Anastasios M. Lekkas

Guidance and Path-Planning Systems for Autonomous Vehicles

NTNU
Innovation and Creativity

*To my uncle, Vassilis Kalliris, for instilling in me
the virtues of curiosity and critical thinking.*

Summary

This thesis is concerned with two interconnected and very important problems regarding the autonomy of vehicles, namely, path planning and guidance. By adopting a modular approach, path planning and guidance can be viewed as two modules which belong to a wider context consisting of four modules, the other two being navigation and control. All four modules interact with each other and none is completely independent. Path planning deals with *what* we want to achieve (by defining spatial and temporal constraints), and guidance dictates *how* we should act in order to achieve it (by generating appropriate reference trajectories to be fed to the corresponding controllers). Therefore it is important to develop: a) path design methodologies, which will generate feasible and safe paths with several desired properties, and b) guidance laws capable of generating reference trajectories which will lead the vehicle on the desired path, even when unknown disturbances (such as ocean currents and wind forces) affect the vehicle's motion.

Four contributions pertaining to the path-planning problem are included in this thesis. The most important is the use of Fermat's spiral (FS) as an alternative to both Dubins paths and clothoids. We show that paths consisting of straight lines and FS arcs are curvature-continuous, computationally inexpensive and can be used for path tracking by changing the parametrization. The second contribution is the development of a number of path-evaluation criteria which aim at providing an onboard computer with sufficient information for selecting the right path for a given application. The methodology is still at its infancy but several improvements, which could result in fast progress, are discussed. The third contribution is the use of a monotone cubic Hermite spline for path-planning purposes. The main advantage is that the method generates very practical paths which do not include wiggles and zig zags between two successive waypoints. Moreover, the method provides the user with better shape control, a property which can turn out to be very useful in real-time collision-avoidance applications. The fourth contribution pertains to a collision-avoidance strategy combining the Voronoi diagrams (VD) method and FS-based path generation. An intuitive and efficient procedure is developed for obtaining smooth paths which keep the vehicle at a safe distance from all obstacles on the map and at the same time avoid unnecessary heading changes.

The thesis also presents a number of guidance-related contributions, each of varying degree of importance and difficulty. The first one is the modification of the line-of-sight (LOS) guidance by introducing a time-varying equation for the lookahead distance Δ. This aims at obtaining a more flexible behavior regarding the steering of the vehicle because for very small Δ the vehicle approaches

the target path at a direction almost normal to the path, whereas for very large Δ it takes a longer time for the vehicle to converge to the path. The effect of the time-varying Δ equation from a stability viewpoint is investigated. The second (and minor) contribution is the consideration of a 5-DOF vehicle kinematic model (common for torpedo-shaped underwater vehicles which do not control the roll angle) and the influence of the coupling between the horizontal and vertical planes on the expression for the sideslip angle. This led to the third contribution, which is a transformation of the LOS guidance in quaternion form for both the uncoupled and the coupled cases. The transformation is based on exploiting very simple trigonometric properties and the geometry of the LOS guidance. The fourth contribution is an integral LOS guidance law capable of eliminating the errors induced by constant external disturbances. The method is formulated using absolute velocity-based vehicle kinematics and simple Lyapunov-based analysis. The fifth contribution moves a few steps further and presents two adaptive integral LOS guidance laws which compensate for the errors induce by ocean currents. These methods are based on the vehicle kinematics in relative-velocity form. This is a very useful result for underwater vehicles, where absolute velocity measurements might not be available. The effect of the current on the direction normal to the direction of motion (that is, the force inducing the cross-track error) is estimated, and stability results for curved paths are also given. The sixth contribution is the development of a guidance technique where, in addition to the LOS guidance for minimizing the cross-track error, surge velocity commands are generated as well in order to minimize the along-track error, hence satisfying constraints related to the path-tracking (or trajectory-tracking) motion control scenario. Finally, the path-tracking solution is combined with the indirect adaptive integral LOS so as to achieve path tracking under the influence of ocean currents, which also results in estimating all the parameters of the current (that is, current velocity and orientation w.r.t. the inertial frame).

In all cases, particular emphasis was placed on finding solutions that are simple and, at the same time, efficient.

Contents

List of figures

List of tables

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of philosophiae doctor (PhD) at the Norwegian University of Science and Technology (NTNU).

The presented work in this thesis has been carried out at the Department of Engineering Cybernetics and the Centre for Ships and Ocean Structures (CeSOS) at NTNU, in the period from August 2009 to October 2013. My supervisor has been Professor Thor I. Fossen of the Department of Engineering Cybernetics, NTNU. This work was supported by CeSOS through the Research Council of Norway and its Centres of Excellence funding scheme.

Acknowledgments

I would like to thank Mauro Candeloro, both for his commitment and punctuality during our collaboration on the Voronoi diagram-based path-planning method and, last but not least, for being such an exciting and uplifting bandmate.

In addition to the aforementioned co-workers and friends, I have been surrounded by many great colleagues and friends who contributed to creating an excellent environment. First and foremost, I am very happy to have had the best officemate ever, Dominik Breu. His positive, optimistic and active personality has not only been the most fitting counteraction to my often negative state of mind, but also a major source of fun. I will always remember our hilarious nights out on the town, and the equally unconventional and amusing travels in France. I have associated my life in Trondheim with Dominik and I am glad that we both will continue to live here, at least for a few years.

I am grateful to Orestis Gorgas for being such a great friend and always there when I needed him. Despite being away from home, his consistent, reliable and loyal personality made me feel that a member of my family was always here with me. Daniele Borri is thanked for being one of the most helpful and entertaining friends ever. Our working environment became so much jollier with his distinct jokes and laughter echoing around the Department of Marine Technology. In addition to Mauro, my warmest regards go to the other bandmates, Daniel de Almeida Fernandes (also for the unforgetable trip to Japan), Geir Johnsen and Matteo Di Capua - the band rehearsals in the container have been a wonderful experience. Many thanks to Erin Bachynski for motivating me to participate in two half marathons and the memorable trip to Tromsø. The gaming nights organized by Christian Holden have been a lot of fun, and the same is true for the interesting discussions (at least the few times he allowed a dialogue to take place!) about his unconventional and detailed views on almost everything. I enjoyed a lot the conversations with Morten Pedersen and appreciated his deep and modest reflections on life, plus the nice trip to Italy. Moreover, it was really nice to have around me people like Jabus De Vaal, Amir Nejad, Drazen Polic and Constantine Michailides, they are acknowledged for a number of enjoyable and relaxing discussions during breaks from work.

When I moved to Norway, there was a number of very old and good friends who I was saddened to leave behind, although most of them had to move somewhere else eventually. Fortunately, after almost five years I feel proud to realize that we managed to keep the distance between us short, by meeting and communicating as frequently as possible. They have contributed significantly to my wellbeing, therefore, many thanks go to Nikos Nezeritis for the innumerable redemptive conversations and the customized guitar lessons, Aris Sfakianakis for the wonderful holidays in Naxos, Panos Siapkaras for the memorable trip to California, Dimitris Kakogiannis for the PhD support hotline and his hospitality in Brussels, Dimitris Sarris and Mike Kalantzis.

Whenever I look back in time I realize that only a few things have remained unchanged and consistent, and this is indisputably true when it comes to the unconditional love and support from my family. For my parents, Stavroula and Michael, me and my sister's education and welfare has been their top priority. Through their countless sacrifices they have always acted as role models for us, without exceptions, and for that reason I can never thank them enough. This includes also my

Part I

Introduction

Chapter 1

Motivation and Contributions

1.1 Path Planning and Guidance: Two Interrelated Problems

1.1.1 Introduction

The compound word "autonomous" is of Ancient Greek origin and composed of the roots $\alpha\upsilon\tau o$- (meaning "self") and $\nu o\mu o\varsigma$ (meaning "law"). In this regard, an *autonomous* vehicle is one that possesses self-governing characteristics which, ideally, allow it to perform pre-specified tasks without human intervention. These characteristics are associated with the vehicle's (or, more generally, system's) available information regarding its position and surroundings, and also the vehicle's ability to use its actuators so as to accomplish a mission. In the context of this thesis, the following four terms summarize the aforementioned properties: a) path planning, b) guidance, c) navigation, and d) control.

Path planning (or Module 1) refers to the system responsible for designing paths to be assigned to the vehicle in order to accomplish its mission. These paths must satisfy several desired properties related to both the vehicle's constraints and the morphology of the environment in which the vehicle navigates. In other words, this module must ensure that the generated path takes into account the dynamic constraints of the vehicle (such as maximum curvature and velocity) while keeping the vehicle at a safe distance from obstacles at all times. Naturally, this process includes two main steps:

1. The determination of a set of points on the map, namely, the waypoints.

2. The generation of a path based on the waypoints.

In practice, both steps should be implemented with the problem constraints in mind. In the robotics literature, the term "motion planning" is used, see for instance [63, 132, 150, 151]. However, motion planning often includes both the design of a suitable path and the actions that should be taken by the robot in order to accomplish the mission. For that reason, we will use the terms "path planning" for the two steps above.

3

Regarding the guidance, navigation and control (GNC) modules, we adopt the following definitions from [88, Ch. 9.2]:

Guidance (or Module 2) is the action or the system that continuously computes the reference (desired) position, velocity and acceleration of a marine craft to be used by the motion control system. These data are usually provided to the human operator and the navigation system. The basic components of a guidance system are motion sensors, external data such as weather data (wind speed and direction, wave height and slope, current speed and direction) and a computer. The computer collects and processes the information, and then feeds the results to the motion control system. In many cases, advanced optimization techniques are used to compute the optimal trajectory or path for the marine craft to follow. This might include sophisticated features such as fuel optimization, minimum time navigation, weather routing, collision avoidance, formation control and synchronization.

Navigation (or Module 3) is the science of directing a craft by determining its position/attitude, course and distance traveled. In some cases velocity and acceleration are determined as well. This is usually done by using a global navigation satellite system (GNSS) combined with motion sensors such as accelerometers and gyros. The most advanced navigation system for marine applications is the inertial navigation system (INS). Navigation is derived from the Latin navis, "ship", and agere, "to drive". It originally denoted the art of ship driving, including steering and setting the sails. The skill is even more ancient than the word itself, and it has evolved over the course of many centuries into a technological science that encompasses the planning and execution of safe, timely and economical operation of ships, underwater vehicles, aircraft and spacecraft.

Control (or Module 4) or more specifically motion control, is the action of determining the necessary control forces and moments to be provided by the craft in order to satisfy a certain control objective. The desired control objective is usually seen in conjunction with the guidance system. Examples of control objectives are minimum energy, setpoint regulation, trajectory-tracking, path-following and maneuvering control. Constructing the control algorithm involves the design of feedback and feedforward control laws. The outputs from the navigation system, position, velocity and acceleration are used for feedback control while feedforward control is implemented using signals available in the guidance system and other external sensors.

According to the definitions above, we may say that the four modules act repeatedly in the following order (see also Fig. 1.1):

- Module 1: The path-planning algorithm generates a path which (if followed without deviations) is guaranteed to be safe and feasible. In the case where the mission involves temporal assignments, this module specifies where on the path the vehicle should be at any time instant.
- Module 3: The navigation system uses the vehicle's sensors in order to determine the vehicle's position, velocity, and attitude.
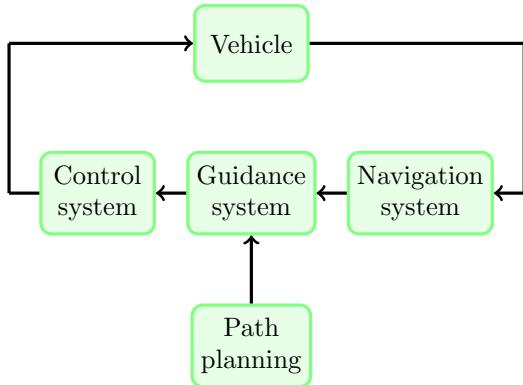
Figure 1.1: Interaction among the 4 main modules

- Module 2: Depending on where the vehicle should be (Module 1) and where it actually is (Module 3), the guidance system determines the reference trajectories to be fed to the control system (Module 4) in order to minimize the error.

- Module 4: Based on the reference trajectories generated by Module 2, the control system calculates the necessary forces that each one of the actuators must produce.

All four modules mentioned above are interconnected and, consequently, it can be very difficult to infer upon the stability of the total system. For that reason, it is often convenient to study the stability of the overall system by viewing it as a cascade system, where the output of one module is the input to another. Therefore, from a stability point of view, we may say that the three GNC modules interact in the following way (see also Fig. 1.2):

- Modules 3 & 2: This pertains to the requirement that the convergence rate of the estimator (that is, the navigation system) must be such that it will allow the guidance system to converge. If the navigation system takes a long time to give good estimates, then the guidance system will have erroneous information about the position of the vehicle and will generate inappropriate reference trajectories.

- Modules 4 & 2: Apparently, the vehicle control system takes time to converge to the desired time-varying values dictated by the guidance system, due to the vehicle's dynamics. If the controller delays are too long, then the vehicle will not reach the goals set by the guidance system, which will be pointlessly generating new trajectories in order to correct the resulting errors.

In this thesis, we are concerned almost exclusively with the first and second modules, namely, path planning and guidance. From an autonomy point of view, these two modules are of great interest because path planning is related to *what* we want to achieve (by defining spatial and temporal constraints), and guidance
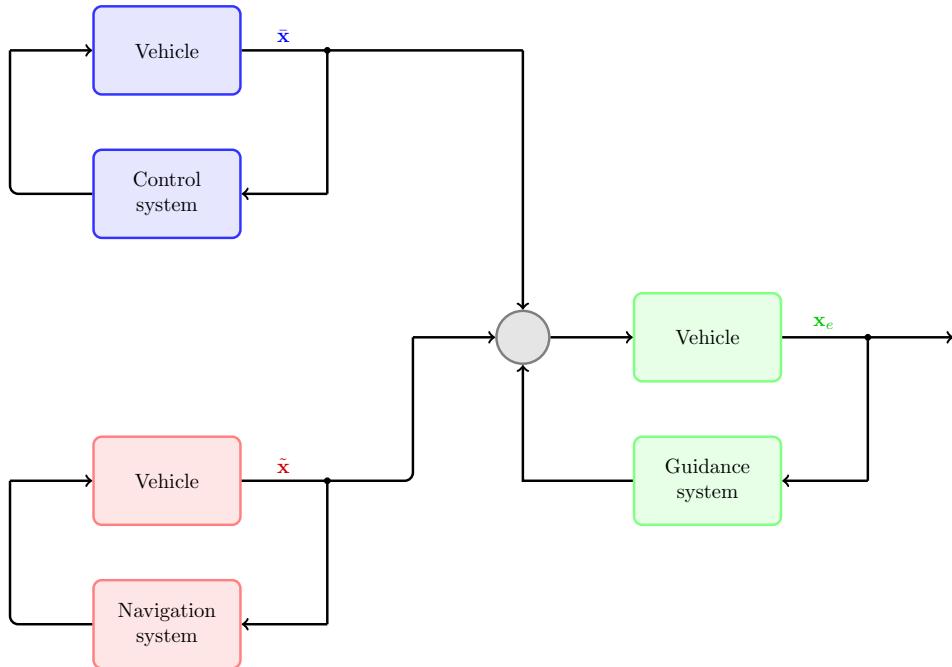
Figure 1.2: Cascaded structure among the three GNC modules. The corresponding output error vectors are denoted by $\mathbf{x_e}$, $\tilde{\mathbf{x}}$, $\bar{\mathbf{x}}$.

dictates *how* we should act in order to achieve it (by generating appropriate reference trajectories). On several occasions though, we will also deal with the stability of the cascaded system formed by the guidance system and the control system. Sections 1.1.2 and 1.1.3 give a more detailed presentation of the two modules and the context within which they are considered in this work.

1.1.2   Path Planning

As the name indicates, path planning pertains to the procedure of determining which route to take when moving from one location to another. Extensive research has been carried out on the topic for many years and the related literature is vast and almost impossible to summarize in a few pages. The majority of important contributions comes from the robotics community because the complexity of the motion of robotic manipulators with many degrees of freedom (DOFs) and the variety of sensors that can be mounted on robotic systems have led to the appearance of many interesting problems, which attracted the attention of many researchers. In addition, since their advent, robots have been expected to reach a point of development where they would be able to imitate the human behavior in a realistic way. This "dream" has been depicted in countless novels, movies and other forms of art, and remains a undepletable source of inspiration because its realization would give

humans the chance to get rid of uncreative and time-consuming tasks and also offer them a better understanding of themselves. Despite the impressive technological advancements that have taken place during the last 20 years, a large number problems, which at a first glance might appear simple, remain very difficult to tackle. One of the main reasons is the uncertainty which is prevalent in the real world. Due to the complicated nature of the problem, the field has undergone through several stages. The early approach was to assume that the robot model was perfectly known and no uncertainty was present, this was the so-called *model-based paradigm* [54, 150, 195], although randomization was used for solving hard motion planning problems [133]. Moreover, the *potential fields* method gave promising results in unknown environments, as long as the obstacles could be sensed, but suffered from singularities [135, 140]. Later on, the idea of internal model was rejected by several researchers who deemed that only sensing was important, an approach which gave some very impressive and promising results but was confined in simple tasks [44, 45]. A more modern approach is *probabilistic robotics*, which has its origins in the Kalman Filter and is placed between the model-based and sensing-based methodologies. The reader is referred to [222] for a more detailed overview of the field and a thorough treatment of modern probabilistic methods.

All the methodologies mentioned above are of unquestionable value and provide the general framework within which every motion-planning problem can be studied. However, every problem has its own peculiarities and, as a consequence, dealing with path-planning applications for vehicles such as AUVs, UAVs etc, entails the involvement of additional factors which have to be taken into account. The most important ones are the dynamic constraints of the vehicles, the environmental forces such as wind, waves and ocean currents, and the limited computational power of small unmanned vehicles, to name a few. It is therefore very important to construct paths which are safe and feasible, smooth to a certain degree, computationally inexpensive, practical, and so on.

In many applications the first step is to introduce a given order of fixed points in space, namely the waypoints, and define the desired path as the sum of the successive straight lines that connect these waypoints. Due to physical constraints though, it is not possible to achieve a smooth transition between two straight lines because, in the general case, such a path has a discontinuous first derivative (and thus velocity function) at the locations of the waypoints. This problem can be avoided by considering a straight-line path between waypoints, as before, but with turning now being achieved by inscribing a circle between two lines to form a curved path [88, Ch. 10]. Note that in this case the vehicle will not pass through the waypoints and this might be an undesired effect, especially if the exact location of the waypoints has been chosen so as, for example, to avoid obstacles. It is therefore reasonable to assume that if the succession between a straight line and a circular arc (or vice versa) occurs at the waypoint location, both the aforementioned requirements (velocity continuity and passing through the waypoints) could be satisfied. Paths consisting of straight lines and circular arcs have been studied extensively in the past. According to Dubins's well-known result for a particle with unity speed, the shortest possible path that meets a maximum curvature bound between a starting position with predefined orientation (starting pose) and a finishing position with predefined orientation (finishing pose) consists of at most

three pieces, each of which is either a straight line or an arc of a circle of radius $R > 0$ [75]. Dubins's work dealt with forward motion of car-like vehicles only, but his result was later extended by Reeds and Shepp in order to take into account backward motion as well, thus allowing to include cusps along the path [194]. Note that, depending on the application, it is not always possible to find a Dubins path, as it was shown in [219]. Extensions of Dubins paths for satisfying curvature constraints in 3-D space have also been developed. Such a methodology was presented in [8] where a nonlinear controller resembling the LOS guidance law was employed for tracking the 3-D Dubins path.

The concatenation of straight and circular segments, however, leads to difficult transition maneuvers between these segments. This problem stems from the fact that a straight line has a curvature $\kappa = 0$, whereas a circle arc has a curvature $\kappa = 1/R$. Hence, there will be a jump in the curvature from 0 to $1/R$ when moving from the straight line to the circle arc. As a result, a sudden change in lateral acceleration will occur and this will lead to deviations from the desired path. This problem was tackled in [28], where continuous curvature paths with a upper-bounded derivative were designed and [94] proceeded further by considering an upper-bounded curvature, forward and backward motions and collision avoidance. The latter approach computed paths that consisted of straight segments, circular arcs and clothoid arcs. These techniques assume car-like vehicles whose models simpler than those used to describe the motion of marine vessels or airplanes, due to the lack of sway motion.

Paths of continuous curvature can also be produced by using other methods. Two of the most popular approaches are the clothoid arcs and the Pythagorean hodographs. The former pertains to paths composed by segments that have equal curvature at their boundaries. Paths based on clothoids consist of straight and arc segments, just like the Dubins path, but the arc segment is now computed with the help of Fresnel integrals instead of being a circular arc. The outcome is a curve with a linearly varying curvature over the path length. The latter approach results in a single well-defined path which satisfies the necessary curvature continuity constraint [82]. Moreover, Pythagorean hodographs employ a polynomial of the fifth degree in order to produce a closed-form solution that gives a flexible path with tangent (velocity) continuity. Further work has been done in [46] so as to guarantee that the curvature constraints are satisfied as well. The main drawback of these approaches is the increased computational cost which can be a heavy burden when implementing them in unmanned vehicles applications, for instance. For a more detailed treatment of the design of Dubins paths, clothoid arcs and Pythagorean hodographs, the reader is referred to [224, Ch. 2-3]. Following the same line of reasoning as the clothoid, the feasibility of constructing paths consisting of straight lines and Fermat's spiral arc segments was studied in [68]. It was shown that Fermat's spiral can be used in order to generate G^2 (curvature-continuous) paths with a very low computational cost compared to clothoids. The approach was further extended in [160] where an alternative parametrization of Fermat's spiral was proposed, hence leading to C^2 paths suitable for path tracking.

Spline interpolation techniques have also been studied extensively in the path-planning literature and, as it is expected, depending on the method chosen the resulting path can have different properties. Bezier curves, for instance, and their

generalizations (B-splines) can give paths of continuous curvature but without passing through all the data points used to define it [128]. Natural splines, on the other hand, pass through the waypoints and also produce curvature continuous paths, but do not possess local control (this refers to the case where relocating one waypoint induces changes throughout a larger part of the path) and the resulting paths are not very practical [88, Ch. 10]. An interesting alternative is the Cubic Hermite Spline Interpolation (CHSI) which passes through all the waypoints and it is possible to assign the derivative values at the control points and also obtain local control over the path.

It should mentioned that path planning is a problem that has been studied extensively in the literature from several different angles and several other approaches have been developed in order to obtain more sophisticated trajectories. Such is the case in [34] where the authors combined the methodologies presented in [9, 95] and developed a path planning strategy which smooths an optimal sequence of waypoints in 3-D space using a highly accurate nonlinear vehicle model which is based on motion primitives. Moreover, a core paths graph (CPG) which aimed at generating optimal flight trajectories that satisfy mission constraints resulting from no-fly zones or obstacles was presented in [169]. In [53], the Fermat spiral path planning strategy developed in [68, 160] was used in combination with the Voronoi diagrams method and resulted in curvature continuous paths while achieving obstacle avoidance.

1.1.3 Guidance

In addition to the path-planning problem, it is also of great importance to devise a guidance algorithm which will provide the vehicle controller with reference trajectories suitable for keeping the vehicle on the path, or leading it toward the path if the position error is nonzero. Proving the stability of a guidance algorithm is the first step to guarantee convergence to the path (at least from a theoretical standpoint), because it implies that the generated reference trajectories are indeed appropriate for the task, if followed accurately and instantly. As mentioned in Section 1.1.1 though, the navigation and control systems will not allow the guidance system to accomplish its task instantly by introducing some errors due to the time it takes them to converge. Apparently, this is even more evident when the reference trajectories are time-varying. Therefore, for a vehicle with a nonzero velocity, the guidance system in combination with the controller should result in a stable overall system in order to ensure convergence to the desired trajectory. This constitutes the second main problem that is tackled in this thesis. Proving the stability of such a system can be a very difficult task if one attempts to find a single Lyapunov function incorporating all the states. However, the problem can be simplified significantly by using theorems from cascaded nonlinear systems theory, and this is the main approach we have adopted on several occasions here.

Similarly to the path-planning problem, there is a vast literature regarding the guidance task. A well-known and relevant theoretical result is the one by Brockett, which states that *point stabilization* (stabilizing the position of a vehicle at a given point and attitude) for vehicles with nonholonomic constraints is not achievable with constant state-feedback control [43]. This is not a problem for fully-actuated

vehicles, contrary to underactuated ones. Each motion control scenario has its own constraints and goals, hence a different guidance law might be more appropriate in each case. In this thesis we deal with the following two motion control scenarios (we employ the definitions found in [42]):

- *Path following*: The control objective is to follow a predefined path, which only involves a spatial constraint. No restrictions are placed on the temporal propagation along the path.

- *Path tracking*: The control objective is to track a target that moves along a predefined path (similar to trajectory tracking). Consequently, it is possible to separate the target-related spatio-temporal constraint into two separate constraints. Still, this scenario can be viewed as a target-tracking scenario and handled with target-tracking methods, thus disregarding any apriori path information that is available.

Both scenarios presuppose the existence of a target path. The only difference, albeit a major one, between the two cases is the temporal constraint, which refers to the requirement that the vehicle be at a specific point on the path at a specific time instant. In the case of an underactuated vehicle, a marine craft for instance, this means that for a path-following scenario is it sufficient to have a nonzero forward speed and generate only the reference trajectories for the rudder so as to steer the vessel appropriately. For the path-tracking scenario though, it is also necessary to generate desired values for the surge velocity. This breaks the overall assignment into two subproblems a) the geometric task, and b) the dynamic task, which constitutes the maneuvering problem, as defined in [208].

Another important distinction from mission to mission, independently of the motion control scenario, is the geometry of the path. In many cases, only straight lines are considered, whereas in others curved paths are assigned. It is worth having a closer look at the distinctive attributes of each occurring case:

- *Straight-line path following*: This is illustrated in Fig. 1.3 and constitutes the simplest case. Assuming (for analysis purposes) that the vehicle velocity is equal to zero, there are no time-varying parameters in the problem. The cross-track error y_e (the shortest distance to the path) remains constant. This changes when the vessel moves, that is $u \neq 0$, since the cross-track error is time-varying now. Moreover, the final desired heading is constant, because the path has a time-invariant path-tangential angle γ_p.

- *Curved path following*: This is similar to the previous case with the exception of the path-tangential angle, which is now time-varying, that is $\dot{\gamma}_p \neq 0$. An illustration can be seen in Fig. 1.4. As a consequence, there is no desired steady-state value for the heading.

- *Straight-line path tracking*: This is depicted in Fig. 1.5. Due to the time constraints involved, minimizing the along-track error x_e becomes part of the problem. It is easier to address the path-tracking problem (both intuitively and mathematically) if we assume that a virtual vehicle (plotted with dashed curves) with kinematics that generate the desired performance is moving on the path. The virtual vehicle can be a particle, since we are only interested in knowing the desired location on the path at any time instant. Assuming zero
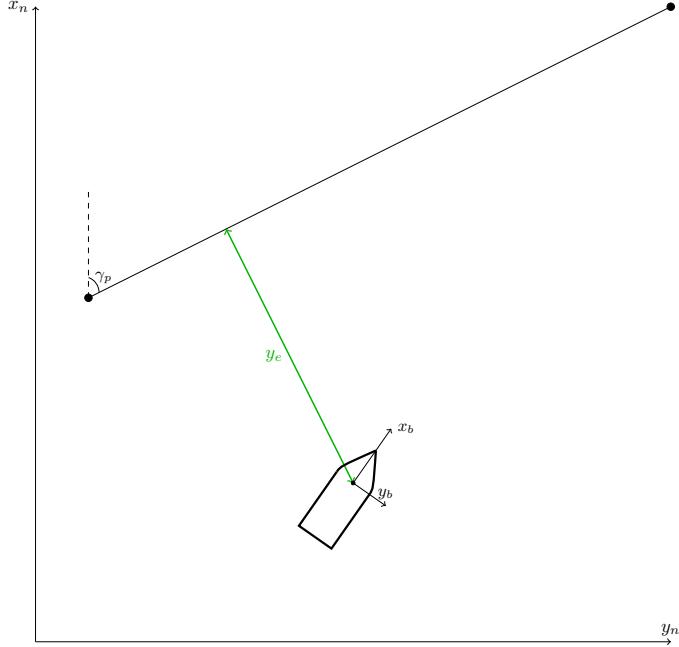
Figure 1.3: Path following: straight-line case

vehicle velocity shows that the along-track error is time-varying, $\dot{x}_e \neq 0$ while the cross-track error is constant, $\dot{y}_e = 0$. We also have constant steady-state value for the heading because $\dot{\gamma}_p = 0$.

- *Curved path tracking*: This is the most complex of the four presented cases, an illustration can be seen in Fig. 1.6. In addition to the challenges posed by straight-line path tracking, the curved path entails a time-varying cross-track error y_e even for a hypothetical zero vehicle velocity. Moreover, $\dot{\gamma}_p \neq 0$.

The descriptions above give useful information regarding the geometry and the challenges posed by each motion control scenario in its simplest form. It goes without saying that there are other factors which can make the problems even harder, also from a kinematic point of view. An example is the presence of external disturbances, such as unknown environmental forces. In that case, a fully-actuated vehicle would have to allocate its thrust in a different way in order to continue moving on the path without compromising its heading. On the other hand, an underactuated vehicle would be compelled to change its heading so as to produce a force component capable of counteracting the influence of the disturbances. Similar issues, in combination with the varying complexity of the motion control scenarios described above, can result in a wide range of problems.

Some of the most popular methods adopted by the marine community stem from the missile guidance literature (see [39, 40, 205]) and typically include the line-of-sight (LOS) guidance (see also [93]), the pure-pursuit guidance and the
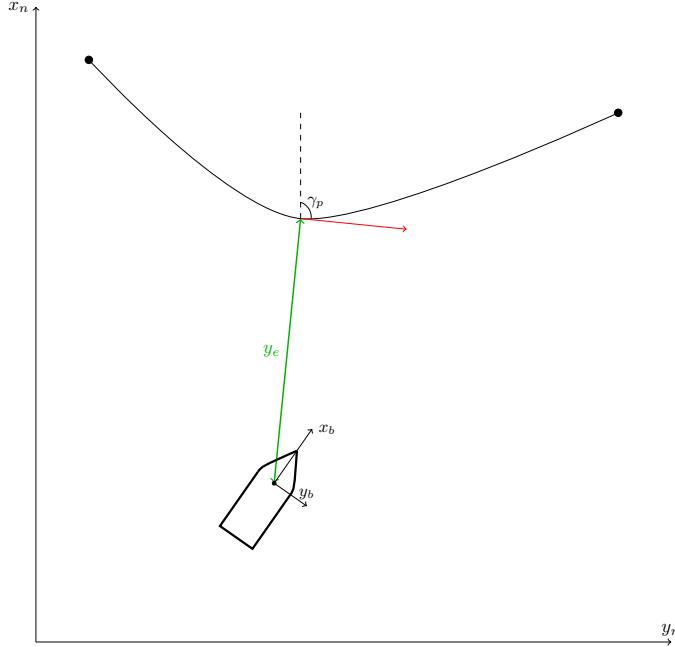
Figure 1.4: Path following: curved-path case

constant-bearing guidance. All these methods are presented in detail in [35]. The augmentation of the proportional navigation (PN) law, which aims at minimizing the LOS between the interceptor and the target, is presented and discussed in [234–236]. An alternative path-following methodology using vector fields and course angle measurements for air vehicles exposed to constant wind was presented in [174] and supported with experimental results. Global convergence using the vector fields method was shown also in [152]. Guidance techniques for UAVs navigating under the influence of wind were also developed in [141, 180, 201]. The use of potential flow for path-following purposes was investigated in [189]. In [185], the authors developed an approach for trajectory tracking which resembles PD control when tracking straight-line paths and incorporates an additional control element when tracking curved segments, further theoretical analysis was provided in [186]. The method was extended in [66] in order to make it more suitable for real-world operations.

Gates uses the notion of a "ghost" (virtual) vehicle which moves on a geometrically defined path in [106]. The ghost vehicle is coupled to the motion of the real vehicle via a fictitious mechanical link which is not of fixed length but has a spring-like nature. A virtual vehicle had been used also earlier, in [149], to avoid the singularities which occur when projecting the position of the guided vehicle on a path, and a kinematic controller was developed for steering the vehicle on the path accurately. In [148], the authors stress the differences between land vehicles and marine craft and develop a nonlinear path-following method which relaxed several
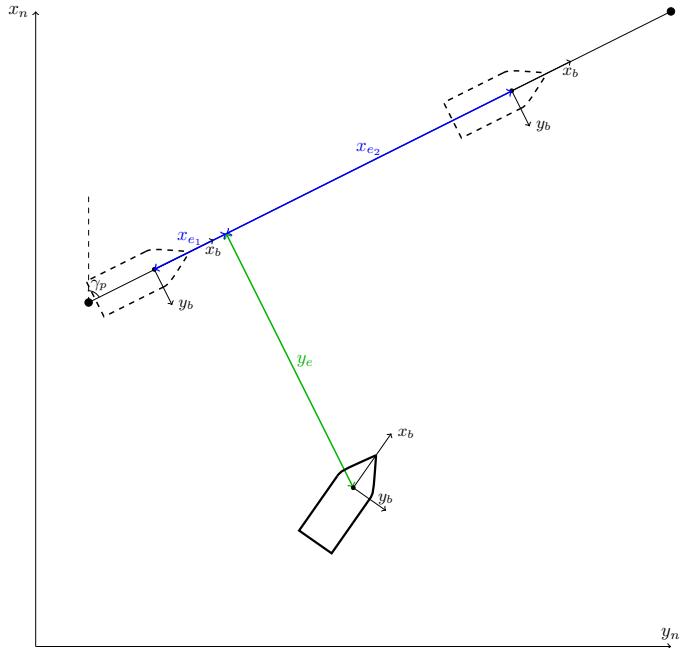
Figure 1.5: Path tracking: straight-line case

assumptions compared to previous approaches, such as constraints regarding the initial position error. The results were validated experimentally in [25], and [148] was further extended in [147] by adding a hybrid parameter adaptation scheme to account for vehicle parameter uncertainty. Furthermore, in [26, 125], the authors develop and validate experimentally a methodology where the underactuation of the vehicle is taken into account explicitly when defining the error variable. The interconnection among the GNC modules is discussed in [27], although the stability problem for the total system is not investigated. The authors also develop a path-following algorithm which employs the virtual vehicle concept, and discuss the effect of ocean currents. Indiveri et al. give a proof of concept for the guidance of underactuated vehicles exposed to constant unknown disturbances in 3-D space by designing an observer for estimating the disturbances [126]. The results indicate convergence of the method, however the authors state that the proof needs further work in order to be completed. Hladio et al. use traverse feedback linearization for path following in [120], hence developing a method which can implemented for a large class of systems. However, feedback linearization depends on accurate knowledge of the system parameters and, as result, the robustness of the method in the case of systems with uncertainties should be investigated. The ability of fuzzy logic to deal with uncertainty is exploited in [14] by defining a set of rules which imitate the behavior of a human driver. The method uses approximate knowledge regarding the curvature of the path ahead of the vehicle, as well as the distance between the vehicle and the next turn. Optimal methods have also been proposed, for more
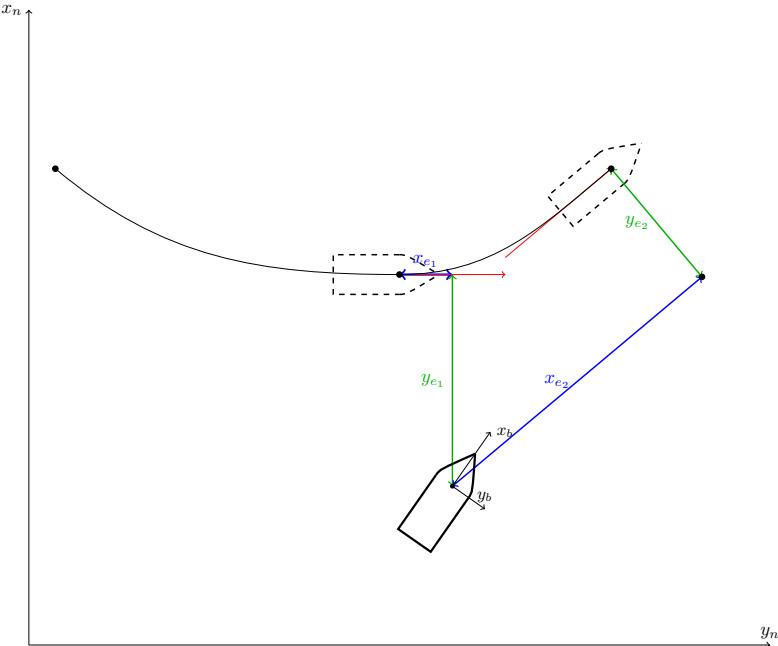
Figure 1.6: Path tracking: curved-path case

details the reader is referred to [199, 200], and formation control application have attracted a lot of attention, resulting in several novel solutions to the problem, see for instance [48, 123, 124, 178, 209]. In this work, we employ the LOS guidance for minimizing the cross-track error, more details regarding the literature relevant to the topic are given in Chapters 7–10.

1.2   List of Publications

The following eight international publications are the outcome of the work related to this thesis.

Book Chapter

[156] A. M. Lekkas and T. I. Fossen. Line-of-Sight Guidance for Path Following of Marine Vehicles. Chapter 5, In: *Advanced in Marine Robotics*. Editor: Oren Gal. LAP LAMBERT Academic Publishing, 2013, pp. 63-92. ISBN 978-3-659-41689-7.

Journal Papers

[158] A. M. Lekkas and T. I. Fossen. Integral LOS Path Following for Curved Paths Based on a Monotone Cubic Hermite Spline Parametrization. *IEEE*

Transactions on Control Systems Technology, 2014.

[160] A. M. Lekkas, A. R. Dahl, M. Breivik and T. I. Fossen. Continuous-Curvature Path Generation using Fermat's Spiral. *Modeling, Identification and Control*, Vol. 34, No. 4, 2013, pp. 183-198.

[91] T. I. Fossen and A. M. Lekkas. Direct and indirect adaptive integral line-of-sight path-following controllers for marine craft exposed to ocean currents. *International Journal of Adaptive Control and Signal Processing*, (submitted), 2014.

Conference Papers

[155] A. M. Lekkas and T. I. Fossen. A time-varying lookahead distance guidance law for path following. In *9th IFAC Conference on Manoeuvring and Control of Marine Craft*, Arenzano, Italy, 2012.

[157] A. M. Lekkas and T. I. Fossen. A quaternion-based LOS guidance scheme for path following of AUVs. In *9th IFAC Conference on Control Applications in Marine Systems*, Osaka, Japan, 2013.

[53] M. Candeloro, A. M. Lekkas, A. J. Sørensen, and T. I. Fossen. Continuous curvature path planning using Voronoi diagrams and Fermat's spirals. In *9th IFAC Conference on Control Applications in Marine Systems*, Osaka, Japan, 2013.

[159] A. M. Lekkas and T. I. Fossen. Minimization of cross-track and along-track errors for path tracking of marine underactuated vehicles. In *European Control Conference*, Strasbourg, France, 2014.

1.3 Thesis Outline

The rest of the thesis in organized in the following way:

Chapter 2 This chapter gives an overview of some previous results which are used in the thesis.

Chapter 3 A number of proposed evaluation criteria, which can be used for determining the most suitable path for a motion control scenario, are the main topic of this chapter. The criteria are related to several path characteristics, among which some can be more easily quantified than others. The overall intention is to initiate an effort that will reach a level of development where an onboard computer will be able to take decisions upon which is the best path to follow, always depending on the problem constraints (minimum allowed distance from obstacles, for example) and the available resources (computational power). The content of this chapter was published, in a simpler form, in [156].

Chapter 4 The idea of using the monotonous cubic Hermite spline interpolation (CHSI) technique developed by Fritsch and Carlson and published in [97] is discussed in this chapter. The main motivation is to construct paths which

are practical and avoid zig zags and wiggles between two successive waypoints even when these waypoints are not the output of a smooth function, something likely to happen in cluttered environments. The method offers better shape control and induces no considerable changes when a waypoint changes location. This chapter is connected with Chapter 8, where the CHSI paths are used for proving stability of the line-of-sight guidance with time-varying lookahead distance. The results of this chapter were published in [158].

Chapter 5 This chapter explores the properties of Fermat's spiral (FS), or parabolic spiral, and proposes FS as an efficient alternative to both Dubins paths and the clothoids. The paths based on FS avoid the curvature discontinuity, which plagues Dubins paths, and are very inexpensive computationally, a property that gives them a strong advantage over the clothoids. An alternative parametrization of the spiral is proposed, which removes the parametric speed singularity at the origin. As a result, FS can result in very practical paths consisting only of straight lines and arc segments, and can be used for both path-following and path-tracking (or trajectory-tracking) applications. The FS is combined in Chapter 6 with the Voronoi diagram method for obstacle avoidance. The content of this chapter is based on [160].

Chapter 6 A methodology for getting useful paths from the roadmaps initially generated by the Voronoi diagrams (VD) method is presented in this chapter. The main goal was to generate paths which are safe, smooth, and do not require unnecessary heading changes. For this reason, the VD is employed in order to initially generate a roadmap where the edges are waypoint candidates for a path connecting one point (departure point) on the map to another (arrival point). A process consisting of several steps is designed so as to choose the waypoints in a way such as the problem requirements are met. After the waypoints have been selected, Fermat's spiral is used to smooth the initial piecewise-linear path connecting the waypoints. The material of this chapter is taken from [53].

Chapter 7 This chapter studies the line-of-sight (LOS) guidance algorithm in the case where no external disturbances act on the vehicle. Initially, the LOS guidance is presented for two uncoupled 3-DOF cases, one pertains to the horizontal plane and one to the vertical plane. Then, a coupled case is considered and it is shown how the coupling influences the sideslip angle equation. Finally, a simple methodology is developed so as to transform the LOS guidance in quaternion form for all the aforementioned cases (coupled and uncoupled). The results of this chapter are based on [156, 157].

Chapter 8 In this chapter, a modification of the LOS guidance is studied. More specifically, the idea of using a time-varying expression for the lookahead distance Δ is explored. It is shown that a time-varying Δ results in a flexible maneuvering behavior and can contribute to reducing vibrations around a desired path. The consequences of introducing the time-varying lookahead distance are also investigated from a stability point of view. In addition, the parametrization of the CHSI paths presented in Chapter 4 is used to show stability of the time-varying Δ LOS guidance in the case of curved paths. The interconnected nature of the system is described and the concept of forward

completeness is employed to ensure that the system can be considered as a cascaded system. The results of this chapter were published in [155, 158].

Chapter 9 The topic of this chapter is the augmentation of the LOS guidance so as to compensate for the influence of constant (with respect to the inertial frame) external disturbances, such as ocean currents and wind forces. First, it is demonstrated in a very simple way why integral action is not necessary when heading control is available (although integral action could always be used to compensate for modeling inaccuracies etc). Next, a integral line-of-sight guidance law is derived from the vehicle kinematics formulated in absolute velocities by using well-known results from Lyapunov stability theory. Finally, the vehicle kinematics based on relative velocities is considered and two novel adaptive techniques for estimating the effects of ocean currents and compensating for them are devised. Once again, the stability proof is based on cascaded nonlinear systems theory. The content of this chapter is based on [91, 156, 158].

Chapter 10 This chapter's contribution is the development of a guidance strategy which generates suitable reference trajectories for tackling the path tracking (trajectory tracking) problem. The LOS guidance is used again to generate the heading reference trajectories and minimize the cross-track error while the new guidance law minimizes the along-track error by generating appropriate reference trajectories for the velocity controller. The vehicle is assigned to pursue a virtual vehicle (or particle) moving on the path. Initially, the novel guidance methodology is based on the vehicle's kinematics formulated in an absolute velocities context. Cascaded nonlinear systems theory is employed to show stability of the total system, including the guidance system and the heading and velocity controllers. Then the approach is extended so as to account for the influence of ocean currents. This is achieved by combining the guidance law formulated in absolute velocity kinematics with the indirect adaptive control scheme from Chapter 9. In this way, it is possible to guarantee the estimation of all the parameters of the current in two dimensions, that is velocity and orientation w.r.t. the inertial frame, instead of just the effect of the current in a specific direction. The guidance system developed in absolute velocities is based on [159], whereas the formulation using relative velocities is unpublished.

Chapter 11 This chapter presents the conclusions and the main ideas for further developing the results of this thesis.

Chapter 2

Mathematical Preliminaries

2.1 Cascaded Nonlinear Systems

In this section we present the theorem which is employed on several occasions in this thesis to show stability of the cascade formed by the guidance system and the vehicle controller. The proof can be found in [181] and in our case we include a reformulated version, like the one which was used in [166].

Consider the cascade system:

$$\Sigma_1 : \dot{\mathbf{x}}_1 = \mathbf{f}(t, \mathbf{x}_1) + \mathbf{g}(t, \mathbf{x})\mathbf{x}_2 \tag{2.1}$$

$$\Sigma_2 : \dot{\mathbf{x}}_2 = \mathbf{f}(t, \mathbf{x}_2), \tag{2.2}$$

where $\mathbf{x}_1 \in R^n, \mathbf{x}_2 \in R^m, \mathbf{x} \triangleq [\mathbf{x}_1, \mathbf{x}_2]^T$. The function $\mathbf{f}(t, \mathbf{x}_1)$ is continuously differentiable in $(t, \mathbf{x}_1)$ and $\mathbf{f}(t, \mathbf{x}_2), \mathbf{g}(t, \mathbf{x})$ are continuous in their arguments and locally Lipschitz.

Theorem A: The cascaded system (8.57)-(8.58) is globally uniformly asymptotically stable if the following three assumptions are satisfied:

A1) The system $f(t, \mathbf{x}_1)$ is globally uniformly asymptotically stable with a Lyapunov function $V(t, \mathbf{x}_1)$, $V : R_{\geq 0} \times R^n \to: R_{\geq 0}$ positive definite (that is $V(t, 0) = 0$ and $V(t, \mathbf{x}_1) > 0 \; \forall \mathbf{x}_1 \neq 0$) and proper (that is, radially unbounded) which satisfies:

$$\left\|\frac{\partial V}{\partial \mathbf{x}_1}\right\| \|\mathbf{x}_1\| \leq c_1 V(t, \mathbf{x}_1) \quad \forall \|\mathbf{x}_1\| \geq \mu \quad \text{where } c_1, \mu > 0. \tag{2.3}$$

We also assume that $(\partial V/\partial \mathbf{x}_1)(t, \mathbf{x}_1)$ is bounded uniformly in t for all $\|\mathbf{x}_1\| \leq \mu$, that is, there exists a constant $c_2 > 0$ such that for all $t \geq t_o \geq 0$

$$\left\|\frac{\partial V}{\partial \mathbf{x}_1}\right\| \leq c_2 \quad \forall \|\mathbf{x}_1\| \leq \mu. \tag{2.4}$$

A2) The function $\mathbf{g}(t, \mathbf{x})$ satisfies

$$\|\mathbf{g}(t, \mathbf{x})\| \leq \theta_1(\|\mathbf{x}_2\|) + \theta_2(\|\mathbf{x}_2\|)\|\mathbf{x}_1\|, \tag{2.5}$$

where $\theta_1, \theta_2 : R_{\geq 0} \to R_{\geq 0}$ are continuous.

