ciega, es decir, no tiene en cuenta el efecto de las corrientes.

3.2 Algoritmos propuestos

A lo largo de este trabajo de tesis se han desarrollado un conjunto de algoritmos de planificación especialmente adaptados a las características de los gliders. A continuación se incluye una descripción de sus principales características.

3.2.1 Adaptación del A*

Se ha adaptado el algoritmo clásico para operar en base a la simulación de las trayectorias posibles del glider sobre el mapa de corrientes. El A* se utiliza comúnmente para resolver el problema del camino más corto, pero utilizando el modelo de movimiento puntual restringido es posible resolver el problema del camino de mínimo tiempo.

El funcionamiento del algoritmo A* se ilustra en el diagrama de la Figura 3.1. Podemos ver como la dirección ψ_g que se fija para el glider debe ser tal que permita seguir el rumbo ψ_e, tal que el vehículo pueda ir del nodo actual a otro nodo de la rejilla del grafo de búsqueda. Es por eso que se requiere del modelo de movimiento restringido. El problema fundamental de esta aproximación es que en zonas de corrientes fuertes,

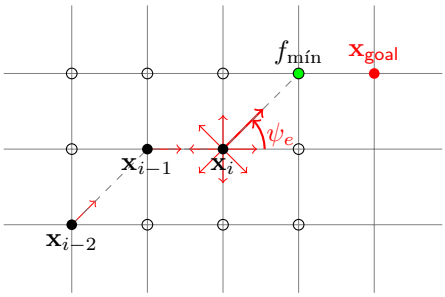


Figura 3.1: Algoritmo A* clásico sobre una rejilla uniforme que define el dominio del espacio de búsqueda Ω. Los ángulos de rumbo ψ_e están restringidos por la rejilla, produciendo trayectorias en línea recta entre nodos y tiempos de emersión no constantes para los problemas de planificación de caminos para gliders.

el cono de accesibilidad se puede estrechar hasta el punto de que no sea posible llegar a ningún nodo vecino desde el nodo actual.

El algoritmo A* emplea un grafo implícito, que se construye evaluando el modelo de movimiento restringido en la función que genera los nodos sucesores —o vecinos— del nodo actual. Como se mencionó anteriormente, en algunos casos el número de vecinos puede ser muy bajo o, incluso nulo, debido al efecto de las corrientes oceánicas.

El algoritmo A* emplea una heurística para acelerar la búsqueda del camino óptimo, siendo así más rápido que el algoritmo de Dijkstra. Para computar la heurística $h(n)$, estimamos la máxima velocidad $U_{c_{\text{máx}}} = \max_{\mathbf{x}\in\mathcal{P}}(\|\mathbf{U}_c(\mathbf{x},t)\|)$ de las corrientes oceánicas a lo largo del camino $\mathcal{P}$ seguido por el glider. Puesto que $\mathcal{P}$ no se conoce de antemano, estimamos $U_{c_{\text{máx}}}$ usando un dominio Ω centrado espacial y temporalmente en la localización actual y tiempo. Dada $U_{c_{\text{máx}}}$ y la velocidad nominal del glider U_g, computamos una subestimación temporal del coste de alcanzar el destino desde al localización actual $\mathbf{x}$, del nodo n, usando la distancia $d = \|\mathbf{x} - \mathbf{x}_{\text{goal}}\|$ al destino,

$$h(n) = \frac{d}{U_g + U_{c_{\text{máx}}}} \tag{3.1}$$

que es una función heurística conservativa pero admisible. Además, mientras usemos la misma estimación de $U_{c_{\text{máx}}}$ para cada nodo n en el grafo, esta heurística también es consistente.

3.2.2 CTS-A*

El patrón de movimiento del glider es recogido con precisión en este algoritmo, de manera que los resultados tienen un mayor plausibilidad. A diferencia del A*, el CTS-A* genera tiempos constantes entre emersiones, al igual que los gliders. Para ello, en la generación de sucesores del grafo implícito de búsqueda, emplea el modelo de movimiento puntual no restringido.

El diagrama de la Figura 3.2 ilustra como ahora se considera un conjunto de direcciones ψ_g y para cada una de ellas se integra el modelo de movimiento no restringido. La posición resultante de la emersión tras el tiempo t_s se lleva a una rejilla sólo con la finalidad de mantener constante el uso de memoria del algoritmo. Así, mantenemos los tiempos entre emersiones constantes y simulamos la verdadera trayectoria del glider realizando los perfiles yo-yo mientras navega sumergido entre emersiones.

En el Algoritmo 3.1 se muestra el procedimiento de generación de sucesores, en el que se considera un muestreo uniforme de direcciones ψ_g —con una determinada discretización— y para cada una se integra el modelo de movimiento no restringido. Este algoritmo también controla el etiquetado de los nodos, que se encarga de guardar las localizaciones de las emersiones en los nodos. Se guardan en base al coste acumulado y una función que puede ser la distancia más cercano al nodo o la propia heurística.