A3) Equation $\dot{\mathbf{x}}_2 = \mathbf{f}(t, \mathbf{x}_2)$ is globally uniformly asymptotically stable and for all $t_o \geq 0$,

$$\int_{t_o}^{\infty} \|\mathbf{x}_2(t, t_o, \mathbf{x}_2(t_o))\| \, dt \leq \phi(\|\mathbf{x}_2(t_o)\|). \tag{2.6}$$

2.2 Quaternion Fundamentals

In this chapter, the quaternion attitude representation is used, where the unit quaternion $\mathbf{q}$ corresponding to a rotation in $\mathbb{R}^3$ is defined as:

$$\mathbf{q} = \eta + \varepsilon_1 \mathbf{i} + \varepsilon_2 \mathbf{j} + \varepsilon_3 \mathbf{k}, \tag{2.7}$$

with $\|\mathbf{q}\| = 1$. The complex conjugate of $\mathbf{q}$ is given by:

$$\mathbf{q}^* = \eta - \varepsilon_1 \mathbf{i} - \varepsilon_2 \mathbf{j} - \varepsilon_3 \mathbf{k}, \tag{2.8}$$

and the inverse quaternion is given by:

$$\mathbf{q}^{-1} = \mathbf{q}^*. \tag{2.9}$$

Following [145, Ch. 5.9], let us associate an angle λ_q with the quaternion $\mathbf{q}$. Then, the image $\mathbf{w} \in \mathbb{R}^3$ of a vector $\mathbf{v} \in \mathbb{R}^3$ due to the rotation by an angle λ_q can be expressed in quaternion form:

$$\mathbf{w}_q = \mathbf{q}\mathbf{v}\mathbf{q}^*. \tag{2.10}$$

The rotation corresponding to the negative of a quaternion is the same as the rotation due to the quaternion:

$$\mathbf{w}_q = \mathbf{w}_{-q}. \tag{2.11}$$

Moreover, the product of two quaternions $\mathbf{q}_1, \mathbf{q}_2$ is defined as:

$$\mathbf{q}_1 \otimes \mathbf{q}_2 := \begin{bmatrix} \eta_1 \eta_2 - \boldsymbol{\varepsilon}_1^{\mathrm{T}} \boldsymbol{\varepsilon}_2 \\ \eta_2 \boldsymbol{\varepsilon}_1 + \eta_1 \boldsymbol{\varepsilon}_2 + \boldsymbol{\varepsilon}_1 \times \boldsymbol{\varepsilon}_2 \end{bmatrix}. \tag{2.12}$$

For more details the reader is referred to [98, 110, 145].

2.3 Parameter Projection

The parameter projection $\mathrm{Proj}(\hat{\theta}, \tau)$ used for ocean current estimation in Chapter 9 is defined as:

$$\mathrm{Proj}(\hat{\theta}, \tau) := \begin{cases} \left(1 - c(\hat{\theta})\right)\tau & \text{if } \left|\hat{\theta}\right| > M_{\hat{\theta}} \text{ and } \hat{\theta}\tau > 0 \\ \tau & \text{otherwise} \end{cases} \tag{2.13}$$

where $c(\hat{\theta}) = \min\{1, (\hat{\theta}^2 - M_\theta^2)/(M_{\hat{\theta}}^2 - M_\theta^2)\}$. This is a special case of the parameter projection from Appendix E of [142]. The following properties hold for the parameter projection: (i) $\mathrm{Proj}(\hat{\theta}, \tau)$ is locally Lipschitz continuous, (ii) $-\tilde{\theta} \, \mathrm{Proj}(\hat{\theta}, \tau) \leq -\tilde{\theta}\tau$.

Part II

Path Planning

Chapter 3

Path-Evaluation Criteria

3.1 Introduction

In the case of a path-following motion control scenario, it is important to decide upon the path to be followed before designing and implementing a guidance algorithm, this is commonly known as the path planning task. It is with respect to this path that the guidance reference trajectories will be generated and, depending on the overall task conditions and constraints, these paths can vary a lot. Frequently, the first step is to introduce a given order of fixed points in space, namely the waypoints, and define the desired path as the sum of the successive straight lines that connect these waypoints. This approach, although simple, might not be sufficient for applications that demand high accuracy because the resulting path is not smooth. There is a vast literature pertaining to this issue, as well as other important factors that arise and affect the performance in each considered case. For some missions, for example, it is critical that the vehicle converges to and stays on the exact path, whereas others are more concerned with finding the minimum length path, and so on. As a consequence, many different solutions have been presented and each one of them satisfies some desired properties that are prioritized.

Dubins showed, for instance, that for a particle with unity speed, the shortest possible path that meets a maximum curvature bound between a starting position with predefined orientation (starting pose) and a finishing position with predefined orientation (finishing pose) consists of at most three pieces, each of which is either a straight line or an arc of a circle of radius $R > 0$ [75]. However, the Dubins methodology does not result in curvature continuous paths due to the fact that a straight line has a curvature $\kappa = 0$, whereas a circle arc has a curvature $\kappa = 1/R$. Hence, there will be a jump in the curvature from 0 to $1/R$ when moving from the straight line to the circle arc. The Euler spirals (also known as clothoids) is an alternative approach that gives curvature continuous paths, but it is also more expensive from a computational point of view because it has an open form solution which includes the calculation of the Fresnel integrals. Other popular alternatives are the Pythagorean Hodographs, first introduced in [82] and the potential field method. For a more detailed and thorough treatment of path planning methods the reader is referred to [224] and [151].

3.2 Path-Evaluation Criteria

Given the large number of methodologies available when it comes to deciding which path is more suitable for an application, it is useful to introduce a number of evaluation criteria which can assist the designer with making a more fitting decision. The development of the path evaluation criteria presented here has benefited a lot from discussions with Andreas Dahl and Morten Breivik, and some of them have been mentioned also in [67, 156, 158].

3.2.1 Smoothness

This property is among the most important ones, due to the fact that it is directly related to the vehicle's dynamic constraints. As a consequence, it has motivated many researchers to investigate new path-design methodologies. In this thesis we are concerned with parametric curves/paths. In this context, two notions can be used to describe the path smoothness, namely, the parametric continuity and the geometric continuity. The main difference is that parametric continuity refers to both the speed and the orientation with which the parameter propagates through the path, whereas the geometric continuity is not concerned with the parameter speed, see also [20, 21, 191]. Therefore, parametric continuity can be considered as a subset of geometric continuity.

Geometric Continuity

Geometric continuity (GC) is a concept that has proved to be very useful in computed-aided geometric design (CAGD) applications because it can be viewed as a measure of continuity that is *parametrization independent*. This implies that, if we think of the word "curve" as the image of a parametrization and the word "parametrization" as an equation which describes a curve, many different parametrizations can result in the same curve [21]. This is a relaxed form of parametric continuity and it actually means that, as long as the curve is geometrically smooth, there is no concern about the speed of the parameter along the curve.

When it comes to motion control scenarios, GC can be used as a requirement in cases where there are no temporal constraints involved in the problem. Path following is such an example, where the vehicle is assigned to converge to the desired path without imposing *where* on the path it should be an any given time instant. GC is denoted G^n, with n specifying the degree of smoothness. We will now briefly give the definitions for GC up to the second degree.

- G^0: this refers to a continuous path, therefore the only requirement is that all subpaths are connected:

$$\mathbf{p}_{p,i}(\varpi_{ub,i}) = \mathbf{p}_{p,i+1}(\varpi_{lb,i+1}) \ \ \forall i \in [1, m-1]. \tag{3.1}$$

  An example is a path consisting of successive straight lines.

- G^1: it means that the two successive subpaths have a common unit tangent vector. In other words, the velocity vectors have the same orientation but

different magnitudes. As a result, the path-tangential angle is continuous. This can be verified by checking the expression for the path-tangential angle of a regular curve:

$$\chi_p(\varpi) = \text{atan2}\left(E_p'(\varpi), N_p'(\varpi)\right), \tag{3.2}$$

where $E_p'(\varpi), N_p'(\varpi)$ are continuous. It can be concluded that, for path-generation purposes, G^1 continuity is equivalent to:

$$\chi_{p,i}(\varpi_{ub,i}) = \chi_{p,i+1}(\varpi_{lb,i+1}) \ \ \forall i \in [1, m-1]. \tag{3.3}$$

Examples of G^1 paths are the Dubins path and the paths that are constructed from cubic Hermite spline interpolation.

- G^2: this implies that two successive subpaths have common unit tangent and curvature vectors. If we consider the curvature equation:

$$\kappa = \frac{N_p'(\varpi)E_p''(\varpi) - E_p'(\varpi)N_p''(\varpi)}{\sqrt[3]{\left(N_p'(\varpi)\right)^2 + \left(E_p'(\varpi)\right)^2}}, \tag{3.4}$$

then, for a regular parametrization, curvature continuity is ensured when $N_p'(\varpi), E_p'(\varpi), N_p''(\varpi), E_p''(\varpi)$ are continuous. Finally, G^2 continuity can be summarized by the expression:

$$\kappa_{p,i}(\varpi_{ub,i}) = \kappa_{p,i+1}(\varpi_{lb,i+1}) \ \ \forall i \in [1, m-1]. \tag{3.5}$$

One very popular example of G^2 continuity is a path which consists of straight lines and clothoids. In Chapter 5, a new methodology for constructing G^2 paths using straight lines and Fermat's spiral is given in detail. The importance of assigning curvature-continuous paths should not be underestimated, especially with respect to the performance of agile vehicles that perform fast maneuvers, an example can be found in [175].

Parametric Continuity

As mentioned earlier, parametric continuity (PC) is a stricter form of continuity which, in addition to GC, imposes constraints on how the parameter propagates along the path. According to [21], this has been a hindrance to many researchers in the field of computer graphics because PC does not necessarily reflect the smoothness of the resulting composite curve. Instead, it is a measure of smoothness for parametrizations. However, the nature of the guidance algorithms related to the motion control scenarios studied in this thesis often requires that parametric-continuous paths are assigned. More specifically about PC:

- C^0: in this case the definition is the same with that of G^0 continuity. Therefore, it pertains to geometrically connected paths, see Eq. (3.1) above.

- C^1: the velocity vector orientation and magnitude are continuous:

$$\mathbf{p}_{p,i}(\varpi_{ub,i}) = \mathbf{p}_{p,i+1}(\varpi_{lb,i+1}) \ \ \forall i \in [1, m-1], \tag{3.6}$$

$$\frac{d}{d\varpi}\mathbf{p}_{p,i}(\varpi_{ub,i}) = \frac{d}{d\varpi}\mathbf{p}_{p,i+1}(\varpi_{lb,i+1}) \ \ \forall i \in [1, m-1]. \tag{3.7}$$

A G^1 curve is visually identical to a C^1 curve, however the speed magnitude is discontinuous for the G^1 curve. Consequently, for a path-following scenario geometric continuity is equivalent to parametric continuity, but the same is not true for a motion control scenario that includes temporal constraints, such as path-tracking.

- C^2: the acceleration is continuous:

$$\mathbf{p}_{p,i}(\varpi_{ub,i}) = \mathbf{p}_{p,i+1}(\varpi_{lb,i+1}) \ \ \forall i \in [1, m-1], \tag{3.8}$$

$$\frac{d}{d\varpi}\mathbf{p}_{p,i}(\varpi_{ub,i}) = \frac{d}{d\varpi}\mathbf{p}_{p,i+1}(\varpi_{lb,i+1}) \ \ \forall i \in [1, m-1], \tag{3.9}$$

$$\frac{d^2}{d^2\varpi}\mathbf{p}_{p,i}(\varpi_{ub,i}) = \frac{d^2}{d^2\varpi}\mathbf{p}_{p,i+1}(\varpi_{lb,i+1}) \ \ \forall i \in [1, m-1]. \tag{3.10}$$

Apparently, acceleration continuity entails curvature continuity. C^2 continuity is always preferred, in some cases (such as path tracking) it is even necessary. However, it can be difficult to satisfy the conditions required for C^2 continuity. Apart from this, some curves which satisfy the conditions (3.8)–(3.10) might result in impractical paths. It is easy to understand that the quest for the "perfect path" is a nontrivial one. As is often the case in any discipline, the designer must eventually consider all the available options and choose to implement the one which fits better the problem in hand.

Discussion

Clearly, when it comes to motion control scenarios, parametric continuity should be the designer's goal. To this end, a very interesting property is reported in [21]: two curves meet with G^n continuity if and only if their arc length parametrizations meet with C^n continuity. This property is exploited in Chapter 5, where an alternative parametrization of Fermat's spiral is proposed and it is shown how this can lead to constructing practical C^2 paths.

3.2.2   Length

Following [165, Ch. 3], a regular arc $\mathbf{x} = \mathbf{x}(t)$, $a \le t \le b$ is rectifiable and its length can be computed by the integral:

$$s = \int_a^b \left\| \frac{d\mathbf{x}}{dt} \right\| dt = \int_a^b \sqrt{\left(\frac{dx_1}{dt}\right)^2 + \left(\frac{dx_2}{dt}\right)^2 + \left(\frac{dx_3}{dt}\right)^2} \, dt. \tag{3.11}$$

In many applications the goal is to design a path that will minimize (3.11) since this can be associated with arriving to the destination faster, or consuming less

energy. However, this is not always true, as it can be demonstrated by the Zermelo navigation problem where the task is to find the trajectory that minimizes the travel time when there is a constant force field (such as a steady current) present. In this case, the straight line is not the optimal path anymore. Moreover, regarding the path length optimality of the Dubins paths, it is worth noting that, depending on the application, it is not always possible to find a Dubins path, as it was shown in [219].

3.2.3 Precision and Allowance

Given a number of ordered waypoints, *path precision* simply investigates whether a path passes through all of them or not. To this end, the following two categories can be distinguished:

- Interpolating paths: it refers to the case where the path under consideration passes through all the waypoints.

- Approximating paths: it refers to the case where the path under consideration passes through some of the waypoints and comes close to, but without passing through, the rest.

This property is important in cases where the vehicle navigates in an area with obstacles that have to be avoided. In addition, it can prove useful in distinguishing paths that have the same properties (regarding the smoothness, for instance) but different geometry. Figs 3.1–3.2 demonstrate the differences between an approximating path (also known in the literature as circular smoothing) and an interpolating one (Dubins path). Both consist of straight lines and circular arcs and are designed by taking into account the same set of waypoints, however their geometries are different.

The term *allowance* refers to the maximum departure from a piecewise-linear path between waypoints [68]. Allowance is also a property which is important when the vehicle navigates in a cluttered environment. By specifying the maximum distance the path deviates from the linear path, the designer is provided with a very useful constraint when deciding the location of the waypoints. If this is combined with the Voronoi diagrams and the clearance constraint, the result can be a completely automated procedure with the following steps:

1. define the obstacles on a map.

2. specify what is the shortest safe distance the vehicle should keep from the obstacles (clearance constraint).

3. generate a set of ordered waypoints, which respect the clearance constraints, using Voronoi diagrams.

4. design a path with allowance equal or lower than the clearance constraint.

The allowance should be explicitly quantified, a process which can be difficult in some cases. Depending on the mission, it might even be useful to specify on which side of the paths tangent line the allowance occurs. In a *North-East* reference frame, for example, the expression $2.1E$, or $+2.1$ could mean that the allowance occurs on the eastern (positive) side of the path's tangent at that point. In a similar way, if the allowance occurred on the western side, the expression could be $2.1W$, or -2.1.
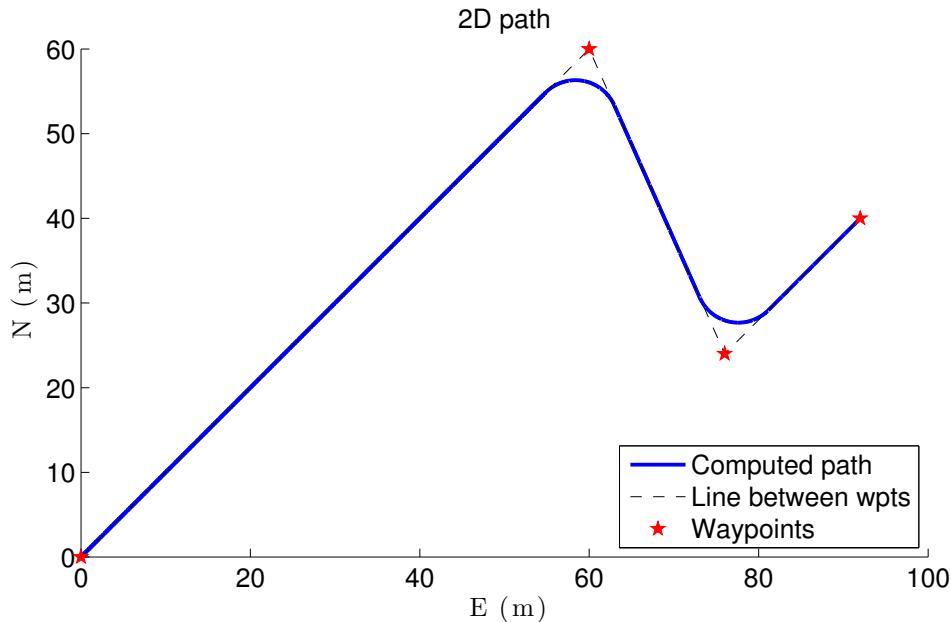
Figure 3.1: A circular smoothing path is an approximating one since is does not pass through all the waypoints

3.2.4   Tractability

This property comprises two rather important issues:

- the *practicality* of the overall shape of a path. This pertains to any unnecessary zig-zagging or wiggling the path might include between two successive waypoints. Such a geometry can be the result of satisfying another property, especially a continuous curvature. An example is the natural cubic spline, which can give curvature-continuous paths but their overall resulting geometry can often be very unsuitable for vehicle navigation. On the other hand, a monotone cubic spline methodology, such as the one presented in [97] and studied in Chapter 4, can give paths with a very practical shape but without preserving curvature continuity at the waypoint locations.

- the *shape control* of the path, which investigates what happens in the rest of the path if one of the waypoints changes location, or if a waypoint is added to the initial waypoint set. This second branch can be of importance in cases where due to updated weather data, for instance, one segment of the path should be avoided and one (or more) waypoint should change location. The question then is what happens to the rest of the path, and especially the part which precedes the newly-assigned waypoint because this might affect the vessel at its current position. Depending on the result, three different path control behaviors can be distinguished: 1) local control, where it is possible
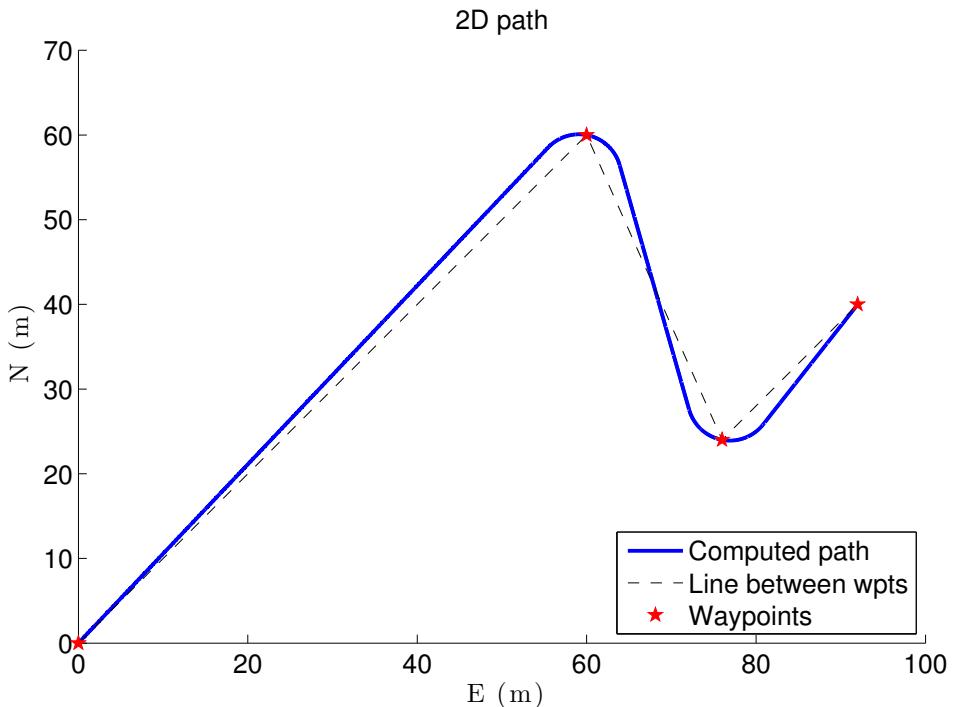
Figure 3.2: Dubins path is an interpolating one because it passes through all the waypoints

to change a waypoint without affecting the rest of the path, 2) global control, where there will be changes through the whole path, and 3) partial control, where in some cases local control is possible. An example of a local control method is the Cubic Hermite Spline Interpolation (CHSI), while natural cubic splines is a global control method and the monotone CHSI is a partial control method. An example is given in Figs 3.3–3.4.

According to [161], similar properties, although not exactly the same, have been used in the past by computer scientists in order to evaluate curves for applications related to computer graphics. An example is the concept of *extensionality*, suggested by Knuth in [139], which Moreton called *consistency* [172].

3.2.5 Computational Time

Each path-generation method is based upon different principles and it comes as no surprise that the involved mathematics might result in algorithms that are computationally more expensive than others. Path planning is often a real-time process and, although modern large vehicles should have available onboard equipment which is powerful enough for such tasks, the same might not be true for smaller
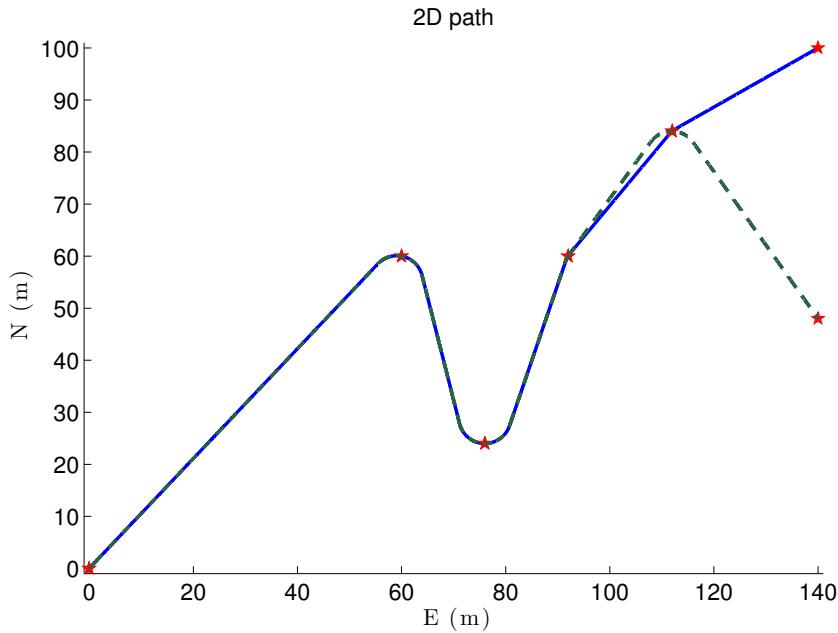
Figure 3.3: Shape control of Dubins path. Changing the last waypoint of the initial path (solid line) results in a path (dashed line) which is almost the same, with the only difference occurring at the neighboring segment. The path shape along the first four waypoints remains unaffected.

unmanned vehicles with limited power. In dynamic environments with moving obstacles, the path generation algorithm must be able to give feasible solutions within limited time and this can make the task a difficult one if the method is based on open form solutions.

When comparing two methods w.r.t the computational time they require, it can be useful to do so for a large number of waypoints in order to amplify their differences and conclude more clearly. An even more complete approach would be to create a figure where the horizontal axis is the increasing number of waypoints and the vertical axis shows the ratio between the computational time of the two methods.

3.2.6   An Example

Table 3.1 summarizes all the topics covered in this chapter in a compact way. The following three paths are compared:

A : Piecewise linear.

B : Dubins path.

C : Dubins path with clothoid transition.
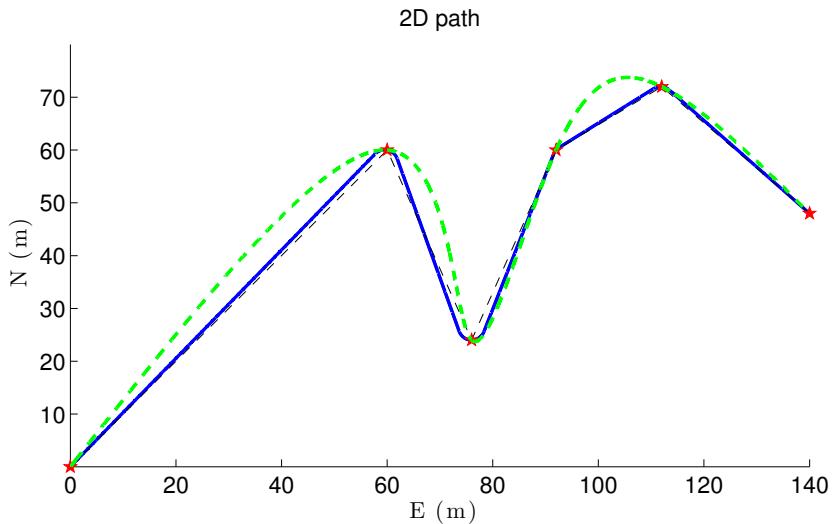
Figure 3.4: Path practicality comparison between Dubins path and a path constructed by natural cubic splines. The latter has a less practical shape, and higher allowance. On the other hand, the natural cubic splines preserve curvature continuity, while the Dubins path does not.

Method	Smoothness	Precision & Allowance	Length	Tractability	Comp. Time (20wpts)
A	C^0	Interp (A_A)	L_A	High/Local	T_A
B	C^1	Interp (A_B)	L_B	M-H/Local	T_B
C	C^2	Interp (A_C)	L_C	M-H/Local	T_C
D	C^2	Approx (A_D)	L_D	M-H/Local	T_D

Table 3.1: Path-evaluation criteria: an example

D : Clothoid smoothing.

Presenting the path characteristics in such a way can help a human operator make decisions, although "complete" autonomy should be the ultimate goal. The user can easily distinguish between interpolating and approximating methods, take into account the length of the path etc. Tractability is a feature that can probably be best handled by a computer because, due to its somehow vague nature, it might be better to quantify the path practicality with the help of fuzzy logic theory. After every criterion has been quantified and the problem constraints have been set, a decision should be taken based on the overall score of each path.

3.3 Conclusions

Autonomy presupposes the ability to take into account several factors during operation and make decisions which are both feasible and efficient. Inevitably this leads to the need for quantification of these factors. In relation to the motion-planning problem, this implies that the selection of a suitable path for a specific application must be based on a number of path-evaluation criteria which will be considered before choosing the most appropriate one. It is not trivial to infer upon the importance of each one of the criteria presented in this chapter and name one of them as the most critical. This will always depend on the problem in hand and whether there are any conditions that can be relaxed or not. For instance, a path consisting of straight lines and clothoid arcs has continuous curvature but is more expensive computationally, which means that it might be difficult to implement it in a dynamic environment where the total path has to be updated frequently. Dubins paths, on the other hand, can be computed almost instantly. So, in this case, the question is whether we can tolerate a relatively short cross-track error at the curvature discontinuity locations. If this is not acceptable maybe it is possible to slow down the vehicle and use the extra time in order to use clothoids.

The main purpose of the criteria presented here is to act as a starting point of a decision-making strategy which will be further extended until the vehicle's onboard computer has sufficient information and therefore is able to pick the best solution available.

Chapter 4

Path Generation Using Monotone Cubic Hermite Spline Interpolation

4.1 Introduction

This short chapter proposes the use of a special case of cubic Hermite splines, introduced by Fritsch and Carlson (FC) in [97], for path-planning purposes. The most important property of the cubic Hermite spline interpolation (CHSI) presented in this chapter is that it preserves *monotonicity* of the waypoints. In fact, the authors of [97] developed this class of splines with the goal of providing reliable interpolating curves to data from different disciplines which were known to be monotone, but the conventional curve fitting techniques would give unrealistic solutions which failed to preserve the shape of the data. It was often necessary to sacrifice interpolation in favor of monotonicity, or the other way around. In the previous chapter it was explained that it is preferred to construct paths which are *practical*, meaning that any unnecessary wiggling must be avoided. Such a property prevents from inducing unwanted path-length increases and, maybe most importantly, helps creating safer paths with lower allowance. As is the case with the conventional CHSI, the monotone version by FC is C^1 and therefore has curvature discontinuities at the locations of the waypoints. Despite this deficiency, the monotone CHSI gives very practical paths that are very difficult, if not impossible, to get with conventional splines. It is also important to note that the monotone CHSI performs very well when the data (waypoints, in our context) are the result of a non-smooth function, which is very likely when using the Voronoi diagram in cluttered environments, for example. Apparently, such performance is very difficult to be achieved by C^2 methods because the constraints on the second derivative will require considerable manipulation of the path shape.

This chapter serves the following purposes:

1. Propose the monotone CHSI by FC as an efficient spline-based methodology for constructing paths. The resulting paths have very convenient shape for path-planning purposes and, in addition, the CHSI is very cheap computationally. The main drawback is the curvature discontinuity, which induces a cross-track error on the waypoints locations.

2. Present a few examples that show *tractability* two different levels of shape control.

3. Modify and prepare the notation in order to be used in Chapter 8 for proving the stability of the line-of-sight guidance with time-varying lookahead distance Δ in the case of curved paths.

The results of this chapter were published in [158].

4.2   Monotone Cubic Hermite Spline Interpolation

It is common practice in many applications to use polynomials as a means of performing interpolation on a given data set. Given the values (also called control points) $f_1, \ldots, f_n$ of the function f at some finite set of points $\{x_1, \ldots, x_n\}$, an interpolating curve is one that passes through each point and an approximating curve is one that gets close but does not necessarily pass through all the points. For path-following applications it is natural to require that the curve connecting the waypoints is an interpolating one, since the initial waypoint planning might have been developed with obstacle avoidance in mind. The polynomial that results in an interpolating curve is called the *Lagrange interpolation polynomial*. In the case where f is differentiable and the problem includes constraints related to the derivative of f at the interpolated points, the polynomial that gives the desired curve is called a *Hermite interpolation polynomial*. Nevertheless, for a high number of waypoints a high degree polynomial is necessary in order to pass through all of them. This is unwanted because the outcome would be a very intractable and of no practical use path, due to Runge's phenomenon.

Therefore, a solution to the path generation problem is to use a Hermite interpolating polynomial of degree 3 (that is, a cubic Hermite spline) for each pair of successive waypoints. Given a nonnegative integer n, let $\mathcal{P}_n$ denote the set of all real-valued polynomials of degree $\leq n$ defined over the set $\mathbb{R}$. A polynomial $p_3 \in \mathcal{P}_3$ guarantees that the velocity function is smooth between successive waypoints and the assigned derivative values at each waypoint ensure that the velocity function is smooth along the whole path. In addition to this, the curvature function is continuous between successive waypoints but, in general, not continuous on the assigned waypoints.

Following [97], we define $\pi : a = x_1 < x_2 < \cdots < x_n = b$ as a partition of the interval $I = [a, b]$ and $f_i : i = 1, 2, \ldots, n$ the corresponding set of monotone data at the partition points (also knots, or control points). The algorithm of Fritsch and Carlson constructs on π a piecewise cubic function $p(x_i) \in C^1[I]$ such that

$$p(x_i) = f_i, \quad i = 1, 2, \cdots, n \tag{4.1}$$

and $p(x)$ is monotone. In each subinterval $I_i = [x_i, x_{i+1}]$, $p(x)$ is a cubic polynomial which can be described by the equation:

$$p(x) = f_i H_1(x) + f_{i+1} H_2(x) + d_i H_3(x) + d_{i+1} H_4(x), \tag{4.2}$$

with $d_j = p'(x_j)$ denoting the corresponding first derivatives at the points $j = i,\ i+1$ and the terms $H_k(x)$ are the cubic Hermite basis functions for the interval

I_i:

$$
\begin{aligned}
H_1(x) &= \zeta((x_{i+1} - x)/h_i) & H_3(x) &= -h_i\eta((x_{i+1} - x)/h_i) \\
H_2(x) &= \zeta((x - x_i)/h_i) & H_4(x) &= h_i\eta((x - x_i)/h_i),
\end{aligned}
\tag{4.3}
$$

where $h_i = x_{i+1} - x_i$, $\zeta(t) = 3t^2 - 2t^3$, $\eta(t) = t^3 - t^2$. In order to preserve monotonicity, the algorithm adjusts the tangents at the control points in the following way [171]:

Let $S_i = (f_{i+1} - f_i)/h_i$ be the slopes of the piecewise linear interpolants. Consequently, if S_i and S_{i-1} have opposite signs, or either of them is zero, this means that x_i is a local minimum or maximum and therefore we set:

$$
d_i = 0.
\tag{4.4}
$$

In the case where S_i and S_{i-1} have the same sign and the corresponding intervals are of the same length, then the tangent is calculated as:

$$
\frac{1}{d_i} = \frac{1}{2}\left(\frac{1}{S_{i-1}} + \frac{1}{S_i}\right).
\tag{4.5}
$$

Finally, if S_i and S_{i-1} have the same sign but the corresponding intervals have different length, we calculate the tangent as:

$$
\frac{w_1 + w_2}{d_i} = \frac{w_1}{S_{i-1}} + \frac{w_2}{S_i},
\tag{4.6}
$$

where $w_1 = 2h_i + h_{i-1}$, $w_2 = h_i + 2h_{i-1}$. The algorithm is also available in Matlab® (function `pchip.m`).

The methodology described above can be extended to parametric splines. This entails the introduction of the independent variable θ and the formulation of one separate equation for each one of the data variables $(x, f(x)) = (x, y)$. Let $\theta_1 < \theta_2 \ldots, \theta_n$ be the path variable or partition of the interval $[\theta_1, \theta_n]$, and let (x_i, y_i) for $i = 1, 2, \ldots, n$ be the corresponding 2-D waypoints. Based on [97] we can interpolate the data between two waypoints (x_i, y_i) and (x_{i+1}, y_{i+1}) according to:

$$
x_d(\theta) = c_{x3}(\theta - \theta_i)^3 + c_{x2}(\theta - \theta_i)^2 + c_{x1}(\theta - \theta_i) + c_{x0},
\tag{4.7}
$$

$$
y_d(\theta) = c_{y3}(\theta - \theta_i)^3 + c_{y2}(\theta - \theta_i)^2 + c_{y1}(\theta - \theta_i) + c_{y0},
\tag{4.8}
$$

where

$$
\begin{aligned}
c_{x0} &= x_i & c_{y0} &= y_i \\
c_{x1} &= x_i' & c_{y1} &= y_i' \\
c_{x2} &= \frac{3S_i^x - x_{i+1}' - 2x_i'}{\Delta\theta_i} & c_{y2} &= \frac{3S_i^y - y_{i+1}' - 2y_i'}{\Delta\theta_i} \\
c_{x3} &= \frac{x_{i+1}' + x_i' - 2S_i^x}{\Delta\theta_i^2} & c_{y3} &= \frac{y_{i+1}' + y_i' - 2S_i^y}{\Delta\theta_i^2},
\end{aligned}
\tag{4.9}
$$

with $(\cdot)'$ denoting differentiation w.r.t. parameter θ, $\Delta\theta_i = \theta_{i+1} - \theta_i$ is the local mesh spacing, and $S_i^x = (x_{i+1} - x_i)/\Delta\theta_{i+1}$ and $S_i^y = (y_{i+1} - y_i)/\Delta\theta_{i+1}$ are the slopes of the piecewise linear interpolants.
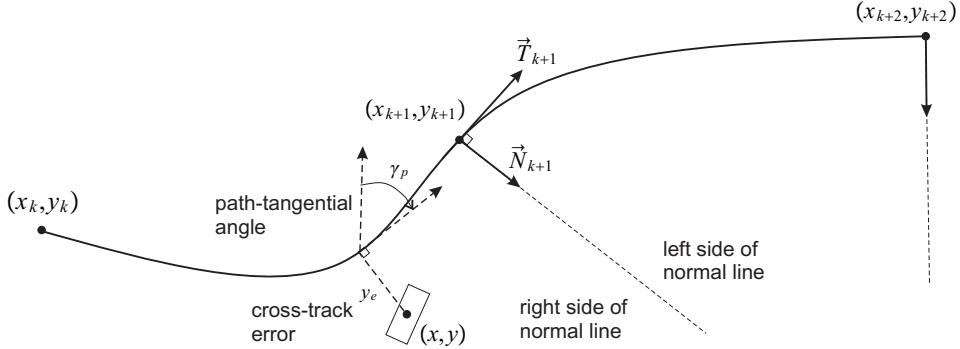
Figure 4.1: Curved path-planning between waypoints.

Discussion regarding the curvature

When it comes to path generation methods, there are two main concerns related to the curvature: a) whether the method is curvature continuous or not, and b) if is it possible to assign a maximum curvature throughout the path. In accordance with the nonparametric case, the interpolants have continuous derivatives, $x_d(\theta) \in C^1$ and $y_d(\theta) \in C^1$, but not necessarily a continuous second derivative. The continuity of the second derivative and the order of accuracy depend on how x_i' and y_i' are calculated. Therefore, the CHSI is, in the general case, discontinuous on the locations of the waypoints. This affects the heading controller input and is further discussed in Section 8.3. In addition to this, and despite the fact that splines have a closed-form expression of position, assigning maximum curvature to a planar path generated by parametric polynomial cubic curves has been reported to be a difficult problem, see for instance [227, 232]. The reason for this is that the curvature of such a parametric curve is a complicated function of the curve's parameters. The CHSI has been employed in this chapter in order to: a) prove stability of the time-varying lookahead distance LOS guidance when converging to curved paths (see Section 8.3), and b) demonstrate the partial/local property of paths (Section 7.5.5) which can be useful for real-time applications.

4.3   Simulations

The practicality and tractability of paths constructed using CHSI can be demonstrated in Figs 4.2 and 4.3. The main idea is initially to construct a path which connects a number of waypoints, and then introduce some changes on the path and observe how the total path is affected.

More specifically, in Fig. 4.2 an initial path (solid line) connecting the seven waypoints $WP1 - WP7_a$ is constructed. Although the path is not the output of a smooth function (for instance, at $WP5$ there is a quite steep turn), the CHSI preserves the monotonicity of the path without problems. Moreover, no significant deviation from the line connecting two successive waypoints can be observed.
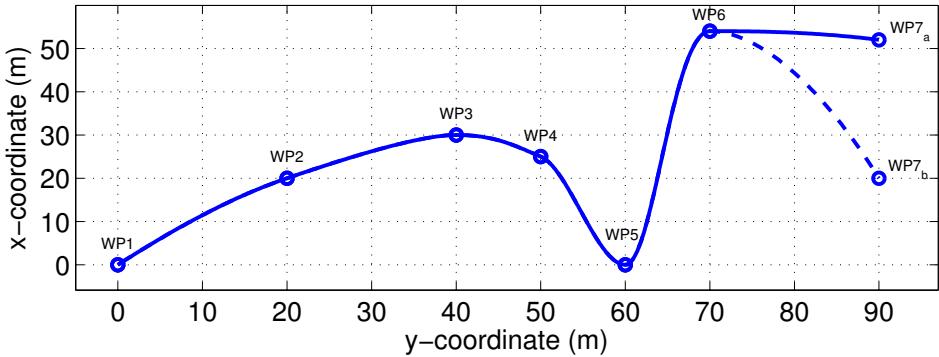
Figure 4.2: Local control and monotonicity of the implemented path-planning method are shown. The last waypoint of the initial path has changed, but in a way such that the lines $WP6 - WP7_a$ and $WP6 - WP7_b$ have slopes of the same sign. This change does not affect the overall shape of the initial path.
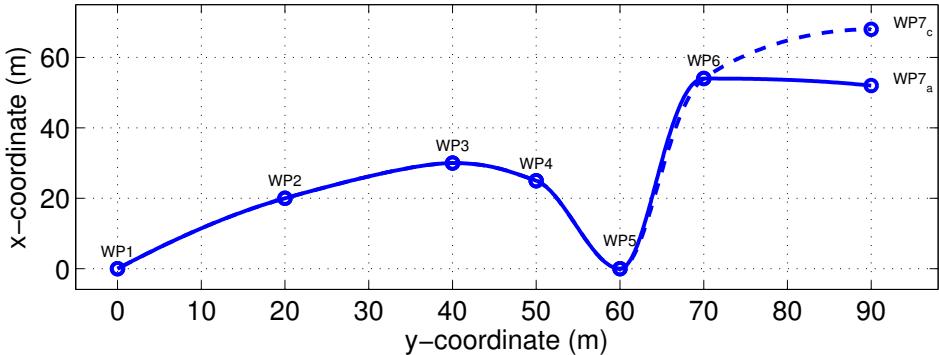
Figure 4.3: Partial local control of the monotone CHSI method is shown. In this case the last waypoint of the initial path changes in a way such that the slopes of $WP6 - WP7_a$ and $WP6 - WP7_c$ have a different sign. This causes a small change at the segment $WP5 - WP6$.

Therefore the path remains practical and safe throughout its length, with its main disadvantage being the curvature discontinuity. Then, we introduce a modification by changing the location of the last waypoint and recompute the total path. Recomputing the whole path is necessary for preserving first derivative smoothness. The new path (dashed line) consists of the waypoints $WP1 - WP7_b$ and it can be seen that the only segment which differs compared to the initial path, $WP1 - WP7_a$, is $WP6 - WP7_b$. This property, which we will refer to as *local control*, can be very useful in real-time applications where a future waypoint might have to be relocated due to unpredictable factors, such as bad weather conditions.

However, note that the CHSI does not always satisfy the local control property, this can be demonstrated in Fig. 4.3. In this case, only the segment $WP5 - WP6$ changes in addition to the relocated waypoint, but very slightly, hence making it almost eligible for satisfying the local control property. This occurred because the relocated waypoint was placed at a location such that the segment $WP6 - WP7_c$ is strictly increasing, contrary to $WP6 - WP7_a$, which is strictly decreasing. Therefore (4.4)–(4.6) induced the small change in segment $WP5 - WP6$.

4.4   Conclusions

This chapter presented a monotone version of the cubic Hermite spline interpolation as a means of constructing practical curved paths. Contrary to what might be the case with conventional splines, the monotone CHSI proved itself successful in providing the user with interpolating curves, which did not include any wiggling or zig zagging even when the waypoints are determined by a non-smooth function. This spline is further used in Chapter 9 in combination with a vehicle guided by the line-of-sight guidance law. It should be mentioned that this is considered to be a stepping stone which will eventually lead to spline-based, curvature-continuous and shape-preserving paths. Although this may sound like an over-optimistic statement, there is evidence in the literature that it might be possible both in 2D and 3D, see [64, 65, 146, 167].

Chapter 5

Continuous-Curvature Path Generation Using Fermat's Spiral

5.1 Introduction

Path-planning systems are of great significance when it comes to the performance and mission accomplishment of practically every type of vehicle, as well as mechatronic devices such as computer numerical control (CNC) machines and robotic manipulators, to name a few. Depending on the task demands, the path-planning algorithms generate appropriate paths or trajectories usually by taking into account both the physical constraints of the system under consideration, and the workspace constraints, such as obstacles or environmental forces. A thorough treatment of the subject can be found in [151].

In the case of underactuated vehicles, especially in path-following and path-tracking motion control scenarios, the shape and properties of the path have a direct influence on the guidance system, which is responsible for generating reference trajectories to be fed to the autopilot. The reason for this is that the path curvature affects the heading angle commands and the parametric speed affects the velocity commands. In these applications, as a rule, the path-planning algorithm will first define a number of ordered waypoints on the map which will have to be connected sequentially so as to form the path. Connecting the waypoints can be achieved in many different ways, with each one having its own advantages and drawbacks. However two main categories can be distinguished:

- Combining straight lines and arc segments.

- Using splines.

Regarding the first category, a simple and intuitive way of getting continuous paths without corners is to inscribe a circle between two lines in order to form a curved path. In 1957, Dubins showed that for a particle that moves forward with unity speed, the shortest possible path that meets a maximum curvature bound between a starting position with predefined orientation (starting pose) and a finishing position with predefined orientation (finishing pose) consists of at most three pieces, each of which is either a straight line or an arc of a circle of radius $R > 0$ [75]. [194] extended Dubins' result for a car-like vehicle by taking into

account backward motion, hence allowing to include cusps along the path. The Dubins car was further extended to include altitude, hence leading to the Dubins airplane [59]. A discrete analogue of Dubins' result using polygonal paths was proved by [78], although the authors state that different definitions of discrete curvature-constrained motion than the one they used might lead to shorter paths. [58] augmented Dubins' result by incorporating a directional-cost element in order to minimize the costs associated with the construction of underground mines. In [122], a modified Dubins problem was considered where a rectilinear path and direction, to which a vehicle should converge, was specified instead of a final point. The goal was to find the optimal 2D path for convergence to a rectilinear path for a vehicle that is initially located at any distance from the desired path. Moreover, [219], [17] and [19] deal with the problem of finding a Dubins path for a vehicle that moves in a constant drift field. In this case, it is not always possible to find a Dubins path.

The main disadvantage of the Dubins path is the curvature discontinuity which occurs at the meeting points of two consecutive path segments, for more details see [224]. This problem can be circumvented by employing a clothoid arc between a straight line and a circular arc [94]. The clothoid (also known as Euler spiral, Cornu spiral and spiro) is useful in path-planning applications due to its property of having its curvature change linearly with arc-length. This notion can be extended to three dimensions, and is consequently also true for the torsion [116]. Clothoids have also been used in formation control applications, see for instance [202] and [204]. The drawback of clothoids is that their coordinates do not have a closed-form expression since they involve computation of the Fresnel integrals.

The second category of methodologies pertains to connecting waypoints using splines. There is a vast literature on this subject, mostly due to the research stemming from the computer graphics community. Some approaches, such as the cubic Hermite spline interpolation (CHSI), result in continuous velocity paths with more tractable shape (without wiggling between the waypoints) but have a discontinuous curvature at the waypoint locations. Others, like the natural cubic splines, give curvature-continuous paths but the resulting shape can be very impractical and inefficient. An interesting alternative is to use Pythagorean-hodograph (PH) curves which are characterized by the special property that their parametric speed is a polynomial (or rational) function of the path parameter [81]. The PH curves were presented for the first time in [82] and the idea occurred while the authors were investigating the existence of planar or spatial curves that have natural parametrizations [83]. PH curves do not necessarily entail curvature-continuous paths, this issue was addressed by [46]. In addition, several other researchers have employed PH curves for path planning, see for instance [223], [203] and [101]. Interestingly, PH curves of monotone curvature have also been used to connect straight lines in a way such that the overall curve is curvature continuous [80].

Contrary to what might be the case in the field of computer graphics, where often the goal is to design aesthetically pleasing curves [79], in motion control scenarios it is preferable to assign paths that consist of straight lines and arc segments rather than splines, mainly because the latter implies that at least one of the control surfaces (a ship's rudder, for instance) will always be active, due to the relentlessly varying curvature. In addition, using straight lines and arc segments

makes it more likely and easier to ensure that the path will not include wiggles or zig-zags between two waypoints. This property, which is also related to *allowance* [68], can be critical when navigating in an area where obstacles are present. Last but not least, moving on straight lines enables adaptive techniques to provide a faster and more robust estimate of unknown external disturbances, such as ocean currents [91].

In this chapter we propose an alternative way of designing curvature-continuous paths, which belongs to the first of the two aforementioned categories. More specifically, Fermat's spiral (FS) is employed as a means to connect successive straight lines in a plane. The main motivation is that the curvature of FS is equal to zero at the origin, a property which makes it suitable for connecting it with a straight line without inducing curvature discontinuities, as is the case with circular arcs. Moreover, contrary to clothoids, FS involves very simple parametric equations that are easy to program and fast to compute. The use of FS for path-planning applications was initially studied by [68] and later implemented for building curvature-continuous and collision-free paths using Voronoi diagrams in [53]. Here we investigate the properties of FS and show that it can accomplish the task of generating curvature-continuous paths and therefore be used in path-following applications. Furthermore, we extend the work done by [68] and [53] in three ways: a) we propose a different parametrization which removes the speed singularity at the origin, hence making FS paths suitable also for path-tracking applications, b) we present a more thorough analysis regarding the calculation of the FS arc length, and c) we show that FS can be used successfully in combination with circular arcs in order to produce Dubins paths with FS arc transition.

5.2 Preliminaries

5.2.1 Basic Definitions

Similarly to [41], a planar path is considered to be a one-dimensional manifold that can be expressed by the set

$$\mathcal{P} := \left\{ \mathbf{p} \in \mathbb{R}^2 \mid \mathbf{p} = \mathbf{p}_p(\varpi) \ \forall \ \varpi \in \mathbb{R} \right\} \tag{5.1}$$

where ϖ is the path parameter and $\mathbf{p}_p(\varpi)$ denotes the position of a point belonging to the path. The generalized path parameter does not necessarily have any physical meaning. It can be the path length, but is usually more convenient to be defined within the unit domain:

$$\varpi \in [0, 1] \,, \tag{5.2}$$

because it can be difficult to construct an elegant basis for a general interval $[\varpi_i, \varpi_{i+1}]$. It is common practice to implement $\mathbf{p}(\cdot)$ as a piecewise-defined function, which reduces the function complexity but demands consideration at the transition points between subpaths. For planar paths consisting of a number of curve segments, each single curve segment can be expressed by the set [117]:

$$\begin{aligned} \mathcal{P}_i := \{ \mathbf{p}_i \in \mathbb{R}^2 \mid \\ \mathbf{p}_i = \mathbf{p}_{i,p}(\varpi) \forall \ \varpi \in \mathcal{I}_i = [\varpi_{i,0}, \varpi_{i,1}] \subset \mathbb{R} \}, \end{aligned} \tag{5.3}$$

and, consequently, the path can be written as a superset of n curve segments:

$$\mathcal{P}_s = \bigcup_{i=1}^{n} \mathcal{P}_i. \tag{5.4}$$

In this work, we consider two-dimensional curves:

$$\mathbf{p}_p(\varpi) = \begin{bmatrix} x_p(\varpi) \\ y_p(\varpi) \end{bmatrix}, \tag{5.5}$$

from which, the path-tangential (or, alternatively, course) angle is computed as:

$$\chi_p(\varpi) = \operatorname{atan2}(y_p'(\varpi), x_p'(\varpi)), \tag{5.6}$$

where $(\cdot)'$ denotes the first derivative w.r.t the path parameter ϖ. In contrast, the derivation w.r.t. time is denoted $(\dot{\cdot})$. The parametric path (5.5) is regular when:

$$|\mathbf{p}_p'(\varpi)| = \sqrt{\left(x_p'(\varpi)\right)^2 + \left(y_p'(\varpi)\right)^2} \neq 0. \tag{5.7}$$

The curvature of the regular path is given by the expression:

$$\kappa = \frac{x_p'(\varpi)y_p''(\varpi) - y_p'(\varpi)x_p''(\varpi)}{\sqrt[3]{\left(x_p'(\varpi)\right)^2 + \left(y_p'(\varpi)\right)^2}}. \tag{5.8}$$

Naturally, a minimum requirement is that subsequent path segments should be connected. The level of connection can be described by the path smoothness. Parametric continuity is denoted C^n, where n is the degree of smoothness, and implies that the first n parametric derivatives match at the common point of two subsequent paths [21]. Similarly, geometric continuity is denoted G^n, and is a relaxed form of parametric continuity, which does not impose constraints on the parameter's speed along the path. For many applications (such as path following of underactuated vehicles) it is sufficient to have geometric continuity, while for others (such as path tracking) parametric continuity is required. For more information on path smoothness as well as other path evaluation criteria, the reader is referred to [68] and [156].

Given a number of ordered points on a plane, it is useful to distinguish the following two types of curves:

- Interpolating curves, which pass through all the given points.
- Approximating curves, which do not pass through all the given points.

This categorization helps describe more accurately methods that use the same building blocks but have different properties. For example, both the Dubins paths and the paths resulting from circular smoothing [88, Sec. 10.3.1] use straight lines and circular arcs, however they result in different paths.

5.2.2 Piecewise Linear Path

The parametric form of a straight line connecting two points in a plane is given by [41]:

$$\mathbf{p}_{\text{line}}(\varpi) = \begin{bmatrix} N_0 + L\varpi\cos(\chi_l) \\ E_0 + L\varpi\sin(\chi_l) \end{bmatrix}, \tag{5.9}$$
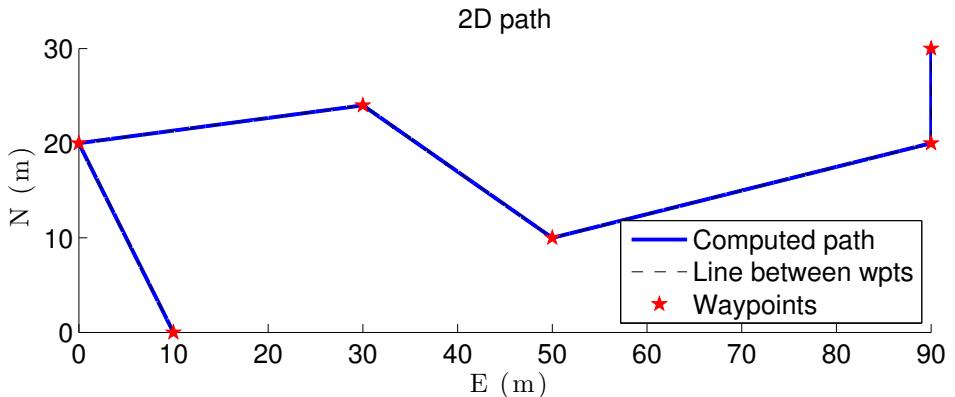
Figure 5.1: Piecewise linear path

where $\mathbf{p}_0 = [N_0, E_0]^T$ is the starting point, L is the length of the path, and χ_l is the path-tangential angle of the straight line. An example of a path consisting only of straight lines can be seen in Fig. 5.1. As it can be observed in Fig. 5.2, the path tangential angle is discontinuous and the curvature cannot be defined at the waypoint locations. Such a path cannot be followed by an underactuated vehicle and should only be assigned to a fully-actuated (3-DOF in the planar case) vehicle, such as a wheeled mobile robot. Even in that case, though, the robot must stop at the waypoint locations and then change its heading so as to adjust its attitude according to the direction of the next linear segment before starting to move along it. This problem can be alleviated by using circular segments to smooth the path.

5.2.3 Circular Arcs

Constructing a path using linear and circular arc segments is an extensively studied and popular approach which also leads to the shortest feasible path, according to Dubins' result [75]. Two path variations are probably the most common ones in this case, and their differences occur due to the different locations where the circular segments are placed. The first variation defines an interpolating curve that passes through all the waypoints (that is, the Dubins path), while the second defines an approximating curve that passes only through the first and last waypoint (also known as circular smoothing path), see [37] and [88, Sec. 10.3]. Both are G^1 paths, but the final geometry is different. A circular arc can be parametrized as follows:

$$\mathbf{p}_{\text{cir}}(\varpi) = \begin{bmatrix} c_N + R\cos(\alpha_0 + \varpi(\alpha_1 - \alpha_0)) \\ c_E + R\sin(\alpha_0 + \varpi(\alpha_1 - \alpha_0)) \end{bmatrix}, \qquad (5.10)$$

where $\mathbf{c} = [c_N, c_E]^T$ and R are the center and radius of the circle which the arc is a segment of, while α_0 and α_1 are the angles at which the arc starts and ends, respectively. Fig. 5.3 shows a straight-line path smoothened by circular arcs. The circular arc radius R is chosen equal to that of the vehicle's minimum turning
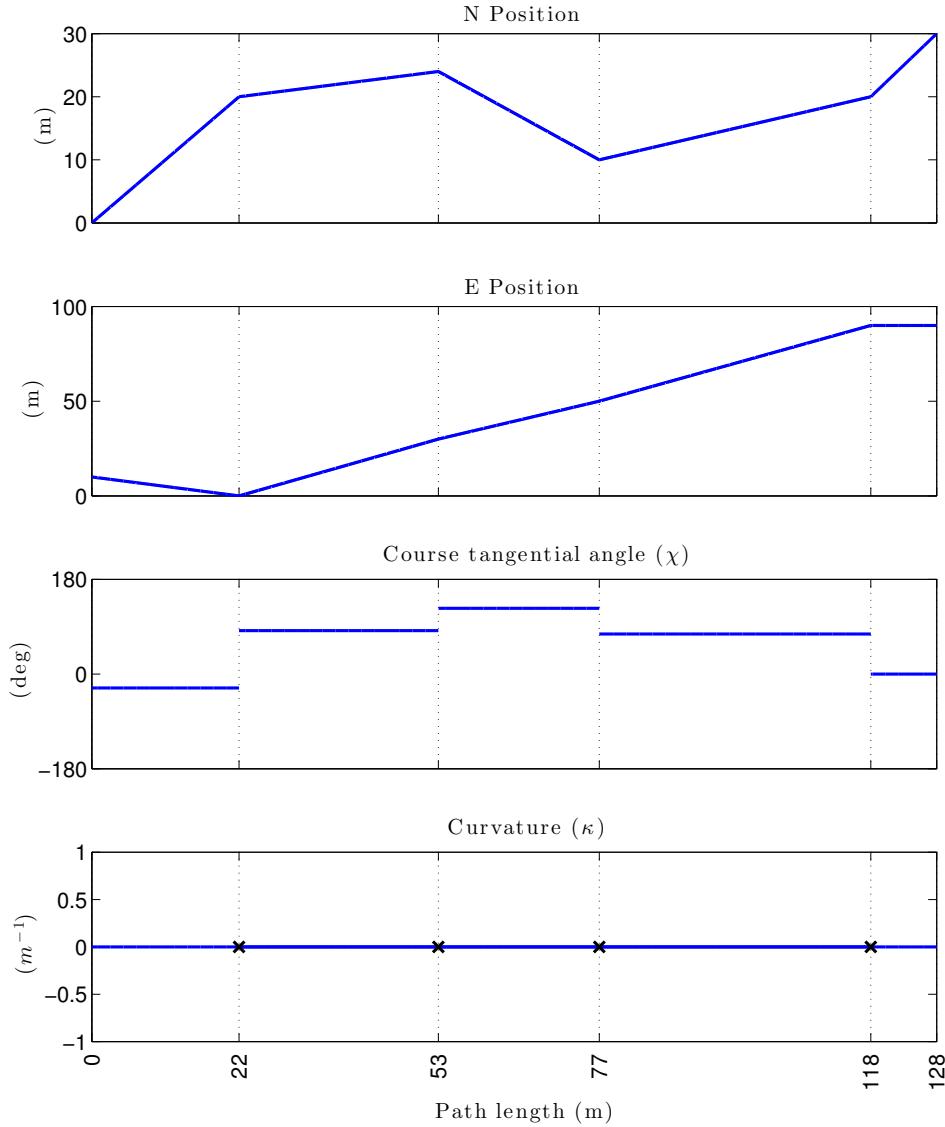
Figure 5.2: Piecewise linear path geometric properties

circle (or, the sharpest turn achievable by the vehicle), which corresponds to the curvature constraint:

$$\kappa_{\max} = \frac{1}{R_{\min}}. \tag{5.11}$$

It can be seen in Fig. 5.4 that the course tangential angle is continuous but the curvature is discontinuous at the locations where the transition between a straight line and a circular arc occurs, which indicates that the path is G^1. The curvature
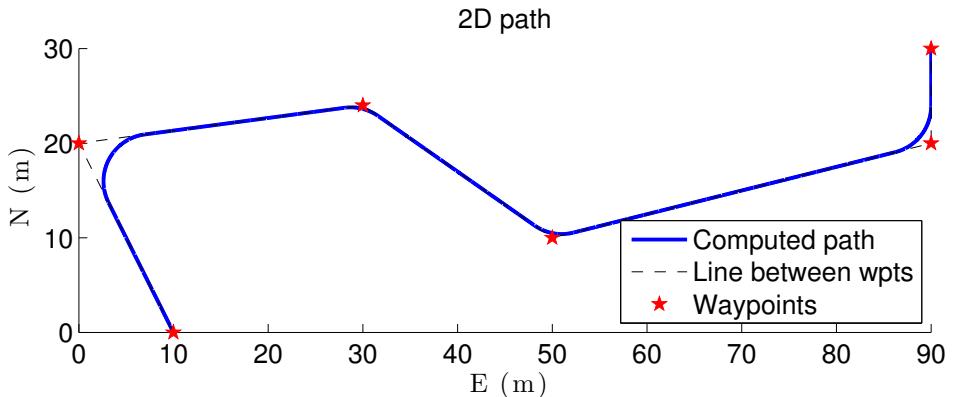
Figure 5.3: Circular smoothing path

discontinuity entails a discontinuity in the desired lateral acceleration of the vehicle because of the relationship [224]:

$$|\boldsymbol{\alpha}| = |\mathbf{u}|^2 \kappa, \tag{5.12}$$

where $\boldsymbol{\alpha}$ is the lateral acceleration vector and $\mathbf{u}$ the velocity vector. This will affect the input to the heading controller and the vehicle performance in general. Finally, from Figs 5.2 and 5.4 it can be observed that the path using circular arcs is shorter. This agrees with intuition since the circular arc operates as a shortcut and avoids going all the way to the second waypoint.

5.2.4 Clothoid

The clothoid is a curve with linear increase in curvature along its length and thus an appealing transition curve. Its parametrization is

$$\mathbf{p}_{cl}(\varpi) = \left[\begin{array}{c} x_0 + \int_0^\varpi \cos\left(\frac{1}{2}c\xi^2 + \kappa_0\xi + \chi_0\right) d\xi \\ y_0 + \int_0^\varpi \sin\left(\frac{1}{2}c\xi^2 + \kappa_0\xi + \chi_0\right) d\xi \end{array} \right], \tag{5.13}$$

where $\mathbf{p}_0 = [x_0, y_0]^{\mathrm{T}}$ is the initial position, c is a sharpness coefficient describing the increase in curvature, κ_0 is the curvature at $\mathbf{p}_0$, χ_0 is the course angle at $\mathbf{p}_0$ and ξ is a dummy integration variable. Unfortunately, the integrals of (5.13) have no analytic solution and a numerical approach must be employed.

There are two ways clothoids can be used as transition curves. The first one is to replace the circular arc with a clothoid. Since the clothoid has an increasing curvature w.r.t. the path length, it is necessary to use two clothoid segments: one with starting point at the end of the first straight line and ending point at the middle of the curved segment (this is the part with increasing curvature), and one with starting point at the middle of the curved segment and ending point at the beginning of the second straight line (this is the part with decreasing curvature). The second clothoid segment is actually a mirrored version of the first one, therefore
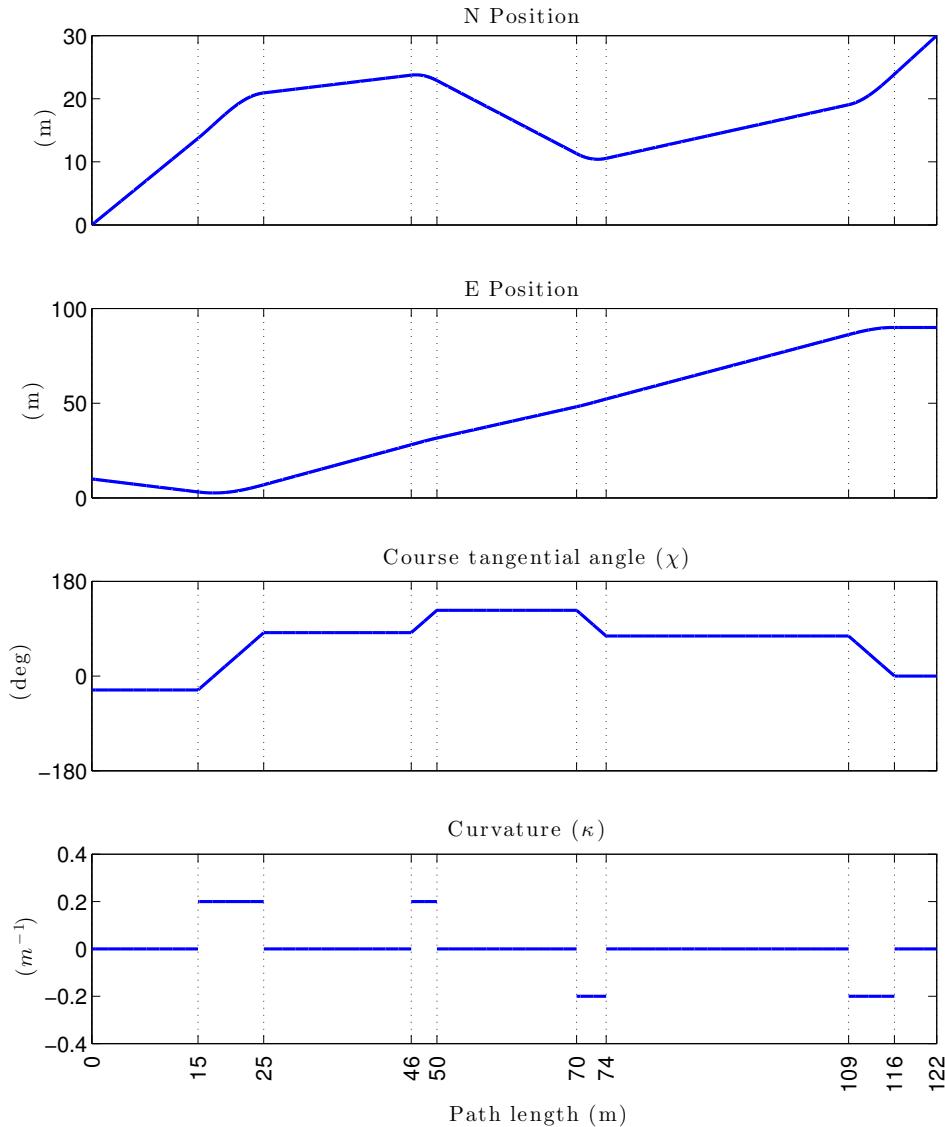
Figure 5.4: Circular smoothing path geometric properties

the computation needs to be done only once. We will refer to this as clothoid smoothing. The second approach is to use the clothoid as a transition curve between the linear segment and the circular arc. To achieve curvature continuity, that is G^2 continuity, the transition curve must have zero curvature at one end and R^{-1} at the other. Similarly to clothoid smoothing, a mirrored curve is necessary for the transition between the circular arc and the second straight line. We will refer to this as circular smoothing with clothoid transition. Figs 5.5–5.6 show a clothoid
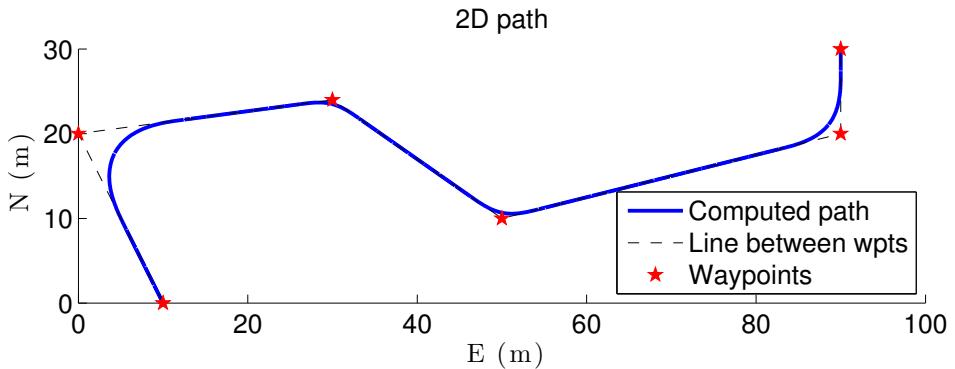
Figure 5.5: Clothoid smoothing path

smoothing path and its geometrical properties, respectively. In this case, the length of the path is even shorter, when compared to the circular smoothing case. This is related to the different wheel over and pull out points of the arc in each case. In accordance with what has been mentioned earlier, this is an approximating path.

The computational cost of (5.13) is the main motivation for investigating alternative solutions. One such alternative is Fermat's spiral, and the contribution of this article is the application of such a spiral for path smoothing.

5.3 Fermat's Spiral

5.3.1 Introduction

The term *Archimedean spirals* refers to a family of curves described by the equation [228]:

$$r = k\theta^{1/n}, \tag{5.14}$$

where r is the radial distance, θ is the polar angle, k is a scaling constant, and n determines how tightly the spiral is wrapped. The Fermat's spiral (FS) is an Archimedean spiral with $n = 2$ and therefore given by the equation:

$$r = k\sqrt{\theta}. \tag{5.15}$$

The spiral curvature is [229]:

$$\kappa\left(\theta\right) = \frac{1}{k} \frac{2\sqrt{\theta}\left(3 + 4\theta^2\right)}{\left(1 + 4\theta^2\right)^{\frac{3}{2}}}, \tag{5.16}$$

which gives $\kappa\left(0\right) = 0$ and $\kappa\left(\theta\right) > 0$ for all $\theta > 0$. Fig. 5.7 shows a full rotation of (5.15), and Fig. 5.8 shows a plot of the curvature for a full rotation with an asterisk indicating the value of θ corresponding to the maximum curvature. The curve was first studied by Fermat in 1636 and published in [84].
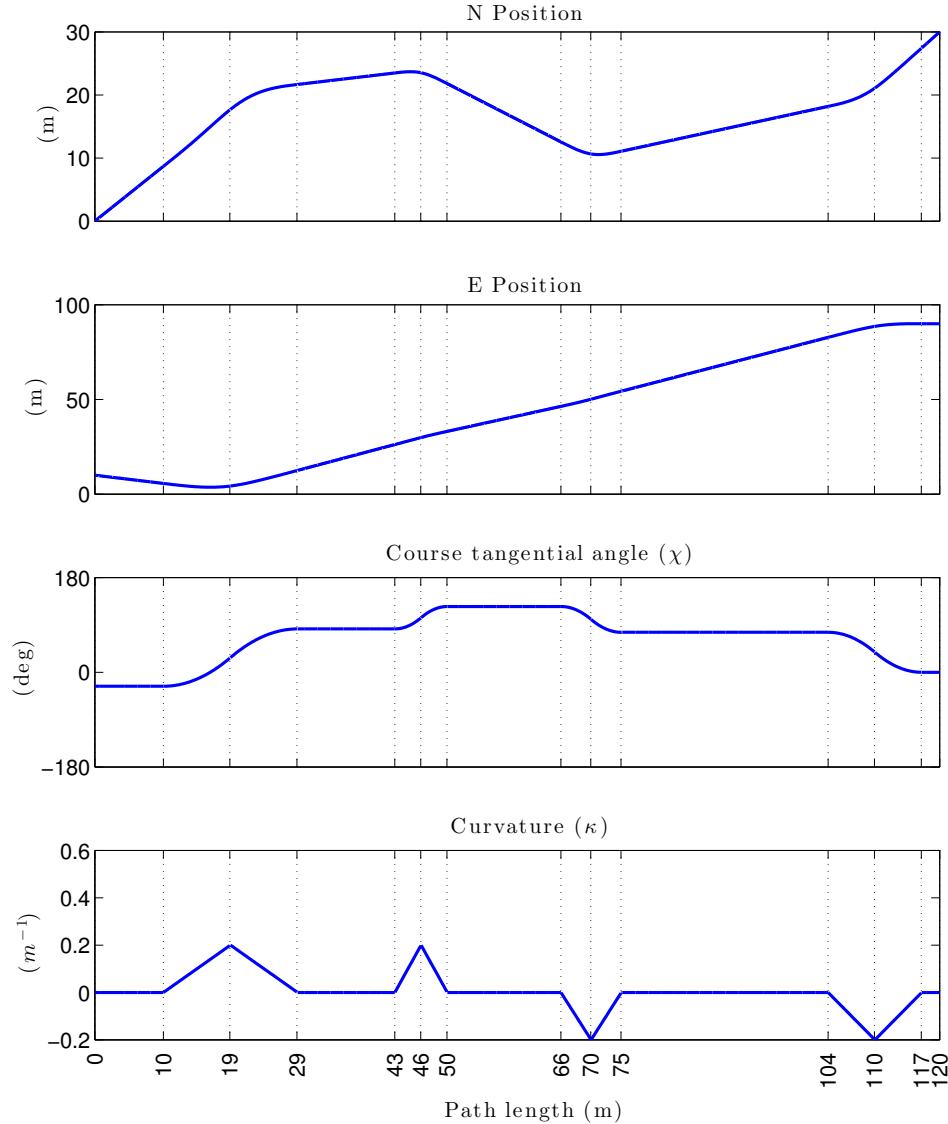
Figure 5.6: Clothoid smoothing path geometric properties
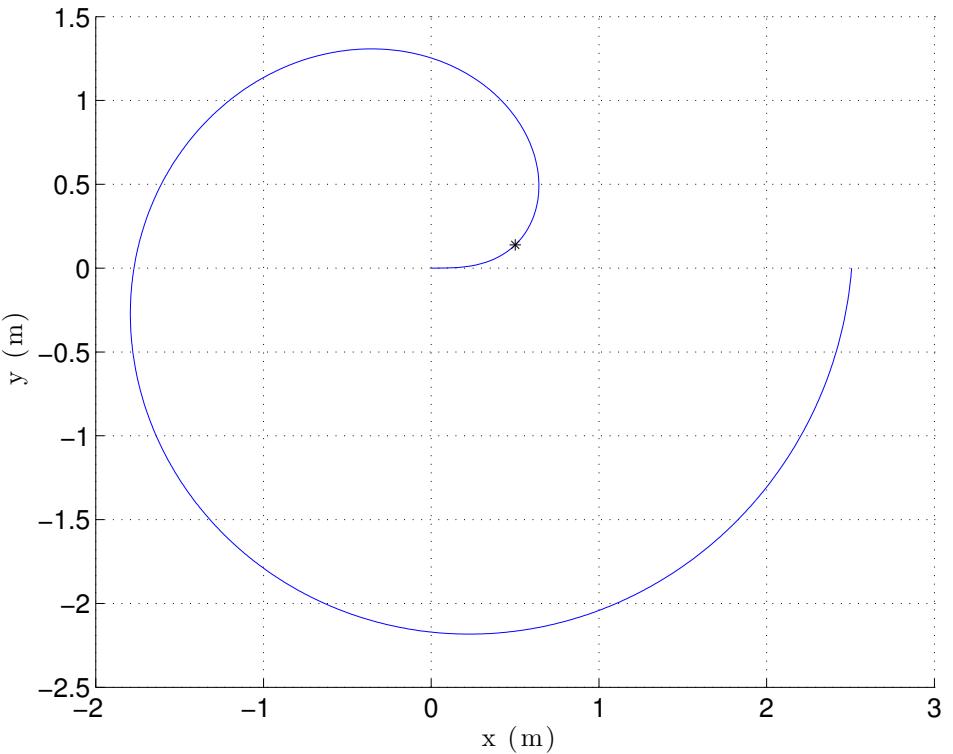
Figure 5.7: Fermat's spiral, one complete rotation. The asterisk marks the point where the curve has the maximum curvature

5.3.2 Cartesian Parametrization

A Cartesian parametrization of (5.15) is [68]:

$$
\left[\begin{array}{c} x \\ y \end{array} \right] = \left[\begin{array}{c} r\cos(\theta) \\ r\sin(\theta) \end{array} \right] = \left[\begin{array}{c} k\sqrt{\theta}\cos(\theta) \\ k\sqrt{\theta}\sin(\theta) \end{array} \right]. \tag{5.17}
$$

To allow for different initial positions $\mathbf{p}_0 = [x_0, y_0]^{\mathrm{T}}$, turning both left and right, and different initial tangent angles χ_0, the following parametrization is proposed:

$$
\mathbf{p}_{\mathrm{FS}}\left(\theta\right) = \left[\begin{array}{c} x_0 + k\sqrt{\theta}\cos\left(\rho\theta + \chi_0\right) \\ y_0 + k\sqrt{\theta}\sin\left(\rho\theta + \chi_0\right) \end{array} \right], \tag{5.18}
$$

where

$$
\rho = \left\{ \begin{array}{ll} 1 & \text{for an anti-clockwise turn, and} \\ -1 & \text{for a clockwise turn.} \end{array} \right. \tag{5.19}
$$

The domain in (5.18) is $\theta \in [0, \theta_{\mathrm{end}}]$, where θ_{end} is the parameter corresponding to the spiral end point, yet to be determined. This parametrization is thus neither
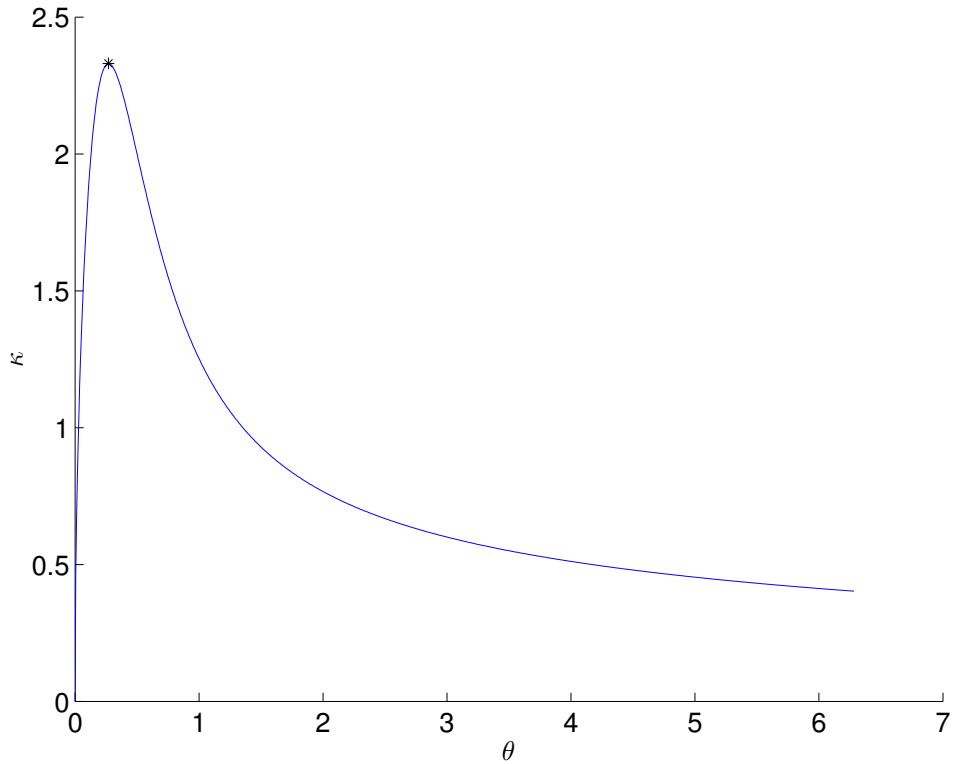
Figure 5.8: Fermat's spiral curvature. The asterisk marks the value of θ for which the curve has the maximum curvature.

unit domain nor unit speed, but it can be made unit domain by setting $\theta = \varpi\theta_{\text{end}}$. Fig. 5.9 shows how changing χ_0 affects the orientation of the spiral and Fig. 5.10 demonstrates the effect of the scaling constant k.

As was the case with the clothoid in Sec. 5.2.4, we will distinguish between FS smoothing and circular smoothing with FS transition. This implies that a mirrored version of (5.18) is needed to describe a curve of curvature decreasing to zero. The following is proposed:

$$\mathbf{p}_{\overline{\text{FS}}}(\theta) = \begin{bmatrix} x_{\text{end}} + k\sqrt{\theta_{\text{end}} - \theta}\cos\left(\rho\left(\theta - \theta_{\text{end}}\right) + \chi_{\text{end}}\right) \\ y_{\text{end}} + k\sqrt{\theta_{\text{end}} - \theta}\sin\left(\rho\left(\theta - \theta_{\text{end}}\right) + \chi_{\text{end}}\right) \end{bmatrix}, \quad (5.20)$$

where $\mathbf{p}_{\text{end}} = [x_{\text{end}}, y_{\text{end}}]^{\text{T}}$ is the position at the end of the curve, i.e. $\mathbf{p}_{\text{end}} = \mathbf{p}_{\overline{\text{FS}}}(\theta_{end})$, and χ_{end} is the course angle at that point.
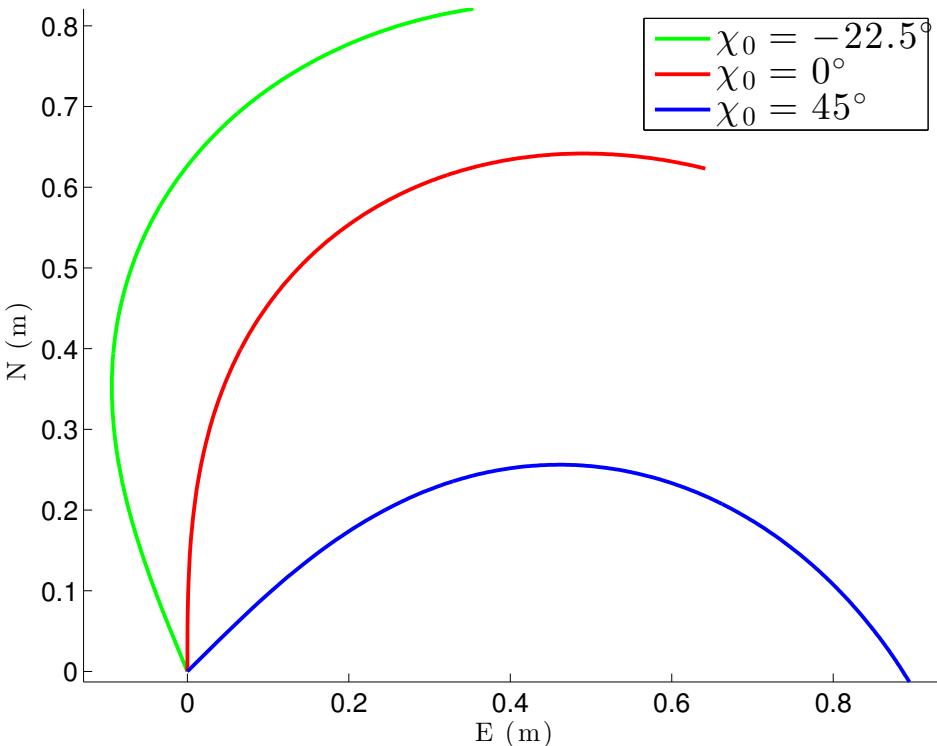
Figure 5.9: Fermat's spiral segment with fixed scaling (k=1) and different initial tangent angles

5.3.3 Parametrized Speed and Acceleration

The first derivative of (5.18) is [68]:

$$\frac{d}{d\theta}\mathbf{p}_{\mathrm{FS}}(\theta) \quad = \quad \frac{k}{2\sqrt{\theta}}\left[\begin{array}{c} \cos\left(\rho\theta + \chi_0\right) - 2\rho\theta\sin\left(\rho\theta + \chi_0\right) \\ \sin\left(\rho\theta + \chi_0\right) + 2\rho\theta\cos\left(\rho\theta + \chi_0\right) \end{array}\right], \quad (5.21)$$

and the second derivative is

$$\frac{d^2}{d\theta^2}\mathbf{p}_{\mathrm{FS}}(\theta) \quad = \quad \frac{-k}{4\theta^{\frac{3}{2}}}\left[\begin{array}{c} \left(4\theta^2 + 1\right)\cos\left(\rho\theta + \chi_0\right) + 4\rho\theta\sin\left(\rho\theta + \chi_0\right) \\ \left(4\theta^2 + 1\right)\sin\left(\rho\theta + \chi_0\right) - 4\rho\theta\cos\left(\rho\theta + \chi_0\right) \end{array}\right]. \quad (5.22)$$

Similarly, for the mirrored curve (5.20), the first and second derivatives are given by (5.34)–(5.35). Equations (5.21)–(5.22) and (5.34)–(5.35) have a singularity at the beginning of the path, i.e. $\theta = 0$. This is a drawback since the velocity and acceleration at the beginning and end of the path are undefined, a property that makes the Fermat spiral segment initially appear unsuitable for path-tracking applications in its current form. However, this is a property of the parametrization (that is, the kinematics and not the geometry of the curve) and it is possible to
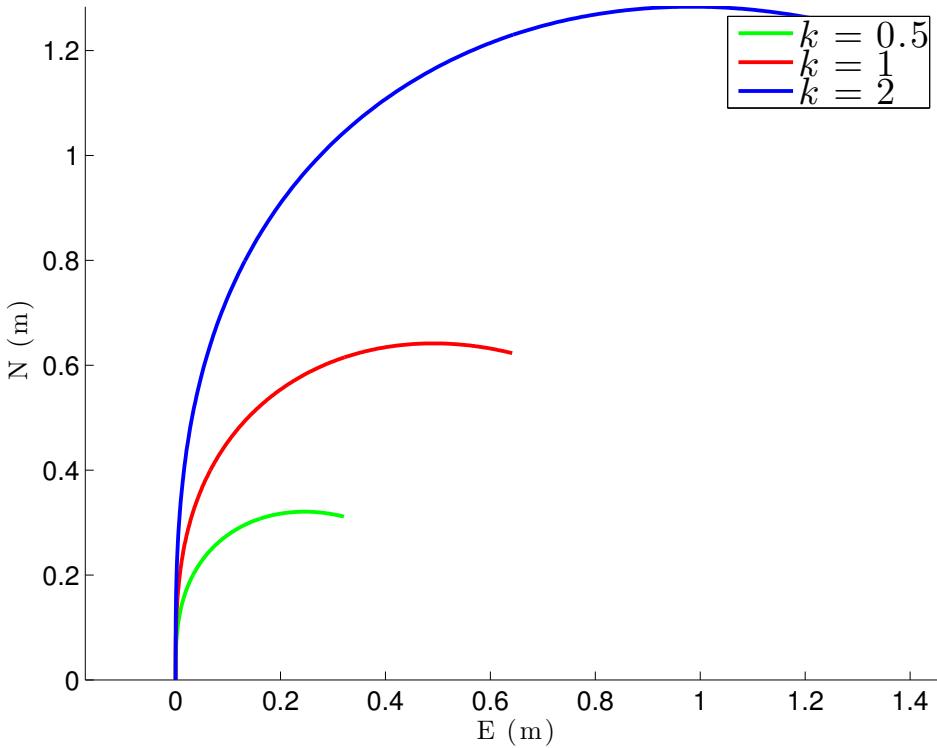
Figure 5.10: Fermat's spiral with fixed initial tangent angle ($\chi_0 = 0$) and different scaling constants

remedy the problem by changing variables, as was discussed by [192]. Given that the problem is caused by the term $\sqrt{\theta}$ in the denominator and that the Cartesian

$$
\frac{d}{d\theta}\mathbf{p}_{\overline{\mathrm{FS}}}(\theta) = -\frac{k}{2\sqrt{\theta_{\mathrm{end}} - \theta}} \left[\begin{array}{c} \cos\left(\rho\left(\theta - \theta_{\mathrm{end}}\right) + \chi_0\right) + 2\rho\left(\theta_{\mathrm{end}} - \theta\right)\sin\left(\rho\left(\theta - \theta_{\mathrm{end}}\right) + \chi_0\right) \\ \sin\left(\rho\left(\theta - \theta_{\mathrm{end}}\right) + \chi_0\right) + 2\rho\left(\theta_{\mathrm{end}} - \theta\right)\cos\left(\rho\left(\theta - \theta_{\mathrm{end}}\right) + \chi_0\right) \end{array} \right]
$$
(5.34)

$$
\frac{d^2}{d\theta^2}\mathbf{p}_{\overline{\mathrm{FS}}}(\theta) =
$$
$$
-\frac{k}{4\left(\theta_{\mathrm{end}} - \theta\right)^{\frac{3}{2}}} \left[\begin{array}{c} \left(4\left(\theta - \theta_{\mathrm{end}}\right)^2 + 1\right)\cos\left(\rho\left(\theta - \theta_{\mathrm{end}}\right) + \chi_0\right) - 4\rho\left(\theta_{\mathrm{end}} - \theta\right)\sin\left(\rho\left(\theta - \theta_{\mathrm{end}}\right) + \chi_0\right) \\ \left(4\left(\theta - \theta_{\mathrm{end}}\right)^2 + 1\right)\sin\left(\rho\left(\theta - \theta_{\mathrm{end}}\right) + \chi_0\right) + 4\rho\left(\theta_{\mathrm{end}} - \theta\right)\cos\left(\rho\left(\theta - \theta_{\mathrm{end}}\right) + \chi_0\right) \end{array} \right]
$$
(5.35)
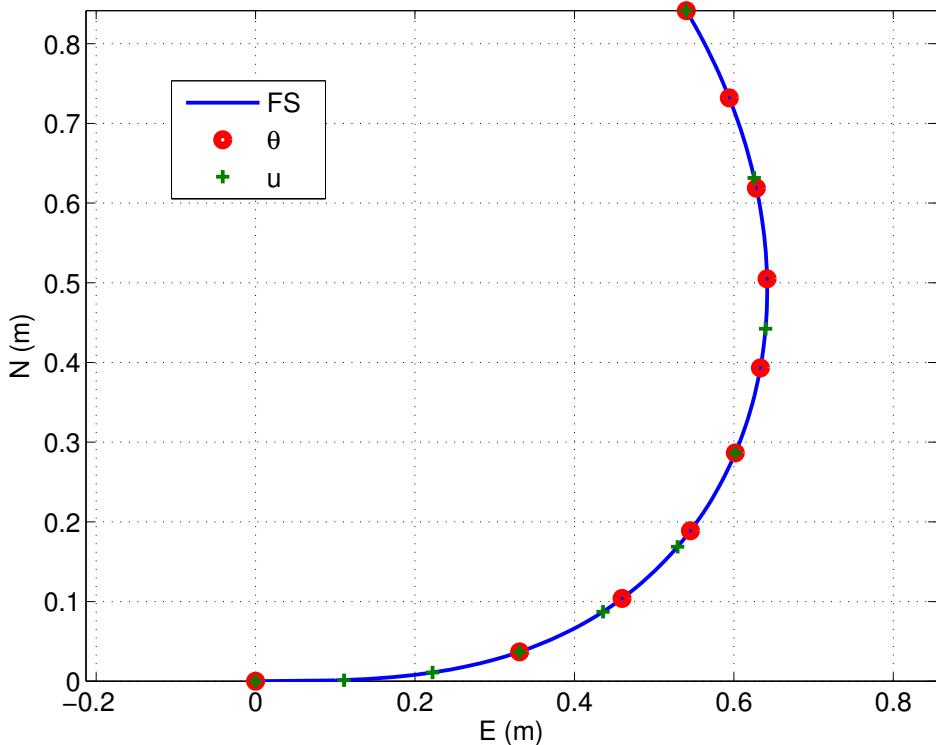
Figure 5.11: Illustration of the two different parametrizations. The geometry of the curve remains unaffected.

parametrization of the FS resembles that of a circle, we change variables as follows:

$$u := \sqrt{\theta} \Rightarrow 0 \leq u \leq \sqrt{\theta_{\max}}, \tag{5.23}$$

and then (5.18) is written:

$$\mathbf{p}_{\text{FS}}(u) = \begin{bmatrix} x_0 + ku\cos(\rho u^2 + \chi_0) \\ y_0 + ku\sin(\rho u^2 + \chi_0) \end{bmatrix}, \tag{5.24}$$

which gives the singularity-free expression for the velocity:

$$\frac{d}{du}\mathbf{p}_{\text{FS}}(u) = k\begin{bmatrix} \cos(\rho u^2 + \chi_0) - 2\rho u^2\sin(\rho u^2 + \chi_0) \\ \sin(\rho u^2 + \chi_0) + 2\rho u^2\cos(\rho u^2 + \chi_0) \end{bmatrix}. \tag{5.25}$$

It is necessary to show that (5.24) results in a regular curve. To this end, we employ the amplitude-phase expressions

$$a\sin(x) + b\cos(x) = \sqrt{a^2 + b^2}\sin(x + \phi), \tag{5.26}$$

$$a\cos(x) + b\sin(x) = \sqrt{a^2 + b^2}\sin(x - \phi), \tag{5.27}$$

53
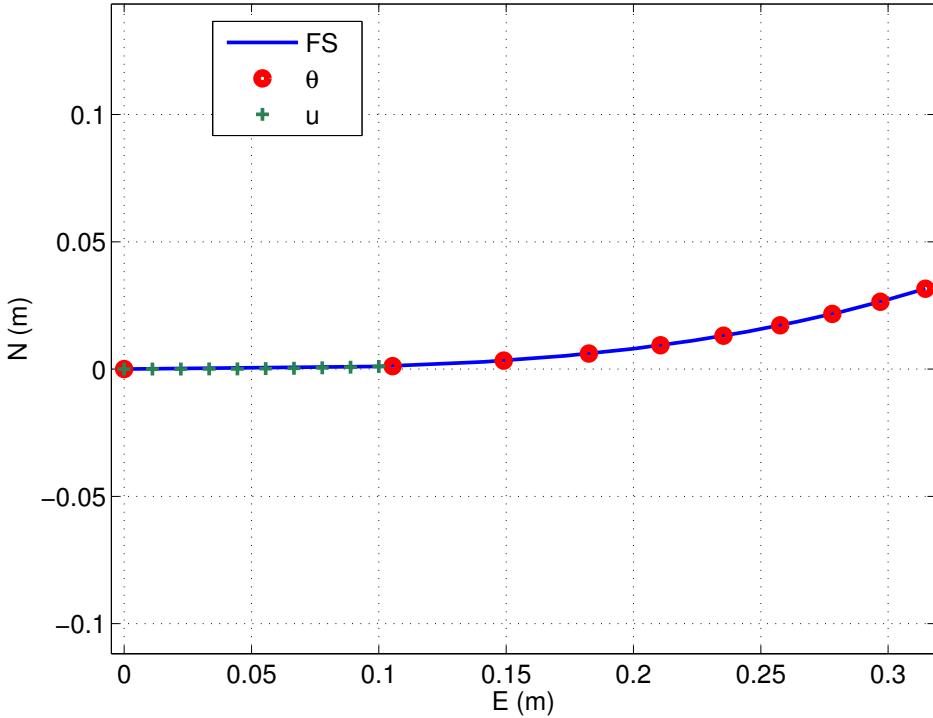
Figure 5.12: Illustration of the two different parametrizations close to the origin. The discontinuity of θ is more evident here.

with $\phi = \mathrm{atan2}(b, a)$, and rewrite (5.25) as:

$$\frac{d}{du}\mathbf{p}_{\mathrm{FS}}(u) = k\sqrt{1 + (2\rho u^2)^2} \begin{bmatrix} \sin\left(\rho u^2 + \chi_0 - \phi\right) \\ \cos\left(\rho u^2 + \chi_0 - \phi\right) \end{bmatrix}, \quad (5.28)$$

with $\phi = \mathrm{atan2}(1, 4\rho^2 u^4)$. For a regular curve it is required that:

$$\left| \frac{d}{du}\mathbf{p}_{\mathrm{FS}}(u) \right| \neq 0. \quad (5.29)$$

For (5.28) we have that:

$$\left| \frac{d}{du}\mathbf{p}_{\mathrm{FS}}(u) \right| = k\sqrt{2}\sqrt{1 + 4\rho^2 u^4}, \quad (5.30)$$

which guarantees that (5.29) is satisfied. As a consequence, the FS can be used for path-tracking applications. In that case, the derivative of the parameter w.r.t time has to be defined. According to [36] and [88, Sec. 10.4], for a desired vehicle speed $U_d(t)$, the parameter derivative w.r.t time is determined by:

$$\dot{u}(t) = \frac{U_d(t)}{\sqrt{x_p'(u)^2 + y_p'(u)^2}}, \quad (5.31)$$

and from (5.30)–(5.31):

$$\dot{u} = \frac{U_d}{k\sqrt{2}\sqrt{1 + 4u^4}},$$

(5.32)

where we used that $\rho^2 = 1$. Fig. 5.11 shows the image of an FS segment when each of the parameters θ and u is sampled at 10 equispaced points.

5.3.4 Path Length

One, often important, consideration when designing paths is the length of the path, an example can be found in [129]. Before moving on, it should be noted that we assume no external disturbances acting on the vehicle, therefore the path length is discussed here from a geometric point of view. As mentioned in Sec. 5.1, the Dubins path is the shortest one between two poses in a plane if we consider a unit speed vehicle with bounded curvature (alternatively, turning radius). Calculating the total length of a Dubins path is trivial since the expressions for the length of straight lines and circular arcs are well known and easy to compute.