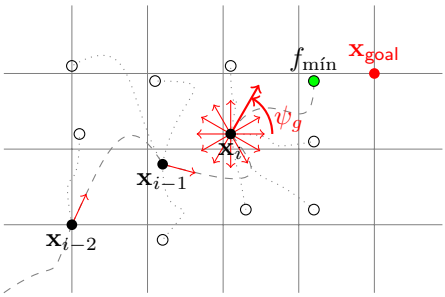


Figura 3.2: Diagrama de funcionamiento del algoritmo Constant-Time Surfacing A*. En cada localización de emersión $\mathbf{x}_i$ diferentes direcciones ψ_g son consideradas, integrando la trayectoria del glider durante el tiempo entre emersiones t_s, o múltiplos $s \cdot t_s$.

Algorithm 3.1 Generación de sucesores en el CTS-A*.

Require: Nodo actual n, el número k de direcciones (*bearings*) a muestrear y el tiempo entre emersiones t_s para integrar la trayectoria. La integración de la trayectoria require el paso de integración dt, la velocidad nominal del glider $\mathbf{v}_g(\psi_g)$ para una dirección ψ_g, y el campo de velocidades de las corrientes oceánicas $\mathbf{v}_c(\mathbf{x})$. Si un nodo vecino m tiene el mismo coste g' que la nueva trayectoria $\mathbf{x}_{\psi_g}$, se usa una función $d(\mathbf{x}, n)$. Si $d(\mathbf{x}, n) = h(n)$, podemos usar $f(n)$ en lugar de $g(n)$, con una sola condición.

Ensure: Lista de sucesors $\mathcal{S}$ y actualización del etiquetado de nodos considerando el coste $g(\mathbf{x})$ y la distancia $d(\mathbf{x}, m)$ de la localización $\mathbf{x}$ emparejada con el nodo m.

Algorithm: successors(n, k, t_s) **return** $\mathcal{S}$

$\mathcal{S} = \varnothing$
for all 1 to k **do**
  $\psi_g \sim [0, 2\pi)$ ▷ muestrear *bearing*
  $s = 0$
  repeat
    $\mathbf{x}_{\psi_g} =$ **trajectory**$(t_s, dt, \mathbf{v}_g(\psi_g), \mathbf{v}_c(\mathbf{x}), \mathbf{x}_n)$
    $m =$ **NN**$(\mathbf{x}_{\psi_g})$ ▷ nodo m más cercano a $\mathbf{x}_{\psi_g}$
    $s = s + 1$
  until $m \neq n$
  $g' = g(n) + s \cdot t_s$
  if $m \notin \mathcal{C} \cup \mathcal{O}$ **then** ▷ m no descubierto hasta ahora
    $m_{\mathbf{x}} = \mathbf{x}_{\psi_g}$
  else
    if $g' < g(m)$ **then** ▷ menor coste
      $m_{\mathbf{x}} = \mathbf{x}_{\psi_g}$
    else if $g' = g(m)$ **then** ▷ igual coste, pero
      if $d(\mathbf{x}_{\psi_g}, m) < d(m_{\mathbf{x}}, m)$ **then** ▷ mejor localización
        $m_{\mathbf{x}} = \mathbf{x}_{\psi_g}$
      end if
    end if
  end if
  $\mathcal{S} = \mathcal{N} \cup \{m\}$ ▷ añadir m a la lista de sucesors $\mathcal{S}$
end for
return $\mathcal{S}$

3.2.3 ABS

Para aliviar el coste computacional del CTS-A*, se emplea un muestreo en base a una función de densidad de probabilidad (PDF) que tiene en cuenta tanto la dirección predominante de la corriente como el rumbo fijado para el glider. Esto mejora el muestreo uniforme de las direcciones ψ_g adoptado en la versión básica del algoritmo CTS-A*.

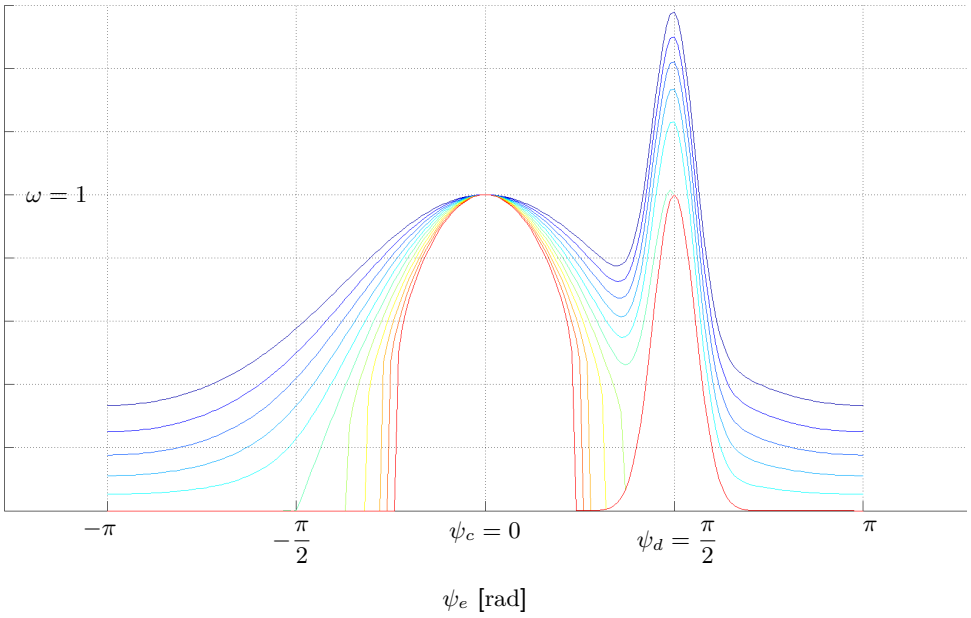


Figura 3.3: Combinación de distribuciones PDF $f_{\psi_c}(\psi_e)$ y una distribución PDF $f_{\psi_d}(\psi_e)$ que considera el rumbo $\psi_d = \frac{\pi}{2}$ directo al destino, modelada con (3.2). $SR \in [50, 150]\,\%$.