Despite their complex Cartesian parametrization, the clothoids have the nice property of a curvature linearly varying w.r.t. the path parameter. As a result, there is a linear relationship between the curvature and the clothoid arc length, meaning that the clothoid arc length is easy to compute.

Computing the length of an FS arc is not as trivial as the aforementioned cases because (using the parametrization of Sec. 5.3.3) it is given by the integral:

$$L_{\mathrm{FS}} = k\sqrt{2} \int_0^{u_{\max}} \sqrt{1 + 4u^4}\,du,$$

(5.35)

which has no closed-form solution. The path length can be expressed as a Gaussian hypergeometric function:

$$L_{\mathrm{FS}}(\theta) = k\sqrt{\theta}\,\, _2F_1\left(-\frac{1}{2}, \frac{1}{4}; \frac{5}{4}; -4\theta^2\right),$$

(5.36)

see [229]. It should be noted that, using the parametrization of Sec. 5.3.3, (5.36) is equivalent to:

$$L_{\mathrm{FS}}(u) = ku\,\, _2F_1\left(-\frac{1}{2}, \frac{1}{4}; \frac{5}{4}; -4u^4\right).$$

(5.37)

According to [1, Ch. 15], the circle of convergence of the Gauss hypergeometric series

$$_2F_1\left(a, b; c; z\right) = \sum_{n=0}^{\infty} \frac{(a)_n (b)_n}{(c)_n} \frac{z^n}{n!}$$

(5.38)

is the unit circle $|z| < 1$, which corresponds to $\theta < 1/2$, or a course change $\Delta\chi < 73.6479°$. The behavior of the series (5.38) on its circle of convergence is *absolute convergence* when $\mathrm{Re}(c-a-b) > 0$, which is true for (5.36)–(5.37) since $5/4+1/2-1/4 = 3/2$. The closer to zero z is, the faster and more accurate the computation of (5.38) becomes.
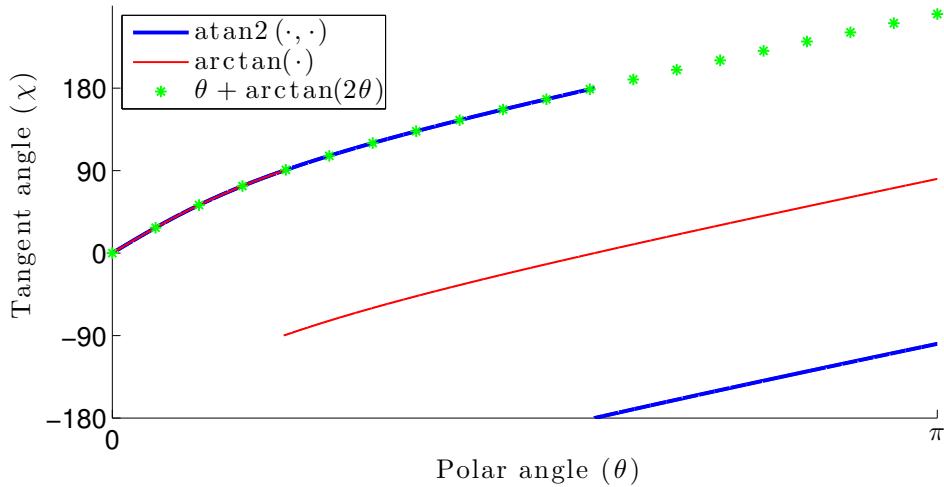
Figure 5.13: Course tangent angle. Only the expression $\theta + \arctan(2\theta)$ is continuous.

The restriction mentioned above regarding the circle of convergence does not forbid us to compute the length of an FS arc for course changes equal to and higher than $73.6479°$. As it can be seen in [188, Sec. 4.6], for $z \in \mathbb{R}$ it is possible to use appropriate transformation formulae in order to map z to a new variable $w \in [0, 1/2]$, hence resulting in a fast and accurate computation for all z, and thus θ.

To conclude, the computation of the length of an FS arc is not as straightforward as that of a circular, or even clothoid, arc. On the positive side, the FS length is given by a hypergeometric series that always converges and is easy to compute.

5.3.5 Course Angle

The course, or direction, along the curve can be determined by (5.6). It is well-defined since the singular terms of (5.21) cancel. Moreover [68]:

$$\chi(\theta) = \arctan\left(\frac{\sin(\theta) + 2\theta\cos(\theta)}{\cos(\theta) - 2\theta\sin(\theta)}\right), \tag{5.39}$$

or, expressed with the two-argument arctangent function,

$$\chi(\theta) = \arctan 2\left(\begin{array}{c} \sin(\theta) + 2\theta\cos(\theta), \\ \cos(\theta) - 2\theta\sin(\theta) \end{array}\right). \tag{5.40}$$

We rewrite (5.39) as follows:

$$\chi(\theta) = \int\left(\frac{d}{d\theta}\arctan\left(\frac{\sin(\theta) + 2\theta\cos(\theta)}{\cos(\theta) - 2\theta\sin(\theta)}\right)\right)d\theta,$$
$$= \int\frac{4\theta^2 + 3}{4\theta^2 + 1}d\theta, \tag{5.41}$$

where, finally, the equations are continuous and equal to:

$$\chi(\theta) = \theta + \arctan(2\theta) \tag{5.42}$$

up to respectively $\chi = 90°$ and $\chi = 180°$. The three expressions (5.39)–(5.40) and (5.42) are plotted in Fig. 5.13 for comparison. In practice, a 180° course change is considered sufficient for one curve because a full rotation is achieved when combined with a mirrored curve.

5.4   Fermat Spiral Path Design

5.4.1   Problem Description

In this section we present a methodology for using the FS in order to construct curvature-continuous paths. Given a piecewise linear path, the vertices can be smoothened by two Fermat curves, one entering and one exiting (mirrored) curve. The vertices then become curvature continuous (G^2) and respect a given curvature constraint $\kappa_{\max}$.

Without loss of generality, we assume the initial piecewise linear path consists of two straight lines connecting three sequential waypoints ($\mathbf{wpt}_{i-1}, \mathbf{wpt}_i, \mathbf{wpt}_{i+1}$) on the plane, see Fig. 5.14. Initially, the course change direction ρ must be determined, for this reason it is useful to define two normalized vectors parallel to the straight lines (see Fig. 5.15):

$$\mathbf{v}_{\text{in}} = \frac{\mathbf{wpt}_i - \mathbf{wpt}_{i-1}}{\|\mathbf{wpt}_i - \mathbf{wpt}_{i-1}\|}, \tag{5.43}$$

$$\mathbf{v}_{\text{out}} = \frac{\mathbf{wpt}_{i+1} - \mathbf{wpt}_i}{\|\mathbf{wpt}_{i+1} - \mathbf{wpt}_i\|}. \tag{5.44}$$

Then, the course change magnitude can be computed as follows:

$$|\Delta\chi| = \arccos(\mathbf{v}_{\text{in}} \cdot \mathbf{v}_{\text{out}}), \tag{5.45}$$

and the course change direction:

$$\rho = -\text{sign}(v_{\text{in,y}}v_{\text{out,x}} - v_{\text{in,x}}v_{\text{out,y}}). \tag{5.46}$$

5.4.2   Domain Determination

The domain of θ in (5.18) must be set according to the desired course change from the beginning to the end of the curve. Similarly, the scaling constant k must be set to respect the curvature constraint $\kappa_{\max}$.

The domain's upper limit θ_{end} determines the course change. Unfortunately, (5.42) is not invertible, thus it is not possible to determine the θ_{end} needed to achieve a desired course χ analytically.

However, (5.42) is continuous and differentiable, even beyond $\chi = 180°$ and numerical solutions are therefore possible. This problem was investigated in [68] where

it was concluded that Halley's method (or, equivalently, Householder's method of order 2), given by:

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2(f'(x_n))^2 - f(x_n)f''(x_n)},$$

(5.47)

was the most efficient. More specifically, Halley's method was sufficiently efficient for tolerance equal to 10^{-3} after only one step.

5.4.3 Scaling Determination

Once the domain of (5.18) is determined, the scaling constant k must be set such that the curvature constraint $\kappa_{\max}$ is satisfied all over the curve. As given by (5.16), the curvature reaches its maximum at:

$$\theta = \sqrt{\frac{\sqrt{7}}{2} - \frac{5}{4}}.$$

(5.48)

In the case where the domain does not include this point, the maximum curvature is at θ_{end}. Thus, the parameter corresponding to the maximum curvature is

$$\theta_{\kappa_{\max}} = \min\left(\theta_{\text{end}}, \sqrt{\frac{\sqrt{7}}{2} - \frac{5}{4}}\right).$$

(5.49)

Solving (5.16) for k yields that, in order for the maximum curvature along the curve to be the same as the curvature constraint, i.e. $\kappa\left(\theta_{\kappa_{\max}}\right) = \kappa_{\max}$, it is necessary with

$$k = \frac{1}{\kappa_{\max}} \frac{2\sqrt{\theta_{\kappa_{\max}}}\left(3 + 4\theta_{\kappa_{\max}}^2\right)}{\left(1 + 4\theta_{\kappa_{\max}}^2\right)^{\frac{3}{2}}}.$$

(5.50)
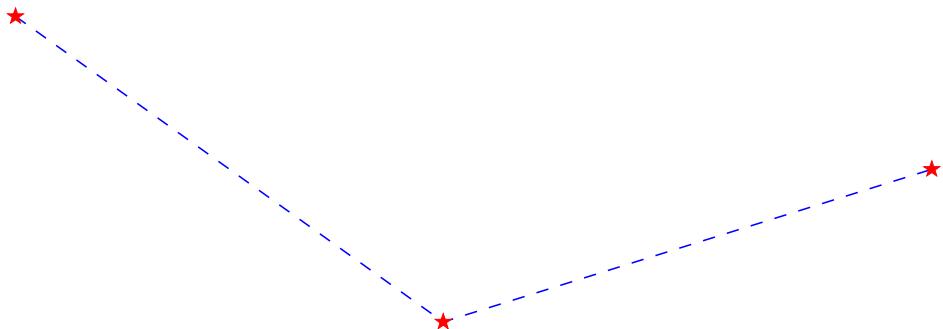
Figure 5.14: Initial path of two straight lines connecting three sequential waypoints on the plane
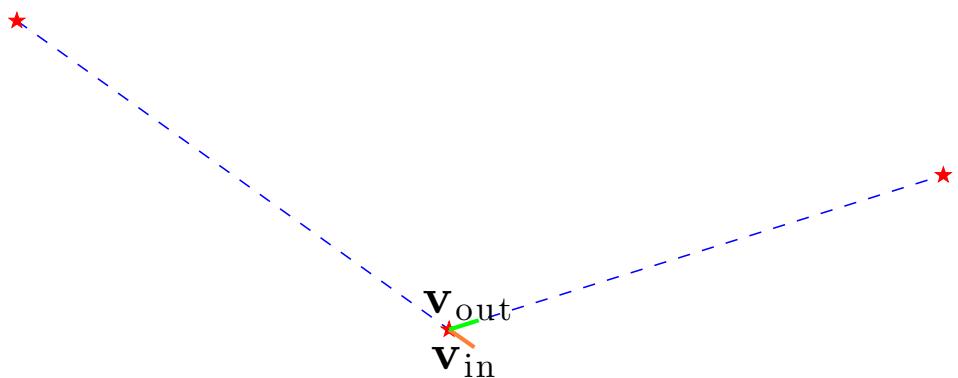
Figure 5.15: Two normalized vectors parallel to the straight lines are defined
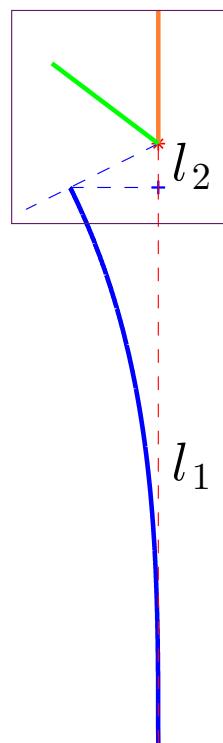
Figure 5.16: Starting point distance
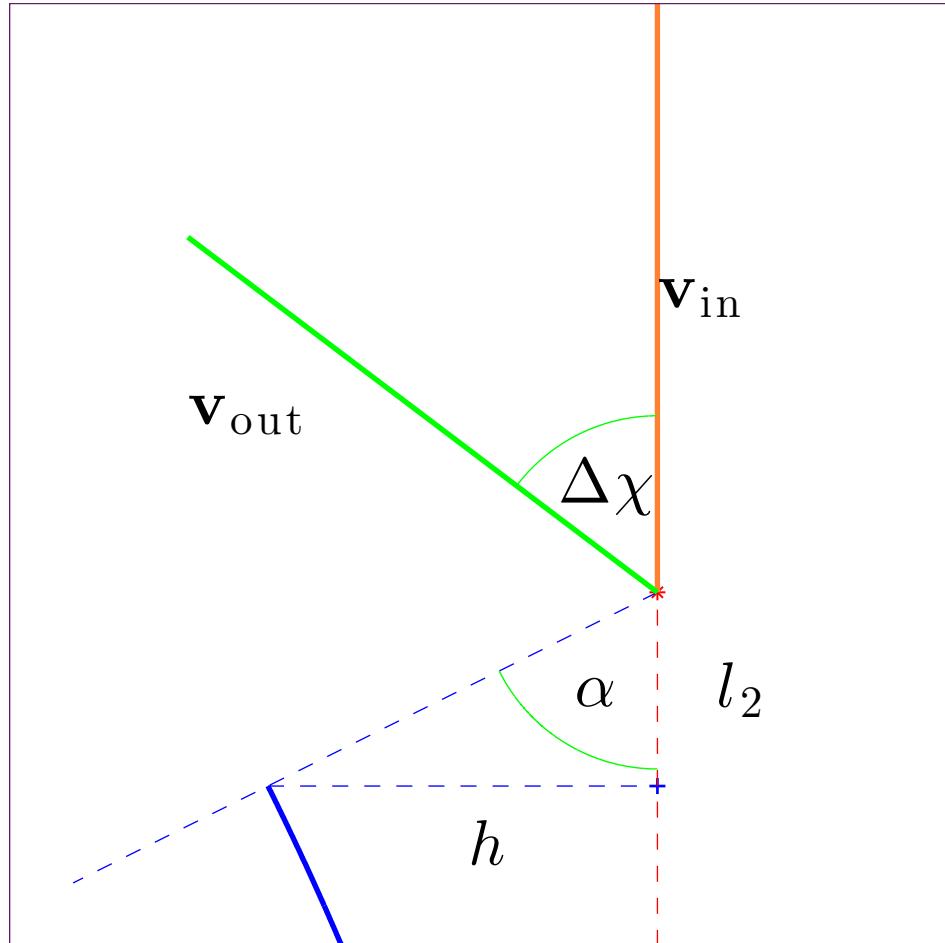
Figure 5.17: Starting point distance (magnified)
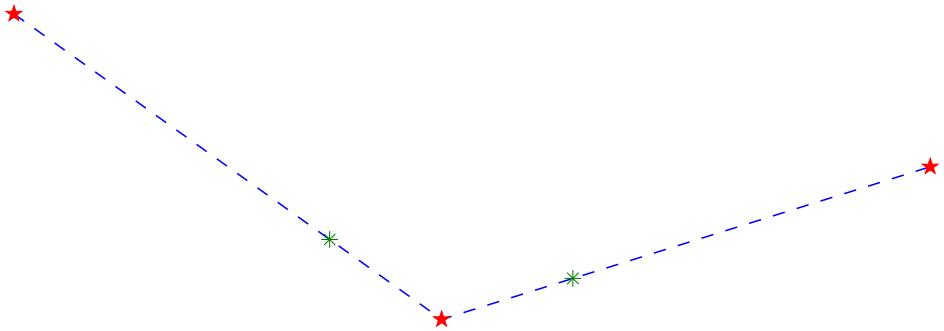
Figure 5.18: Wheel over and pull out points
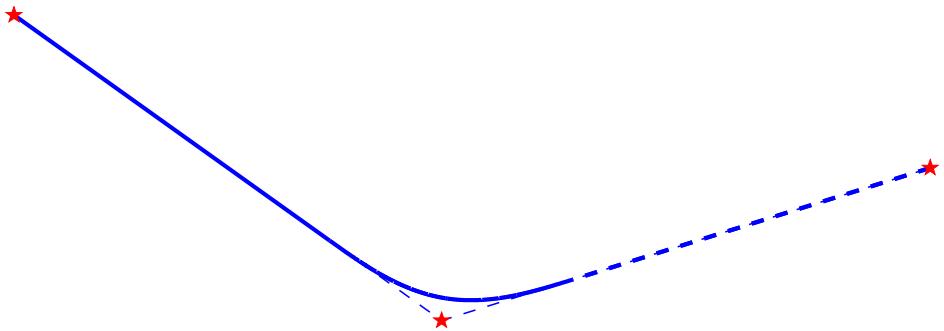
Figure 5.19: Final path

5.4.4 Length and Height Calculations

The projected length l_1 of the FS curve along the initial course and the height h perpendicular to it are useful for determining the initial position $\mathbf{p}_0$. Both are found by considering (5.18) with $\chi_0 = 0$. The length is then the x-component (see Fig. 5.16):

$$l_1 = k\sqrt{\theta_{\text{end}}}\cos(\theta_{\text{end}}), \tag{5.51}$$

and the height is the y-component (see Fig. 5.17):

$$h = k\sqrt{\theta_{\text{end}}}\sin(\theta_{\text{end}}). \tag{5.52}$$

5.4.5 Fermat Smoothing

The initial and final course angles, χ_0 and χ_{end}, can be determined in many ways. For instance, given tangent vectors:

$$\chi_0 = \text{atan2}\left(v_{\text{in,y}}, v_{\text{in,x}}\right), \tag{5.53}$$

$$\chi_{\text{end}} = \text{atan2}\left(v_{\text{out,y}}, v_{\text{out,x}}\right). \tag{5.54}$$
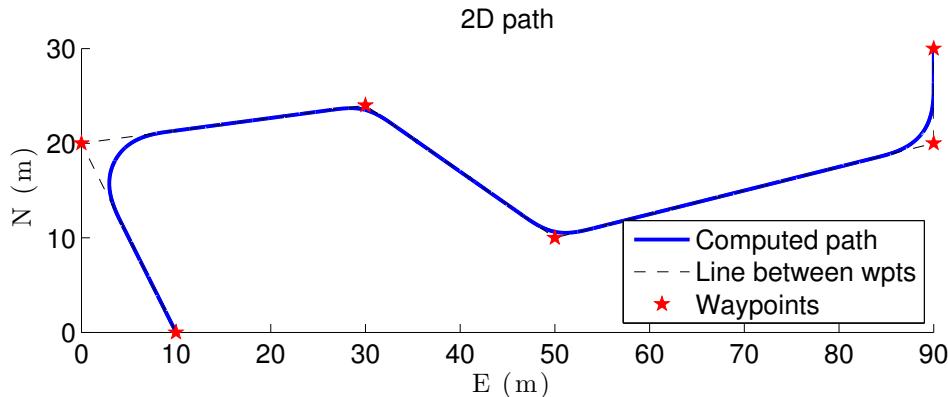
Figure 5.20: Fermat-smoothened path

Finally, the initial and final positions $\mathbf{p}_0$ and $\mathbf{p}_{\mathrm{end}}$ must be determined such that the curves meet. Fig. 5.16 shows that these must be at a distance

$$l = l_1 + l_2 \tag{5.55}$$

along the linear path from the waypoint. The distance l_1 is given by (5.51). From the figure it can be seen that

$$l_2 = \frac{h}{\tan(\alpha)}, \tag{5.56}$$

where h is given by (5.52) and

$$\alpha = \frac{180° - \Delta\chi}{2}, \tag{5.57}$$

as seen from Fig. 5.17. There is no danger for the angle α being equal to zero, and hence inducing a singularity at Eq. (5.56). This would imply a course change of $180°$, which will never occur for two successive waypoints. Figs 5.18–5.19 show the wheel over and pull out points, and the FS arc connecting them, respectively.

Figs 5.20–5.21 show the Fermat smoothing path and its properties. Similarly to the clothoid, due to the numeric solution of θ_{end}, the apparent continuity is not true: a gap exists at the transition between the Fermat curves. Still, any numeric precision can be achieved, consequently making the discontinuity correspondingly small and practically neglectable. Except for this, the path is visibly both tangent and curvature continuous, thus G^2.

It should be noted that the path curvature is κ_{max} at two points, unless the required course change is $\Delta\chi \leq 43.83°$[1]. However, it is more reasonable to stay at maximum curvature for as long as possible once reaching it. Such a path is discussed in Sec. 5.4.6.

---

[1]When $\theta_{\mathrm{end}} > \sqrt{\frac{\sqrt{7}}{2} - \frac{5}{4}} = 0.27$, the curvature κ_{max} is reached at $\theta = \sqrt{\frac{\sqrt{7}}{2} - \frac{5}{4}}$ both on the entering and exiting curve. When $\theta_{\mathrm{end}} \leq \sqrt{\frac{\sqrt{7}}{2} - \frac{5}{4}}$, the curvature κ_{max} is reached only at the transition between the curves.
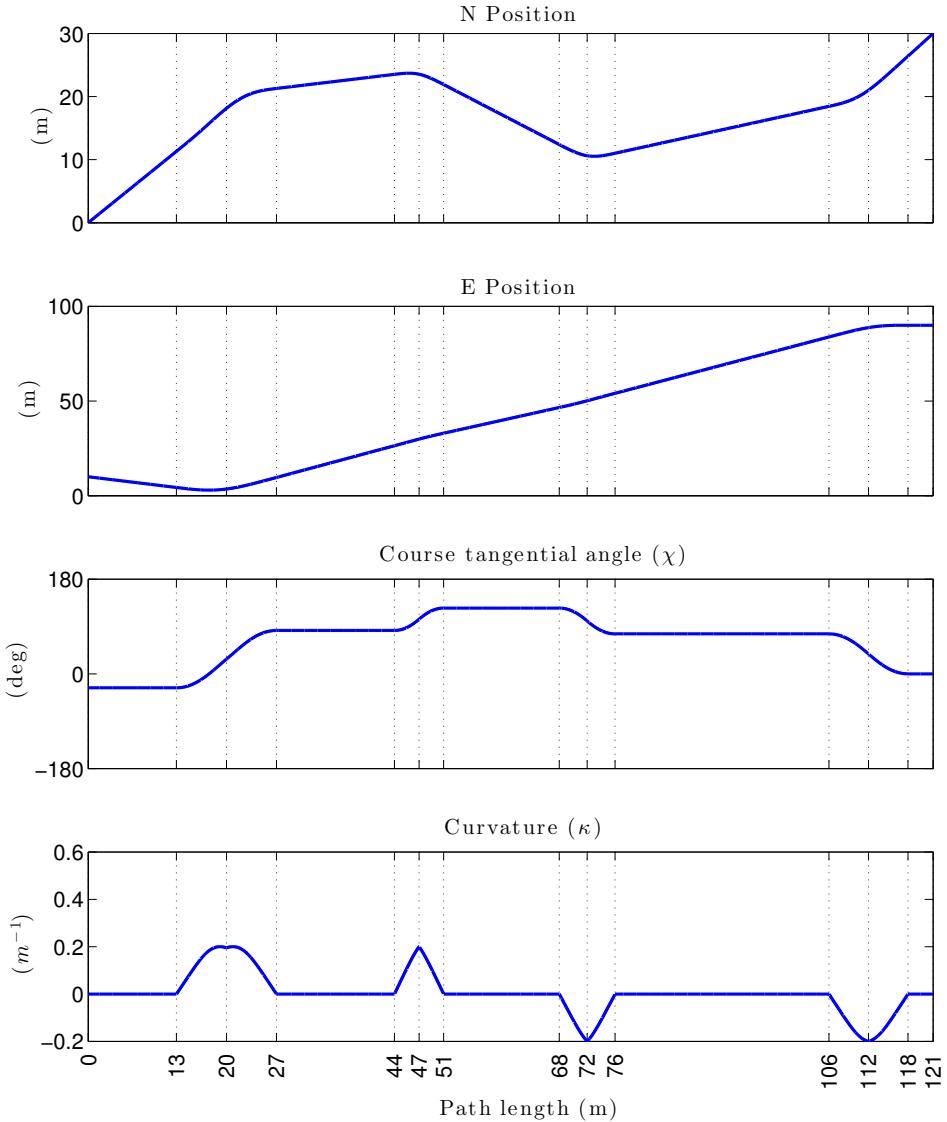
Figure 5.21: Fermat smoothing path properties
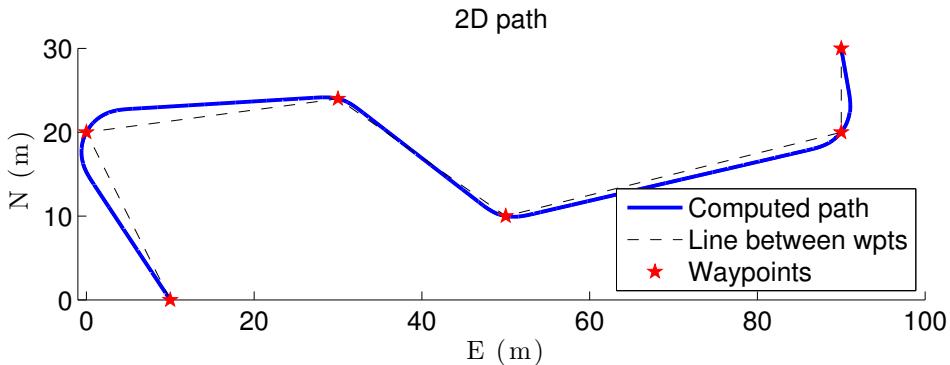
Figure 5.22: Dubins path with Fermat transition

5.4.6 Dubins Path with Fermat Transition

As mentioned earlier, in addition to using FS for connecting straight paths, FS can be used as a transition curve between a straight line and a circular arc. In this way, the resulting path can be the same as a Dubins path with the exception of the transitional FS segment. A similar approach where the clothoid was used as the transition curve was proposed in [94]. A Dubins path with FS transition and its properties can be seen in Figs 5.22–5.23. The curvature plot shows a more realistic and attractive behavior, compared to that of the previous section. Similarly to the original Dubins path case, this is an interpolating path.

5.5 Fermat Spiral Path Evaluation

In [156] and [68], the authors developed a number of path evaluation criteria that can be used for comparing paths depending on the application. Those criteria were:

- Smoothness
- Length
- Allowance
- Tractability
- Algorithm Complexity

Regarding the FS smoothing path, we conclude the following:

Smoothness: The path has a continuous curvature and, in addition, it was shown in Sec. 5.3.3 that a different parametrization results in parametric continuity as well. Table 5.1 shows how the FS path compares with the other paths considered in this chapter, when it comes to geometric continuity. Note that this is true for both interpolating and approximating versions of the path.

Length: The length of an FS arc can be computed by a hypergeometric series which converges for any value of θ, as it was shown in Sec. 5.3.4. Naturally, a path consisting of straight lines and FS segments will be longer than a Dubins path.
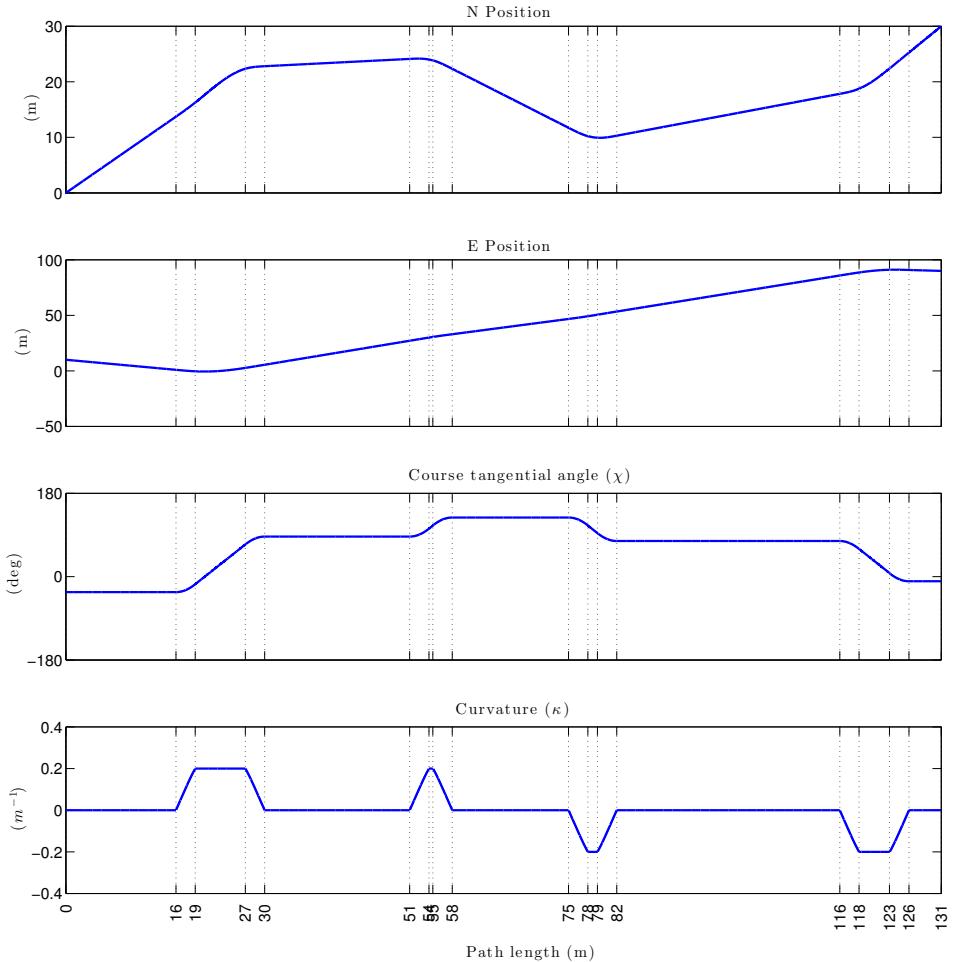
Figure 5.23: Dubins path with Fermat smoothing properties

Allowance: Allowance refers to how much the smoothing arc deviates from the initial piecewise linear path. This can be useful when the vehicle navigates in an area with obstacles. In this case the allowance is computed as:

$$a = k\sqrt{\theta_{\text{end}}} \sin(\theta_{\text{end}}). \tag{5.58}$$

Fig. 5.24 shows the allowance a for different course change magnitudes $\Delta\chi$. The comparison includes the FS, the clothoid and the circular arc. It can be observed that the Fermat smoothing allowance is almost the same as that of the clothoid for course changes up to $30°$. Above that value, the FS allowance is smaller than the clothoid's, and always larger than that of the circle.

Tractability: The effect of moving a waypoint is the same as for the clothoid, resulting in local influence, which means that if one waypoint changes location only
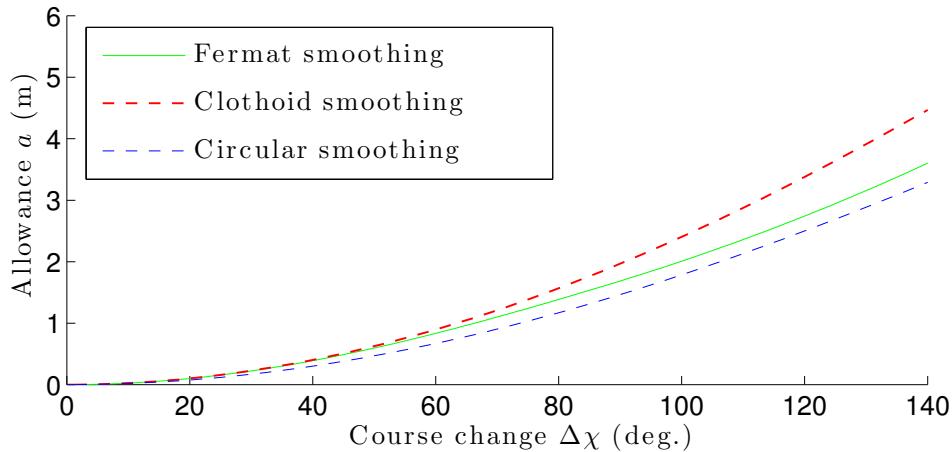
Figure 5.24: Fermat, clothoid and circular smoothing allowance comparison

Method	G^0/C^0	G^1	G^2
Piecewise linear	✓	✗	✗
Circular arc smoothing	✓	✓	✗
Clothoid smoothing	✓	✓	✓
Fermat smoothing	✓	✓	✓

Table 5.1: Geometric continuity comparison

the neighboring path segments will be affected. This is a useful property when a vehicle navigates in an area with obstacles and a waypoint has to be relocated during operation, for instance, due to weather conditions.

Algorithm Complexity: The process of computing an FS is significantly less computationally expensive than the clothoid. The exact computational times will vary (depending on the implementation platform), but FS involves only the numerical solution of (5.42), while the clothoid requires calculating the Fresnel integrals.

5.6 Conclusions

This work has considered the development of a new methodology, based on Fermat's spiral (FS), for generating curvature-continuous paths that can be used for path-following and path-tracking applications for underactuated vehicles. The main motivation has been to construct paths of simple geometry (preferably consisting of straight lines and arc segments) which avoid the curvature discontinuity of Dubins paths and the computational intensity of the clothoids. The FS was considered as a good candidate because it has a zero curvature at its origin, hence making it possible to connect it smoothly with straight lines. The properties of the

FS have been explored and an alternative parametrization has been proposed for guaranteeing parametric speed continuity, an important constraint when it comes to path-tracking applications. Despite the fact that the parameter end point (depending on the course change between two sequential straight lines) can only be computed numerically, the computational load is much lighter than that of the clothoid.

Further work includes a generalization to 3D paths and the design of parallel FS paths for formation control of multiple vehicles.

Chapter 6

Obstacle Avoidance for Underactuated Vehicles Using Voronoi Diagrams

6.1 Introduction

As explained in Chapter 1.1, the natural first step in a path-following scenario is to determine the vehicle's initial and final locations (alternatively, waypoints), or poses, if the initial and final vehicle orientations are predefined as well. Then, the path-planning algorithm is assigned to design a path which not only connects the two poses, but also satisfies several other constraints. These constraints can be both vehicle-related (such as the turning radius) and environment-related (such as avoiding obstacles). The final path must therefore be such that the vehicle is able to follow the path without deviating from it, unless an external factor interferes, and without colliding with any of the obstacles present. Examples of external factors that can cause unwanted deviations are environmental forces, such as waves, ocean currents, and gusts of wind, to name a few. For this reason, it is necessary to make the approach more robust by ensuring that the resulting path will always be at least at a minimum pre-specified distance from every obstacle. It is then said that the path has a certain amount of *clearance*, see [107, 108].

In this chapter, the Voronoi diagram is employed as a means of developing a methodology which not only generates collision-free paths and takes into account the required clearance, but also satisfies the vehicle's curvature constraints. The results presented here are based on joint work with Mauro Candeloro and were first published in [53].

6.2 Voronoi Diagrams for Path Planning

6.2.1 A Brief Literature Review

Voronoi diagrams (VD) were named after Georgy Voronoi who studied them in 1908 [226], although [230] reports that they were considered by Descartes in 1644

and used by Dirichlet in 1850 [73]. The VD is a tool to divide a geometrical space in regions determined by the distribution of the objects present in that space and, due to its simplicity and efficiency, the method has been used extensively since the beginning of the 20th century in a wide range of applications such as networking, chemistry, data classification, crystallography and biology. An in-depth description of Voronoi diagrams and their possible applications can be found in [15].

One of the most popular applications of the VD, though, is path planning and environment exploration for mobile robots. This is true mainly due to following three reasons:

- The Voronoi diagram is a strong tool which divides the geometrical space in a way such that the borders of the regions are maximally distant from all the obstacles in the cluttered environment.

- It is an algorithm that has $\mathbf{O}(n)$ complexity, while the majority of other mathematical tools can solve the same problem with quadratic complexity.

- The kinematics of mobile robots allows them to change heading without affecting the other degrees-of-freedom (DOFs), so they can easily run a path composed only by a sequence of straight lines.

The use of VD for mobile robot obstacle avoidance has been through several stages of development, with many milestones over the last 20 years. As reported by Thrun [220], the simplest approach is to use the VD as a roadmap [55, 150], where the robot is assigned to move along the VD, see also [60, 62, 143, 144, 177]. This requires a robot that is fully actuated in 3-DOF environments because the resulting path contains cusps and sharp edges. As a consequence, the robot would have to stop at each point where two successive lines meet, change its orientation, and then continue moving along the next line. The main difference between these works is that in [150, 177] the authors assume that the environment is perfectly known, whereas [60, 62, 143, 144] deal with uncertainty by extending the approach to sensor-based motion planning. A description of how to incrementally construct the roadmap using only range information was presented in [61]. A sensor-based approach, which was able to incorporate new obstacle information, was also used by the authors in [103–105], where VD and Fast Marching were combined so as to give more efficient paths. The suboptimality of the paths resulting from using the VD as roadmaps was tackled in [220, 221]. An efficient approach which gives optimal paths and respects clearance-related constraints was presented in [23, 24]. More recently, considerable effort has been put in order to enable the construction of VD over an unknown area in finite time using range measurements, see [136, 137]. Dynamic VD have also been studied, see [3, 4, 114, 197, 198] for some earlier work, and [16, 18, 216, 217] for more recent approaches.

Apart from mobile robots, many researchers have exploited the properties of VD so as to construct paths for other types of vehicles. This is necessary since the edges and sharp corners of the Voronoi paths do not allow most vehicles to move along them without deviating from the paths. This refers to both underactuated vehicles and fully-actuated vehicles which are assigned to follow the paths without stopping at every sharp corner in order to change their heading. In other words, when velocity and curvature constraints are present, it is important to modify the VD accordingly in order to satisfy them. This problem was addressed for a car-

like vehicle in [173]. The authors of [138] use the VD, assisted by the Eppstein search algorithm presented in [77], for waypoint generation. Based on the resulting waypoints, they construct Dubins paths for small fixed wing UAVs. However, the Dubins paths have a curvature discontinuity at the transition points and the kinematic model used is that of a particle. A different strategy, based on virtual forces, was followed in [33], where the UAV path was represented by a chain of masses connected to one another by springs and dampers. In [102], VD were used for motion planning of rigid robots in dynamic $3D$ environments by taking into account constraints translated as virtual forces. A Bezier curve was used in [121, 240] for designing smooth paths from the VD waypoints. One drawback of this approach is that Bezier curves require the definition of control points which are not located on the path. In addition, several other drawbacks can be associated to the paths consisting of splines, as discussed in Section 5.1. Naturally, applications of VD to cooperative control have also appeared in the literature, see for instance [57, 164].

Regarding marine applications, both the static and dynamic VD have been considered, the reader is referred to [51, 109, 111, 112]. An interesting augmentation of VD was developed in [168], where clothoids and circular arcs were used to smooth the Voronoi paths, hence resulting in a curvature-continuous path. However, [168] focuses on generating differentiable VD and neither deals with how the waypoints should be selected, nor considers clearance constraints.

The rest of this chapter presents the methodology presented in [53]. The main goal was to construct curvature-continuous paths which respect both curvature and clearance constraints. At the same time, it was considered important to implement a path planning method which is not computationally expensive because, for a given map with a relatively large number of waypoints, this might induce a significantly inferior performance. To this end, Fermat's spiral, see [68, 160], was preferred over the clothoids as a transition curve. The problem was not investigated from an optimal length point of view, however it would be highly inefficient to just smoothen the Voronoi roadmap and assign it as the desired path. Consequently, a combination of the algorithms of Dijkstra and Yen [72, 237], along with a few ad-hoc techniques were used so as to get a useful and practical path.

6.2.2   Mathematical Description

The basic idea behind Voronoi diagrams is explained in the following: a finite set of points $\mathbf{P} = \{p_1, ..., p_2\}$, called *generator points*, are contained in a space $\mathbf{X}$ (in this work $\mathbf{X} \subset \mathbb{R}^2$) where a metric function $d(\cdot)$ is defined. The procedure associates to each point $p_i \in \mathbf{P} \subset \mathbf{X}$ a *Voronoi region* $\mathbf{R}_i$ defined as the set of points $x_i \in \mathbf{X}$ such that their distance to p_i is lower than the distance from x_i to any other point of $\mathbf{P}$. In mathematical terms, if:

$$d(x, p) = \inf\{d(x, p) | p \in \mathbf{P}\} \tag{6.1}$$

the Voronoi region is defined as:

$$\mathbf{R}_k = \{x \in X | d(x, \mathbf{P}_k) \leq d(x, \mathbf{P}_j) \forall j \neq k\} \tag{6.2}$$

The final Voronoi diagram $\mathbf{V}(p_i)$ will consist in the set of the Voronoi regions borders, that is the intersection of the tuple of cells $(\mathbf{R}_k)_{k \in \mathbf{K}}$: $\mathbf{V}(p_i) = \underset{i}{\cap} \mathbf{R}_i$, where
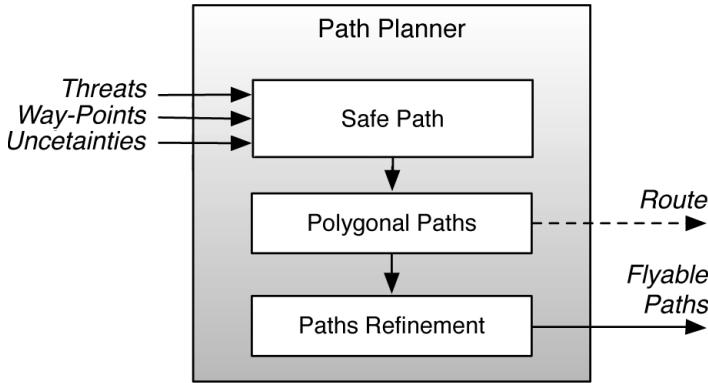
Figure 6.1: Approach to path planning [224]

K is a set of indexes. It can be easily observed that the shape and the properties of the final diagram will depend on the definition of the metric used in the process, see for instance the case of the curved Voronoi diagrams presented in [29].

6.2.3 Obstacles and Metric Definition

Implementing Voronoi diagrams for path planning and obstacle avoidance is straightforward when assuming that the locations of the generator points p_i coincide with the locations of the obstacles. In this case the borderlines of the Voronoi regions $\mathbf{R}_i$ are composed by the points that are maximally distant from the obstacles or, in other words, the points that are the least dangerous for the vehicle to follow. Some implementations of the VD also take into account the determination of a *clearance factor* [23] to reduce the distance between the path and the obstacles, hence decreasing the length of the final path, the time required to arrive to the destination, and the fuel consumption of the vehicle.

In this work we consider a real map where the obstacles consist of both the islands and the mainland. For our implementation, the obstacles are defined as simple geometrical representations of the aforementioned obstacles, where the vertices of the objects have been chosen to be the generator points of the Voronoi diagram.

The L_2-distance function (alternatively, the Euclidean distance) has been considered as the metric function $d(\cdot)$:

$$d\{(x_1, y_1), (x_2, y_2)\} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \qquad (6.3)$$

this metric generates Voronoi regions with straight-line borders. A roadmap then consists of a number of successive straight lines, the Voronoi borderlines, which connect the starting waypoint with the final waypoint. Such a piecewise-linear path is the initial step, which will be further refined in order to achieve the performance constraints.

6.3   Path-Planning Algorithm

The path-planning algorithm consists of 5 main steps (see also Fig. 6.2):

System parameters setting: includes definition of the obstacles, and specification of the vehicle constraints and the clearance factor;

Obstacles definition: generates the obstacles map and the obstacles edges;

PL Path generation: defines the piecewise-linear path connecting the starting and ending point;

Fermat's spirals design: smooths the path edges assuring curvature continuity;

Plotting: plots the final path.

It should be mentioned that, since the implementation pertains to underactuacted vehicles, one additional goal of this algorithm is to obtain a *practical* path. This implies avoiding unnecessary heading changes and/or moving along longer paths when there is a shorter (while still safe) one available. The steps to achieve this goal are contained in the *Path generation* module. As a consequence, especially Step 3 involves some extra processing before the piecewise-linear path to be smoothed is determined. More specifically, right after the initial VD has been constructed, the following actions occur:

- 3.a) The Voronoi diagram gives a raw obstacle-free roadmap.

- 3.b) The map borders are defined.

- 3.c) The point of departure and the point of arrival are inserted.

- 3.d) The paths that cross the borders of the map, or any obstacles, are removed. The map borders that are obstacle-free are defined in the system parameters settings phase, and modeled as obstacles, not to be crossed. In this way any it is avoided to accept as safe any path that is leading outside the map.

- 3.e) The Dijkstra-Yan algorithm is implemented on the remaining safe paths in order to find the shortest one.

- 3.f) In-line, or almost-collinear, waypoints are removed. The Voronoi diagram creates a high number of vertices, which is proportional to the number of points that represent the obstacles in the map. If all the vertices defining a path were to be considered, this would result in two negative effects: the first one is of geometrical nature and can induce the definition of spiral segments with high curvature that exceeds the vehicle's potential. The second one is practical and can be associated to the fact that there is no meaning to assign the path $A - B - C$ if A and C can be connected with a straight line without decreasing the quality of the path. This means that, if A, B and C are almost collinear, where "almost" refers to a certain threshold which depends on the problem in hand, then B can be eliminated. A similar approach was implemented in [240].

- 3.g) The clearance constraint is checked and an alternative path is chosen if the safety constraints are not met.

- 3.h) Finally, unnecessary waypoints are removed.
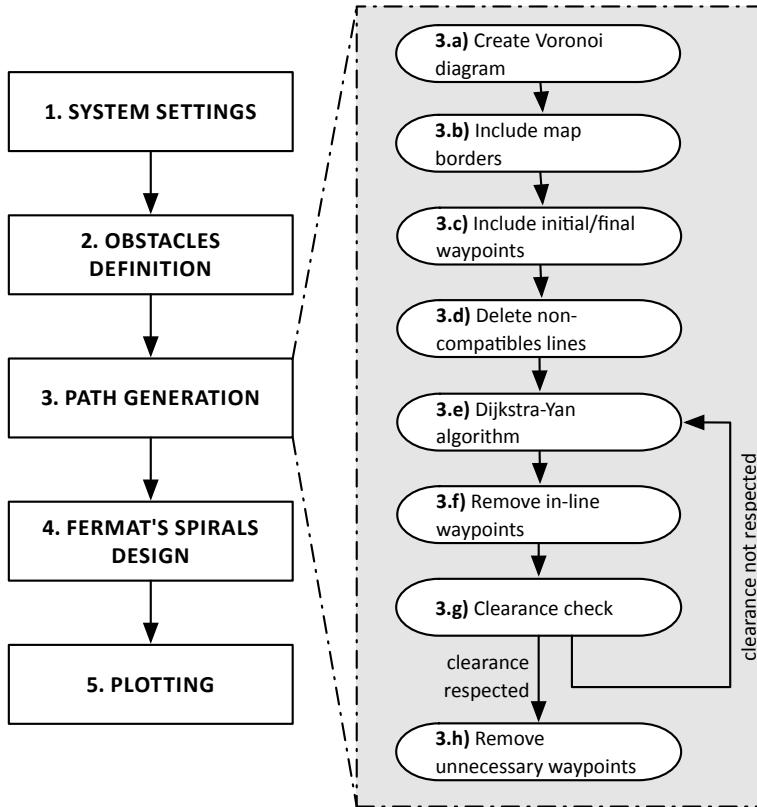
Figure 6.2: Algorithm Structure

Notice that the clearance check should be repeated after Step 3 because the spirals might violate the clearance constraints. In this case, the unacceptable spiral segments should be designed again with a higher curvature, until the clearance constraints are satisfied along the whole length of the curved path. If this is not possible then other waypoints should be selected from the Voronoi diagram and the procedure be repeated.