En la estrategia de muestreo adaptativo de direcciones (ABS) es posible modelar la PDF considerando la velocidad efectiva U_e y el rumbo ψ_d al destino. Primeramente, U_e se modela con el modelo de movimiento restringido.

Se construye otra PDF $f_{\psi_d}(\psi_e)$ considerando la diferencia entre el rumbo ψ_e y ψ_d. Proponemos la siguientes distribución Normal para modelarla,

$$\mathcal{N}(\psi_e;\ \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(\psi_e - \mu)^2}{2\sigma^2}\right\} \quad \psi_e \in [0, 2\pi] \tag{3.2}$$

con media $\mu = \psi_d$ in rad y varianza σ^2.

Componiendo ambas PDFs, obtenemos la PDF propuesta tal y como se muestra en la Figura 3.3 para varios ratios de velocidad SR. A partir de esta nueva PDF, una vez normalizada se muestrean los rumbos ψ_e, y a partir de ellos se tienen las direcciones ψ_g que se usarán para integrar el modelo de movimiento no restringido tal y como se indica en el algoritmo CTS-A*. Así, podemos emplear un menor número de direcciones, reduciendo el tiempo de cómputo, pero manteniendo o mejorando la calidad de la solución, puesto que exploramos en las direcciones más prometedoras.

3.2.4 Adaptación del RRT

En este algoritmo se generan de forma rápida las trayectorias más probables que puede seguir el glider en una exploración dirigida de su entorno. El algoritmo RRT es aplicable en la planificación de caminos para gliders. Es importante indicar que este algoritmo no busca la solución óptima, sino una solución factible en poco tiempo de cómputo en problemas de gran dimensionalidad.

En la práctica, hemos observado que en el problema de planificación de rutas para gliders, al menos se requieren dos árboles de búsqueda, una lanzado desde la localización inicial $\mathbf{x}_0$ y otro desde el destino $\mathbf{x}_{\text{goal}}$ que irá en sentido contrario —i.e. utilizando los valores de corrientes invertidos. Esta aproximación sólo es válida con campos de corrientes estáticas, ya que de ser dinámicas no sería posible determinar en que instante de tiempo se lanza el árbol desde el destino, pues para ello hay que resolver el problema.

3.2.5 Optimización

Este algoritmo permite afrontar planificaciones complejas a partir de un sistema de inicialización adecuado al problema seguido por un esquema de optimización iterativo.

Para resolver el problema de mínima distancia restante al destino, definimos una función objetivo basada en el punto final $\mathbf{x}_{n-1}$ de integración de la trayectoria del glider para todas las direcciones $\mathcal{B}$ comandadas desde el waypoint inicial $\mathbf{x}_0$,

$$\mathbf{x}_{n-1} = f(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G}) \tag{3.3}$$

donde $\mathcal{C}$ es el mapa de corrientes oceánicas y $\mathcal{G}$ es la configuración del glider —i.e. la velocidad nominal U_g y otros parámetros usados en el modelo de movimiento no restringido. Puesto que queremos minimizar la distancia restante al waypoint destino $\mathbf{x}_{\text{goal}}$, construimos nuestra función objetivo como la distancia restante entre $\mathbf{x}_{n-1}$, dado por (3.3), y $\mathbf{x}_{\text{goal}}$. Es decir, la función objetivo f' es

$$f'(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G}) = \|\mathbf{x}_{\text{goal}} - f(\mathbf{x}_0, \mathcal{B}, \mathcal{C}, \mathcal{G})\| \tag{3.4}$$

donde $\|\cdot\|$ es una distancia métrica, que en nuestro caso es la longitud del arco del *great circle* que pasa por $\mathbf{x}_{n-1}$ y $\mathbf{x}_{\text{goal}}$, ya que navegamos una geometría esférica que aproxima a La Tierra; para misiones cortas, con distancias pequeñas, podemos usar otra aproximación o incluso la distancia Euclídea (véase la Sección A.2).

3.2.5.1 Optimización Iterativa

En el caso del problema del camino de mínima distancia, no se conoce el tiempo que se tardará en llegar al destino, pues es lo que buscamos resolver. Por tanto, tampoco conocemos el número de direcciones $\psi_g \in \mathcal{B}$, que forman parte de la función objetivo. Para solucionar este problema planteamos un método de optimización iterativo.

El principal problema de las técnicas de optimización es que requieren una buena solución inicial para no verse atrapadas en mínimos locales. En el caso de entornos con obstáculos se requiere de un procedimiento de inicialización que encuentra una solución suficientemente buena, para luego ser optimizada.

En general, para entornos libres de obstáculos se puede utilizar la aproximación del DtG como solución inicial. Sin embargo, en presencia de obstáculos proponemos un algoritmo para la fase de inicialización que emplea una versión *gruesa* del algoritmo CTS-A*. Con ella conseguimos evitar obstáculos y así proporcionar una solución inicial al algoritmo de optimización.

3.3 Planificación de Rutas sobre Estructures Oceánicas

A parte de las técnicas comentadas hasta ahora, es posible planificar rutas teniendo en cuenta estructuras oceánicas presentes en el océano, en lugar de todas las corrientes. Para ello podemos aplicar técnicas de

identificación y seguimiento de dichas estructuras, como veremos a continuación.

3.3.1 Análisis de Estabilidad

El análisis de estabilidad del campo vectorial de corrientes oceánicas permite determinar qué zonas son estables y cuáles inestables, utilizando tanto corrientes estáticas como dinámicas. Para ello computamos el campo FTLE con la siguiente expresión,

$$\sigma_{t_0}^{T}(\mathbf{x}) = \frac{1}{|T|} \ln \sqrt{\lambda_{\text{máx}}(\Delta)} \tag{3.5}$$

que representa el FTLE más grande para un tiempo de integración finito T, asociado al punto $\mathbf{x}$ y el instante de tiempo t_0.

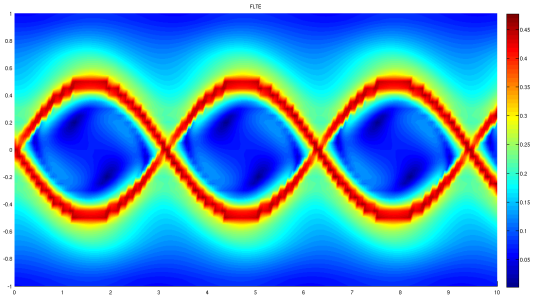


Figura 3.4: Análisis de Estabilidad computando el campo FTLE de un vórtice Stuart. Las regiones inestables se corresponden con LCSs, las cuales aparecen en tonos rojos en los bordes de los giros/vórtices.

El resultado del FTLE puede verse en la Figura 3.4 para un vórtice Stuart, generado sintéticamente. Podemos observar claramente como las zonas del borde de los giros son altamente inestables. Este información no es fácil de apreciar directamente sobre el campo de corrientes, especialmente si es dinámico, como en este caso. De ahí surge la utilidad de computar el campo FTLE. A partir de este campo, es posible segmentar las zonas inestables y así obtener Estructuras Coherentes Lagrangianas (LCSs). Así, es posible planificar rutas evitando estas zonas, puesto que su inestabilidad indica que el siguiente punto de emersión puede variar bastante del esperado, es decir, aumentar su incertidumbre.

3.3.2 Giros Oceánicos

Otro tipo de estructuras muy comunes en el océanos son los giros. Nos encontramos giros de distintas escalas espaciales y temporales. Así, los giros mesoescalares son interesantes para el estudio de múltiples procesos oceánicos.

Se ha realizado un modelado tridimensional de un giro mesoescalar en sectores. Este modelo se puede incorporar a un Algoritmo Genético para realizar el seguimiento de la estructura, tras su identificación. Se ha desarrollado una técnica para dar los primeros pasos en el muestreo de estas estructuras móviles, optimizando la estrategia de muestreo de una determinada zona del giro —v.g. centroide, borde, etc.—, así como su caracterización, lo cual tiene especial utilidad en la asimilación de datos por parte de modelos oceánicos.

Capítulo 4

Resultados Experimentales

4.1 Camino de Mínimo Tiempo

Para el camino de mínimo tiempo hemos probado los algoritmos DtG, RRT, A*, CTS-A* con $M = 18$ muestras, y Optimización Iterativa.

Se han lanzado 20 casos de test que muestran como el DtG y RRT no son capaces de encontrar ninguna solución en algunos casos. Por contra, el A*, CTS-A* y el método de Optimización Iterativa, siempre encuentra solución, siendo la de éste último ligeramente mejor que los métodos basados en grafos.

4.2 Camino de Mínima Distancia Restante al Destino

Para el camino de mínima distancia restante al destino hemos probado los algoritmos DtG, A*, CTS-A* con $M = 18$ muestras, y Optimización; no se prueba el algoritmo RRT porque manejamos corrientes dinámicas.