6.3.1 Suppression of waypoints in excess

The final path can still be refined by eliminating all those waypoints that do not influence the quality of the path in terms of safety and clearance to the obstacles. The procedure can be summarized with the following pseudo-code:

```
function removeExcess(wayPoints)
    for each waypoint do
        Q1=waypoint(i);
        Q2=waypoint(i+2);
```

```
        P =waypoint(i+1);
        if {clearance of ((Q1,Q2),Obst.)>thres.}
            remove P from the list;
        else
            i=i+1;
        end if;
    end for;
end function.
```

6.3.2 Fermat's spiral joints algorithm

The implementation algorithm needs to receive as input the two straight segments that should be connected with the FS arc. Notice that in this chapter the FS is constructed so that that the intersection point will correspond to the middle point of the shortest segment. This assures the continuity of the path in the case where two FS need to be connected without a straight line in the middle. The algorithm follows the following steps (the step numbers and symbols refer to Fig. 6.11):

1. Find the initial and final tangents $\mathbf{v}_{in}$ and $\mathbf{v}_{out}$, and course change magnitude $\Delta\chi = \chi_{endpoint} - \chi_{startpoint}$;

2. Find endpoint tangent $\chi_{\theta_{max}} = \frac{\Delta\chi}{2}$ and the related θ (numerically).;

3. Find intersection point with the shortest segment, that will be its mean point;

4. Find the intersection point $\mathbf{q}_{st}$ with the other segment if the first or the second segment is the shortest (Figure 6.11):

$$\begin{cases} l = \dfrac{-\mathbf{p}_{st} + \mathbf{P}_2}{\mathbf{v}_{in}} \\ \mathbf{q}_{st} = \mathbf{P}_2 + l \cdot \mathbf{v}_{out} \end{cases} \tag{6.4}$$

$$\begin{cases} l = \dfrac{-\mathbf{q}_{st} - \mathbf{P}_2}{\mathbf{v}_{out}} \\ \mathbf{q}_{st} = \mathbf{P}_2 - l \cdot \mathbf{v}_{in} \end{cases} \tag{6.5}$$

5. Compute the scaling factor corresponding to the desired curvature;

6. Compute the maximum curvature;

7. Calculate a sufficient number of points to approximate the curve from zero curvature to maximum curvature;

8. Calculate the curve from maximum curvature back again to zero curvature (the last part of the link) by mirroring the formula used in Step 6.

6.4 Simulations

In the next pages, the results of a simulation based on a real map of the Norwegian coastline are illustrated. In particular, a part of Fensfjorden, which is located 50 km north of Bergen, is considered. The vehicle, which could be a fishing boat, a research

vessel, or a ferry boat traveling in the fjord, is assigned to begin its trip at the *starting point* and finally reach the *final point* while at the same time keeping a certain distance (the clearance) from the obstacles and avoiding unnecessary turns or unjustifiably long paths.

Figs 6.3–6.8 depict the main results of the algorithm steps, which lead to the determination of the final waypoints based on which the path is constructed. It can easily be observed that the performance requirements are met, hence providing a feasible and practical path. The final path can be seen in Fig. 6.9, while a plot of the curvature of the final path is shown in Fig. 6.10. Once again, the performance requirements are satisfied, it can be confirmed that the curvature is continuous and that its highest value is $0.04 \, \mathrm{m}^{-1}$, corresponding to a turning radius of $25 \, \mathrm{m}$. This value was chosen because it is a typical one for vessels belonging to the aforementioned categories.

6.5 Conclusions

This chapter has dealt with a very important part of motion planning, namely, the determination of the waypoints. The main goal was to develop a simple strategy so as was to pick the waypoints in a way such that a certain clearance constrained is satisfied and the overall resulting path is practical and curvature-continuous. To this end, several actions were taken in order to pick up suitable a set of waypoints from the large number initially generated by the Voronoi diagram. Fermat's spiral arcs were employed to smooth the piecewise-linear path which occurs when connecting the final waypoints. This led to a methodology which achieves the following:

- Generates a path which guarantees that the vessel will be always at a safe distance from the obstacles in a cluttered environment.

- The whole process is very efficient computationally even for a large number of waypoints. This occurs due to the low number of computations required for both the generation of the Voronoi diagram and the calculation of the path table for the FS segments.

Figure 6.3: The first step involves generating the Voronoi diagram after the obstacles have been defined on the map.

Figure 6.4: The second step involves choosing the waypoints which give the shortest piecewise-linear path.
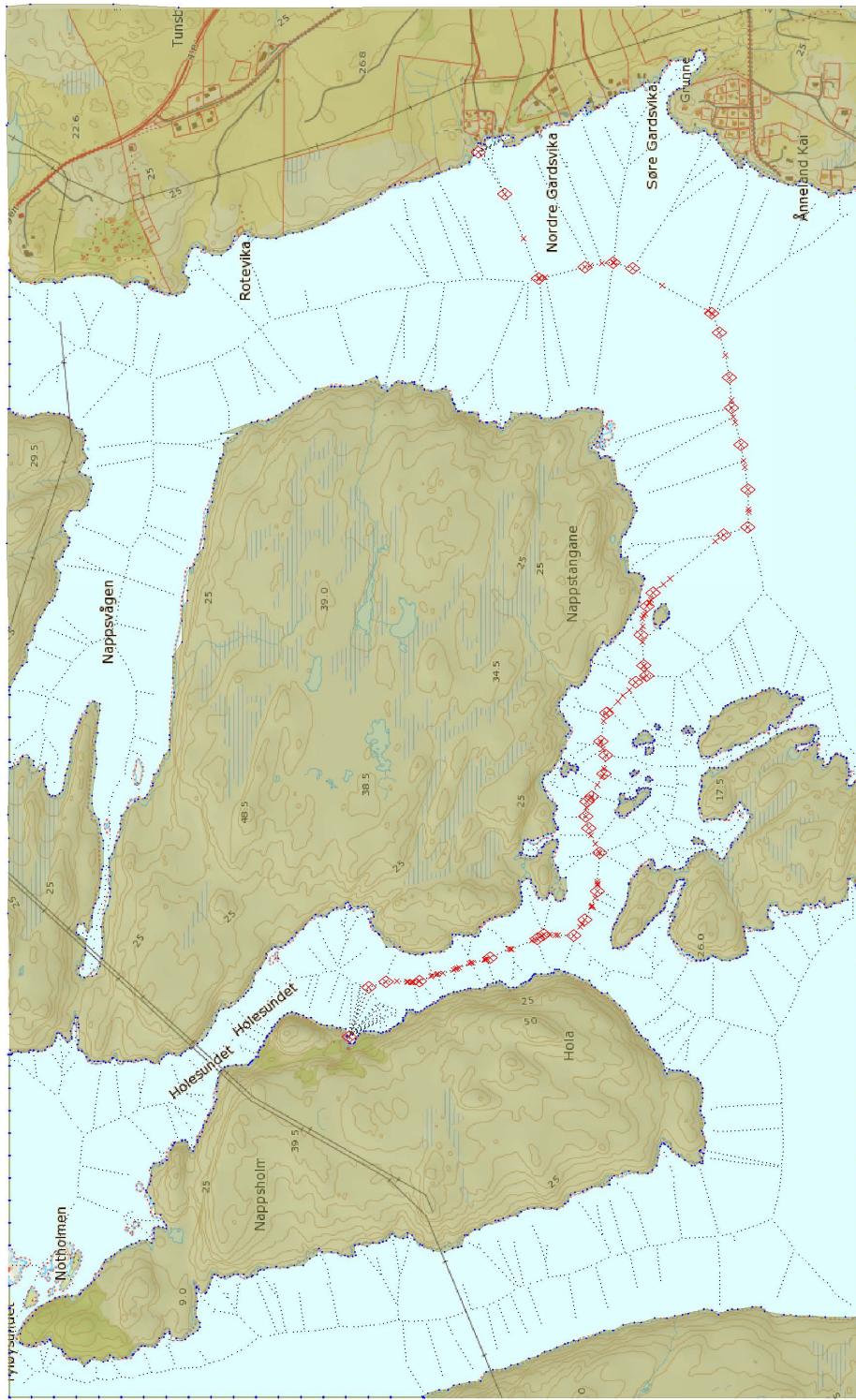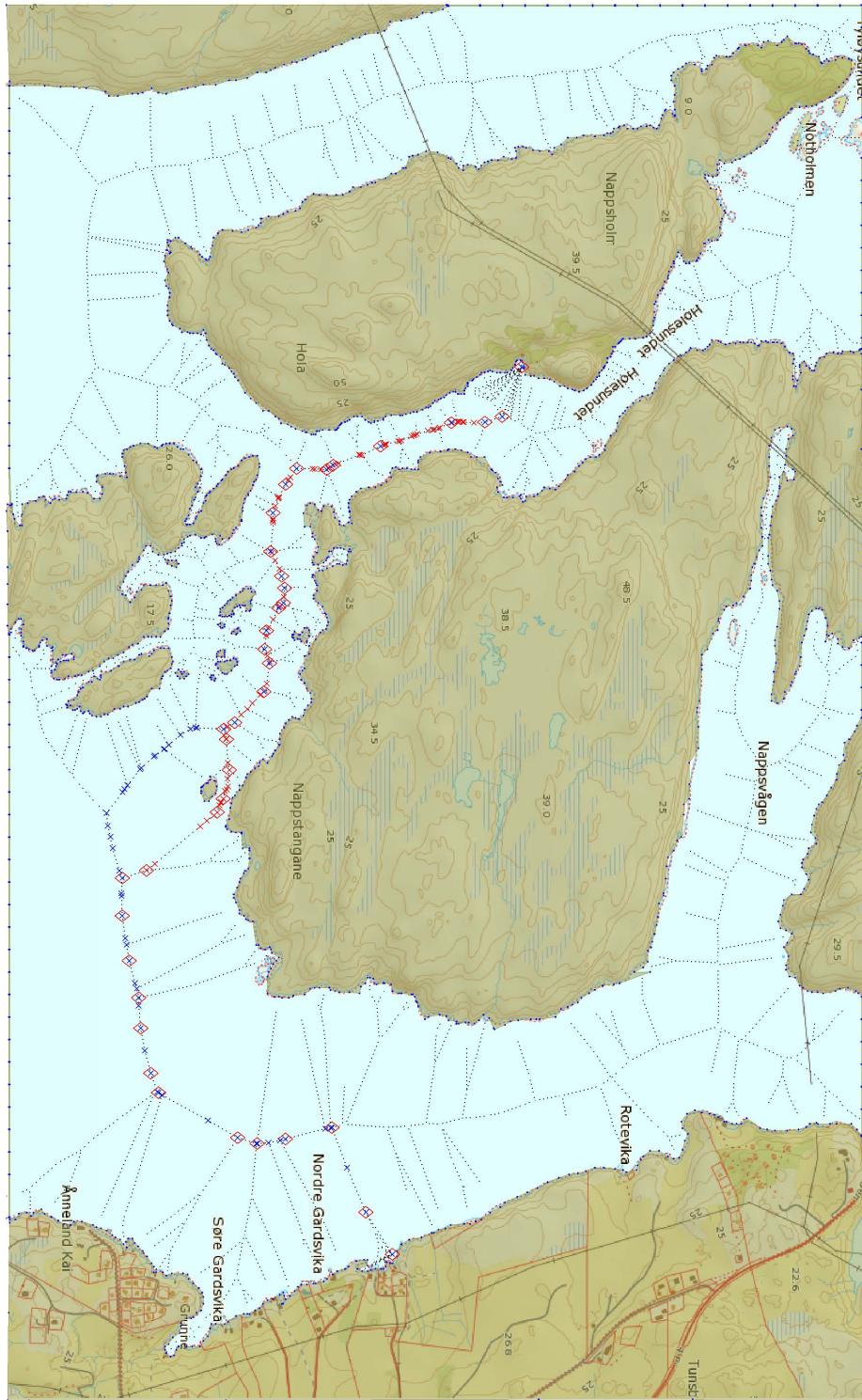
Figure 6.5: In Step 3, the inline, or almost collinear waypoints are removed (only the diamond-shaped points are kept) and a clearance check is performed.

Figure 6.6: Step 4: The first path fails to meet the clearance constraints and is rejected. The second shortest path (blue waypoints) is now defined.
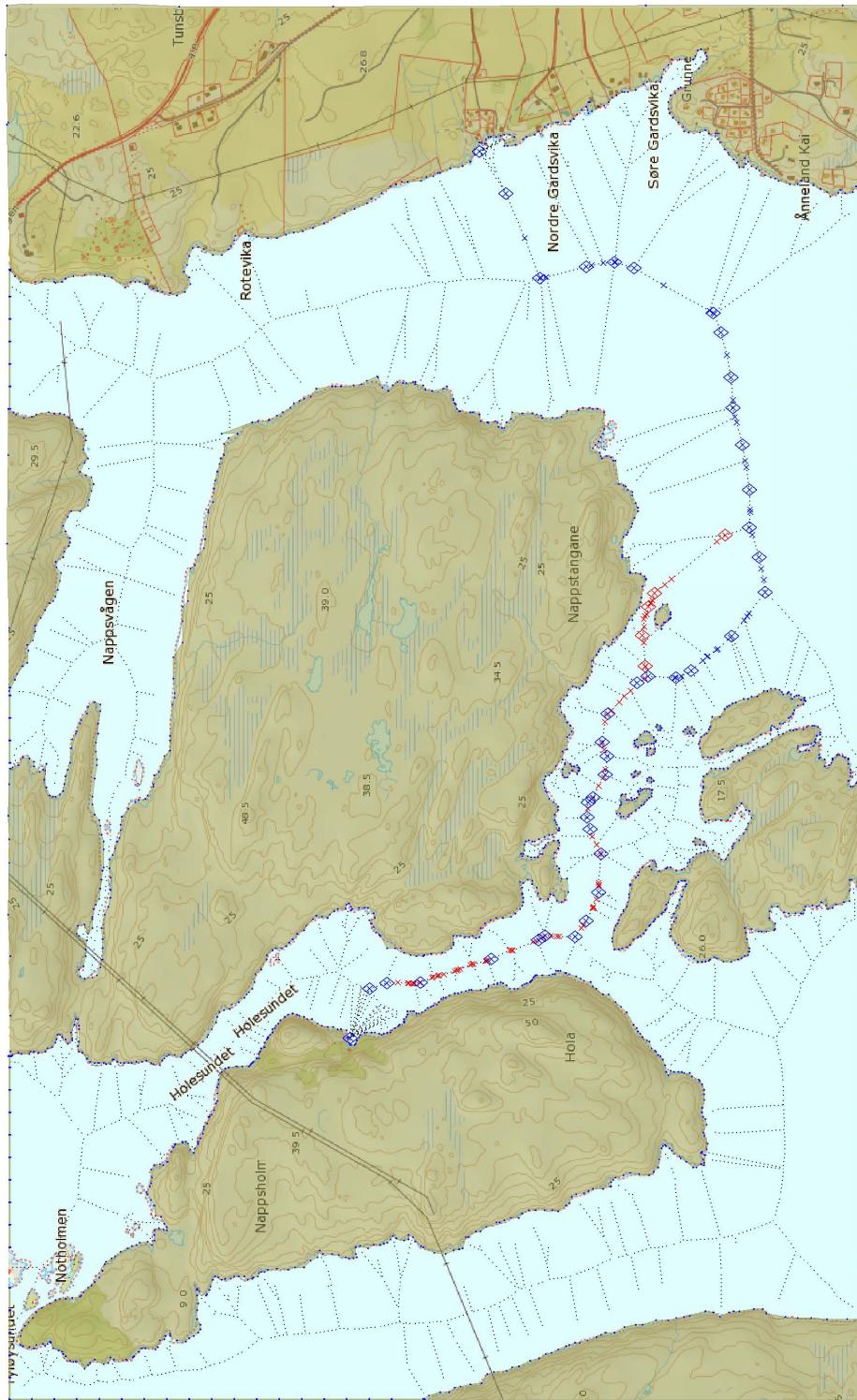
Figure 6.7: Step5: The second shortest path in now undergoing the same procedure. The number of waypoints is reduced and clearance check is performed.
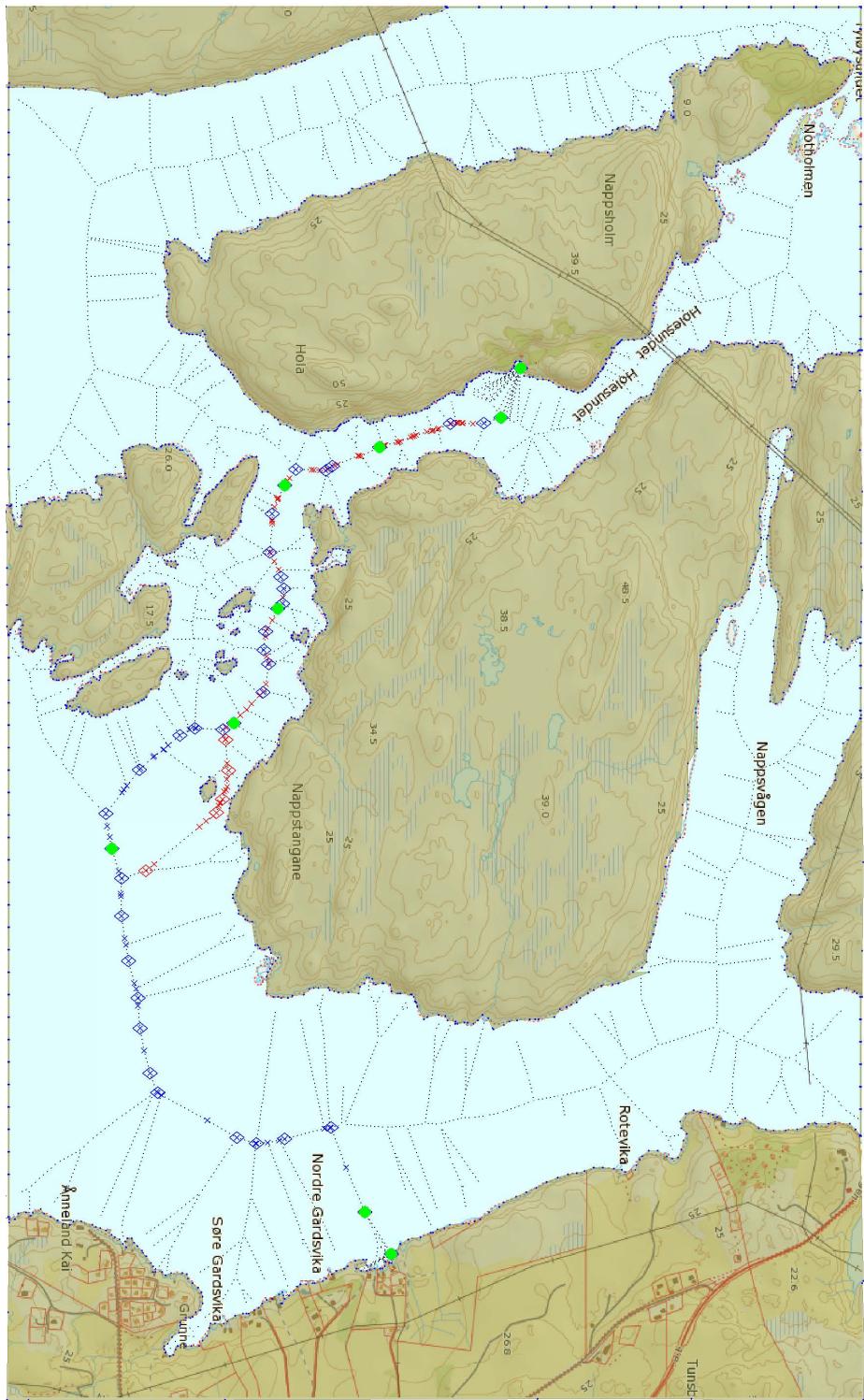
Figure 6.8: The second path satisfies the constraints and is accepted. Additional checks are performed in order to further reduce the number of waypoints and, hence, the number of heading changes.
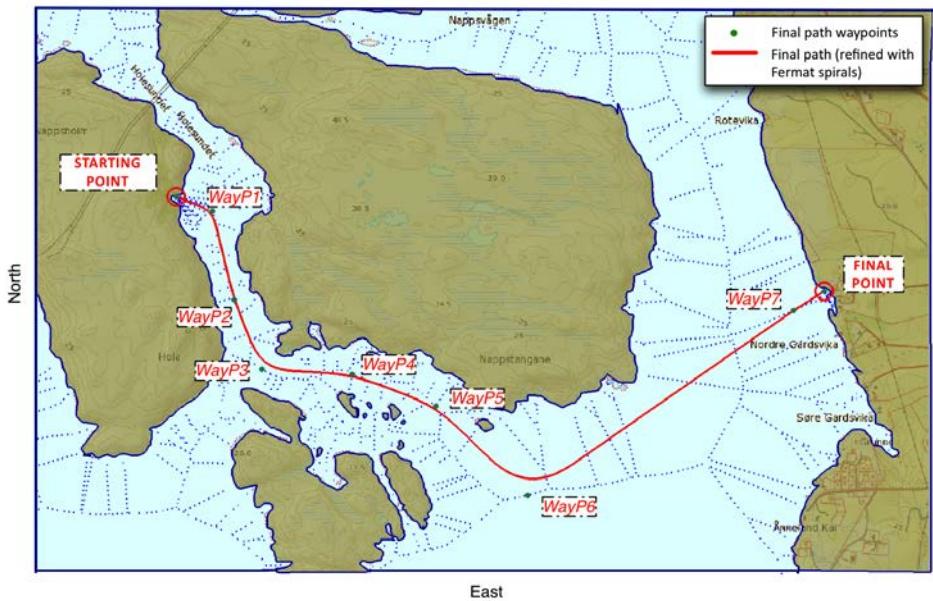
Figure 6.9: The waypoints of the refined path define a discontinuous-curvature path. Fermat's spirals are then added providing continuous curvature. The resulting path is practical, safe and defined by a few waypoints that makes unnecessary the presence of high-curvature turns.
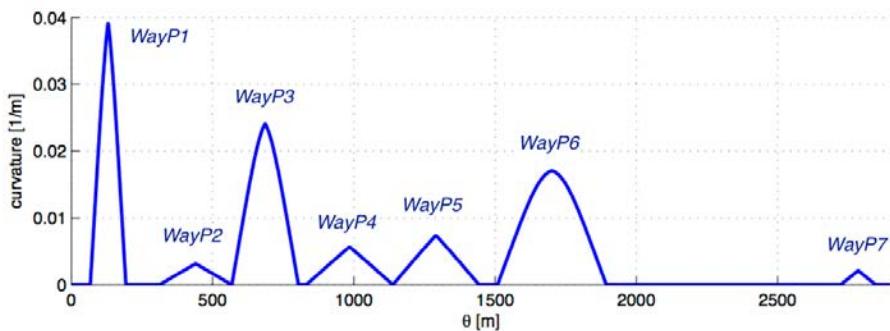
Figure 6.10: Curvature as a function of distance from the starting point. It can be verified visually that curvature-continuity is preserved. Peaks corresponds to waypoints.
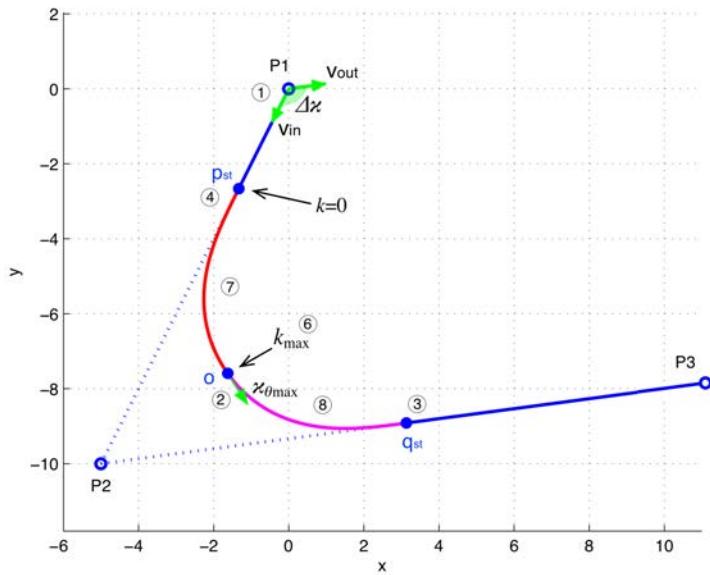
Figure 6.11: An illustration showing how FS smooths the sharp corner formed by three successive waypoints. The two mirrored segments (orange vs pink segment) are also depicted here.

Figure 6.12: A depiction of how the final path is based on the selected waypoints, while at the same time it does not cross all of them. However, the clearance constraints are always satisfied although the path is an approximating one.

Part III

Guidance

Chapter 7

Line-of-Sight Guidance

7.1 Introduction

Guidance systems are of utmost importance when considering the performance of aerial, surface and underwater vehicles, regardless of the motion control scenario involved. In addition to the introductory information given in Section 1.1.3, more information on control motion scenarios, such as target tracking, path following, path tracking and path maneuvering can be found in [35, 42, 233]. Path-following refers to the case where the control objective is to converge to and follow a predefined path without involving any requirements pertaining to temporal constraints, which means that it is sufficient for a ship to have a nonzero forward speed and use its rudder for minimizing the error w.r.t. the path, to give an example. In order to achieve this, several guidance laws have been studied and implemented in the past, with the missile community being most likely the oldest and most active one on the field. As a matter of fact, some of the most commonly used approaches by the marine control community have been directly influenced by their missile community counterparts, such as the Line-of-Sight (LOS) guidance, the Pure Pursuit (PP) guidance and the Constant Bearing (CB) guidance. The implementation of the three aforementioned methodologies in the case of autonomous underwater vehicles has been discussed extensively in [41].

The LOS guidance algorithm and its properties have been studied thoroughly in the literature. Frequently, the LOS guidance is considered in connection with the heading autopilot due to the fact that the two systems form together a cascade structure that needs to be stabilized. One of the first efforts of this kind was presented in [118]. In [93], an implementation of the LOS guidance law for straight line path following can be found, whereas in [36] the motion of a particle was considered and the method was implemented for curved path following. A cascaded systems approach was also developed by [30] where the guidance system is interconnected with a sliding mode controller in order to achieve global κ-exponential stability when the desired path is a straight line in 3-D space. The concept of global κ-exponential stability was introduced in [213] and is equivalent to Uniform Global Asymptotic Stability (UGAS) and Uniformly Local Exponential Stability (ULES) combined. All the aforementioned approaches assumed a constant looka-

head distance. In general, a small lookahead distance will induce more aggressive steering, hence resulting in reaching the desired path faster, but it might also be the reason for unwanted oscillations around the path. On the other hand, a large lookahead distance results in a smoother steering which prevents unwanted oscillations, but the downside is slower convergence to the path. The idea of combining these behaviors in a complementary manner was investigated in recent works, such as [38],[187],[179] and [155] where the authors discussed and implemented the possibility of using a time-varying lookahead distance. These works, regardless of the theory each one of them was based on, indicated that a time-varying lookahead distance can contribute to converging to the desired path faster while at the same time reducing oscillations around the path.

Although the LOS guidance algorithm is efficient and has a very simple structure, when unknown forces (due to the wind or sea currents, for example) act on the vehicle it is not possible (in the general case) to succeed in accomplishing the motion control task, i.e. to converge to the desired path. A common and reasonable strategy is then to augment the LOS guidance law by adding integral action in order to eliminate any constant offsets induced by the constant, or slow time-varying, environmental forces. In [41] the addition of a simple integrator is discussed, whereas in [31] a more sophisticated approach was followed resulting in a method that avoided the integrator wind-up phenomenon.

This chapter presents an overview of the lookahead-based LOS guidance law in its conventional form, when implemented for marine applications, and follows a constructive approach. Initially we consider two uncoupled 3-DOF models, where the first pertains to guidance on the horizontal plane and the second pertains to guidance on the vertical plane. Stability proofs are given in both cases, assuming perfect heading and pitch tracking respectively, and the result shows how the sideslip angle and the angle of attack affect the performance. Next, a more general case in 3-dimensional space is studied. The task is to converge to a straight line which is at a known depth underwater. In this case the horizontal plane LOS guidance system is coupled with the depth controller and the stability analysis is more complicated and it also depends on the type of depth controller used.

In path-following applications, the shape and the properties of the path itself have a great influence on the reference trajectories generated by the LOS algorithm. Therefore, the path-planning problem is directly correlated to the guidance problem. It is often sufficient to design paths that consist of successive straight lines, this selection offers simplicity since the path-tangential angle is constant for each line and it also makes it easier to compensate for constant environmental disturbances, such as winds and ocean currents. Stability proofs can also be more tractable when using straight line paths. On the other hand, such paths result in a non-smooth velocity function and, consequently, curvature, resulting in sudden increases of the cross-track error when switching to the next waypoint. There is a vast literature on the properties of the several path-planning methods and therefore a wide variety of paths to choose from. Depending on the application and the constraints, properties such as curvature continuity or minimum length might be more important to satisfy than others. In this chapter, paths based on straight lines are used to demonstrate the efficiency of the LOS guidance.

The rest of this chapter is organized as follows: Section 7.2 presents the dy-

namics and kinematics of the vehicle. In Section 7.3, the 3-DOF LOS guidance is presented for a vessel navigating on the horizontal, as well as the vertical, plane. Here the two cases are assumed to be uncoupled. In Section 7.4, a three-dimensional scenario is studied and in this case the horizontal LOS guidance system is coupled with the depth controller. In Section 7.5, we present a transformation methodology for expressing the LOS guidance in quaternion form. Finally, Section 7.6 gives a brief summary of the chapter. Section 9.1 deals with guidance under the influence of constant environmental disturbances and reviews two LOS-based methods to tackle the problem. Furthermore, it gives an explanation of why no integral action needs to be added when the course angle, instead of the yaw angle, is controlled.

7.2 Vehicle Models

This section presents the vessel model and the related assumptions that are considered in this chapter. Two reference frames, namely, the *North-East-Down* (NED) coordinate system $\{n\} = (x_n, y_n, z_n)$ and the body-fixed reference frame $\{b\} = (x_b, y_b, z_b)$ is adopted in this chapter to describe the motion, location and orientation of the vehicle. The NED frame is defined as a tangent plane on the surface of the Earth moving with the vehicle and is sufficient for local operations. Its origin is o_n and the x axis points towards the true *North*, the y axis points towards the true *East* and the z axis points *downwards*, normal to the Earth's surface. The body-fixed frame is moving with the vehicle and its origin o_b coincides with the center of gravity of the vehicle, see also [88, Ch. 2].

7.2.1 Vehicle Dynamics

Similarly to [30], for the path-following task we can neglect the roll angle, hence for an underactuated autonomous vehicle the following 5-DOF dynamic model can be used:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\boldsymbol{\nu}, \tag{7.1}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}, \tag{7.2}$$

where $\mathbf{M}$ is the mass and inertia matrix, $\mathbf{C}(\boldsymbol{\nu})$ is the Coriolis and centripetal matrix, $\mathbf{D}(\boldsymbol{\nu})$ is the damping matrix, $\mathbf{g}(\boldsymbol{\eta})$ describes the gravitational and buoyancy forces, and $\boldsymbol{\tau}$ includes the control forces and moments.

Accordingly, the generalized position and velocity are recognized as:

$$\boldsymbol{\eta} = (x,\ y,\ z,\ \theta,\ \psi)^{\mathrm{T}},\ \ \boldsymbol{\nu} = (u,\ v,\ w,\ q,\ r)^{\mathrm{T}}, \tag{7.3}$$

where (x, y, z) is the vehicle's inertial position in Cartesian coordinates, θ is the pitch angle and ψ is the yaw angle. In addition, u is the surge velocity, v is the sway velocity, w is the heave velocity, q is the pitch rate and r is the yaw rate.

7.2.2 Kinematic Models

There are two kinematics models corresponding to (7.1) that are common to use. The main difference between them is the incorporation of the effects of ocean

currents. The first model considers only absolute velocities and is the following:

$$\dot{x} = u\cos(\psi)\cos(\theta) - v\sin(\psi) + w\cos(\psi)\sin(\theta), \tag{7.4}$$

$$\dot{y} = u\sin(\psi)\cos(\theta) + v\cos(\psi) + w\sin(\psi)\sin(\theta), \tag{7.5}$$

$$\dot{z} = -u\sin(\theta) + w\cos(\theta), \tag{7.6}$$

$$\dot{\theta} = q, \tag{7.7}$$

$$\dot{\psi} = \frac{1}{\cos(\theta)}r, \quad \cos(\theta) \neq 0. \tag{7.8}$$

As it is shown in [89], by incorporating relative velocities, the kinematics can be rewritten as:

$$\dot{x} = u_r\cos(\psi)\cos(\theta) - v_r\sin(\psi) + w_r\cos(\psi)\sin(\theta) + u_c^n, \tag{7.9}$$

$$\dot{y} = u_r\sin(\psi)\cos(\theta) + v_r\cos(\psi) + w_r\sin(\psi)\sin(\theta) + v_c^n, \tag{7.10}$$

$$\dot{z} = -u_r\sin(\theta) + w_r\cos(\theta) + w_c^n, \tag{7.11}$$

$$\dot{\theta} = q, \tag{7.12}$$

$$\dot{\psi} = \frac{1}{\cos(\theta)}r, \quad \cos(\theta) \neq 0. \tag{7.13}$$

where the relative velocities in surge, sway and heave (u_r, v_r, w_r) and their absolute velocities counterparts are related as follows:

$$u_r = u - u_c^b, \quad v_r = v - v_c^b, \quad w_r = w - w_c^b, \tag{7.14}$$

with (u_c^b, v_c^b, w_c^b) denoting the ocean current velocities in the body-fixed frame and (u_c^n, v_c^n, w_c^n) denoting the ocean current velocities in the NED frame.

It is worth noting that, even in the presence of ocean currents, it can be easily shown that the two kinematic models, (7.4)–(7.8) and (7.9)–(7.13) are equivalent. However, the overall analysis of the system, including the stability results of the LOS algorithm, can change a lot depending on which one of the two is adopted. The main criterion for choosing either one is the available measurements and what states of the system are controlled. For instance, if Global Navigation Satellite System (GNSS) measurements are available, then the absolute velocities are measured. Consequently, even if ocean current forces are present, it is possible to reduce the uncertainty and simplify the analysis by controlling the course angle and using conventional LOS guidance. For an underwater vehicle with only relative measurements available though, the relative velocities model with yaw control and an augmented LOS guidance law with integral action is is a more fitting option. These issues will be discussed in detail in Chapter 9.

7.3 LOS Guidance Designs for the Decoupled Horizontal and Vertical Planes

7.3.1 Introduction

This section deals with path-following in two dimensions by considering two independent and decoupled 3-DOF cases, namely, the horizontal plane guidance and
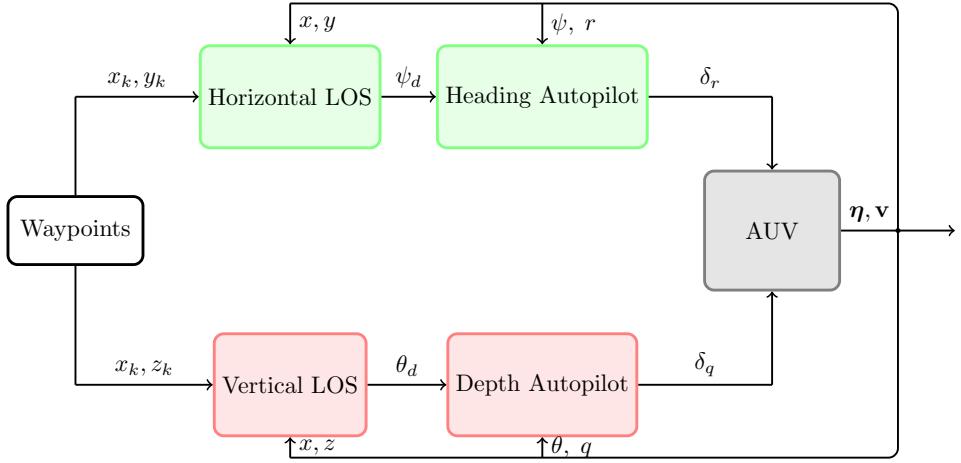
Figure 7.1: Block diagram of the decoupled LOS guidance for the horizontal and vertical planes.

the vertical plane guidance. The former is concerned with generating appropriate heading reference trajectories in order for the vehicle to converge to a straight line on the xy-plane and, similarly, the latter generates pitch reference trajectories in order to converge to a straight line on the xz-plane. For either of them a LOS guidance law that results in a κ-exponentially stable equilibrium point is derived. The reason for studying these decoupled cases is twofold: first, the horizontal plane case is exactly how the problem is stated and tackled for a surface vessel performing path-following, and second, breaking the problem in two parts contributes to presenting a more constructive approach and serves as a smooth transition to the 3-D case. In this section it is assumed that no environmental disturbances are present. Moreover, a discussion on the cascaded system formed by the LOS guidance law and the heading, or pitch, angle controller is included at the end of the section.

7.3.2 Horizontal-Plane Path Following

In the case of decoupled horizontal plane path-following we assume that $\theta = 0°$, consequently the kinematics equation to be considered is:

$$\dot{x} = u\cos(\psi) - v\sin(\psi), \tag{7.15}$$

$$\dot{y} = u\sin(\psi) + v\cos(\psi), \tag{7.16}$$

$$\dot{\psi} = r. \tag{7.17}$$

The horizontal speed U_h is given by:

$$U_h := \sqrt{u^2 + v^2}, \tag{7.18}$$

and is assumed to be positive and bounded:

$$U_{h,\min} \leq U_h \leq U_{h,\max}, \quad 0 < U_{h,\min}. \tag{7.19}$$

From (7.18)–(7.19) it is implied that the vessel always has at least a nonzero surge speed. The reason for setting a minimum positive speed $U_{h,\min}$ is related to the stability proof of the LOS guidance, as it is shown later in this section. The model (7.15)–(7.17) includes only absolute velocities and describes the motion of an underactuated vehicle since two out of three DOF's can be controlled independently, namely the yaw angle and the surge velocity.

Path Following Objective

Assuming that the vehicle is assigned to converge to the line connecting the waypoints WP_k–WP_{k+1}, the along-track and the cross-track error for a given vehicle position (x, y) are given by:

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} = \mathbf{R}^\top(\gamma_p) \begin{bmatrix} x - x_k \\ y - y_k \end{bmatrix}, \tag{7.20}$$

where (x_k, y_k) is the position of the k-th waypoint expressed in the NED frame, and the rotation matrix from the inertial frame to the path-fixed reference frame is given by:

$$\mathbf{R}(\gamma_p) = \begin{bmatrix} \cos(\gamma_p) & -\sin(\gamma_p) \\ \sin(\gamma_p) & \cos(\gamma_p) \end{bmatrix} \in SO(2). \tag{7.21}$$

Moreover,

$$x_e = (x - x_k)\cos(\gamma_p) + (y - y_k)\sin(\gamma_p), \tag{7.22}$$
$$y_e = -(x - x_k)\sin(\gamma_p) + (y - y_k)\cos(\gamma_p), \tag{7.23}$$

where γ_p is the horizontal path-tangential angle:

$$\gamma_p = \mathrm{atan2}(y_{k+1} - y_k, x_{k+1} - x_k), \tag{7.24}$$

where the two-argument function atan2 is a generalization of the $\arctan(y/x)$ that takes into account the signs of both x and y in order to determine the quadrant of the result, hence making it possible to distinguish between diametrically opposite directions. Finally, the associated control objective for horizontal plane straight-line path-following is:

$$\lim_{t \to +\infty} y_e(t) = 0. \tag{7.25}$$

Note that the along-track error x_e does not need to be minimized in a path-following scenario, the contrary is true for applications that impose temporal constraints.

Horizontal LOS Guidance Law

Figure 7.2 depicts the geometry of the LOS guidance and some of the main variables that are involved in the problem. Before moving on it is useful to mention that the LOS vector is often defined differently in marine applications compared to the definition adopted by the aircraft and missile communities. According to the definition in [233], the LOS is the line that starts at the reference point (that is, the aircraft or the missile) and passes through the objective of the guidance (i.e the
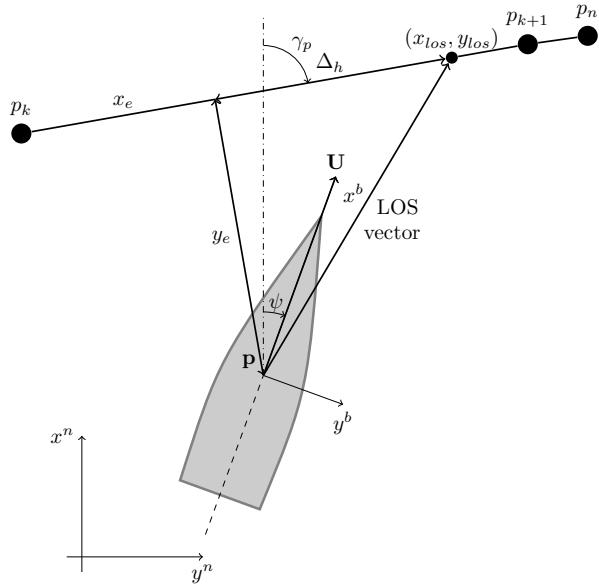
Figure 7.2: Line-of-sight guidance geometry for straight lines in the xy plane. Here the sideslip angle is equal to zero.

target). On the other hand, in marine guidance applications the LOS vector starts at the vessel and passes through a point $\mathbf{p}(x_{\text{los}}, y_{\text{los}})$ which is located on the path-tangential line at a lookahead distance $\Delta_h > 0$ ahead of the direct projection of the vessel's position $\mathbf{p}(x, y)$ on to the path. The latter definition seems to be more congruent with the path-following task and, as a consequence, in this chapter the lookahead-based steering method will be considered. Depending on the lookahead distance value, the maneuvering characteristics of the vehicle can vary significantly. More specifically, a low Δ_h value will induce more aggressive steering compared to a larger value. For this reason, algorithms proposing a time-varying Δ_h have been implemented in the past in order to obtain a more flexible behavior, see for instance [187], [179] and [155]. However, we will consider the different guidance laws with a constant Δ_h in order to reduce the overall complexity of the problem and be able to compare their performance more reliably. The lookahead-based guidance law is given by (see [41]):

$$\psi_d = \gamma_p + \arctan\left(\frac{-y_e}{\Delta_h}\right).$$

$$(7.26)$$

In the presence of external disturbances, or during turns, the heading angle ψ_d and the course angle χ_d are not aligned anymore and are related in the following way:

$$\chi_d = \psi_d + \beta,$$

$$(7.27)$$
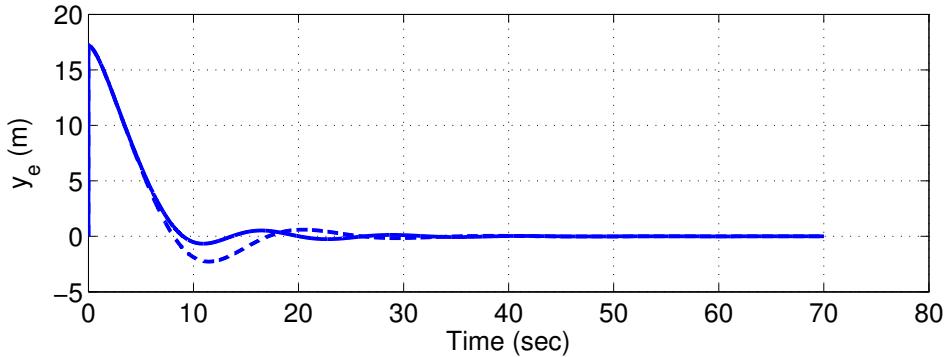
Figure 7.3: Cross-track error comparison with β compensation (solid line) and without compensating for β (dashed line)

and therefore the desired heading angle is:

$$\psi_d = \gamma_p + \arctan\left(\frac{-y_e}{\Delta_h}\right) - \beta. \tag{7.28}$$

By differentiating (7.23) with respect to time we get:

$$
\begin{aligned}
\dot{y}_e &= -\dot{x}\sin(\gamma_p) + \dot{y}\cos(\gamma_p), \\
&= -\left(u\cos(\psi) - v\sin(\psi)\right)\sin(\gamma_p) \\
&\quad + \left(u\sin(\psi) + v\cos(\psi)\right)\cos(\gamma_p), \\
&= u\sin(\psi - \gamma_p) + v\cos(\psi - \gamma_p), \tag{7.29}
\end{aligned}
$$

and by transforming (7.29) in amplitude-phase form we get:

$$
\begin{aligned}
\dot{y}_e &= \sqrt{u^2 + v^2}\sin(\psi - \gamma_p + \beta), \\
&= U_h\sin(\psi - \gamma_p + \beta), \tag{7.30}
\end{aligned}
$$

where

$$\beta = \operatorname{atan2}(v, u), \tag{7.31}$$

which is equal to the orientation of the vehicle's velocity vector U_h with respect to the body-fixed frame. In other words, (7.31) is the angle between the vehicle's velocity orientation and the vehicle's heading. This is the commonly known as sideslip, or drift, angle.

Proposition 1: Under the assumption that the desired heading is perfectly tracked such that $\psi = \psi_d$, the system (7.30) has a κ-exponentially stable equilibrium point at $y_e = 0$ if the desired heading angle is given by (7.28).

Proof. If we assume that the desired heading is perfectly tracked at all times and choose the desired heading angle as in (7.28), the derivative of the cross-track error
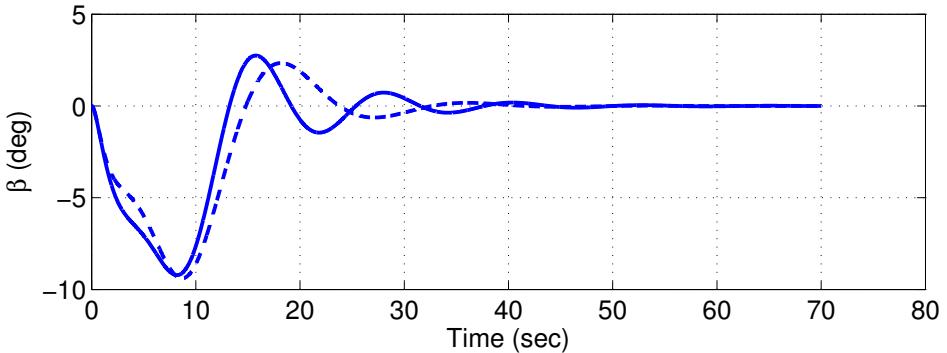
Figure 7.4: Sideslip angle comparison with β compensation (solid line) and without compensating for β

becomes:

$$\dot{y}_e = U_h \frac{-y_e}{\sqrt{\Delta_h^2 + y_e^2}}. \tag{7.32}$$

The Lyapunov Function Candidate (LFC)

$$V_1 = \frac{1}{2} y_e^2, \tag{7.33}$$

has the time-derivative:

$$\dot{V}_1 = U_h \frac{-y_e^2}{\sqrt{\Delta_h^2 + y_e^2}}. \tag{7.34}$$

which is negative for $U_h > 0$. Hence, the origin $y_e = 0$ is a UGAS equilibrium of the system (7.32). Moreover, on the ball $D_1 = \{y_e \in R | y_e| \leq \mu_1\}$, $\mu_1 > 0$, we have that

$$\dot{V}_1 = -\frac{U_h y_e^2}{\sqrt{\mu_1^2 + \Delta_h^2}} \leq -k_1 y_e^2, \tag{7.35}$$

for some $0 < k_1 < U_h/(\sqrt{\mu_1^2 + \Delta_h^2})$, which entails that the origin is a ULES equilibrium. The combination of UGAS and ULES implies global κ-exponential stability, as it was shown in [154]. □

Discussion regarding the sideslip angle β: It is worth clarifying that the sideslip angle that appears in (7.30) is not induced by any environmental forces, since in this section they are assumed to be zero. The sideslip angle in the present case occurs due to the nonzero sway velocity during a turn, this is what causes a difference between the orientation ψ (heading angle) of the surge velocity u and the orientation χ (course angle) of the total speed U_h. This is depicted in Fig. 7.5. When the vehicle is moving forward in an environment without external disturbances (i.e. the vessel following the x^n axis), the total velocity U_h is equal to the surge velocity u and there is no sideslip angle. However, during a turn (i.e the vessel moving on the curved path), a part of the total velocity is transferred into sway velocity and
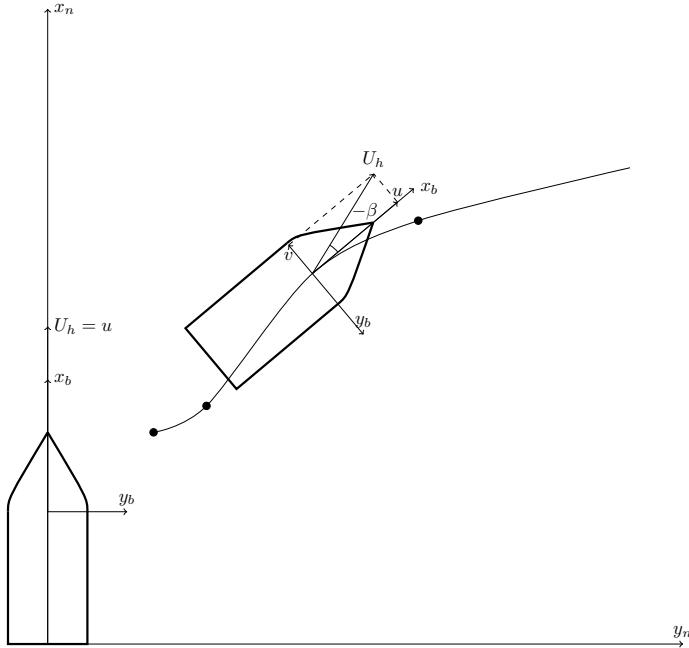
Figure 7.5: Sideslip angle during a turn. No external disturbances are present in this case.

the sideslip angle is nonzero. Consequently, it is this sideslip angle that this section refers to. It is worth noting that in the past there have been proofs of the LOS guidance without including this component of β, see for instance [30]. In that case it was still possible to prove stability because this effect is relatively small (although it can be larger for very agile maneuvering) and the related terms can be modeled as part of the interconnecting term of the cascade structure which was shown to satisfy the growth conditions given in [181]. Ideally, however, incorporating (7.31) in the desired heading command should increase performance, see Figs. 7.3 and 7.4.