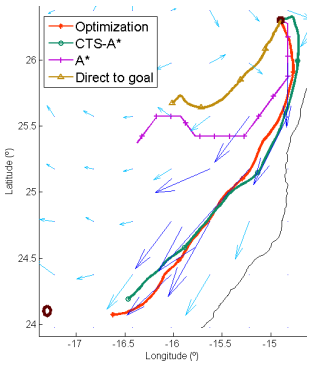


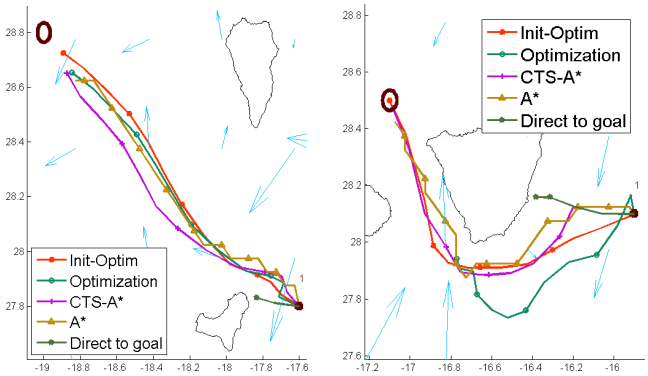
Figura 4.1: Captura de un caso de test para 4 días; la última predicción se repite para rellenar las no disponibles. La figura muestra el camino sobre las corrientes oceánicas $\longrightarrow$ (las corrientes que exceden la velocidad del glider $U_g = 0{,}4\mathrm{m/s}$ están resaltadas $\longrightarrow$) desde el waypoint inicial ● al de destino ○. La distancia desde el waypoint inicial al de destino es $344{,}6\mathrm{km}$. La distancia restante para alcanzar el destino es, para cada algoritmo: Optimización $= 68{,}9\mathrm{km}$; CTS-A* $= 85{,}1\mathrm{km}$; A* $= 169{,}4\mathrm{km}$; DtG $= 217{,}6\mathrm{km}$.

Se han lanzado 65 casos de test, divididos en corrientes débiles y fuertes. La mejora respecto al DtG, que se muestra en el Cuadro 5.2, es claramente mayor con corrientes fuertes. Asimismo, el algoritmo de Optimización es el que deja al vehículo más cerca del destino, después del tiempo de misión T dado.

4.3 Evitación de Obstáculos

Para la evitación de obstáculos se han seleccionado escenarios extraídos del mar de Canarias, donde la presencia de las islas permite definir configuraciones complejas de resolver para la planificación.

Se han analizado los algoritmos tanto para misiones largas con corrientes estáticas, como misiones cortas con corrientes dinámicas. Esto es equivalente a decir que se ha probado la evitación de obstáculos para el problema del camino de mínimo tiempo y el de mínima distancia restante al destino, respectivamente.



(a) Caso I. (b) Caso II.

Figura 4.2: Comparativa del problema de Mínima Distancia Restante al Destino para el problema de Evitación de Obstáculos. Comparativa de 2 casos de test realizados $U_g = 0{,}4\mathrm{m/s}$. La distancia restante al waypoint de destino is (SL si Stop in Land): (a) Optimización-CTS-A* $= 13{,}3\mathrm{km}$, Optimización $= 22{,}1\mathrm{km}$, CTS-A* $= 20{,}6\mathrm{km}$, A* $= 25{,}9\mathrm{km}$, DtG $= 157{,}1\mathrm{km}$ (SL); (b) Optimización-CTS-A* $= 0{,}0\mathrm{km}$, Optimización $= 69{,}7\mathrm{km}$ (SL), CTS-A* $= 3{,}2\mathrm{km}$, A* $= 8{,}7\mathrm{km}$, DtG $= 80{,}0\mathrm{km}$ (SL); Respectivamente, la distancia total desde el waypoint inicial al de destino es: $176{,}5\mathrm{km}$, $125{,}8\mathrm{km}$.

Los resultados se han separado para zonas de costa y *offshore*, comparando cada técnica con la del método Optimización-CTS-A*, que aplica el método de inicialización inteligente utilizando una versión *gruesa* del algoritmo CTS-A*, tal y como se explica en la Sección 4.3.5.2. Observamos como lejos de costa el resultado sin esta fase de inicialización es equivalente a la usada en el método de Optimización, lo cual indica su validez en todos los escenarios. En zonas costeras, esta inicialización mejora cerca de $20\mathrm{km}$ la solución del método de Optimización. Además, se consiguen resultados en torno a $\approx 5\mathrm{km}$ mejores que con las técnicas basadas en grafos. Más lejos aún queda la aproximación DtG.

En la Figura 4.2 se muestran 4 casos de evitación de obstáculos cerca de islas. Podemos observar como en algunos casos de test la mayoría de técnicas quedan atrapadas; nótese que las técnicas basadas en grafos pueden verse atrapadas debido las corrientes en torno a determinadas zonas.

4.4 Seguimiento de Trayectorias

En el caso del seguimiento de trayectorias se ha comprobado la capacidad de las soluciones propuestas para mantener la ruta del glider sobre la ruta deseada. También se ha comprobado la mejora en el rendimiento que se consigue relajando ligeramente las condiciones del seguimiento.

Con la relajación de la condición de seguimiento de la trayectoria obtenemos varias soluciones, que definen una curva de Pareto. La curva de Pareto muestra los dos objetivos contrapuestos uno frente al otro, i.e. el tiempo viajado (eje y) frente al promedio de la distancia al camino deseado (eje x) —que es la línea recta en

este ejemplo. Todos los caminos intentan mantener el vehículo cerca de la línea recta, pero esta restricción se relaja en cada simulación incrementando el tamaño del pasillo alrededor de la línea. Como muestra la figura, mientras más estrecho es el pasillo, mayor es el tiempo viajado. Mientras tanto, al agrandarse el pasillo, nos aproximamos al camino de mínimo tiempo de la optimización sin restricciones.