Comment regarding the lower speed bound $U_{h,\min}$: It is also important to mention that in order for the proof above to be consistent with stability theory, it is necessary to set a specific positive lower bound $U_{h,\min}$ instead of just assuming that $0 < U_h$. This necessity occurs due to the fact that if the velocity is decreasing in a way such that it converges asymptotically to zero, then it is not possible to conclude that the system (7.32) converges.

7.3.3 Vertical-Plane Path Following

This problem is similar to the one presented in Section 7.3.2. In the case of decoupled vertical plane path-following of underwater vehicles, it is common to assume that the yaw angle $\psi = 0°$, consequently the kinematic equations to be considered

are:

$$\dot{x} = u\cos(\theta) - w\sin(\theta), \tag{7.36}$$

$$\dot{z} = -u\sin(\theta) + w\cos(\theta), \tag{7.37}$$

$$\dot{\theta} = q. \tag{7.38}$$

Apparently, for this problem the vertical speed is defined as:

$$U_{\mathrm{v}} := \sqrt{u^2 + w^2} > 0, \tag{7.39}$$

and is assumed to be positive and bounded:

$$U_{\mathrm{v,min}} \le U \le U_{\mathrm{v,max}}, \quad 0 < U_{\mathrm{v,min}}. \tag{7.40}$$

Path-Following Objective

Similarly to the case for surface vessels, we assume that the vehicle is supposed to converge to the line connecting the waypoints WP_k–WP_{k+1}, the along-track and the cross-track error for a given vehicle position (x, z) are given by:

$$\begin{bmatrix} x_e \\ z_e \end{bmatrix} = \mathbf{R}^\top(\alpha_p) \begin{bmatrix} x - x_k \\ z - z_k \end{bmatrix}, \tag{7.41}$$

where (x_k, z_k) is the position of the k-th waypoint expressed in the NED frame, and the rotation matrix from the inertial frame to the path-tangential frame is given by:

$$\mathbf{R}(\alpha_p) = \begin{bmatrix} \cos(\alpha_p) & \sin(\alpha_p) \\ -\sin(\alpha_p) & \cos(\alpha_p) \end{bmatrix} \in SO(2). \tag{7.42}$$

Moreover,

$$x_e = (x - x_k)\cos(\alpha_p) - (z - z_k)\sin(\alpha_p), \tag{7.43}$$

$$z_e = (x - x_k)\sin(\alpha_p) + (z - z_k)\cos(\alpha_p), \tag{7.44}$$

where α_p is the vertical path-tangential angle:

$$\alpha_p = \mathrm{atan2}(-(z_{k+1} - z_k), (x_{k+1} - x_k)). \tag{7.45}$$

Consequently, the associated control objective for vertical plane straight-line path-following is:

$$\lim_{t \to +\infty} z_e(t) = 0. \tag{7.46}$$

Vertical LOS Guidance Law

The time-derivative of the vertical cross-track error gives:

$$\begin{aligned} \dot{z}_e =& \dot{x}\sin(\alpha_p) + \dot{z}\cos(\alpha_p), \\ =& -(u\cos(\theta) - w\sin(\theta))\sin(\alpha_p) \\ & -(u\sin(\theta) - w\cos(\theta))\cos(\alpha_p), \\ =& \sqrt{u^2 + w^2}\cos(\theta - \mathrm{atan2}(w, u))\sin(\alpha_p) \\ & - \sqrt{u^2 + w^2}\sin(\theta - \mathrm{atan2}(w, u))\cos(\alpha_p), \\ =& -U_{\mathrm{v}}[\sin(\theta - \alpha)\cos(\alpha_p) - \cos(\theta - \alpha)\sin(\alpha_p)] \end{aligned} \tag{7.47}$$

and by transforming (7.47) according to the common trigonometric property regarding the sinus of a difference of angles:

$$\dot{z}_e = -U_v \sin(\theta - \alpha - \alpha_p), \tag{7.48}$$

where

$$\alpha = \mathrm{atan2}(w, u) \tag{7.49}$$

is the commonly known as the angle of attack. Similarly to the horizontal cross-track error case, we assume that the desired pitch angle is perfectly tracked at all times and choose the desired pitch angle as:

$$\theta_d = \alpha_p + \alpha + \arctan\left(\frac{z_e}{\Delta_v}\right), \tag{7.50}$$

with Δ_v denoting the lookahead distance for the vertical-plane LOS.

Proposition 2: Under the assumption that the desired pitch angle is perfectly tracked such that $\theta = \theta_d$, The system (7.48) has a κ-exponentially stable equilibrium point at $z_e = 0$ if the desired pitch angle is given by (7.50).

Proof. Under the assumption of perfect pitch angle tracking, adopting (7.50) as the desired pitch angle gives the following derivative of the vertical cross-track error:

$$\dot{z}_e = U_v \frac{-z_e}{\sqrt{\Delta_v^2 + z_e^2}}. \tag{7.51}$$

The Lyapunov Function Candidate (LFC)

$$V_2 = \frac{1}{2}z_e^2, \tag{7.52}$$

has the time-derivative:

$$\dot{V}_2 = U_v \frac{-z_e^2}{\sqrt{\Delta_v^2 + z_e^2}}. \tag{7.53}$$

which is negative for $U_v > 0$. Hence, the origin $z_e = 0$ is a UGAS equilibrium of the system (7.51). Moreover, on the ball $D_2 = \{z_e \in R | |z_e| \le \mu_2\}$, $\mu_2 > 0$, we have that

$$\dot{V}_2 = -\frac{U_v z_e^2}{\sqrt{\mu_2^2 + \Delta_v^2}} \le -k_2 z_e^2, \tag{7.54}$$

for some $0 < k_2 < U_v/(\sqrt{\mu_2^2 + \Delta_v^2})$, which implies that the origin is a ULES equilibrium point and therefore a κ-exponentially stable equilibrium point. □

The same comments and observations mentioned at the end of Section 7.3.2 apply in this case too for both the total speed U_v and the angle of attack α.
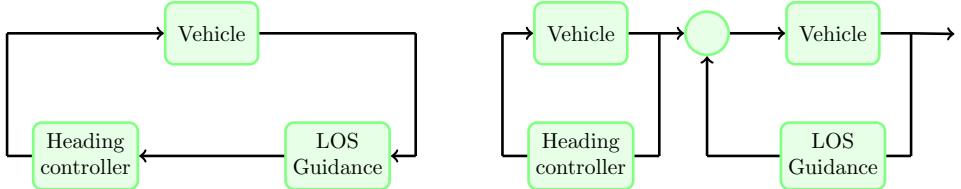
Figure 7.6: Cascade system equivalence between the horizontal LOS guidance and the heading controller. The total system can be studied as a cascade structure where the perturbing system (vehicle + heading controller) affects the convergence of the perturbed system (vehicle + LOS guidance) via the heading error dynamics.

7.3.4 Cascade Structure Formed by the Horizontal/Vertical LOS Guidance System and the Heading/Pitch Controller

In the previous sections the assumption of perfectly tracked heading (or pitch, henceforth this clarification will be omitted), such as $\psi = \psi_d$, has been mentioned several times and the stability results have relied on this in order to hold true. Although such an assumption is not congruent with reality, it is not an oversimplification because the guidance system is seen as one of the two systems of a cascaded system. The other subsystem is the heading dynamics, which, apparently, includes the heading controller. Therefore when analyzing the stability of the cascade structure, the theorems and the procedure introduced in [181] are used. Part of this procedure is to analyze the stability of each system independently before taking into account the interconnection between them. Sections 7.3.2 and 7.3.3 dealt with analyzing only the guidance system. The remaining steps depend on the heading autopilot and are outside the scope of this section, examples can be found in [30] and [155]. A depiction of the cascade systems equivalence is shown in Fig. 7.6. In Chapters 8–10, the cascade interconnection will be studied in details for a number of cases.

7.4 LOS Guidance Design for 3-D Coupled Motions

7.4.1 Introduction

This section considers a 3-D motion control scenario where an underwater vehicle is assigned to achieve straight-line path following at a predetermined depth. Contrary to Section 7.3, the horizontal LOS guidance is now coupled with the vertical motion of the AUV. Hence 5-DOF's are considered and the kinematics is given by (7.4)–(7.8) The total speed in this case is:

$$U := \sqrt{u^2 + v^2 + w^2}, \tag{7.55}$$

where, similarly to the decoupled planar cases of Section 7.3:

$$U_{\min} \leq U \leq U_{\max}, \quad 0 < U_{\min}. \tag{7.56}$$
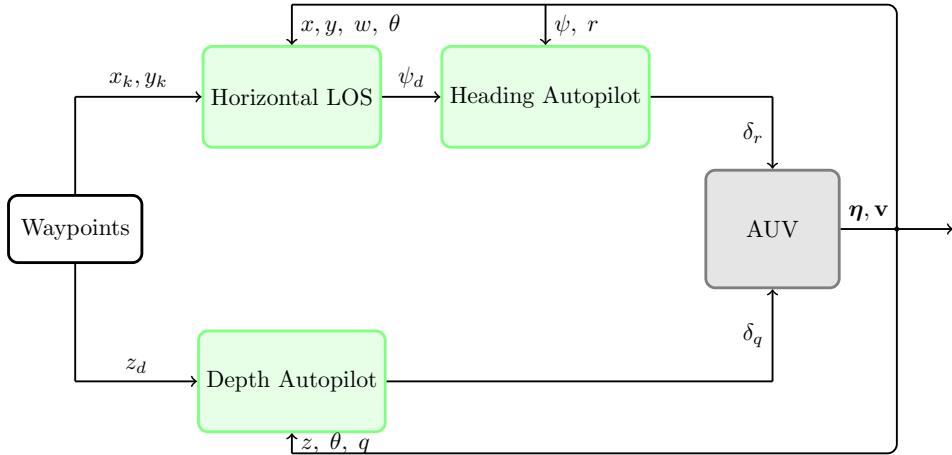
Figure 7.7: Block diagram of the 3-D path following scenario. The horizontal plane motion (AUV + LOS guidance) is perturbed by the vertical plane motion (AUV + depth controller). The latter, however, is independent.

Comment on the system's structure: The task is to converge to a horizontal straight line at a depth $z = z_d$. Consequently, it is reasonable to consider the two following subsystems in order to solve the problem: a) a depth controller for (7.6)–(7.7), and b) a LOS guidance for (7.4)–(7.5) and (7.8). From (7.6)–(7.7) we observe that the heave-pitch subsystem is uncoupled with respect to the rest of the states, whereas (7.4)–(7.5) and (7.8) indicates that the horizontal motion is coupled with the vertical plane motion via the pitch angle θ and the heave velocity w. A block diagram of the total system structure can be seen in Fig. 7.7.

7.4.2   First Subsystem: Horizontal LOS Guidance Excited by the Heave and Pitch Dynamics

The second subsystem is given by the equations:

$$\dot{x} = u \cos(\psi) \cos(\theta) - v \sin(\psi) + w \cos(\psi) \sin(\theta), \tag{7.57}$$

$$\dot{y} = u \sin(\psi) \cos(\theta) + v \cos(\psi) + w \sin(\psi) \sin(\theta), \tag{7.58}$$

$$\dot{\psi} = \frac{1}{\cos(\theta)} r, \quad \cos(\theta) \neq 0. \tag{7.59}$$

This system is apparently different compared to the horizontal LOS solved in Section 7.3.2 because the system is now coupled with the depth control system via the pitch angle θ and the heave velocity w. The cross-track error is still given by:

$$y_e = -(x - x_k) \sin(\gamma_p) + (y - y_k) \cos(\gamma_p), \tag{7.60}$$

however, in this case it propagates differently due to the coupling with θ and w:

$$
\begin{aligned}
\dot{y}_e ={}& - \dot{x}\sin(\gamma_p) + \dot{y}\cos(\gamma_p), \\
={}& - u\cos(\psi)\cos(\theta)\sin(\gamma_p) + v\sin(\psi)\sin(\gamma_p) \\
& - w\cos(\psi)\sin(\theta)\sin(\gamma_p) + u\sin(\psi)\cos(\theta)\cos(\gamma_p) \\
& + v\cos(\psi)\cos(\gamma_p) + w\sin(\psi)\sin(\theta)\cos(\gamma_p).
\end{aligned}
\tag{7.61}
$$

Using several trigonometric identities yields:

$$
\begin{aligned}
\dot{y}_e ={}& - \cos(\psi)\sin(\gamma_p)[u\cos(\theta) + w\sin(\theta)] \\
& + \sin(\psi)\cos(\gamma_p)[u\cos(\theta) + w\sin(\theta)] + v\cos(\psi - \gamma_p), \\
={}& \sqrt{u^2 + w^2}\cos(\theta - \alpha)[\sin(\psi)\cos(\gamma_p) - \cos(\psi)\sin(\gamma_p)] + v\cos(\psi - \gamma_p), \\
={}& U_{\mathrm{v}}\cos(\theta - \alpha)\sin(\psi - \gamma_p) + v\cos(\psi - \gamma_p),
\end{aligned}
$$

and, finally, we have the compact form:

$$
\dot{y}_e = \sqrt{(U_{\mathrm{v}}^2 \cos^2(\theta - \alpha) + v^2)}\sin(\psi - \gamma_p + \beta_{\mathrm{v}})
\tag{7.62}
$$

where:

$$
\beta_{\mathrm{v}} = \operatorname{atan2}(v, U_{\mathrm{v}}\cos(\theta - \alpha)).
\tag{7.63}
$$

This result shows that the first subsystem (it intervenes here through θ and w) does not prevent the minimization of the horizontal cross-track error as long as the yaw controller can compensate for the generalized sideslip angle β_{v}. This agrees with intuition because if the target line is at a high depth, the horizontal LOS guidance system will probably converge first while the AUV continues to submerge. The desired yaw angle is:

$$
\psi_d = \gamma_p - \beta_{\mathrm{v}} + \arctan\left(\frac{-y_e}{\Delta_h}\right).
\tag{7.64}
$$

Before proceeding with the theorem and the proof pertinent to the stability of (7.62), the following assumptions are made:

- *A1:* The desired heading ψ_d is perfectly tracked at all times.

- *A2:* The depth controller ensures that w and θ are bounded states.

- *A3:* The pitch angle satisfies the condition $\theta \neq \pm\dfrac{\pi}{2}$.

Theorem 1: The system (7.62) has a globally κ-exponentially stable equilibrium point at $y_e = 0$ if all the assumptions A1-A3 are satisfied and the desired heading angle is given by (7.64).

Proof. This is similar to the cases studied in Section 7.3. The Lyapunov Function Candidate (LFC):

$$
V_3 = \frac{1}{2}y_e^2,
\tag{7.65}
$$

has the time-derivative:

$$\dot{V}_3 = \sqrt{(U_v^2 \cos^2 (\theta - \alpha) + v^2)} \frac{-y_e^2}{\sqrt{\Delta_h^2 + y_e^2}}. \tag{7.66}$$

which is negative. Hence, the origin $y_e = 0$ is a UGAS equilibrium of the system. In addition, on the ball $D_3 = \{y_e \in R | y_e| \leq \mu_3\}$, $\mu_3 > 0$, we have that

$$\dot{V}_3 = -\frac{\sqrt{(U_v^2 \cos^2 (\theta - \alpha) + v^2)} y_e^2}{\sqrt{\mu_3^2 + \Delta_h^2}} \leq -k_3 y_e^2, \tag{7.67}$$

for some $0 < k_3 < \sqrt{(U_v^2 \cos^2 (\theta - \alpha) + v^2)/(\mu_3^2 + \Delta_h^2)}$, which implies that the origin is a ULES equilibrium point and therefore a κ-exponentially stable equilibrium point. □

7.4.3 Second Subsystem: Depth Control

This section discusses two popular approaches when it comes to depth control of AUVs, namely, PID control and sliding mode control. Note that several other methodologies have been studied in the past, such as adaptive linear controllers [162], model predictive controllers [214] and H_∞ control [99], [163], [206]. More detailed information on similar control techniques for AUVs can be found in [153]. At the end of the section we discuss the possibility of the depth reference trajectories being generated by a vertical LOS guidance algorithm. Although such a concept can be characterized as redundant when the overall task of the depth controller is to achieve setpoint regulation, it constitutes an important step toward an even more general path-following scenario where the task is to converge to any straight line in space (i.e. γ_p, $\alpha_p \neq 0$), hence a time-varying depth reference signal is required.

Conventional PID Controllers

Controlling depth using PID controllers is a common approach that has been reported in the literature, see for instance [127], [231], [170]. Usually, this approach employs a linearized model for the pitch dynamics and conventional PID controllers are tuned properly in order to stabilize the system. As it is always the case with linearized systems, the approach might fail for large pitch angles, or time-varying velocity. However, if the linearization assumptions are satisfied then the PID controller will result theoretically in a GES equilibrium point. Experimental tests using PID controllers for depth control have reported satisfactory performance and this fact in combination with their simplicity makes them a more attractive approach compared to a more complicated one like, for instance, H_∞ control [153].

Sliding Mode Controllers

Depth control using sliding modes is also a methodology that has been implemented extensively in the past, an early reference is that of [74], in-water tests were reported in [119], also a well-known approach was the one presented in [118]. This controller is based on the concept of the sliding surface to increase robustness and its goal

is to provide good performance even under the presence of modeling uncertainties and environmental disturbances. A sliding mode controller demonstrates better performance and is faster compared to a PID controller, but the downside is that it is more complicated and requires a complete model. An example of a pitch and depth autopilot for underwater vehicles based on sliding mode control can be found in [88, pp. 526–528].

Vertical LOS for Depth Control

Adopting vertical LOS coupled with horizontal LOS is the next reasonable step in order to achieve path following for any straight path in space, with the exception of $\alpha_p \pm \pi/2$ due to the singularity of the Euler angle representation. It is possible to avoid this singularity by using quaternions. Due to the fact that the kinematics (7.6)–(7.7) is uncoupled, the analysis from Section 7.3.3 still holds. Therefore, combining (7.62) and (7.50) and using the property

$$\cos\left(\arctan(x)\right) = 1/(\sqrt{1+x^2}), \tag{7.68}$$

gives:

$$\dot{y}_e = \frac{U_\mathrm{v}\Delta_\mathrm{v}}{\sqrt{\Delta_\mathrm{v}^2 + z_e^2}} \sin\left(\psi - \gamma_p\right) + v\cos\left(\psi - \gamma_p\right), \tag{7.69}$$

and finally we get the compact form:

$$\dot{y}_e = \sqrt{\left(\frac{U_\mathrm{v}^2\Delta_\mathrm{v}^2}{\Delta_\mathrm{v}^2 + z_e^2} + v^2\right)} \sin\left(\psi - \gamma_p + \beta_\mathrm{v}\right), \tag{7.70}$$

with

$$\beta_\mathrm{v} = \mathrm{atan2}\left(v, \frac{U_\mathrm{v}\Delta_\mathrm{v}}{\sqrt{\Delta_\mathrm{v}^2 + z_e^2}}\right). \tag{7.71}$$

As expected, (7.70) shows that the horizontal cross-track error is now a function of the vertical cross-track error as well. Note, however, that z_e does not prevent y_e from converging because the stability result for the vertical LOS guarantees that z_e is bounded and has a κ-exponentially stable equilibrium point at $z_e = 0$. Eq. (7.71) shows an generalized formulation of the sideslip angle. Similarly to the 2-D sideslip angle case, this has an effect if the sway motion is nonzero, i.e. on a turn. Apparently, the system (7.70) is stabilized with a heading command:

$$\psi_d = \gamma_p - \beta_\mathrm{v} + \arctan\left(\frac{-y_e}{\Delta_h}\right). \tag{7.72}$$

This can be shown similarly to the previous proofs, for a LFC:

$$V_4 = \frac{1}{2}y_e^2, \tag{7.73}$$

we derivate V_4 w.r.t time and use (7.72) as the desired heading command. The final result is:

$$\dot{V}_4 = -\sqrt{\left(\frac{U_\mathrm{v}^2\Delta_\mathrm{v}^2}{\Delta_\mathrm{v}^2 + z_e^2} + v^2\right)}\left(\frac{y_e^2}{\Delta_h^2 + y_e^2}\right). \tag{7.74}$$

7.5 A Quaternion-Based LOS Guidance Scheme

7.5.1 Introduction

It is a well known fact that, in the general case, the Euler angles representation of the 6-DOF kinematics involves singularities for the pitch angles $\theta = \pm 90°$ [88]. Consequently, it is useful to derive a quaternion version of the conventional LOS guidance for AUVs. Moreover, the quaternion representation is more computationally efficient compared to Euler angles since it does not include trigonometric functions. This makes it even more suitable for applications involving unmanned vehicles where the on-board computational power might be more limited. This paper serves as the first step toward this direction. The quaternion representation of the LOS algorithm is derived for two uncoupled 3-DOF cases: a) the horizontal xy-plane, and b) the vertical zx-plane, and for a coupled case where the sideslip angle is also a function of the vertical motion. For each case, the terms of the guidance law are transformed from Euler angles to quaternion by taking into account the nature of quaternions that correspond to rotations and using simple trigonometric identities.

7.5.2 Horizontal LOS Guidance in Quaternion Form

This section deals with transforming the horizontal LOS guidance (7.28) in quaternion form. A quaternion q_ψ representing a rotation of an angle ψ around the z-axis can be expressed as follows [145]:

$$q_\psi = \underbrace{\cos\left(\frac{\psi}{2}\right)}_{\eta} + \mathbf{k}\underbrace{\sin\left(\frac{\psi}{2}\right)}_{\varepsilon_3}. \tag{7.75}$$

Transforming ψ_r to the quaternion q_{ψ_r}

Before moving on, it must specified that $\psi_r = \arctan(-y_e/\Delta)$. First we define:

$$y_e^2 + \Delta_h^2 := R_h^2 \tag{7.76}$$

$$\sin\left(\psi_r\right) = \frac{-y_e}{R_h}, \quad \cos\left(\psi_r\right) = \frac{\Delta_h}{R_h}, \tag{7.77}$$

Then, since

$$\sin\left(\psi_r\right) = 2\sin\left(\frac{\psi_r}{2}\right)\cos\left(\frac{\psi_r}{2}\right), \tag{7.78}$$

$$\cos\left(\psi_r\right) = 2\cos^2\left(\frac{\psi_r}{2}\right) - 1, \tag{7.79}$$

we get:

$$\sin\left(\psi_r\right) = 2\varepsilon_3\eta, \tag{7.80}$$

$$\cos\left(\psi_r\right) = 2\eta^2 - 1. \tag{7.81}$$

Combining (7.76)–(7.77) with (7.78)–(7.79) yields:

$$2\varepsilon_{3_r}\eta_r = \frac{-y_e}{\sqrt{\Delta_h^2 + y_e^2}}, \tag{7.82}$$

$$2\eta_r^2 - 1 = \frac{\Delta_h}{\sqrt{\Delta_h^2 + y_e^2}}, \tag{7.83}$$

and solving w.r.t. η_r and ε_{3_r} gives:

$$\eta_r = \pm\sqrt{\frac{\Delta_h + R_h}{2R_h}}, \tag{7.84}$$

$$\varepsilon_{3_r} = \frac{-y_e}{2R_h\eta} \tag{7.85}$$

$$= \frac{-y_e}{\sqrt{2R_h(\Delta_h + R_h)}}. \tag{7.86}$$

It is important to note that either the positive or negative value for η_r can be chosen, hence affecting the sign of ε_r accordingly. Both cases correspond to the same rotation, as it is shown by (2.11). Therefore, the corresponding quaternion, q_{ψ_r} is written:

$$q_{\psi_r} = \sqrt{\frac{\Delta_h + R_h}{2R_h}} + \mathbf{k}\frac{-y_e}{\sqrt{2R(\Delta_h + R)}}. \tag{7.87}$$

It can be confirmed that (7.87) corresponds to a rotation quaternion by computing the norm:

$$\|q\| = \eta^2 + \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon} \tag{7.88}$$

$$= \frac{\Delta_h + R_h}{2R_h} + \frac{y_e^2}{2R_h(\Delta_h + R_h)} \tag{7.89}$$

$$= 1. \tag{7.90}$$

Transforming γ_p to the quaternion q_{γ_p}

Similarly to the calculations for q_{ψ_r} we have:

$$q_{\gamma_p} = \cos\left(\frac{\gamma_p}{2}\right) + \mathbf{k}\sin\left(\frac{\gamma_p}{2}\right), \tag{7.91}$$

$$\gamma_p = \mathrm{atan2}\left(\frac{\Delta y}{\Delta x}\right), \tag{7.92}$$

and, for the sake of notational brevity, we define:

$$R_{xy} = \sqrt{\Delta x^2 + \Delta y^2}, \tag{7.93}$$

consequently the quaternion transformation gives:

$$\eta_{\gamma_p} = \pm\sqrt{\frac{\Delta x + R_{xy}}{2R_{xy}}}, \tag{7.94}$$

$$\varepsilon_{3_{\gamma_p}} = \frac{\Delta y}{\sqrt{2R_{xy}(\Delta x + R_{xy})}}. \tag{7.95}$$

Transforming β

Implementing the same procedure for the sideslip angle β yields:

$$q_\beta = \cos\left(\frac{\beta}{2}\right) + \mathbf{k}\sin\left(\frac{\beta}{2}\right), \tag{7.96}$$

$$\beta = \text{atan2}\left(\frac{v}{u}\right). \tag{7.97}$$

which gives:

$$\eta_\beta = \pm\sqrt{\frac{u+U}{2U}}, \tag{7.98}$$

$$\varepsilon_{3_\beta} = \frac{v}{\sqrt{2U(u+U)}}. \tag{7.99}$$

Horizontal Rotation Quaternion

A quaternion-based control law system will make use of the horizontal rotation quaternion, q_h, which is given by the quaternion products of q_{ψ_r}, q_β, q_{γ_p}. Therefore:

$$q_h = q_{\psi_r} \otimes q_\beta \otimes q_{\gamma_p}. \tag{7.100}$$

Due to the fact that all the rotations considered in (7.100) are around the z-axis, the order of the multiplication is not important. Note, however that this is a special case.

7.5.3 Vertical LOS Guidance in Quaternion Form

The transformation procedure is similar to the one in Section 7.5.2. A quaternion q_θ representing a rotation of an angle θ around the y-axis can be expressed as follows [145]:

$$q_\theta = \underbrace{\cos\left(\frac{\theta}{2}\right)}_{\eta} + \underbrace{\mathbf{j}\sin\left(\frac{\theta}{2}\right)}_{\varepsilon_3}. \tag{7.101}$$

Transforming θ_r to the quaternion q_{θ_r}

$$z_e^2 + \Delta_v^2 = R_v^2 \tag{7.102}$$

$$\sin(\theta_r) = \frac{z_e}{R_v}, \quad \cos(\theta_r) = \frac{\Delta_v}{R_v}, \tag{7.103}$$

Then, since

$$\sin(\theta_r) = 2\sin\left(\frac{\theta_r}{2}\right)\cos\left(\frac{\theta_r}{2}\right), \tag{7.104}$$

$$\cos(\theta_r) = 2\cos^2\left(\frac{\theta_r}{2}\right) - 1, \tag{7.105}$$

we get:

$$\sin(\theta_r) = 2\varepsilon_3\eta, \tag{7.106}$$

$$\cos(\theta_r) = 2\eta^2 - 1. \tag{7.107}$$

Combining (7.102)–(7.103) with (7.104)–(7.105) yields:

$$2\varepsilon_{3_r}\eta_r = \frac{z_e}{\sqrt{\Delta_v^2 + z_e^2}}, \tag{7.108}$$

$$2\eta_r^2 - 1 = \frac{\Delta_h}{\sqrt{\Delta_v^2 + z_e^2}}, \tag{7.109}$$

and solving w.r.t. η_r and ε_{3_r} gives:

$$\eta_r = \pm\sqrt{\frac{\Delta_v + R_v}{2R_v}}, \tag{7.110}$$

$$\varepsilon_{3_r} = \frac{z_e}{2R_v\eta} \tag{7.111}$$

$$= \frac{z_e}{\sqrt{2R_v(\Delta_v + R_v)}}. \tag{7.112}$$

Therefore, the corresponding quaternion, q_{θ_r} is written:

$$q_r = \sqrt{\frac{\Delta_v + R_v}{2R_v}} + \mathbf{j}\frac{z_e}{\sqrt{2R(\Delta_v + R_v)}}. \tag{7.113}$$

Transforming α_p to the quaternion q_{α_p}

Similarly to Section 7.5.3 we have:

$$q_{\alpha_p} = \cos\left(\frac{\alpha_p}{2}\right) + \mathbf{j}\sin\left(\frac{\alpha_p}{2}\right), \tag{7.114}$$

$$\alpha_p = \text{atan2}\left(\frac{-\Delta z}{\Delta x}\right), \tag{7.115}$$

and, for the sake of notational brevity, we define:

$$R_{zx} = \sqrt{\Delta z^2 + \Delta x^2}, \tag{7.116}$$

consequently the quaternion transformation gives:

$$\eta_\alpha = \pm\sqrt{\frac{\Delta x + R_{zx}}{2R_{zx}}}, \tag{7.117}$$

$$\varepsilon_{3_{\alpha_p}} = \frac{\Delta x}{\sqrt{2R_{zx}(\Delta z + R_{zx})}}. \tag{7.118}$$

Transforming α to the quaternion q_α

Implementing the same procedure for the angle of attack α yields:

$$q_\alpha = \cos\left(\frac{\alpha}{2}\right) + \mathbf{j}\sin\left(\frac{\alpha}{2}\right), \tag{7.119}$$

$$\alpha = \text{atan2}\left(\frac{w}{u}\right). \tag{7.120}$$

which gives:

$$\eta_\alpha = \pm\sqrt{\frac{u + U_\text{v}}{2U_\text{v}}}, \tag{7.121}$$

$$\varepsilon_{3_\alpha} = \frac{w}{\sqrt{2U_\text{v}(u + U_\text{v})}}. \tag{7.122}$$

Vertical Rotation Quaternion

As was the case in the previous section, we now calculate the quaternion q_v that describes the total rotation around the y-axis for the vertical LOS case:

$$q_\text{v} = q_{\theta_r} \otimes q_\alpha \otimes q_{\alpha_p}. \tag{7.123}$$

7.5.4   LOS Guidance for 3-D Coupled Motions in Quaternion Form

As discussed earlier, the only difference in this case is the coupled sideslip angle β_v. Assuming perfect pitch tracking ($\theta = \theta_d$), we have from (7.50):

$$\begin{aligned} \theta - \alpha &= \arctan\left(-\Delta z/\Delta x\right) + \arctan\left(z_e/\Delta_\text{v}\right), \\ &= \arctan\left(\underbrace{(z_e\Delta x - \Delta_\text{v}\Delta z)/(\Delta x\Delta_\text{v} + z_e\Delta z)}_{\lambda_1}\right), \end{aligned} \tag{7.124}$$

then we calculate the quantity:

$$\lambda_2 := \cos(\theta - \alpha) = 1/\left(\sqrt{1 + \lambda_1^2}\right). \tag{7.125}$$

The transformation in this case gives:

$$\eta_{\beta_\text{v}} = \pm\sqrt{\frac{U_\text{v}\lambda_2 + \zeta}{2\zeta}}, \tag{7.126}$$

$$\varepsilon_{3_{\beta_\text{v}}} = \frac{\nu}{\sqrt{2\zeta(U_\text{v}\lambda_2 + \zeta)}}. \tag{7.127}$$

7.5.5   Simulations

In order to test the validity of the transformation methodology presented in the previous sections, a simple numerical example was constructed and implemented.
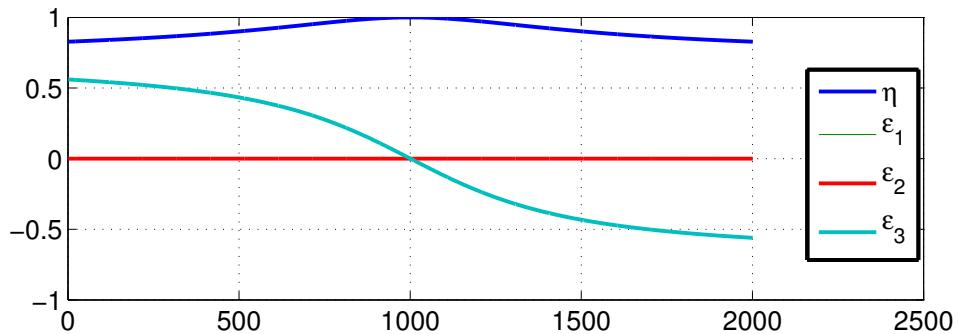
Figure 7.8: Quaternion elements computed by the transformation algorithm.
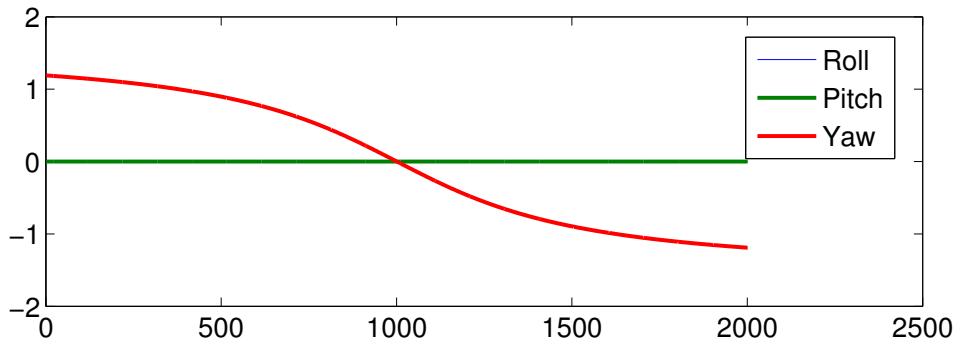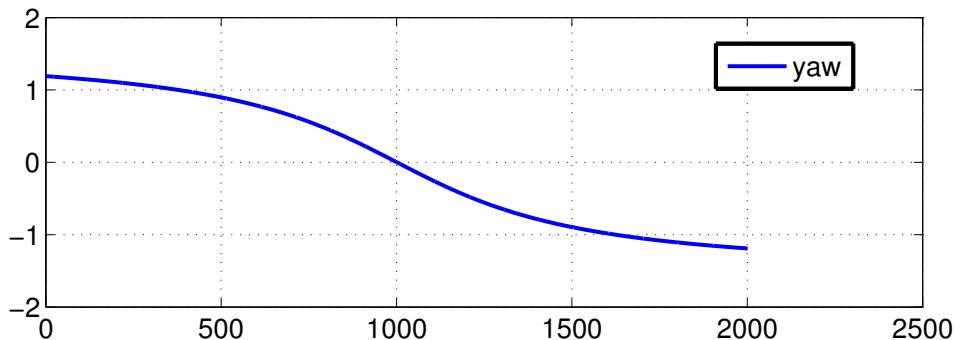
Figure 7.9: Euler angles corresponding to the quaternion computed by the transformation algorithm

.

Figure 7.10: Euler angles computed by $\arctan(-y_e/\Delta_h)$

.

The lookahead distance was chosen as $\Delta_h = 4$ m and the cross-track error increased linearly from -10 m to $+10$ m with a step of .01 m, hence generating 2001 samples. In Fig. 7.8 the quaternion computed by (7.87) is presented and Fig. 7.9 shows the corresponding Euler angles. It can be observed that the computed quaternion corresponds to a rotation around the z-axis. Moreover, Fig. 7.10 shows the values computed by the term $\arctan(-y_e/\Delta_h)$ and the result is that the plot is the same as the one in Fig. 7.9, hence confirming the validity of the transformations leading to (7.87).

7.6 Conclusions

This chapter has presented the line-of-sight guidance for a number of cases, both in 2-D and 3-D space. First, two uncoupled 2-D cases were studied, followed by a 3-D coupled case. It was shown that the sideslip angle can be affected because of the coupling. Stability results were given for all cases. Then the problem of the interconnection between the LOS guidance system and the vehicle's control system was discussed. This last section presented a simple technique in order to transform the conventional LOS guidance algorithm in quaternion form. The main motivation is to avoid singularities in the more general 6-DOF case and also get a more computationally-efficient algorithm. The transformation methodology was presented here for the uncoupled motions in the horizontal and vertical planes, as well as a coupled case. Future work includes the transformation of the LOS guidance corresponding to the coupled 6-DOF motions.

Chapter 8

Curved-Path LOS Guidance with Time-Varying Lookahead Distance

8.1 Introduction

In the previous chapter, the LOS guidance problem was presented for a number of cases in its simplest form, that is, without considering curved paths, dealing with the stability of the guidance-controller cascade system, or other aspects of the problem. Moreover, performance improvement can be achieved by modifying the initial method. A possible modification is to use a time-varying lookahead distance so as to get a more efficient vehicle maneuvering behavior. As mentioned earlier, the main task of the LOS algorithm is to steer the vehicle in a way such that it stays on the path, or lead it toward the path if the cross-track error (that is, the shortest distance to the path) is nonzero. Therefore, for a vehicle with a nonzero forward velocity, the guidance system in combination with the attitude controller should result in a stable overall system in order to ensure convergence to the desired trajectory. There is a vast literature regarding the LOS guidance, since it is one of the most popular methods adopted by the marine community (see also [93]) along with the pure pursuit guidance and the constant bearing guidance, all these methods are presented in detail in [35]. Optimal methods have also appeared in the literature, for more details the reader is referred to [199] and [200].

In [118], the authors addressed the problem of guiding an underwater vehicle with a line-of-sight (LOS) guidance algorithm and designing a sliding mode controller for stabilizing the combined speed, steering and diving response. In [36], a guidance-based path-following scheme which is singularity-free for all regular paths in conjunction with a nonlinear backstepping controller was presented and Uniform Global Asymptotic Stability (UGAS) and Uniformly Local Exponential Stability (ULES) was proven by using cascade theory. A cascaded systems approach was also developed by [30] where the guidance system is interconnected with a sliding mode controller in order to achieve global κ-exponential stability to straight lines in 3-D space. The concept of κ-exponential stability implies both UGAS and ULES and was introduced in [213]. A control strategy for formation control and straight-line path following for underactuated surface vessels was developed in [32]

where the cascade system formed by the closed-loop path following dynamics and the closed-loop synchronization dynamics was shown to be UGAS and ULES. In all the aforementioned approaches a constant lookahead distance was considered. The importance of using a variable lookahead distance was demonstrated in the κ-exponentially stable guidance laws which were developed at a kinematics level and proposed in [38]. The work of [187] presented a UGAS nonlinear Model Predictive Control (MPC) approach where the lookahead distance, Δ, was optimized in order to achieve a combination of fast convergence and small overshoot compared to the constant lookahead LOS algorithms. In [179], the LOS guidance parameter was embedded in the linear MPC controller design as as additional decision variable and the simulations indicated a smoother convergence to a straight line compared to the linear MPC controller with a fixed lookahead distance. In [155] a different direction was followed by proposing a time-varying lookahead distance Δ equation which is dependent on the cross-track error. This results in lower values for Δ (and thus a more agile and aggressive response) when the vehicle is far from the desired path, and greater Δ values when the vehicle is closer to the path and less abrupt behavior is needed so as to avoid oscillating around the path. The minimum and maximum values for Δ can be determined by the user and should depend on the maneuvering characteristics of the vehicle. The method was shown to be κ-exponentially stable and the simulations of the motion of a Mariner class vessel indicated that the time-varying lookahead distance guidance algorithm can improve performance. Further work on this for curved paths was done in [158].

Furthermore, several modifications have been proposed in the past so as to obtain a modified version of the LOS guidance law that can compensate for the influence of constant external disturbances, such were the integral LOS algorithms studied in [41] and [31]. Such modifications will be discussed in the next chapter.

Regarding the time-varying lookahead distance, this chapter is based on the work of [155, 158]. A globally exponentially stable (GES) sliding mode controller is designed for stabilizing the yaw angle of the vehicle and acts as the perturbing system in the cascade system that it forms along with the well-known LOS guidance law [184] which now incorporates the time-varying lookahead distance equation. The cascade is shown to be globally κ-exponentially stable when the task is to converge to the curved path which is constructed by the monotone version of the Hermite spline interpolation [97], which we presented in Chapter 4.

The proposed algorithm has a small computational footprint compared to non-linear constrained optimization methods which require a fast computer. Consequently, it is possible to implement the path-following controller and cubic spline algorithms onboard a small autonomous vehicle using an embedded computer with limited clock frequency to reduce the energy consumption. Typical applications are path-following control systems for autonomous underwater vehicles (AUVs), unmanned aerial vehicles (UAVs), unmanned surface vehicles (USVs) as well as conventional ships, aircraft and land-based vehicles. Simulations of the proposed method indicate that smooth convergence, without oscillations around the desired path, is achieved.

8.2 The Vehicle Model and Control Objective

8.2.1 The vehicle model

The *North-East-Down* (NED) coordinate system $\{n\} = (x_n, y_n, z_n)$ and the body-fixed reference frame $\{b\} = (x_b, y_b, z_b)$ will be used in this chapter to describe the motion, location and orientation of the vehicle. The NED frame is defined as a tangent plane on the surface of the Earth moving with the vehicle and is sufficient for local operations. Its origin is o_n and the x axis points towards the true *North*, the y axis points towards the true *East* and the z axis points *downwards*, normal to the Earth's surface. The body-fixed frame is moving with the vehicle and its origin o_b coincides with the center of gravity of the vehicle, see also [88, Ch. 2]. Consider a surface vehicle at the position (x, y) moving with the speed

$$U = \sqrt{u^2 + v^2}, \tag{8.1}$$

where u and v are the velocities in surge and sway respectively. The speed U is assumed to be positive and bounded

$$U_{\min} \leq U \leq U_{\max}, \ \ 0 < U_{\min}. \tag{8.2}$$

The three degree-of-freedom (DOF) horizontal dynamics of the surface vehicle can be represented by three differential equations

$$\dot{u} = f_u(u, v, r, \tau), \tag{8.3}$$
$$\dot{v} = f_v(u, v, r, \tau), \tag{8.4}$$
$$\dot{r} = f_r(u, v, r, \tau), \tag{8.5}$$

where $\dot{(\cdot)}$ denotes differentiation w.r.t. time and τ is the vector of the control forces and moments generated by the actuators. The kinematic equations are:

$$\dot{x} = u\cos(\psi) - v\sin(\psi), \tag{8.6}$$
$$\dot{y} = u\sin(\psi) + v\cos(\psi), \tag{8.7}$$
$$\dot{\psi} = r, \tag{8.8}$$

where ψ is the yaw angle relative to the NED reference frame (true North) and r is the yaw rate of the vehicle. The model (8.6)–(8.8) describes the motion of an underactuated vehicle since two out of three DOF's can be controlled independently, namely the yaw angle and the surge velocity. In this work we are dealing with only the path-following task, which (following the definition given in [42]) does not impose temporal restrictions and therefore we assume that the speed is measured and manually controlled, hence the speed control problem will not be addressed. Furthermore, (8.6)–(8.8) assume that there are no environmental forces acting on the system and thus ignore the relative velocities that would appear as a result of the vehicle's motion with respect to the wind or ocean currents. This assumption will be relaxed in the next chapter.

8.2.2 Control objective

The minimum distance between the vehicle and the monotonic curve between two waypoints will be used as cross-track error. Since the vehicle moves with speed U, the position (x, y) of the vehicle will be time-varying. Hence, an analytical expression for the cross-track error will be derived such that the LOS path-following controller can minimize the path error. Differentiation of the Hermite interpolants (4.7)–(4.8) w.r.t. the parameter θ gives the first derivatives

$$x_d'(\theta) = 3c_{x3}(\theta - \theta_i)^2 + 2c_{x2}(\theta - \theta_i) + c_{x1}\theta, \tag{8.9}$$

$$y_d'(\theta) = 3c_{y3}(\theta - \theta_i)^2 + 2c_{y2}(\theta - \theta_i) + c_{y1}\theta, \tag{8.10}$$

and the second derivatives

$$x_d''(\theta) = 6c_{x3}(\theta - \theta_i) + 2c_{x2}, \tag{8.11}$$

$$y_d''(\theta) = 6c_{y3}(\theta - \theta_i) + 2c_{y2}. \tag{8.12}$$

The tangent-line and normal-line through the point $(x_d(\theta), y_d(\theta))$ are given by

$$y_t - y_d(\theta) = \frac{y_d'(\theta)}{x_d'(\theta)}(x_t - x_d(\theta)), \tag{8.13}$$

$$y_n - y_d(\theta) = -\frac{1}{\frac{y_d'(\theta)}{x_d'(\theta)}}(x_n - x_d(\theta)). \tag{8.14}$$

The θ value corresponding to the path-normal that intersects the vehicle is found by requiring that $(x_n, y_n) = (x, y)$. Moreover, from (8.14) it follows that:

$$y_d'(\theta^*)(y - y_d(\theta^*)) + x_d'(\theta^*)(x - x_d(\theta^*)) = 0. \tag{8.15}$$

This involves solving the roots of the third-order cubic function for θ^*. Instead of using an analytical solution a numerical solution based on Newton-Raphson method will converge quite fast. For instance,

$$\theta_{j+1}^* = \theta_j^* - \frac{f(\theta_j^*)}{f'(\theta_j^*)}, \tag{8.16}$$

with

$$f(\theta_j^*) = y_d'(\theta_l^*)(y - y_d(\theta_j^*)) + x_d'(\theta_j^*)(x - x_d(\theta_j^*)), \tag{8.17}$$

$$f'(\theta_j^*) = y_d''(\theta_j^*)(y - y_d(\theta_j^*)) + x_d''(\theta_j^*)(x - x_d(\theta_j^*)) - x_d'(\theta_j^*)^2 - y_d'(\theta_j^*)^2, \tag{8.18}$$

will converge in a few iterations if the the initial path variable θ_0^* is taken as the last θ_i value when moving along the path between two waypoints parametrized on the interval $[\theta_1, \theta_n]$.