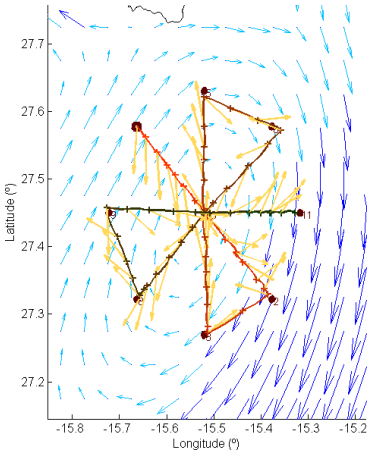


Figura 4.3: Ensamblaje de problema de seguimiento de trayectorias para realizar una misión de muestreo y caracterización de un giro mesoescalar. La trayectoria de molino se descompone en líneas rectas que cruzan el giro múltiples veces desde diferentes direcciones, para las cuales se resuelve el problema de seguimiento de trayectorias usando el método de Optimización Iterativa. Se muestran las direcciones del glider → en cada punto de emersión, junto con las corrientes oceánicas → desde el waypoint inicial ■ al de destino ○.

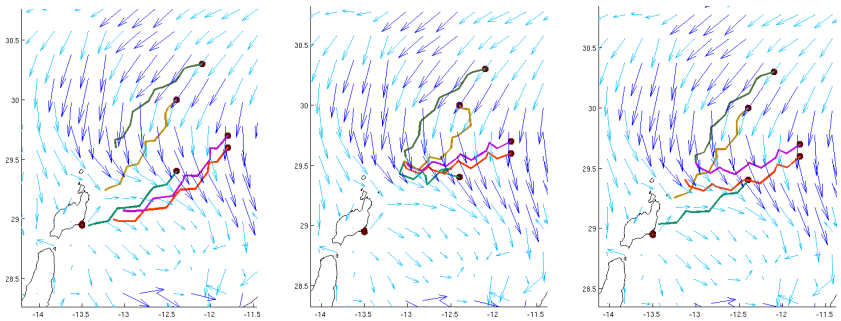
Finalmente, es posible concatenar múltiples problemas de seguimiento de trayectorias, como muestra el patrón en forma de molino de la Figura 4.3. Éste está compuesto por múltiples trayectorias en forma de líneas rectas que deben ser seguidas. Así, con la solución a este problema se pueden realizar aplicaciones de muestreo de giros siguiendo, por ejemplo, este patrón de molino.

4.5 Planificación Multi-glider

Para la planificación multi-glider se han seleccionado escenarios en los que los gliders debían mantener una posición relativa fija con respecto a un vehículo líder. También se han planteado problemas de optimización del tiempo de recuperación de múltiples unidades desplegadas.

Para la recuperación de múltiples gliders tras una misión, planteamos el problema de su recogida eficiente con un barco. Este problema requiere la planificación coordinada de los caminos de todos los gliders de la flota, para así reducir el tiempo de recogida y, por consiguiente, el gasto económico de la mismo, atendiendo al combustible consumido por el barco.

En la Figura 4.4 se muestran varias soluciones para el problema de recogida eficiente utilizando el método de Optimización. Se busca la recogida de mínimo tiempo, a la vez que se intenta resolver el problema de mínima distancia restante al destino para un período de 4 días. Los gliders viajan a $U_g = 1\mathrm{km/h}$, con un tiempo entre emersiones de $t_s = 8\mathrm{h}$. En (a), tenemos las solución con planificación de caminos individual. Por el contrario, las otras dos figuras muestran los resultados usando planificación de caminos multi-glider. En (b), todos los gliders intentan encontrarse en un punto para la recogida. Finalmente, en (c), el camino del barco se incluye en el proceso de optimización. Esta última aproximación es la que da mejores resultados.



(a) Individual. (b) Encontrarse juntos. (c) Camino del barco.

Figura 4.4: Planificación de caminos multi-glider y coordinación para una flota de 5 gliders que tienen que ser recuperados del mar por un barco en el mínimo tiempo. El barco tiene que viajar: (a) $192{,}0\mathrm{km}$; (b) $167{,}5\mathrm{km}$; (c) $165{,}4\mathrm{km}$, respectivamente.

4.6 Seguimiento y Muestreo de Estructuras Oceánicas Móviles

Finalmente, hemos realizado algunos simulaciones preliminares de la técnica comentada en la Sección 4.4.2, para realizar el seguimiento y muestreo de estructuras oceánicas móviles. Concretamente, hemos modelado giros mesoescalares e intentamos seguirlos al mismo tiempo que se muestran determinadas zonas de interés de los mismos.

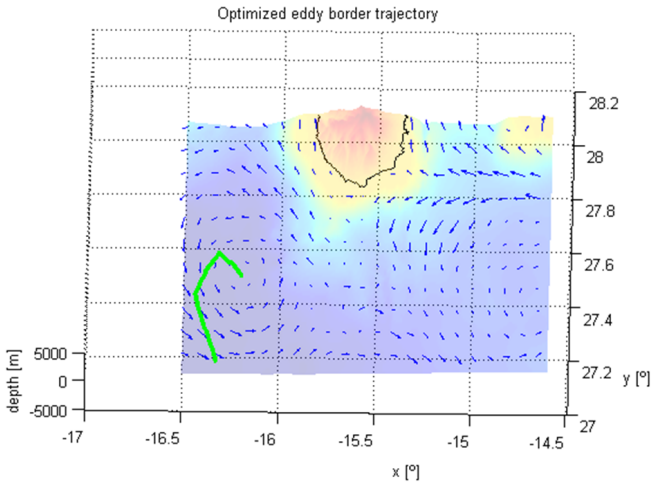


Figura 4.5: Ejemplo en el que el contorno de un giro mesoescalar se sigue usando nuestra aproximación con un Algoritmo Genético y el modelo discreto del giro.