The normal line from the point $(x_d(\theta^*), y_d(\theta^*))$ on the path through the point (x, y) on the vehicle defines the along-track and cross-track errors (x_e, y_e). Moreover,

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} = \mathbf{R}^\top(\gamma_p) \begin{bmatrix} x - x_d(\theta^*) \\ y - y_d(\theta^*) \end{bmatrix}, \tag{8.19}$$

$$\mathbf{R}(\gamma_p) = \begin{bmatrix} \cos(\gamma_p) & -\sin(\gamma_p) \\ \sin(\gamma_p) & \cos(\gamma_p) \end{bmatrix}, \tag{8.20}$$

where $\mathbf{R}(\gamma_p) \in SO(2)$ is the rotation matrix in yaw. In algebraic form, the equations of the along-track and the cross-track error for a given vehicle position (x, y) become

$$x_e = (x - x_d(\theta^*))\cos(\gamma_p) + (y - y_d(\theta^*))\sin(\gamma_p), \tag{8.21}$$

$$y_e = -(x - x_d(\theta^*))\sin(\gamma_p) + (y - y_d(\theta^*))\cos(\gamma_p), \tag{8.22}$$

where γ_p is the path-tangential angle:

$$\gamma_p = \text{atan2}(y_d'(\theta^*), x_d'(\theta^*)), \tag{8.23}$$

and the two-argument function atan2 is a generalization of the $\arctan(y/x)$ that takes into account the signs of both x and y in order to determine the quadrant of the result, hence making it possible to distinguish between diametrically opposite directions.

Finally, the associated control objective for curved path-following is:

$$\lim_{t \to +\infty} y_e(t) = 0. \tag{8.24}$$

Note that in the case where temporal constraints are needed (for example in a path-tracking or target-tracking scenario) then it is necessary to include the along-track error dynamics in our study as well.

8.3 Time-Varying Lookahead Distance Guidance Law

8.3.1 Line-of-sight guidance law

Figure 8.1 depicts the geometry of the LOS guidance problem and some of the main variables that are involved in it. As it was mentioned again in Chapter 7, the LOS vector is defined differently in marine applications compared to the aircraft and missile communities. According to the definition in [233], the line-of-sight is the line that starts at the reference point (that is, the aircraft or the missile) and passes through the objective of the guidance, i.e the target. On the other hand, in marine guidance applications the LOS vector starts at the vessel and passes through a point $\mathbf{p}(x_{\text{los}}, y_{\text{los}})$ which is located on the path-tangential line at a lookahead distance $\Delta(t) > 0$ ahead of the direct projection of the vessel's position $\mathbf{p}(x, y)$ on to the path. In this chapter the lookahead-based steering method will be considered. The corresponding guidance law is given by the equation:

$$\chi_d = \gamma_p + \arctan\left(\frac{-y_e}{\Delta}\right), \tag{8.25}$$

where χ_d is the desired course angle of the vehicle:

$$\chi_d = \psi_d + \beta \tag{8.26}$$

where β is the sideslip angle of the vehicle. These variables will be explained in more detail in Section 9.2. In other words, the LOS guidance ensures that the vehicle's velocity is directed toward the moving point $\mathbf{p}(x_{\text{los}}, y_{\text{los}})$ until the vehicle
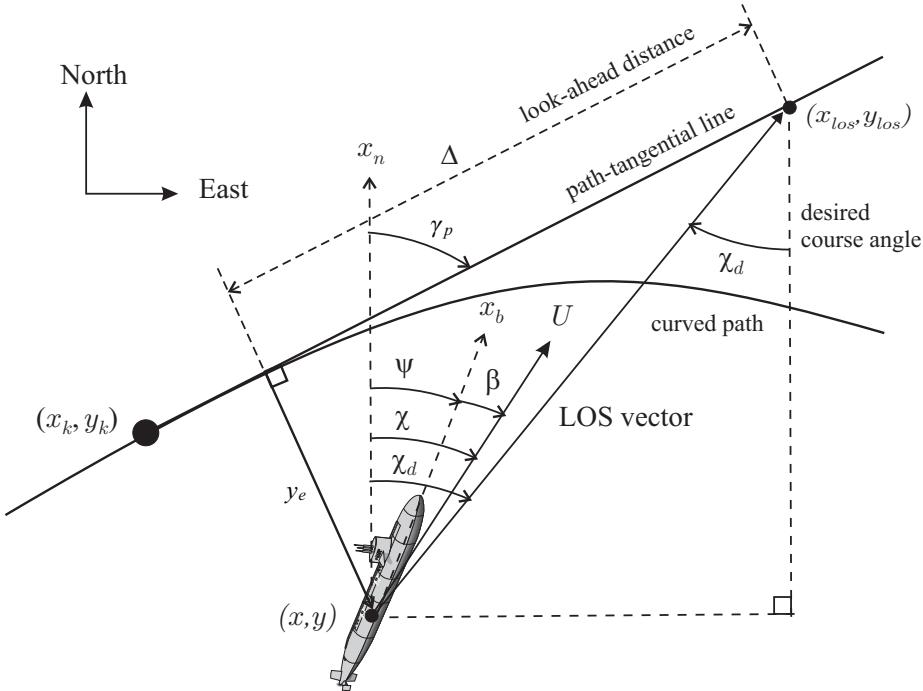
Figure 8.1: Line-of-sight guidance geometry for curved paths.

converges to the path, which indicates that the control objective (8.24) has been satisfied.

By differentiating (8.22) with respect to time we get:

$$
\begin{aligned}
\dot{y}_e = & -(\dot{x} - \dot{x}_d(\theta^*))\sin(\gamma_p) - (x - x_d(\theta^*))\cos(\gamma_p)\dot{\gamma}_p \\
& + (\dot{y} - \dot{y}_d(\theta^*))\cos(\gamma_p) - (y - y_d(\theta^*))\sin(\gamma_p)\dot{\gamma}_p \\
= & \underbrace{u\sin(\psi - \gamma_p) + v\cos(\psi - \gamma_p)}_{n_1} + \underbrace{\dot{x}_d(\theta^*)\sin(\gamma_p) - \dot{y}_d(\theta^*)\cos(\gamma_p)}_{n_2} \\
& - \dot{\gamma}_p \underbrace{((x - x_d(\theta^*))\cos(\gamma_p) + (y - y_d(\theta^*))\sin(\gamma_p))}_{\text{Along-track distance } x_e}.
\end{aligned}
\tag{8.27}
$$

The sums n_1 and n_2 can be transformed in amplitude-phase form, consequently we get:

$$
n_1 = \sqrt{u^2 + v^2}\sin(\psi - \gamma_p + \beta),
\tag{8.28}
$$

where

$$
\beta = \text{atan2}(v, u),
\tag{8.29}
$$

which is equal to the orientation of the vehicle's velocity vector $U(u, v)$ with respect to the body-fixed frame. Equation (8.29) is the angle between the vehicle's velocity orientation and the vehicle's heading, i.e. the sideslip, or drift, angle.

The second term can be rewritten as:

$$n_2 = \dot{\theta}\sqrt{x'(\theta)^2 + y'(\theta)^2}\sin(\gamma_p + \phi), \qquad (8.30)$$

where,

$$\phi = \mathrm{atan2}(-y'(\theta), x'(\theta)) \qquad (8.31)$$

$$= -\gamma_p. \qquad (8.32)$$

From (8.30) and (8.32) we conclude that $n_2 = 0$. Regarding the along-track error, x_e, we can conclude from (8.14) that:

$$(y - y_d) = -\frac{1}{\tan(\gamma_p)}(x - x_d) \Rightarrow \qquad (8.33)$$

$$x_e = 0 \qquad (8.34)$$

and consequently

$$\dot{y}_e = U\sin(\psi - \gamma_p + \beta). \qquad (8.35)$$

If we assume that the desired heading is perfectly tracked at all times and choose the desired heading angle as:

$$\psi_d = \gamma_p + \arctan\left(\frac{-y_e}{\Delta}\right) - \beta, \qquad (8.36)$$

the derivative of the cross-track error becomes:

$$\dot{y}_e = U\frac{-y_e}{\sqrt{\Delta^2 + y_e^2}}. \qquad (8.37)$$

The aforementioned assumption is not an oversimplification because the overall system (i.e. the guidance system and the heading controller) will be analyzed as a cascade structure where (8.35) constitutes the nominal system Σ_1 and the heading error dynamics constitutes the perturbing (or driving) system Σ_2. As a consequence, the stability analysis will show whether the time that the controller needs in order to converge can have a destabilizing effect on the guidance system. This will be further explained later on.

The Lyapunov Function Candidate (LFC) $V_1 = (1/2)y_e^2$ has the time-derivative:

$$\dot{V}_1 = U\frac{-y_e^2}{\sqrt{\Delta^2 + y_e^2}}. \qquad (8.38)$$

which is negative definite since $U > U_{\min}$. Hence, the origin $y_e = 0$ is a UGAS equilibrium of the nominal system Σ_1. Moreover, on the ball $D = \{y_e \in R | y_e| \leq \mu\}$, $\mu > 0$, we have that

$$\dot{V}_1 = -\frac{Uy_e^2}{\sqrt{\mu^2 + \Delta^2}} \leq -ky_e^2, \qquad (8.39)$$

for some $0 < k < U/(\sqrt{\mu^2 + \Delta^2})$, which entails that the origin is a ULES equilibrium. The combination of UGAS and ULES implies global κ-exponential stability, according to [154].

8.3.2 Time-varying lookahead distance

So far in our analysis we have considered a constant lookahead distance Δ. If we neglect, temporarily, the path-tangential angle γ_p and focus on the part of the LOS guidance that is a function of Δ, we can write

$$\chi_r = \arctan(-K_p y_e) \tag{8.40}$$

with $K_p = 1/\Delta$ which implies that the lookahead-based steering guidance law is equivalent to a saturated proportional control law [88, Ch. 10]. Moreover, as it can be deduced from Fig. 8.1, more aggressive steering will occur for a lower Δ value compared to a greater one. This fact motivates us to propose the following formula for the lookahead distance:

$$\Delta(y_e) = (\Delta_{\max} - \Delta_{\min})e^{-K_\Delta y_e^2} + \Delta_{\min}. \tag{8.41}$$

where $\Delta_{\min}$ and $\Delta_{\max}$ are the minimum and maximum allowed values for Δ respectively and, along with the convergence rate $K_\Delta > 0$, constitute the design parameters. The idea behind (8.41) is intuitive and simple and it can be summarized by the fact that a small Δ value is assigned when the vehicle is far from the desired path, (thus resulting in a more aggressive behavior that tends to decrease the cross-track error faster) and a large value for Δ is assigned when the vehicle is close to the path and overshooting needs to be avoided. The concepts "far" and "close" with respect to the desired trajectory are relative and the designer should take several considerations into account when determining $\Delta_{\min}$, $\Delta_{\max}$ and K_Δ, such as the maneuvering characteristics of the vehicle. For the sake of notational brevity we assign $\Delta_r := \Delta_{\max} - \Delta_{\min}$ and $\Delta := \Delta(y_e)$. In [155] we proposed a formula for $\Delta(y_e)$ which is slightly different than (8.41) since it included the absolute value instead of the square of the cross-track error. However, the time-derivative of Δ is required when designing the vehicle's heading controller and this can be the reason for a non-smooth controller input. Adopting (8.41) circumvents this problem.

How the time-varying Δ affects the stability properties of the guidance system

Equation (8.39) can now be rewritten as:

$$\dot{V}_1 = -\frac{U y_e^2}{\sqrt{\mu^2 + \Delta_{\max}^2}} \leq -k y_e^2. \tag{8.42}$$

From (8.42) we see the effect of the variable lookahead formula on the stability properties of the guidance system, compared to what would have been the case with a constant Δ. More specifically, the region where the guidance system is ULES is constrained by $\Delta_{\max}$. Therefore, the larger $\Delta_{\max}$ is, the more limited the region where the system is ULES becomes.

8.4 Heading Autopilot Design

In this Section, the stability of the vehicle's heading dynamics with the aid of a sliding mode controller is presented. In order to describe the yaw dynamics (8.5) of a marine craft, the *nonlinear extension of Nomoto's 1st-order model* [176] is considered:

$$T\dot{r} + H_N(r) = K\delta, \tag{8.43}$$

$$H_N(r) = n_3 r^3 + n_2 r^2 + n_1 r + n_0, \tag{8.44}$$

where r is the yaw rate, δ the rudder control input, T and K the Nomoto time and gain constants respectively, and $H_N(r)$ is a nonlinear function describing the maneuvering characteristics of the ship. For a course stable craft $n_1 > 0$, $T, K > 0$ and assuming symmetry in the hull implies $n_2 = 0$. The bias term n_0 can be treated as an additional rudder offset in the case where a constant rudder angle is required to compensate for constant environmental forces. Consequently, a large number marine of craft can be described by the equation:

$$\dot{r} + \alpha_1 r^3 + \alpha_2 r = b\delta, \tag{8.45}$$

where $\alpha_1 = n_3/T > 0$, $\alpha_2 = n_1/T > 0$ and $K/T = b > 0$. Equivalently, in state–space form:

$$\dot{r} = -\alpha_1 r^3 - \alpha_2 r + b\delta, \tag{8.46}$$

$$\dot{\psi} = r. \tag{8.47}$$

This formulation is a simplification compared to a real ship. In reality, the rudder input often does not affect the dynamics linearly and the Nomoto coefficients change as a function of the forward ship speed. Equations (8.46)–(8.47) constitute the perturbing system Σ_2. In order to control the system (8.46)–(8.47), we will construct a sliding mode controller. As mentioned before, the goal is to stabilize the heading angle at the desired value (8.36), consequently the desired heading rate can be computed as follows:

$$r_d = \dot{\psi}_d,$$
$$= \frac{\Delta^2}{\Delta^2 + y_e^2}\left(\frac{-\dot{y}_e}{\Delta}\right) + \dot{\gamma}_p - \dot{\beta}. \tag{8.48}$$

Assuming that the reference signals ψ_d, r_d are smooth, we define the sliding surface:

$$s := \bar{r} + \lambda\bar{\psi}, \tag{8.49}$$

where,

$$\bar{\psi} = \psi - \psi_d, \quad \bar{r} = r - r_d, \tag{8.50}$$

and $\lambda > 0$ is a design parameter reflecting the bandwidth of the controller. Next, we propose the LFC $V_2 = (1/2)s^2$ and by differentiating along the trajectories of s, we get

$$\dot{V}_2 = s(-\alpha_1 r^3 - \alpha_2 r + b\delta - \dot{r}_d + \lambda\bar{r}). \tag{8.51}$$

Consequently, the control law

$$\delta = (1/b)(\alpha_1 r^3 + \alpha_2 r + \dot{r}_d - \lambda \bar{r} - k_d s - k_s \mathrm{sgn}(s)), \tag{8.52}$$

with $k_d > 0$, $k_s \geq 0$ gives

$$\dot{V}_2 = -k_d s^2 - k_s |s| \leq -k_d s^2 \tag{8.53}$$

Therefore the equilibrium point $s = 0$ is GES . From (8.49) it follows that $\bar{\psi} = 0$, $\bar{r} = 0$ are also GES equilibria, see [225], [71]. It should be noted that autopilots can be sensitive to parametric uncertainty and unmodeled dynamics, hence the inclusion of the sliding mode term $k_s \mathrm{sgn}(s)$ in (62).

Discussion on the continuity of the reference signals

As it was shown in Section 4.2, the CHSI does not result in paths with continuous second derivative due to the discontinuities that occur at the waypoint locations. Given the fact that the curvature of a parametrized curve $\mathbf{r}(\theta) = (x(\theta), y(\theta))$ is calculated as

$$\kappa_r = \frac{x' y'' - x'' y'}{(x'^2 + y'^2)^{3/2}}, \tag{8.54}$$

and the lateral acceleration α_L of the vehicle can be computed as

$$\alpha_L = U^2 \kappa_r, \tag{8.55}$$

we can easily conclude that, in the general case, there will be a step in the desired lateral acceleration of the vehicle when passing through a waypoint. This can also be verified by the desired heading rate (8.48) which inherently is a function of the path's second derivative via the term $\dot{\gamma}_p$. As a result, ψ_d is always a continuous signal but the same is not true for r_d. For vehicles with low turning rates, such as marine craft, it is not critical to include r_d in the controller and, thus, setting $r_d = 0$ will not affect the performance significantly. This will be demonstrated by simulating accordingly in Section 8.6.2. For highly accelerated vehicles, however, it is recommended to use a filter before feeding the heading rate to the controller, or, alternatively, use a different parametrization.

8.5 Interconnection Between the Guidance System and the Heading Autopilot

The two nonlinear systems (8.37) and (8.46)–(8.47) are interconnected and form a cascade structure. The driving system is the sliding mode controller since the convergence to the desired yaw angle ψ_d affects the stability of the guidance system via the state $\bar{\psi} = \psi - \psi_d$. However, the guidance system perturbs the yaw control system as well, not only via the desired yaw angle ψ_d but also due to the fact that the cross-track error appears in the desired heading rate equation. This implies that apart from the three assumptions that need to be satisfied in order to infer upon the stability of the cascade system (for the theoretical background and the

proofs of the related theorems the reader is referred to [181]), it is necessary to prove that the system is forward complete.

A system is called forward complete if for every initial condition r and every input signal u, the corresponding solution is defined for all $t \geq 0$, i.e. $\sigma^{\max_{r,u}} = +\infty$ [10]. We achieve this in a similar manner as in [166], but before proceeding with the proof we compute the interconnecting term:

$$g(y_e, s) = U(\sin(\psi - \gamma_p) - \sin(\psi_d - \gamma_p)),$$
$$= 2U \sin\left(\frac{\bar{\psi}}{2}\right) \cos\left(\gamma_p - \frac{\psi + \psi_d}{2}\right). \tag{8.56}$$

This equation shows how the heading error dynamics acts and prevents $U \sin(\psi - \gamma_p)$ from becoming equal to $U \sin(\psi_d - \gamma_p)$ instantaneously. Obviously, if at each time instant we had $\psi = \psi_d \Rightarrow \bar{\psi} = 0$, then $g(y_e, s) = 0 \; \forall t$. But this perfect heading tracking scenario is never possible in reality due to the yaw inertia of the vehicle, hence the cascaded nature of the overall system.

8.5.1 Forward completeness of the closed-loop system

Equation (8.56) allows the two systems (8.37) and (8.46)–(8.47) to be rewritten as follows:

$$\Sigma_1 : \dot{y}_e = f_g(t, y_e) + g(y_e, s), \tag{8.57}$$
$$\Sigma_2 : \dot{s} = f_c(y_e, s), \tag{8.58}$$

where $f_g(t, y_e) = U \sin(\psi_d - \alpha)$. By proving that (8.57)–(8.58) is forward complete we can consider $f_c(s, y_e)$ to be a time-varying function so as $f_c(s, y_e) = f_c(s, t)$:

$$\Sigma_1 : \dot{y}_e = f_g(t, y_e) + g(y_e, s) \tag{8.59}$$
$$\Sigma_2' : \dot{s} = f_c(s, t) \tag{8.60}$$

This equivalence is depicted in Fig. 8.2. Then it is possible to use the theorems from [181]. According to [10], if a system is forward complete, then there exist a nonnegative, radially unbounded, smooth function $V : \mathbb{R}^n \to \mathbb{R}_+$ and a class-$\mathcal{K}_\infty$ function σ such that:

$$\frac{\partial V(x)}{\partial x} f(x, u) \leq V(x) + \sigma(|u|) \tag{8.61}$$

$\forall \; x \in \mathbb{R}^n$ and $\forall \; u \in \mathbb{R}$. In order to show that the system (8.57)–(8.58) is forward complete we employ the LFC:

$$V_{fc} = \frac{1}{2} y_e^2 + \frac{1}{2} s^2, \tag{8.62}$$

which gives

$$\frac{\partial V_{fc}}{\partial x} f(x, u) = -\frac{U y_e^2}{\sqrt{y_e^2 + \Delta^2}} + g(y_e, s) y_e - k_d s^2. \tag{8.63}$$

The first and third term of the right-hand side of the equation are negative. Regarding $g(y_e, s)$, from (8.56) we can write

$$g(y_e, s) \leq 2U_{\max}, \tag{8.64}$$
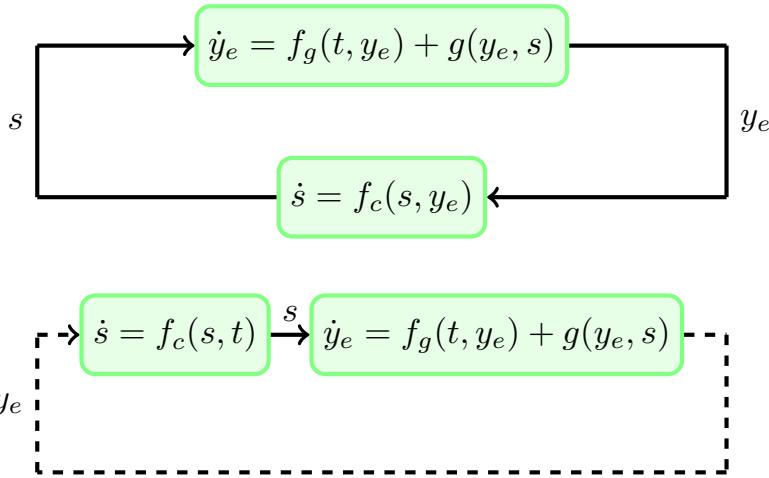
Figure 8.2: Equivalence between the closed-loop system and the cascade structure. This holds because the closed-loop system is forward complete.
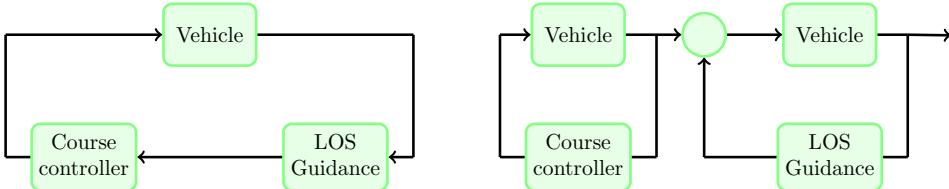
Figure 8.3: The overall system is a cascade where the vehicle along with the heading controller is the driving system Σ_2' and the vehicle in combination with the LOS guidance constitutes the driven system Σ_1. The yaw angle tracking error affects the convergence of the guidance system's objective, which is to minimize the cross-track error.

consequently (8.63) becomes

$$\frac{\partial V_{fc}}{\partial x} f(x, u) \leq V_{fc} + 2U_{\max}|y_e|. \tag{8.65}$$

Hence, we have shown that the system is forward complete. This result implies that we can proceed and analyze the overall system as a cascaded system. This fact leads to a separation principle, the structure of which can be seen in Fig. 8.3 .

8.5.2 Stability of the cascade structure

We choose the state vector that contains the error states of both the control and guidance systems that form the cascade:

$$\mathbf{x} = [y_e, \bar{\psi}, \bar{r}]^{\mathrm{T}} \tag{8.66}$$

Hence we continue by stating the following theorem:

Theorem 8.1 (*K*-exponentially Stable Cascade). *The origin* $\mathbf{x} = \mathbf{0}$ *of the cascade structure (8.59)-(8.60) (formed by the perturbing system (8.46)–(8.47) and the perturbed system (8.37)) is globally κ-exponentially stable if the control law is given by (8.52), and the desired yaw angle is described by (8.36).*

Proof. The proof consists of showing that the three assumptions of Theorem 1 in [181] are satisfied. For this chapter, however, we will also use the formulation given in [182] in order to prove global κ-exponential stability. *Assumption A1:* We already showed that the equilibrium point $y_e = 0$ is globally κ-exponentially stable. We also have that $V_1 = (1/2)y_e^2$, and

$$\left|\frac{\partial V_1}{\partial y_e}\right| |y_e| = |y_e||y_e| \Rightarrow$$
$$\left|\frac{\partial V_1}{\partial y_e}\right| |y_e| \leq c_1 V_1(y_e) \quad \text{for} \quad c_1 \geq 2 \quad \text{and} \quad \forall |y_e| \geq 0. \tag{8.67}$$

The condition

$$\left|\frac{\partial V_1}{\partial y_e}\right| \leq c_2 \quad \forall |y_e| \leq \mu \tag{8.68}$$

is also satisfied $\forall |y_e| \leq \mu, \quad \mu > 0$.

Assumption A2: This condition is apparently satisfied because of (8.64).

Assumption A3: It has already been proved that the equilibrium point $s = 0$ is GES. This means that if we rewrite the time derivative of the LFC as

$$\dot{V}_2 \leq -k_d s^2 \quad \forall s, \tag{8.69}$$

then the solutions will satisfy

$$|s(t)| \leq \lambda |s(t_o)| e^{-2(t-t_o)}, \tag{8.70}$$

and therefore by choosing $\nu(|s(t_o)|) = (\lambda/2)|s(t_o)|$ the integrability condition is satisfied.

Since all three assumptions are satisfied and, in addition to this, the nominal system Σ_1 has a globally κ-exponentially stable equilibrium and the system Σ_2 has a GES equilibrium, we conclude from Lemma 8 in [182] that the cascade system has a globally κ-exponentially stable equilibrium at $\mathbf{x} = \mathbf{0}$. $\qquad \square$
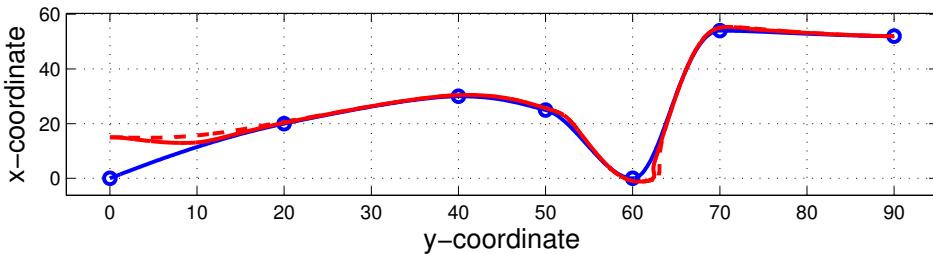
Figure 8.4: Path following comparison between the constant Δ (solid red line) and the variable Δ (dashed red line) approach.

8.6 Simulations

8.6.1 Curved Path

A nonlinear 3-DOF model has been used for simulating the vehicle's motion. The simulation was implemented using integration in continuous time. The ability of the proposed variable Δ method to guide the vehicle on the curved path is illustrated in Fig. 8.4 and compared to the constant Δ algorithm. In this case we have omitted the compensation of β since it often might not be available in practice, but this omission downgrades the performance of both guidance methods and therefore the comparison can be considered as a fair one. The parameters were chosen as $\Delta_{\min} = 4$ m, $\Delta_{\max} = 10$ m and $\kappa_\Delta = 1$. For the conventional LOS implementation we set $\Delta = 7$ m. It can be observed that due to the availability of a range of values for Δ, the variable lookahead method can contribute to meeting the path a bit faster than the constant Δ case. In addition to this, the proposed method results in improved performance when more difficult maneuvering is needed. This is depicted more clearly in Fig. 8.5 which shows a zoomed in portion of the previous picture close to waypoint 5, but this time, for the sake of completeness, the case of a time-varying Δ plus β compensation has also been included. As it is expected, we observe that the time-varying Δ approach induces a smaller cross-track error than constant $\Delta = 7$ does. It is also natural that the compensation of the sideslip angle (purely due to sway in this case) offers extra information to the control system and improves the performance even further, this can be of critical importance for applications to highly accelerated vehicles. Figure 8.6 shows how the lookahead distance varies during the operation. When the simulation starts, the craft is far from the desired path and, consequently, a low Δ is assigned. As mentioned before, this explains why the craft meets the path faster than it does with a constant Δ. It should also be noted that if a constant low Δ value is assigned, then the vehicle will have an overall aggressive behavior and this will induce an oscillatory behavior around the path. This was demonstrated in [155] where the task for a Mariner class vessel was to converge to a straight line. Moreover, Fig. 8.6 shows that when the marine craft is close to the path, Δ is close to its maximum value and usually drops at the waypoints. This happens because the second derivative continuities
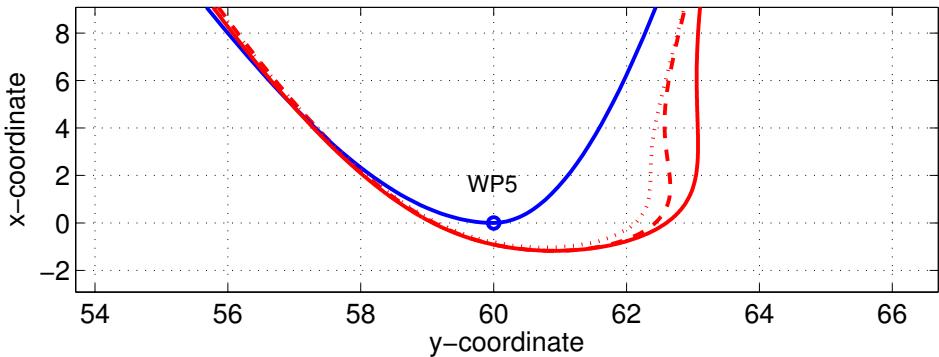
Figure 8.5: Performance comparison on a steep turn at waypoint 5 between the constant Δ (solid red line), the variable Δ (dashed red line) and the variable Δ including β (due to turning) compensation (dotted red line) approach.
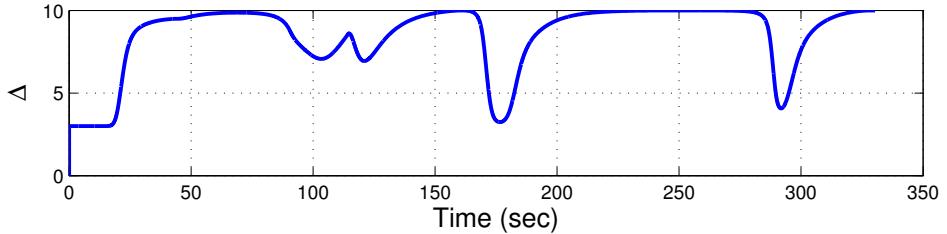
Figure 8.6: Time-varying lookahead distance.

result in step increases of the cross track error and the Δ value decreases in order to compensate for this.

It should be mentioned that in real life applications it would be normal to see a sudden drop of Δ from $\Delta_{\max}$ when the craft meets the path, especially during the first approach. The reason for this is that, especially for large and heavy craft, the heading dynamics will take more time to converge, hence the true heading will not be equal to the desired heading when the craft reaches the path. Consequently, the craft will deviate a bit from the path right after reaching it and Δ will react by decreasing right after reaching its maximum value. This behavior is also demonstrated in the next section (taken from [155]), where a more complete model with actuator constraints is used to show convergence on a straight line.

8.6.2 Straight path

The proposed guidance law was simulated in Matlab on a 3-DOF model of a Mariner class vessel with length $L = 160.93$ m which can be found in the MSS toolbox (Perez et al. 190). We chose the parameters in (8.41) as $\Delta_{\min} = 200$ m,
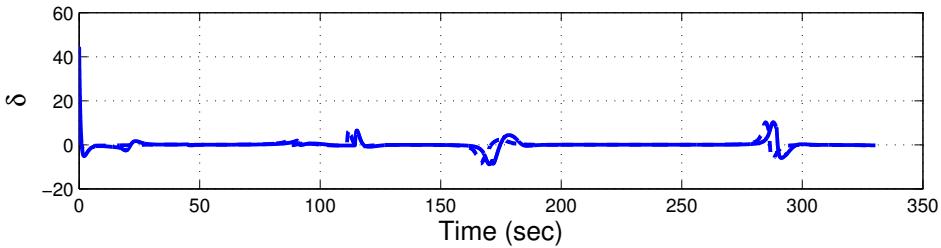
Figure 8.7: Comparison of the control variable δ between the variable lookahead distance method (solid blue line) and the conventional algorithm (dashed blue line).
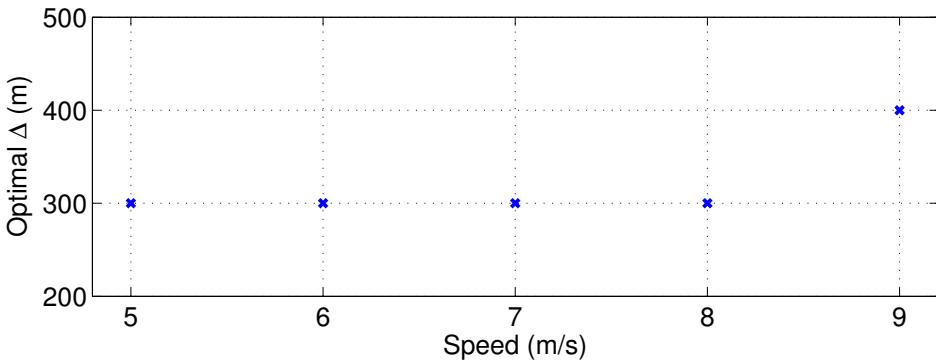
Figure 8.8: Optimal constant lookahead distance for different speeds.

$\Delta_{\max} = 1\,000\,\mathrm{m}$ and $\gamma = 1/300$. The simulation was implemented in discrete time with a time step of 0.01 sec.

In order to obtain a criterion with respect to which the effectiveness of the proposed method could be tested, we implemented Monte Carlo (MC) simulations where the ship was guided by the LOS algorithm with a constant Δ. The range of the MC simulations was $200 \le \Delta \le 1\,000\,\mathrm{m}$ with a step of $50\,\mathrm{m}$ and $5 \le U \le 9\,\mathrm{m/s}$ with a step of $1\,\mathrm{m/s}$. For each speed, the optimal lookahead was defined as the distance which induced the lowest value for the expression $\int y_e^2 dt$, the results are presented in Fig. 8.8. It is reasonable that for higher speeds a greater lookahead distance value will be the optimal one due to the fact that the craft is approaching the target path faster and small lookahead distances will result in a greater over-shoot. For $U = 7\,\mathrm{m/s}$ the optimal lookahead distance was $\Delta_{7,\mathrm{opt}} = 300\,\mathrm{m}$. This result fits nicely with the fact that in practice Δ is often chosen as $\Delta = 2L$.

The square of the cross track error induced by the two approaches can be seen in Fig. 8.9. The plot indicates that the constant lookahead method causes the vessel to reach the target line earlier than the proposed method. However, $\Delta_{7,\mathrm{opt}} = 300\,\mathrm{m}$ is responsible for an oscillatory behavior around the path before converging which
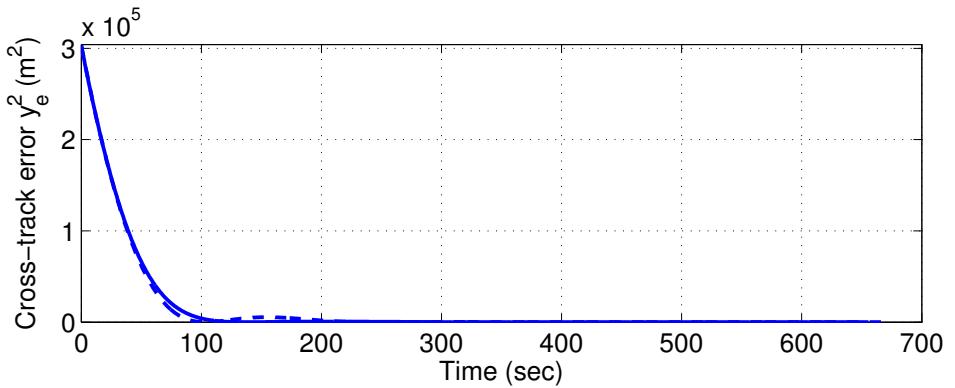
Figure 8.9: Squared cross-track error comparison between the constant LOS algorithm with $\Delta_{7,\text{opt}} = 300\,\text{m}$ (dashed line) and the proposed method (solid line).
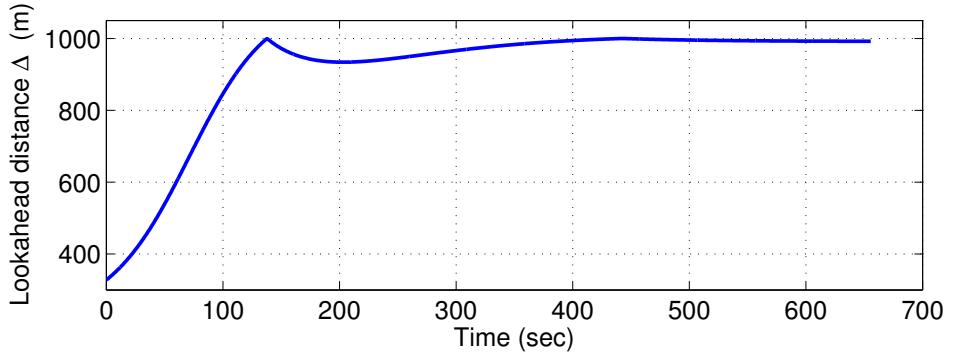
Figure 8.10: Time-varying lookahead distance according to the proposed method.

does not appear in the time-varying Δ case. The suggested method leads the craft in a way such that it meets the path with a small delay, but without overshooting. The explanation for this can be found by observing the time-varying lookahead distance plot in Fig. 8.10. When the simulation commences, the craft is far away from the target line, hence (8.41) indicates that a low lookahead distance value will be used. The closer the craft gets to the target line, the higher the lookahead distance becomes since a less agile behavior is needed in order to avoid overshooting. The abrupt drop of Δ that occurs at around 140 sec is a corrective action against a small increase of the cross-track error. The reason for this increase is that the heading controller has not converged entirely to the desired angle by the time the vessel meets the desired path (see Fig. 8.12) and as a result the ship tends to deviate from the path. This corrective behavior indicates that the method is robust with respect to tracking errors. It is worth noting that parameter γ must be chosen
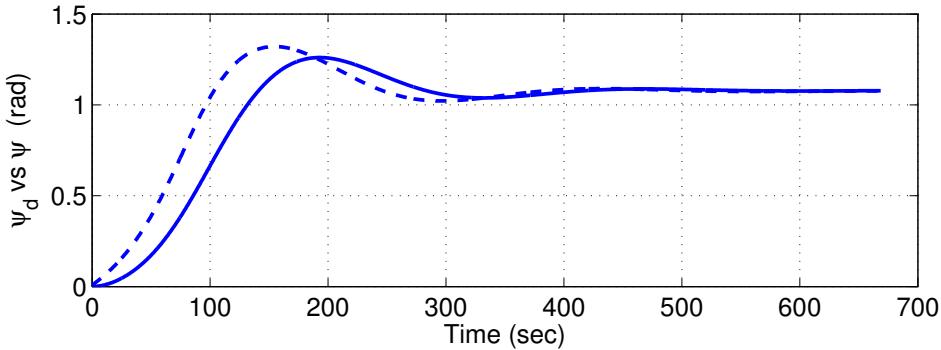
Figure 8.11: Actual ship heading (solid line) vs desired (dashed line) for $\Delta_{7,\text{opt}} = 300\,\text{m}$.
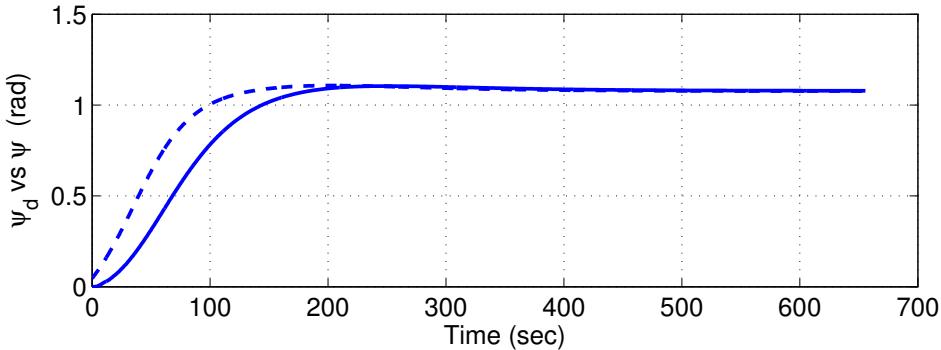
Figure 8.12: Actual ship heading (solid line) vs desired (dashed line) for time-varying Δ.

carefully in order to get a more efficient Δ. This happens due to the nature of (8.41), the function will give values only close to $\Delta_{\min}$ if $\gamma = 1$.

Figs. 8.11 and 8.12 illustrate the desired versus the true craft heading for the constant lookahead distance and the suggested method respectively. As expected, avoiding overshooting results in a smoother heading plot, whereas the conventional method requires that more heading adjustments are to be commanded before the craft has converged to the desired line and, thus, to a constant heading ($\psi = \psi_d = \alpha_k$). The reason for the large difference between the desired and actual heading is that a vessel of this size (and thus yaw inertia) cannot turn instantaneously to the heading indicated by the guidance system. Moreover, the rudder input is saturated and therefore the heading values that are required by the guidance system and are outside the feasible range of the ship will be taken up (or down) to the minimum (or maximum) allowed values.

Chapter 9

Augmented LOS Guidance for Unknown Disturbance Rejection

9.1   Introduction

In Chapters 7–8 we dealt with path-following applications in 3-DOF under the assumption that no external disturbances affect the vehicle. It comes as no surprise that in real life this is rarely, if ever, the case because in practice there is always influence due to the waves, wind and ocean currents. The natural outcome then is that the LOS guidance will fail to succeed in achieving the motion control objective. To illustrate this in a more intuitive way, when the desired path is a straight line these (hypothetically constant, or slow time-varying) forces will result in a constant cross-track error. This occurs because, in the general case, the external force has a component which is normal to the path, thus "pushing" the vehicle away from it. The magnitude of this error depends on the magnitude of the total external force, as well as the force's orientation with respect to the target line. To make things worse, if the desired path is a curved one then this cross-track error does not even reach a steady state because it will vary as fast as the path curvature.

Adding an integrator to the system in order to eliminate constant offsets is a very common practice in control systems and, unsurprisingly, this technique has been implemented by other researchers in the past so as to create an augmented version of the LOS guidance. The idea is to accumulate the error with an integral term and use it to correct the heading (or pitch) reference trajectories until the error has been eliminated. The outcome of this can be confirmed visually because, when the vehicle eventually converges on the target line, its heading will not be the same as the course angle. In other words, a sideslip angle is inevitable in order for the vehicle to converge to and follow the path. Extra care should be taken when referring to sideslip angle in this case. The steady state sideslip angle β due to the disturbances is, in the general case, different compared to the sideslip angle β_r which occurs during turning. This distinction is illustrated in Fig. 9.1. In connection with the discussion on the sideslip angle (10.23) due to turning in Sec. 7.3.2, β_r is a function of the relative velocities v_r, u_r instead of their absolute counterparts u, v.
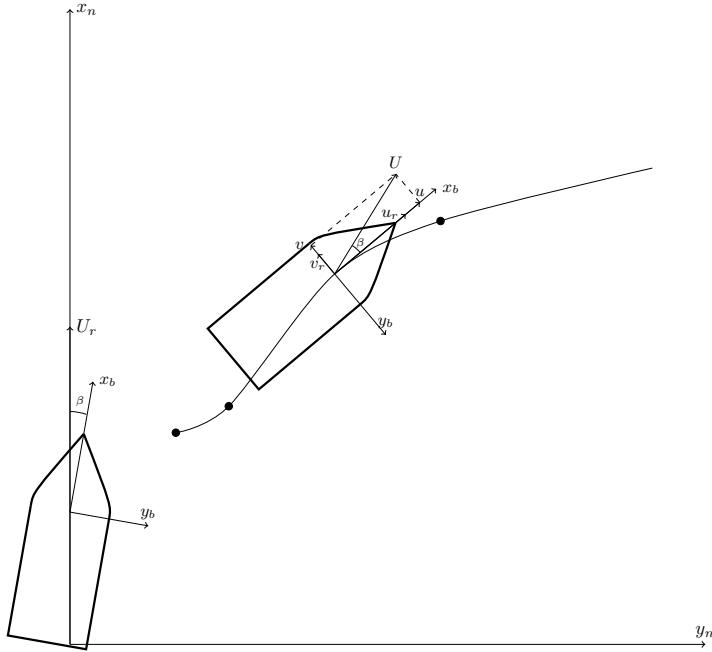
Figure 9.1: Sideslip angle, with a current coming from the east, hence affecting the velocity components.

It should be noted that incorporating the integral term in the LOS guidance implies that the modification is done at a kinematics level. It is only the heading reference trajectories which will be fed to the heading controller that change. Similar techniques were presented in [31, 41, 50]. In this way, the main advantages are that the guidance algorithm can compensate for not only the current effects but also modeling uncertainties while at the same time the structure of the vehicle controller remain unaffected. A different approach would be to add the integral action to the vehicle velocity/attitude controller. This approach is discussed in [11] and a number of such controllers which had appeared earlier in the literature, see [12, 13, 85, 86, 90, 218, 239] is compared.

The remaining part of this chapter is organized as follows: In Section 9.2, a brief discussion is given regarding how the sideslip angle can be compensated for using direct measurements. Section 9.3 revises two integral LOS guidance laws pertaining to marine vehicles and shows how course control (whenever applicable) circumvents this problem by ensuring that the course angle of the ship coincides with the path-tangential angle. Section 9.4 proposes an integral law stemming from very simple analysis based on the vehicle kinematics. In Sec. 9.5, we move a few steps further and develop an adaptive control-based methodology for compensating for the effects of ocean currents using a relative velocity kinematic model. The material presented in this chapter has been published in [91, 156, 158].

9.2 Compensation of Sideslip Angle for Heading Control

9.2.1 Introduction

The drift, or sideslip, angle β is defined as the angle from the x_b axis of the body-frame o_b to the velocity vector of the vehicle [88, Ch. 2]. In plain words, the drift angle is the angle between the direction towards which the vehicle is looking (i.e. the orientation described by the heading angle ψ) and the direction towards which the vehicle is moving (i.e. the orientation described by the course angle χ):

$$\beta = \chi - \psi. \tag{9.1}$$

This deviation between the heading and the course angle can be attributed to the sway velocity component which contributes to a variation of the overall velocity vector orientation. In the case where no external forces affect the motion of the vehicle, the sideslip angle occurs due to the lateral acceleration while turning and can be computed as in (10.23), repeated here:

$$\beta = \mathrm{atan2}(v, u). \tag{9.2}$$

In the presence of external forces, such as currents with velocity components u_c, v_c for instance, the drift angle will also depend on the relative velocities:

$$\beta_r = \mathrm{atan2}(v_r, u_r). \tag{9.3}$$

where $u_r = u - u_c$ and $v_r = v - v_c$.

In section 8.3 it turned out from the stability analysis of the guidance system that, for a vehicle governed by the kinematic equations (8.6)–(8.8), the LOS steering law is required to include a term related to the drift angle in order to stabilize the cross-track error dynamics around the desired equilibrium point (that is, $y_e = 0$). It is therefore important to stress the fact that the drift angle compensation needs to be done whether there are external forces acting on the vehicle or not. The latter case is not always treated in a similar way in the literature, perhaps because it is possible to consider the effect of the sway velocity due to turning on the guidance system as a perturbation, see for instance [30], [96]. In these papers it was still possible to prove κ-exponential stability of the cascade structure consisting of the guidance system and the heading controller. This is a very useful result, especially since it is not always possible to measure or estimate the drift angle. However, taking into account (9.2) whenever possible should improve the rate of convergence of the path-following method.

The rest of this chapter deals with the following two alternatives that make it possible to tackle the drift angle issue and thus compensate for its influence:

1. Direct measurement of β

2. Integral line-of-sight guidance

The feasibility and/or effectiveness of each one of the aforementioned techniques largely depends on the available instrumentation, among other things.