La Figura 4.5 muestra una captura de las corrientes oceánicas y la trayectoria optimizada al final de la misión; nótese que el giro mueve mientras se muestrea. En este ejemplo se intenta muestrear de manera óptima un región situada en el borde del giro.

Capítulo 5

Conclusiones

En esta tesis se ha abordado el estudio del problema de la planificación automática de rutas para gliders a partir de modelos de predicción numérica. Se ha demostrado que es posible mejorar las capacidades operativas de estos vehículos introduciendo herramientas de asistencia al pilotaje.

5.1 Aportaciones Originales

Como principales aportaciones conviene destacar el conjunto de algoritmos de planificación desarrollados. Las herramientas complementarias generadas constituyen asimismo aportaciones válidas que facilitan el trabajo en este ámbito de manera importante.

Además de lo anterior, en la tesis se han realizado diferentes análisis cuyas conclusiones pueden considerarse contribuciones en sí mismas, como es el caso de la revisión bibliográfica comentada, los estudios sobre las alternativas de modelado o el análisis de aplicabilidad de las técnicas de optimización.

De hecho, tras el análisis bibliográfico hemos visto que el problema del camino de mínimo tiempo no es común en la robótica de vehículos móviles terrestres, donde el problema de planificación de caminos más común es el del camino más corto. La principal diferencia radica en el hecho de que en el camino de mínima tiempo no sólo afecta la distancia entre origen y destino, sino también la velocidad del vehículo, que en el caso de la planificación de rutas para gliders se ve afectada por las corrientes U_c.

El algoritmo A* se usa ampliamente para el problema del camino más corto. Varios autores lo han adaptado al problema del camino de mínimo tiempo para gliders (Garau et al., 2005, 2009; Soulignac, 2010). Básicamente, se incorpora el modelo de movimiento restringido, para encontrar el camino óptimo dentro de la rejilla del grafo de búsqueda del algoritmo. La discretización de la rejilla obliga a usar el modelo de movimiento restringido, el cual en presencia de corrientes puede llevar a caminos donde no se puede expandir ningún nodo. Además, el tiempo entre un nodo y su vecino no es constante ni igual al tiempo entre emersiones t_s, no reflejando así la forma en que navega un glider. Igualmente, tampoco se integra la trayectoria del perfil yo-yo. El análisis de estas limitaciones es una primera contribución de este trabajo de tesis.

El estudio de la literatura de planificación de caminos para gliders, así como la de problemas de planificación de caminos con costes variables y asimétricos, muestra pocas contribuciones en este área. Los problemas más similares —i.e. que manejan costes asimétricos variables— son: el Problema del Viajante Canadiense (CTP), rutas de tráficos dinámicas, y mapas de coste o planificación con costes.

En teoría de grafos, el CTP es una generalización del problema del camino más corte a grafos parcialmente observables, i.e. el grafo se obtiene mientras es explorado (Papadimitriou, 1991). El concepto de observabilidad parcial cubre los costes variables del problema del camino de mínimo tiempo para gliders. De manera similar, en los problemas de rutas de tráfico dinámicas, la densidad de vehículos produce costes variables para viajar dentro de la red de rutas. Esto es equivalente a los mapas de coste, que habitualmente se asocian a la planificación

de caminos basada en relieve. De hecho, podríamos construir un mapa de coste a partir del campo vectorial de corrientes oceánicas, pero esto sería equivalente a aplicar el modelo de movimiento restringido usado por el A*. Aunque, en nuestra humilde opinión, esta es la categoría en que se ubican los problemas de planificación de caminos, en realidad no representan la navegación del glider ni ciertas características de su funcionamiento con facilidad. Más concretamente, estos problemas no incluyen la restricción de emersiones de tiempo constante.

Antes que un análisis teórico del problema y los algoritmos de planificación de caminos, este trabajo adopta un enfoque pragmático con el estudio de métodos aplicables a la planificación de rutas para gliders. Hemos adaptado métodos clásicos basados en grafos como el A*, y técnicas de muestreo probabilístico como el RRT, siguiendo trabajos previos de Rao and Williams (2009). Debido a las limitaciones del A* y el RRT, hemos desarrollado varias técnicas novedosas para eliviar sus defectos. Aplicamos el modelo de movimiento no restringido y reflejamos la restricción de emersiones separadas un tiempo constante. Hacemos estos tanto para el algoritmo CTS-A* como para las diferentes variantes de los métodos de optimización descritos en la tesis.

Hemos realizados varios experimentos para evaluar la validez, optimalidad y tiempo de cómputo. Para casi todos los casos de test, los mejores caminos son encontrados con los métodos de optimización, que además es el más rápido. El CTS-A* también encuentra caminos muy cerca del óptimo, aunque con algo más de tiempo de cómputo. Con la estrategia de muestra ABS se reduce su tiempo de cómputo, puesto que permite reducir el número de muestras para las direcciones que se integran desde cada nodo. Además, un versión *gruesa* del CTS-A* se usa en la fase de inicialización de los métodos de optimización para evitar obstáculos en entornos costeros.

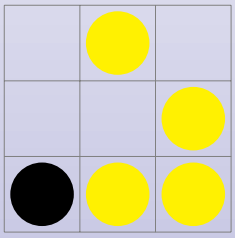
Los resultados experimentales también muestran que la técnica de optimización es fácil de adaptar a un buen número de problemas de planificación de caminos para gliders. Esto incluye el seguimiento de trayectorias, las planificación multi-glider y el seguimiento de estructuras oceánicas móviles. Sin embargo, las ventajas de nuestro esquema de optimización residen en las particularidades de los gliders. Por ello, tales beneficios se perderían si se aplica a la planificación de rutas con otro tipo de vehículo. Es en las emersiones separadas un tiempo constante donde encontramos la razón de esta falta de generalización. La optimización se realiza en el espacio de direcciones, en lugar de en el espacio de localizaciones. En consecuencia, gracias los tramos entre emersión de $t_s \approx 8\mathrm{h}$, la dimensionalidad del problema de optimización se reduce drásticamente, lo cual ayuda a obtener tiempos de cómputo bajos. Claramente, si usásemos vehículos con una tasa de control —i.e. con un tiempo t_s menos—, el número de direcciones $|\mathcal{B}|$ aumentaría, perdiendo esta ventaja.

En el caso de aproximaciones como los métodos basados en grafos, éstos tienen que manejar la gran dimensionalidad del problema, definido en 4 dimensiones en el caso más general —i.e. 3 dimensiones espaciales y tiempo. Además, varias direcciones ψ_g deben considerarse al generar los sucesores cuando se expande un nuevo nodo en la rejilla de búsqueda. Como resultado, el tiempo de cómputo es grande en estas aproximaciones.

Hemos usado modelos de movimiento puntuales en nuestros algoritmos de planificación de caminos, puesto que resulta una elección razonable, dada la incertidumbre de los modelos oceánicos y la localización de los gliders. Además, nos consideramos cambios de cabeceo θ o límites de profundidad $[z_{\mathrm{mín}}, z_{\mathrm{máx}}]$, puesto que generalmente no se modifican en la práctica. En algunos casos simulamos la deriva del glider mientras espera en superficie, pero normalmente sólo cuando el resultado se usa posteriormente para asistir en el pilotaje de gliders en misiones reales. En cualquier caso, también hemos evaluado un modelo cinemático de equilibrio de fuerzas y comparado sus resultados con los puntos de emersión del glider RU27 en la misión trans-Atlántica. Como ya se ha mencionado, la precisión de estos modelos no se requiere para la planificación de rutas con gliders, aumentarían el tiempo de cómputo del planificador sin ningún beneficio tangible, y requieren un mayor número de parámetros —muchos de ellos difíciles de cuantificar.

Finalmente, hemos avanzado en los primeros pasos en el muestreo de estructuras oceánicas móviles. En particular, hemos desarrollado una técnica para seguir giros mesoescalares y mejorar la estrategia de muestreo (véase la Sección 4.4.2). Este problema tiene enorme interés en la asimilación de datos, y su uso para mejorar el modelado de estas estructuras en los modelos oceánicos.

Contributions to Underwater Glider Path Planning



Path or Motion Planning is a Robotics discipline that deals with the search of feasible or optimal paths. For most vehicles and environments, it is not a trivial problem and therefore we find a great diversity of algorithms to solve it, not only in Robotics and Artificial Intelligence, but also as part of the Optimization literature, with Numerical Methods and Bio-inspired Algorithms, like Genetic Algorithms and Ant Colony Optimization.

The particular case of variable cost scenarios is considerably difficult to address because the environment where the vehicle moves varies over time. The present thesis work studies this problem and proposes a number of practical solutions for Underwater Robotics applications. The thesis focuses on Autonomous Underwater Glider (AUG) Path Planning, considering its motion model and the effect of ocean currents. Gliders are a type of Autonomous Underwater Vehicle (AUV) that moves with a relatively slow nominal speed when compared with ocean currents. For this reason, they conform an interesting research field from the perspective of path planning. Ocean current maps provide the snapshot of this physical parameter at different time instants, which along with other physical parameters, are computed with Ocean Models. In those zones close to the coastline they exhibit a high temporal variability, so we have to perform path planning in a variable cost environment, where the cost function is also asymmetric.

This work attacks the problem using different approaches. Starting with classical path planning algorithms adapted to the problem, like A*, up to optimization techniques, Genetic Algorithms, Probabilistic Sampling methods, among others. The thesis also provides a number of proposals to solve this particular problem. For instance, some variants of A* combined with probabilistic sampling techniques and several proposals of optimization algorithms within the configuration space of the feasible bearings the vehicle can follow from each surfacing point, are described and analyzed. A thorough comparison of these techniques is covered by this document, with the intention of assessing which ones are more suitable for the given problem, as well as analyzing their computational cost. Finally, the work explores sampling strategies for movable ocean structures.

Enrique Fernández Perdomo
Antonio Carlos Domínguez Brito ❖ Daniel Hernández Sosa ❖ José Isern González

ULPGC SIANI