9.2.2   Direct measurement of β

The most straightforward and effective way of solving the drift angle problem would be to measure β and use this value in (8.36). This is the approach we implemented in Section 8.4. In some cases this could be achieved by using optical correlation sensors but these are very expensive, thus making this option difficult to realize. In the absence of environmental forces the sideslip angle is given by (9.2) and this means that knowing the values of u, v is sufficient. The simplest way to get these values is to use accelerometers in order to take measurements of the longitudinal and the lateral acceleration and then calculate the respective velocities by integrating these measurements , see for instance [5] , [115]. The main drawback of this approach, though, is the fact that accelerometer measurements are noisy and also include errors due to the accelerometer bias, hence resulting in large accumulated errors during long-term operations.

A global navigation satellite system (GNSS) can be employed so as to calculate the drift angle more accurately. A GNSS receiver is able to give velocity measurements by either measuring the Doppler shift of the GNSS carrier wave or by measuring the carrier phase difference between two successive samples [131]. This means that it is not necessary to differentiate the GNSS position measurements. In [69] the authors show that it is possible to determine experimentally the sideslip angle by taking the difference between the GNSS velocity heading and the vehicle heading calculated by integrating the yaw rate measurements of a gyroscope:

$$\beta = \psi_{gyro} - \psi_{GNSS}. \tag{9.4}$$

In this case, the gyro bias should be eliminated while moving along straight segments of the path, otherwise the error due to the integration will accumulate fast. Apart from the simple treatment described by (9.4), more accurate solutions can be attained by inertial navigation systems (INS) sensors aided by GNSS velocity measurements, an implementation of which can be found in [22]. This method uses a state estimator to fuse the several measurements and has the advantage that it does not require a model of the vehicle. It is also worth noting that a GNSS system with two antennas can compute the vehicle heading. This could be an alternative so as to avoid using the gyroscope in (9.4). Observer-based techniques have been used extensively in the past in order to to address the problem of estimating the drift angle. The interested reader can find more detailed information in [215], [113].

Despite the variety of the aforementioned options, the problem is far more difficult to tackle when environmental forces act on the vehicle. To illustrate this, imagine a ship that moves forward with zero sway velocity. The existence of a current with $v_c \neq 0$ can introduce a sway velocity component and eventually a drift angle, as it can be concluded by (9.3). The GNSS approach shown in (9.4) would not provide the user with sufficient accuracy anymore, because in addition to the ground speed components, the current velocity components should be measured as well. Measuring ocean currents from a moving ship is often a difficult and expensive task. Acoustic Doppler Current Profilers (ADCPs) and Electromagnetic Current Meters (ECMs) are nowadays frequently installed on marine craft to attain estimations of current velocities for a range of depths [52], [100]. It is not difficult to realize that when more factors enter our problem, most commonly wave or wind

forces, the level of complexity increases a lot, especially when we are interested in unmanned vehicles applications where space limitations often impose constraints on the allowed onboard equipment.

9.3 A Short Revision of Two Existing ILOS Methods

An alternative approach that aims at alleviating the sideslip angle effects is the integral LOS guidance. As the name indicates, the conventional LOS guidance is extended by adding integral action, more specifically the integral of the cross-track error y_e or a function of it. As a result, even when $y_e = 0$ there will be a sideslip angle generation due to the nonzero integral term which accumulates because of the action of constant disturbances that tend to take the vehicle away from the desired path. The concept of adding integral action to the LOS guidance law is discussed in [41]. In [31] the authors proposed a new integral LOS approach and considered both the kinematic and dynamic aspects of path-following when the task is to converge to a straight-line. By implementing this technique we avoid the instrumentation requirements presented in section 9.2.2. In this section, we implement two Integral Line-of-Sight (ILOS) method which have been presented earlier in the literature. They both succeed in eliminating constant disturbances, albeit with different efficiency. The purpose of this section is to serve as an introduction to ILOS methods and not as a thorough comparison of the two methods.

9.3.1 Conventional Integral LOS Guidance

Following [41], by rewriting (8.36) as

$$\psi_d = \gamma_p + \underbrace{\arctan(-K_p y_e)}_{\psi_r} - \beta, \ K_p = (1/\Delta), \tag{9.5}$$

we observe that the lookahead-based steering law has the same form as a saturated proportional control law, effectively mapping $y_e \in R$ into $\psi_r(y_e) \in (-\pi/2, \pi/2)$. Following the same line of reasoning, it is straightforward to add integral action in order to compensate for the cross-track error caused by a constant disturbance:

$$\psi_d = \gamma_p + \arctan\left(-K_p y_e - K_i \int_{t_0}^{t} y_e d\tau\right). \tag{9.6}$$

where $K_i > 0$ denotes the integral gain. As is usually the case with integral control action, careful design is necessary so as to avoid undesired effects such as wind-up and overshooting. Wind-up refers to the case where the integral term increases to large values and, as a consequence, has negative effects on the system's performance, such as very long convergence time and overshooting. Suggested methodologies pertaining to avoiding such phenomena can be found in [70], [47]. Equation (9.6) can be rewritten as follows

$$\psi_d = \gamma_p + \arctan\left(-K_p y_e - K_i y_{\text{int}}\right), \tag{9.7}$$

$$\dot{y}_{\text{int}} = y_e. \tag{9.8}$$
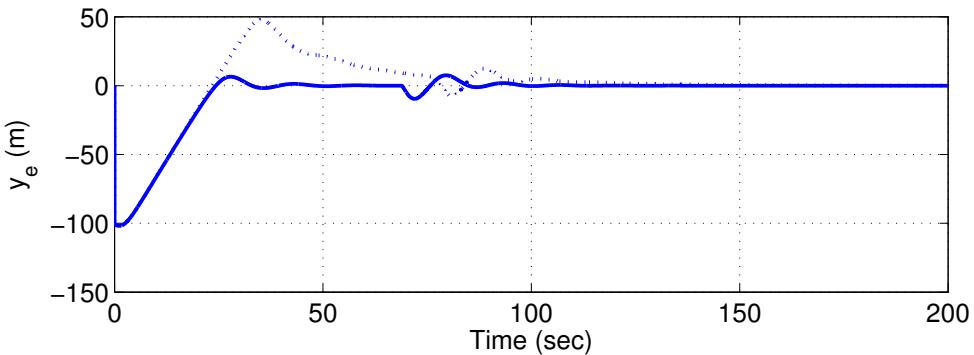
Figure 9.2: Cross-track error comparison between the conventional integral LOS (dotted line) and the integral LOS proposed by Børhaug et al. (solid line).

9.3.2 Integral LOS Guidance by Børhaug et al.

In [31] the authors followed a different approach and presented the following modified LOS guidance law with integral action:

$$\psi_d = \gamma_p - \arctan(K_p y_e + K_i y_{\text{int}}) \tag{9.9}$$

$$\dot{y}_{\text{int}} = \frac{y_e \Delta}{\Delta^2 + (y_e + \kappa y_{\text{int}})^2}, \tag{9.10}$$

where $K_p = (1/\Delta)$, $K_i = K_p\kappa$ and $\kappa > 0$ is a design parameter. Equation (9.10) has been designed in a way such that the influence of the integrator diminishes when the cross-track error increases, hence the wind-up risk is reduced. The authors focus on marine surface vessels and they pair the proposed guidance law (9.9)–(9.10) with a set of adaptive tracking controllers. After a long and extensive analysis they show that, after explicit bounds on the parameters of the guidance law have been satisfied, their proposed strategy results in globally asymptotic path following. The analysis in [31] was based on the 3-DOF kinematic model (7.15)–(7.17) with absolute velocities. In [49] the authors presented a proof of the same integral LOS guidance law using the relative velocity kinematics (9.26)–(9.27) and (7.13) (with $\theta = 0$) and showed that this simplifies the overall analysis and results in less complicated controllers. Fig. 9.2 compares the performance between the conventional integral LOS and the method developed by Børhaug et al.

9.3.3 LOS Guidance using Course Control

In a path-following operation of a surface vehicle it is natural to assume that GNSS (Global Navigation Satellite System) measurements are available, otherwise it is not possible to compute the cross-track error, for instance. Consequently, it is feasible to get absolute velocity measurements that can be used to control the course angle χ of the ship. Especially in the case of devices using GNNS Doppler data, the achievable accuracy can be very high, as it is reported in [56]. The ability
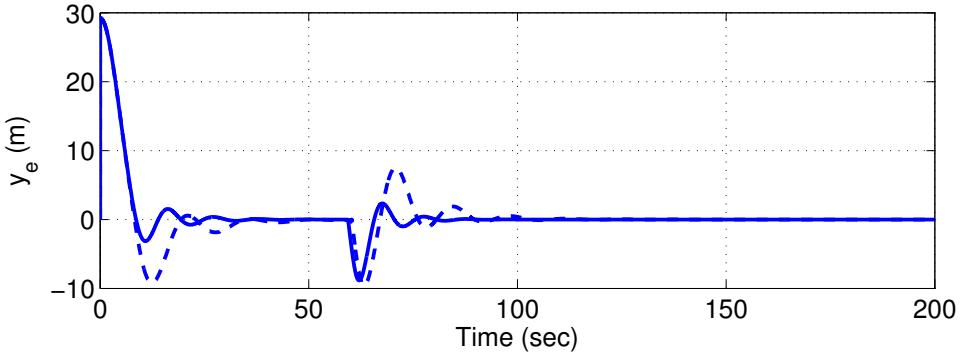
Figure 9.3: Cross-track error comparison between the course control case (solid line) and the integral LOS proposed by Børhaug et al. (dashed line).

to control χ allows to tackle the problem in a different way by rewriting the cross-track error (8.35) as

$$\dot{y}_e = U \sin(\chi_d - \gamma_p), \qquad (9.11)$$

which, similarly to the proof in Section 7.3, can be easily shown to have a κ-exponentially stable equilibrium point at $y_e = 0$ by choosing the desired course angle as:

$$\chi_d = \gamma_p + \arctan\left(\frac{-y_e}{\Delta}\right). \qquad (9.12)$$

Interestingly, it can be concluded by comparing (9.12) with (9.9) that, by controlling the course angle instead of the heading angle, the integral term is not necessary anymore. This is a natural outcome of expressing the cross-track error in the form (8.35) because in that way it becomes entirely known how the sideslip angle affects y_e. As a result, the necessity to deal with the current effect uncertainty is circumvented because the law (9.12) attempts to lead the overall velocity vector orientation to the right value, hence there is no need for an extra state that will accumulate the cross-track error so as to compensate for the heading control deviation from that orientation. It is worth noting that in previous works the desired course command fed into the course controller included integral action as well, see for instance [207] and [42]. In practice, however, it will probably still be useful to include integral action because there are always model uncertainties and other unknown factors present that have to be compensated for. Fig. 9.3 compares the performance between the course control case and the integral LOS guidance law developed by Børhaug et al. As it was expected, the fact that there is no uncertainty from the beginning in the course control method leads to better results.

9.4 Integral LOS Guidance Based on Absolute Velocities

In this section we propose a modified version of the integral LOS algorithm based on the system kinematics expressed using absolute velocities. The sideslip angle is

treated as a completely unknown variable here and no effort is put in driving the integral state to a specific equilibrium point. The method achieves to accomplish the task and it is stepping stone between the conventional integral action and the methodology we will present in the next sections of this chapter.

The following integral LOS guidance system is proposed:

$$
\chi = \gamma_p - \arctan(K_p y_e + K_i y_{\text{int}})
$$
$$
= \gamma_p - \arctan\left(\frac{1}{\Delta}(y_e + \kappa y_{\text{int}})\right), \tag{9.13}
$$
$$
\dot{y}_{\text{int}} = \frac{U y_e}{\sqrt{\Delta^2 + (y_e + \kappa y_{\text{int}})^2}}, \tag{9.14}
$$

where $K_p = (1/\Delta)$, $K_i = K_p \kappa$ and $\kappa > 0$ is a design parameter. Note that β is unknown, since only heading control is assumed.

Theorem 9.1 (Globally κ-exponentially Stable Integral LOS Guidance Law). *The origin $y_e = 0$ of the system* (8.35) *is globally κ-exponentially stable under the influence of constant environmental disturbances if the desired course angle is given by* (9.13) *and the time derivative of the integral term is described by* (9.14).

Proof. We can rewrite (8.35) as follows

$$
\dot{y}_e = U \sin(\psi - \gamma_p + \beta)
$$
$$
= U \sin(\chi - \gamma_p)
$$
$$
= U \sin\left(-\arctan\left(\frac{y_e + \kappa y_{\text{int}}}{\Delta}\right)\right), \tag{9.15}
$$

and since

$$
\sin(\arctan(x)) = x/(\sqrt{x^2 + 1}), \tag{9.16}
$$

we get

$$
\dot{y}_e = -U \frac{y_e + \kappa y_{\text{int}}}{\sqrt{\Delta^2 + (y_e + \kappa y_{\text{int}})^2}}. \tag{9.17}
$$

Next, we propose the LFC

$$
V_{\text{int}} = \frac{1}{2} y_e^2 + \frac{1}{2} \kappa y_{\text{int}}^2 \tag{9.18}
$$

and its time-derivative is computed as

$$
\dot{V}_{\text{int}} = y_e \left(-U \frac{y_e + \kappa y_{\text{int}}}{\sqrt{\Delta^2 + (y_e + \kappa y_{\text{int}})^2}}\right) + \kappa y_{\text{int}} \dot{y}_{\text{int}}
$$
$$
= -U \frac{y_e^2}{\sqrt{\Delta^2 + (y_e + \kappa y_{\text{int}})^2}} + \frac{\kappa y_{\text{int}}\left(\dot{y}_{\text{int}} \sqrt{\Delta^2 + (y_e + \kappa y_{\text{int}})^2} - U y_e\right)}{\sqrt{\Delta^2 + (y_e + \kappa y_{\text{int}})^2}}. \tag{9.19}
$$
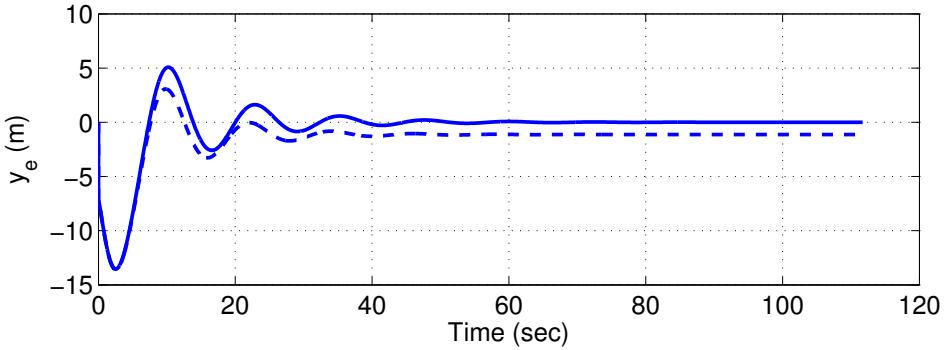
Figure 9.4: Cross-track error induced by constant environmental disturbances for conventional LOS guidance (dashed line) and the proposed integral LOS guidance law (solid line).

Choosing $\dot{y}_{\text{int}}$ as in (9.14) finally yields

$$\dot{V}_{\text{int}} = -U \frac{y_e^2}{\sqrt{\Delta^2 + (y_e + \kappa y_{\text{int}})^2}}$$

$$\leq 0. \tag{9.20}$$

From (9.20) we can conclude that under the integral LOS guidance law (9.13)–(9.14), the system (9.17) has a UGAS and ULES equilibrium point at $y_e = 0$. □

Finally, Fig. 9.4 shows the effectiveness of the integral LOS guidance law derived in Section 9.4. In this test, the task was chosen to be convergence to a straight line so as to get a better idea about the offset created by the time-invariant external disturbances, as well as the time the integral LOS described by (9.13)–(9.14) needs in order to converge. The external disturbance is considered to be a constant sea current coming from the East and the desired path is a straight line passing through the waypoints $WP_1(0,0)$ and $WP_2(300,300)$. The initial position and orientation of the marine craft are $P_0(20,10)$ and $\psi_0 = 0$ rad respectively. The total speed is $U = 5$ m/sec. If no integral action is applied, the induced cross-track error is approximately 1.5 m on the "west" side of the target line, hence the negative sign. The integral LOS, on the other hand, with $\kappa = 0.1$ manages to eliminate this offset and the vehicle converges to the desired line in approximately 60 sec. The simulations were implemented in using Euler integration, with time step 0.01 sec.

9.5 Direct and Indirect Adaptive Integral Line-of-Sight Path-Following Controllers

In this section, we move several steps further, compared to the approach of Section 9.4. Two novel methodologies based on adaptive control are developed in order to estimate and compensate for the effects of ocean currents. The vehicle dynamics

is incorporated in the stability analysis, hence the total cascade system is studied. The methods are developed using relative velocity kinematics, which them possible to be implemented by underwater vehicles, or in cases where absolute velocities might not be available in general.

9.5.1 Kinematics

We will consider a marine craft that is assigned to converge to a 2-D parametrized path specified by straight lines or curves. 2-D paths are commonly used for surface vessels, while for underwater vehicles it is assumed that the depth is controlled independently such that the path-following control problem is limited to motions in the horizontal plane. Without loss of generality, the presented methods can be extended to 3-D motions by following a similar approach as Lekkas and Fossen [156].

Cross-track error

A 2-D continuous path $(x_p(\theta), y_p(\theta))$ where $\theta \geq 0$ denotes the path variable is assumed to go through a set of successive waypoints (x_k, y_k) for $k = 1, ..., N$. The path can be constructed as straight lines or curves. For a vehicle located at (x, y) the cross-track error is computed as the orthogonal distance to the path-tangential reference frame where $\theta = \theta^*$ defines the point $(x_p(\theta^*), y_p(\theta^*))$ on the path. Consequently, the cross-track error satisfies:

$$\begin{bmatrix} 0 \\ y_e \end{bmatrix} = \mathbf{R}^\top(\gamma_p) \begin{bmatrix} x - x_p(\theta^*) \\ y - y_p(\theta^*) \end{bmatrix} \tag{9.21}$$

where

$$\mathbf{R}(\gamma_p) = \begin{bmatrix} \cos(\gamma_p) & -\sin(\gamma_p) \\ \sin(\gamma_p) & \cos(\gamma_p) \end{bmatrix} \tag{9.22}$$

Moreover, the path-tangential reference frame is rotated an angle γ_p from the North-East-Down (NED) reference frame using the rotation matrix $\mathbf{R}(\gamma_p) \in SO(2)$. Expanding (9.21) gives the normal line

$$y - y_p(\theta^*) = -\frac{1}{\tan(\gamma_p)}(x - x_p(\theta^*)) \tag{9.23}$$

through $(x_p(\theta^*), y_p(\theta^*))$ and the cross-track error:

$$y_e = -(x - x_p(\theta^*))\sin(\gamma_p) + (y - y_p(\theta^*))\cos(\gamma_p) \tag{9.24}$$

The path-tangential angle is recognized as:

$$\gamma_p = \arctan\left(y_p'(\theta^*), x_p'(\theta^*)\right) \tag{9.25}$$

where $x_p'(\theta) = \partial x_p/\partial\theta$, $y_p'(\theta) = \partial y_p/\partial\theta$ and $\arctan(y, x)$ is the angle between the positive x-axis of a plane and the point given by the coordinates (x, y) on it. For a straight line between two waypoints $\gamma_p = \arctan(y_{k+1} - y_k, x_{k+1} - x_k)$ is constant, while for a curved parametrized path γ_p will vary according to (9.25).

Equations of relative motion

The kinematic equations can be expressed in terms of the relative *surge* and *sway* velocities $u_r = u - u_c$ and $v_r = v - v_c$ according to Fossen [89]:

$$\dot{x} = u_r \cos(\psi) - v_r \sin(\psi) + V_x \tag{9.26}$$

$$\dot{y} = u_r \sin(\psi) + v_r \cos(\psi) + V_y \tag{9.27}$$

$$\dot{\psi} = r \tag{9.28}$$

where ψ and r are the yaw angle and rate, respectively. The body-fixed velocities (u_c, v_c) and North-East current velocities (V_x, V_y) satisfies:

$$[u_c, v_c]^\top = \mathbf{R}^\top(\psi)[V_x, V_y]^\top \tag{9.29}$$

Notice that the pair (V_x, V_y) is constant in NED, while the body-fixed current velocities (u_c, v_c) depend on the heading angle ψ. Time differentiation of (9.24) gives:

$$\dot{y}_e = -(\dot{x} - \dot{x}_p(\theta^*)) \sin(\gamma_p) + (\dot{y} - \dot{y}_p(\theta^*)) \cos(\gamma_p)$$
$$- [(x - x_p(\theta^*)) \cos(\gamma_p) + (y - y_p(\theta^*)) \sin(\gamma_p)] \dot{\gamma}_p \tag{9.30}$$

The last bracket in (9.30) is zero as seen from (9.23) and

$$\dot{x}_p(\theta^*) \sin(\gamma_p) - \dot{y}_p(\theta^*) \cos(\gamma_p) = 0 \tag{9.31}$$

according to (9.25). Consequently, (9.26), (9.27) and (9.30) give:

$$\dot{y}_e = -\dot{x} \sin(\gamma_p) + \dot{y} \cos(\gamma_p)$$
$$= -(u_r \cos(\psi) - v_r \sin(\psi) + V_x) \sin(\gamma_p) + (u_r \sin(\psi) + v_r \cos(\psi) + V_y) \cos(\gamma_p) \tag{9.32}$$

This can be written in *amplitude-phase form*:

$$\dot{y}_e = U_r \sin(\psi + \beta_r - \gamma_p) + U_c \sin(\beta_c - \gamma_p) \tag{9.33}$$

where the relative speed and direction are recognized as $U_r = \sqrt{u_r^2 + v_r^2}$ and $\beta_r = \arctan(v_r, u_r)$, respectively. Similar, $U_c = \sqrt{u_c^2 + v_c^2}$ and $\beta_c = \arctan(V_y, V_x)$. From the *reverse triangle inequality* if follows that the body-fixed velocities, $\mathbf{v}_i = [u_i, v_i]^\top$, satisfy:

$$\|\mathbf{v}_r\| = \|\mathbf{v} - \mathbf{v}_c\| \geq \|\mathbf{v}\| - \|\mathbf{v}_c\| \tag{9.34}$$

Since normal operation of a vehicle implies that the vehicle is moving faster than the current, $U > U_c$. Consequently, the relative velocity satisfies:

$$U_r \geq U - U_c > 0 \tag{9.35}$$

If the vehicle is moving at forward speed $U > 2U_c$ the relative velocity will satisfy $U_r > U_c > 0$. These properties will be exploited when designing the adaptive integral LOS guidance laws.

Integral LOS guidance law

The marine craft LOS algorithms for path following are usually employed at a kinematic level where the goal is to prescribe a value ψ_d for the heading angle ψ in (9.33). The hydro-acoustic reference system measures U_r, β_r and the cross-track error y_e. Since, the path tangential angle γ_p is known, we choose:

$$\psi_d = \gamma_p - \beta_r + \tan^{-1}\left(-\frac{1}{\Delta}(y_e + \alpha)\right) \tag{9.36}$$

where $\Delta > 0$ is the user specified look-ahead distance and α is a control input, which can be designed to obtain integral action. The heading angle tracking error is denoted as:

$$\tilde{\psi} = \psi - \psi_d \tag{9.37}$$

Substituting (9.36) into (9.33) gives:

$$\dot{y}_e = U_r \sin\left(\tilde{\psi} + \tan^{-1}\left(-\frac{1}{\Delta}(y_e + \alpha)\right)\right) + U_c \sin(\beta_c - \gamma_p) \tag{9.38}$$

The kinematic property: $\sin(a+b) = \sin(a)\cos(b) + \cos(a)\sin b$, gives:

$$\begin{aligned}
\dot{y}_e &= U_r \sin(\tilde{\psi})\cos\left(\tan^{-1}\left(-\frac{1}{\Delta}(y_e + \alpha)\right)\right) + U_r \cos(\tilde{\psi})\sin\left(\tan^{-1}\left(-\frac{1}{\Delta}(y_e + \alpha)\right)\right) \\
&\quad + U_c \sin(\beta_c - \gamma_p)
\end{aligned} \tag{9.39}$$

which reduces to

$$\dot{y}_e = U_r \sin(\tilde{\psi})\frac{\Delta}{\sqrt{\Delta^2 + (\alpha + y_e)^2}} - U_r \cos(\tilde{\psi})\frac{\alpha + y_e}{\sqrt{\Delta^2 + (\alpha + y_e)^2}} + U_c \sin(\beta_c - \gamma_p) \tag{9.40}$$

This can be rewritten as

$$\dot{y}_e = -\frac{U_r(\alpha + y_e)}{\sqrt{\Delta^2 + (\alpha + y_e)^2}} + U_c \sin(\beta_c - \gamma_p) + U_r \phi(y_e, \tilde{\psi})\tilde{\psi} \tag{9.41}$$

where

$$\phi(y_e, \tilde{\psi}) := \frac{\sin(\tilde{\psi})}{\tilde{\psi}}\frac{\Delta}{\sqrt{\Delta^2 + (\alpha + y_e)^2}} - \frac{\cos(\tilde{\psi}) - 1}{\tilde{\psi}}\frac{(\alpha + y_e)}{\sqrt{\Delta^2 + (\alpha + y_e)^2}} \tag{9.42}$$

Property 9.1 (Boundedness of $\phi(y_e, \tilde{\psi})$). Assume that $|\alpha| \leq \alpha_{\max}$ and $0 < \Delta_{\min} \leq \Delta \leq \Delta_{\max}$. Hence, the function $|\phi(y_e, \tilde{\psi})| \leq c$ for all y_e and $\tilde{\psi}$ since $|\sin(x)/x| \leq 1$, $|(\cos(x) - 1)/x| < 0.73$ for all x and

$$\left|\frac{\Delta}{\sqrt{\Delta^2 + (\alpha + y_e)^2}}\right| \leq 1, \quad \left|\frac{\alpha + y_e}{\sqrt{\Delta^2 + (\alpha + y_e)^2}}\right| \leq 1 \tag{9.43}$$

Moreover, $c = 1.73$ will be an upper bound for $|\phi(y_e, \tilde{\psi})|$.

9.5.2 Heading autopilot design

The yaw dynamics of a marine craft is usually modeled by using the Nomoto model
(Fossen [88])

$$\dot{\psi} = r \tag{9.44}$$

$$T\dot{r} + r = K\delta + b \tag{9.45}$$

where T and K are the Nomoto time and gain constants, respectively and $|b| \leq b_{\max}$
is a bias term due to environmental disturbances and unmodeled dynamics. For
simplicity a single rudder producing a rudder angle δ is used to steer the vehicle.
The Nomoto model can easily be modified to include other actuators. Define a
sliding surface:

$$s_0 := \tilde{\psi} + \lambda \int_0^t \tilde{\psi}(\tau)d\tau \tag{9.46}$$

where $\lambda > 0$ is a design constant. Hence,

$$s = \dot{s}_0 + \lambda s_0 = \dot{\tilde{\psi}} + 2\lambda\tilde{\psi} + \lambda^2 \int_0^t \tilde{\psi}(\tau)d\tau \tag{9.47}$$

represents a sliding surface. The error dynamics can be expressed in state-space
form as:

$$\begin{bmatrix} \dot{\tilde{\psi}} \\ \dot{s}_0 \end{bmatrix} = \underbrace{\begin{bmatrix} -\lambda & -\lambda \\ 0 & -\lambda \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \tilde{\psi} \\ s_0 \end{bmatrix} + \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{\mathbf{b}} s \tag{9.48}$$

It is convenient to define the signal $r_r := r - s$ and the heading controller as:

$$\delta = \frac{1}{K}\left(T\dot{r}_r + r_r - K_d s - \eta \operatorname{sgn}(s)\right) \tag{9.49}$$

where $\operatorname{sgn}(s)$ is the signum function and $K_d > 0$. The gain η is determined by
Lyapunov stability analysis. Since,

$$T\dot{s} + (1 + K_d)s + \eta \operatorname{sgn}(s) = b \tag{9.50}$$

we propose the Lyapunov function candidate:

$$V_1 = \mathbf{x}^\top \mathbf{P} \mathbf{x} + \frac{1}{2}Ts^2 \tag{9.51}$$

where $\mathbf{x} = [\tilde{\psi}, s_0]^\top$ and $\mathbf{P} = \mathbf{P}^\top > 0$ is given by

$$\mathbf{P}\mathbf{A} + \mathbf{A}^\top \mathbf{P} = -q\mathbf{I}_2 \tag{9.52}$$

for an user specified $q > 0$. Time differentiation of V_1 and substitution of (9.48)
and (9.50) into the expression for $\dot{V}_1$ under the assumption that $\eta \geq b_{\max}$ gives

$$\dot{V}_1 = -q\mathbf{x}^\top \mathbf{x} + 2\mathbf{x}^\top \mathbf{P}\mathbf{b}s - (1 + K_d)s^2 + bs - \eta|s|$$
$$\leq -q||\mathbf{x}||^2 + 2||\mathbf{P}|| \, ||\mathbf{x}|| \, |s| - (1 + K_d)s^2 \tag{9.53}$$

Let $\lambda_{\max}(\mathbf{P})$ denote the maximum eigenvalue of $\mathbf{P}$. Hence, the matrix

$$\begin{bmatrix} q & -\lambda_{\max}(\mathbf{P}) \\ -\lambda_{\max}(\mathbf{P}) & 1 + K_d \end{bmatrix} > 0 \tag{9.54}$$

for $K_d > \lambda_{\max}(\mathbf{P})^2/q - 1 > 0$. This particular choice for K_d implies that $\dot{V}_1 < 0$ and consequently the equilibrium point $[\tilde{\psi}, s_0, s]^\top = \mathbf{0}$ is globally exponentially stable (GES) according to Theorem 4.10 in Khalil [134].

9.5.3   Straight-Line Path Following

The heading autopilot of Section 9.5.2 can be used in cascade with an adaptive LOS guidance law for path following. Consider a straight-line path for which $\gamma_p = $ constant and define:

$$\theta := U_c \sin(\beta_c - \gamma_p) = \text{constant} \tag{9.55}$$

Consequently, the cross-track error (9.41) forms a cascade with the heading autopilot system in Section 9.5.2:

$$\Sigma_1 : \quad \dot{y}_e = \underbrace{-\frac{U_r(y_e + \alpha)}{\sqrt{\Delta^2 + (y_e + \alpha)^2}}}_{f_1(t,y_e)} + \underbrace{U_r \phi(y_e, \tilde{\psi})}_{g(t,y_e,\tilde{\psi})} \tilde{\psi} + \theta \tag{9.56}$$

$$\Sigma_2 : \quad \dot{\tilde{\psi}} = f_2(t, \tilde{\psi}) \tag{9.57}$$

where $f_2(t, \tilde{\psi})$ defines the yaw angle error dynamics corresponding to (9.48) and (9.50). The stability properties of the non-linear system (9.56)–(9.57) are given by Lemma 9.2:

Lemma 9.2 (Unforced cascaded system ($\theta = 0$)).   *Assume that ψ_d is computed using (9.36) and that (9.49) is used for heading control. Furthermore assume that $U > U_c$ and that $0 < \Delta_{\min} \le \Delta \le \Delta_{\max}$. Then the equilibrium point $(y_e, \tilde{\psi}) = (0, 0)$ of the unforced system (9.56)–(9.57), that is $\theta = 0$, with a bounded control signal α is globally $\kappa-$exponentially stable.*

Proof. From Section 9.5.2 we have that the equilibrium point $\tilde{\psi} = 0$ of the heading autopilot system Σ_2 given by (9.57) is GES. The equilibrium point $y_e = 0$ of the nominal system (Σ_1-system with $\tilde{\psi} = 0$):

$$\dot{y}_e = -\frac{U_r(y_e + \alpha)}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} \tag{9.58}$$

is UGAS and ULES or global $\kappa-$exponential stable as defined by Sørdalen and Egeland [213]. This follows from $V_2 = (1/2)y_e^2$, which after time differentiation along the trajectories of y_e gives:

$$\dot{V}_2 = -\frac{U_r}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} y_e^2 \le 0 \tag{9.59}$$

Since $U > U_c$, the relative velocity $U_r > 0$. Finally, the linear growth rate condition $|\phi(y_e, \tilde{\psi})| \le c$ (see Property 1) implies that all conditions of Lemma 8 in Panteley et al. [182] are satisfied and the cascade Σ_1–Σ_2 is globally $\kappa-$exponentially stable. $\square$

Indirect adaptive control

In this section we propose an adaptive disturbance observer, which can estimate the unknown parameter θ in (9.56) induced by ocean currents. The observer can be combined with a control signal α to obtain asymptotic tracking, see Figure 9.5.
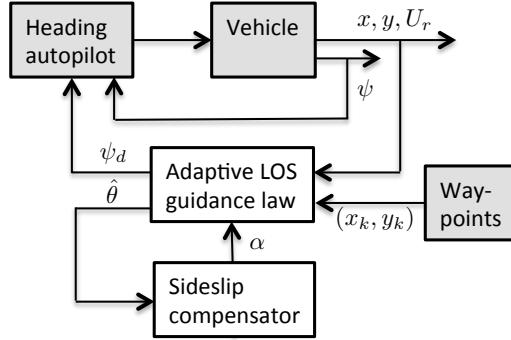
Figure 9.5: Adaptive integral LOS guidance law and heading autopilot.

Theorem 9.3 (Adaptive disturbance observer). *Assume that ψ_d is computed using (9.36) and that (9.49) is used for heading control. Furthermore assume that α is bounded and that $U > U_c$. Then the adaptive observer:*

$$\dot{\hat{y}}_e = -\frac{U_r(\hat{y}_e + \alpha)}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} + \hat{\theta} + K_1(y_e - \hat{y}_e) \tag{9.60}$$

$$\dot{\hat{\theta}} = K_2(y_e - \hat{y}_e) \tag{9.61}$$

with $0 < \Delta_{\min} \leq \Delta \leq \Delta_{\max}$, and adaptation gains $K_1 > 0$ and $K_2 > 0$ renders the equilibrium point $(\tilde{\psi}, \tilde{y}_e, \tilde{\theta}) = (\psi - \psi_d, y_e - \hat{y}_e, \theta - \hat{\theta}) = (0, 0, 0)$ globally $\kappa-$exponentially stable.

Proof. The observer error dynamics for the systems (9.56)–(9.57) and (9.60)–(9.61) is a cascaded system:

$$\Sigma_1^* : \begin{cases} \dot{\tilde{y}}_e = -\frac{U_r}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} \tilde{y}_e + \tilde{\theta} - K_1 \tilde{y}_e + U_r \phi(y_e, \tilde{\psi}) \tilde{\psi} \\ \dot{\tilde{\theta}} = -K_2 \tilde{y}_e \end{cases} \tag{9.62}$$

$$\Sigma_2 : \quad \dot{\tilde{\psi}} = f_2(t, \tilde{\psi}) \tag{9.63}$$

where $\dot{\tilde{\theta}} = -\dot{\hat{\theta}}$. The heading autopilot of Section 9.5.2 renders the equilibrium point $\tilde{\psi} = 0$ of the subsystem Σ_2 GES. For the first subsystem Σ_1^* we consider the

nominal system:

$$\dot{\tilde{y}}_e = -\frac{U_r}{\sqrt{\Delta^2 + (y_e + \alpha)^2}}\tilde{y}_e + \tilde{\theta} - K_1\tilde{y}_e \tag{9.64}$$

$$\dot{\tilde{\theta}} = -K_2\tilde{y}_e \tag{9.65}$$

In order to proof stability of (9.64)–(9.65) let $V_3 = (1/2)\tilde{y}_e^2 + 1/(2K_2)\tilde{\theta}^2$ be a Lyapunov function candidate. Consequently,

$$\dot{V}_3 = -\frac{U_r\tilde{y}_e^2}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} - K_1\tilde{y}_e^2 + \tilde{\theta}\left(\tilde{y}_e + \frac{1}{K_2}\dot{\tilde{\theta}}\right)$$

$$= -\left(\frac{U_r}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} + K_1\right)\tilde{y}_e^2$$

$$\leq 0 \tag{9.66}$$

Since $U_r > 0$ the signals $\tilde{\theta}$ and $\tilde{y}_e$ are bounded. In addition, the equilibrium point $(\tilde{y}_e, \tilde{\theta}) = (0, 0)$ of (9.64)–(9.65) is UGAS/ULES (global $\kappa-$exponential stable). This is seen by writing the error dynamics (9.64)–(9.65) in the following form:

$$\dot{x}_1 = f(x_1, t) + g(\mathbf{x}, t)x_2 \tag{9.67}$$
$$\dot{x}_2 = -K_2 g(\mathbf{x}, t)x_1 \tag{9.68}$$

where $x_1 = \tilde{y}_e$, $x_2 = \tilde{\theta}$, $\mathbf{x} = [x_1, x_2]^\top$ and

$$f(x_1, t) = -\left(\frac{U_r}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} + K_1\right)x_1 \tag{9.69}$$

$$g(\mathbf{x}, t) = 1 \tag{9.70}$$

Since $g^2(\mathbf{x}, t) = 1 > 0$, the *persistency of excitation* condition is satisfied and consequently all conditions of Theorem 1 in Fossen et al. [92] (alternatively Panteley et al. [183] are satisfied. Then we have proven that the nominal system (9.64)–(9.65) corresponding to Σ_1^* is globally $\kappa-$exponentially stable. Since Σ_2 is GES, the linear growth rate condition $|\phi(y_e, \tilde{\psi})| \leq c$ (see Property 1) is satisfied for subsystem Σ_1^* and all conditions of Lemma 8 in Panteley et al. [182] are satisfied it follows that the cascade Σ_1^*–Σ_2 is globally $\kappa-$exponentially stable. $\square$

Corollary 9.4 (Indirect adaptive control and asymptotic tracking)**.** *The adaptive observer (9.60)–(9.61) can be used together with a control signal α for cancellation of the drift term θ in (9.56) asymptotically. Let the control objective be to drive $y_e \to 0$ when $\hat{\theta} \to \theta$. From (9.56) it is seen that perfect asymptotic tracking $y_e = \tilde{\psi} = 0$ and cancellation of θ are obtained for:*

$$\alpha = \Delta\frac{\hat{\theta}/U_r}{\sqrt{1 - (\hat{\theta}/U_r)^2}} \tag{9.71}$$

where the signal $|\hat{\theta}/U_r| < 1$ must be saturated in a practical implementation.

Remark 9.1. Since $\hat{\theta} \to \theta$ exponentially (9.71) will not saturate during normal operation and the saturating element is only needed for numerical robustness. Alternatively, the adaptive observer can be implemented using a projection algorithm to ensure that the estimate $|\hat{\theta}| < U_r$.

In the next section, we present a globally convergent direct adaptive control law, which makes use of a projection algorithm to regulate y_e and $\tilde{\psi}$ to zero.

Direct adaptive control

A direct adaptive controller can be designed such that the input α cancels the unknown ocean current θ in (9.56). According to Lemma 9.2 the cross-track error y_e goes to zero exponentially for the unforced system Σ_1. The parameter θ in (9.56) can be cancelled by choosing α such that:

$$\frac{\alpha}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} = \frac{\hat{\theta}}{U_r} := \theta_n \tag{9.72}$$

Solving for α gives one feasible solution (the negative root) given by:

$$\alpha = \frac{y_e \theta_n^2 - \theta_n \sqrt{\Delta^2(1 - \theta_n^2) + y_e^2}}{1 - \theta_n^2} \tag{9.73}$$

Hence, the requirement $|\theta_n| < 1$ must be enforced to ensure that α is bounded when estimating θ_n. Consider (9.56) in the form:

$$\dot{y}_e = -\frac{U_r y_e}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} - \frac{U_r \alpha}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} + (\tilde{\theta} + \hat{\theta}) + U_r \phi(y_e, \tilde{\psi})\tilde{\psi} \tag{9.74}$$

where $\tilde{\theta} = \theta - \hat{\theta}$. If we choose α as (9.73), the cross-track error (9.74) becomes:

$$\dot{y}_e = -\frac{U_r y_e}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} + \tilde{\theta} + U_r \phi(y_e, \tilde{\psi})\tilde{\psi} \tag{9.75}$$

Theorem 2 (Adaptive integral LOS guidance law)**:** Assume that ψ_d is computed using (9.36) and (9.73), and that (9.49) is used for heading control. Furthermore, assume that $\gamma > 0$, $0 < \Delta_{\min} \leq \Delta \leq \Delta_{\max}$, $U_r > 0$ and that the ocean current is constant and there exists a known constant $M_\theta > 0$ such that $|\theta| \leq M_\theta < U_r$. Hence, the parameter adaptation law[1]:

$$\dot{\hat{\theta}} = \mathrm{Proj}(\hat{\theta}, -\gamma y_e) \tag{9.76}$$

where $|\hat{\theta}(0)| \leq M_{\hat{\theta}}$ ensures that $y_e \to 0$, $\tilde{\psi} \to 0$ and that $\tilde{\theta}$ is bounded.

---

[1]$\mathrm{Proj}(\,\cdot,\cdot)$ denotes a parameter projection (Krstic et al. [142, App. E]), which ensures that $|\theta|$ remains smaller than some design constant $M_{\hat{\theta}} > M_\theta$. The details of the parameter projection are given in

Proof. Eq. (9.75) is forced by $\tilde{\theta}$. From Lemma 9.2 it follows that the equilibrium point $(y_e, \tilde{\psi}) = (0, 0)$ of the unforced system, that is the nominal system:

$$\dot{y}_e = -\frac{U_r}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} y_e + U_r \phi(y_e, \tilde{\psi}) \tilde{\psi} \tag{9.77}$$

$$\dot{\tilde{\psi}} = f_2(t, \tilde{\psi}) \tag{9.78}$$

is globally κ-exponentially stable. Next, the forcing term is included in the analysis by writing (9.75) and (9.76) as a cascade. Let $\mathbf{z} = [z_1, z_2]^\top = [y_e, \tilde{\psi}]^\top$. Hence, (9.75)–(9.76) can be written:

$$\Sigma_{1p} : \dot{\mathbf{z}} = \mathbf{F}(t, \mathbf{z}) + \mathbf{G}\tilde{\theta} \tag{9.79}$$

$$\Sigma_{2p} : \dot{\tilde{\theta}} = -\operatorname{Proj}(\hat{\theta}, -\gamma y_e) \tag{9.80}$$

where

$$\mathbf{F}(t, \mathbf{z}) := \begin{bmatrix} -\frac{U_r}{\sqrt{\Delta^2 + (z_1 + \alpha)^2}} z_1 + U_r \phi(\mathbf{z}) z_2 \\ f_2(t, z_2) \end{bmatrix}, \quad \mathbf{G} := \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{9.81}$$

In order to proof stability of (9.79)–(9.80) let $V_4 = (1/2)\mathbf{z}^\top \mathbf{z} + 1/(2\gamma)\tilde{\theta}^2$ with $\gamma > 0$ be a Lyapunov function candidate. Consequently,

$$\dot{V}_4 = \mathbf{z}^\top \left(\mathbf{F}(t, \mathbf{z}) + \mathbf{G}\tilde{\theta} \right) + \frac{1}{\gamma}\tilde{\theta}\dot{\tilde{\theta}} \tag{9.82}$$

Since $\dot{\tilde{\theta}} = \dot{\theta} - \dot{\hat{\theta}} = -\operatorname{Proj}(\hat{\theta}, -\gamma y_e)$ and $\mathbf{z}^\top \mathbf{F}(t, \mathbf{z}) \leq 0$ it follows that:

$$\dot{V}_4 = \mathbf{z}^\top \mathbf{F}(t, \mathbf{z}) + \tilde{\theta} \left(-\frac{1}{\gamma} \operatorname{Proj}(\hat{\theta}, -\gamma y_e) + y_e \right)$$

$$\leq 0 \tag{9.83}$$

where we have exploited the fact that $-\tilde{\theta} \operatorname{Proj}(\hat{\theta}, \tau) \leq -\tilde{\theta}\tau$ (see Section 2.3). It is easy to verify that $\ddot{V}_4$ is bounded and consequently global convergence of $y_e \to 0$ and $\tilde{\psi} \to 0$ as well as boundedness of the parameter estimation error $\tilde{\theta}$ follows from Barbălat's lemma (Lemma 8.2 in Khalil [134]). $\qquad\square$

9.5.4 Curved-Path Path Following

For the curved-path path following case γ_p will be time varying. The results of Theorem 2 can be extended to curved paths by using over-parametrization. Consider the cross-track error (9.33) in the following form:

$$\dot{y}_e = U_r \sin(\psi - \gamma_p + \beta_r) + U_c \sin(-\gamma_p) \cos(\beta_c) + U_c \cos(-\gamma_p) \sin(\beta_c)$$

$$:= U_r \sin(\psi - \gamma_p + \beta_r) + \sin(\gamma_p)\theta_1 + \cos(\gamma_p)\theta_2 \tag{9.84}$$

where $\theta_1 = U_c \cos(\beta_c)$ and $\theta_2 = -U_c \sin(\beta_c)$ are two unknown constants. Choosing the desired yaw angle according to (9.36) gives:

$$\dot{y}_e = -\frac{U_r(y_e + \alpha)}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} + \sin(\gamma_p)\theta_1 + \cos(\gamma_p)\theta_2 + U_r\phi(y_e, \tilde{\psi})\tilde{\psi} \qquad (9.85)$$

The control input α canceling the drift term is chosen as:

$$\frac{\alpha}{\sqrt{\Delta^2 + (y_e + \alpha)^2}} = \frac{\sin(\gamma_p)\hat{\theta}_1 + \cos(\gamma_p)\hat{\theta}_2}{U_r} := \theta_n \qquad (9.86)$$

where $\hat{\theta}_1$ and $\hat{\theta}_2$ are the parameter estimates, and α is computed using (9.73). The parameter update laws are:

$$\dot{\hat{\theta}}_1 = \mathrm{Proj}(\hat{\theta}_1, -\gamma \sin(\gamma_p)y_e) \qquad (9.87)$$

$$\dot{\hat{\theta}}_2 = \mathrm{Proj}(\hat{\theta}_2, -\gamma \cos(\gamma_p)y_e) \qquad (9.88)$$

where $|\hat{\theta}_1(0)| \leq M_{\hat{\theta}}$ and $|\hat{\theta}_2(0)| \leq M_{\hat{\theta}}$ ensure that $|\theta_1|$ and $|\theta_2|$ remain smaller than some design constant $M_{\hat{\theta}} > M_\theta$. The constraint $|\theta_n| < 1$ in (9.73) is satisfied if M_θ is chosen such that $|\theta_i| \leq M_\theta < (1/2)U_r$ for $i = 1, 2$. Again Barbălat's lemma can be used to prove that $\tilde{y}_e$ and $\tilde{\psi}$ go to zero and the signals $\tilde{\theta}_1$ and $\tilde{\theta}_2$ are bounded by following a similar approach as Theorem 2.

9.5.5 Simulations

We consider straight-line path following ($\gamma_p = 45.0°$) of an AUV exposed to an unknown constant ocean current: $U_c = 1.0$ m/s and $\beta_c = -40.0°$, while $\hat{\theta}(0) = 0.0$. For both methods (Theorems 9.3 and 2) a step input in the cross-track error is injected at 50 seconds to demonstrate the effect of saturation and parameter projection.

Case study 1: Adaptive disturbance observer

The adaptive disturbance observer (Theorem 9.3) is simulated using the kinematic model (9.26)–(9.28). The control law was chosen as (9.71) with a saturating element $|\theta_n| \leq 0.99$ to ensure that α given by (9.71) is bounded for all $\hat{\theta}$. The vehicle speed is chosen as $U = 3.0$ m/s during path following. The observer gains are $K_1 = K_2 = 1.0$, while $\Delta = 10.0$ m. The simulation study shows that the estimation errors converge exponentially to zero as expected; see Figs 9.6–9.7. Accurate path following is also obtained when using (9.71) and the cross-track error y_e converges to zero as shown in Fig. 9.8. Computer simulations show that saturation is not a problem since it only affects the convergence time of y_e to zero. Moreover, the adaptive observer including the parameter estimator is not affected (Fig. 9.9) by how the control signal α is chosen (Fig. 9.10). This verifies the main results of Theorem 9.3.
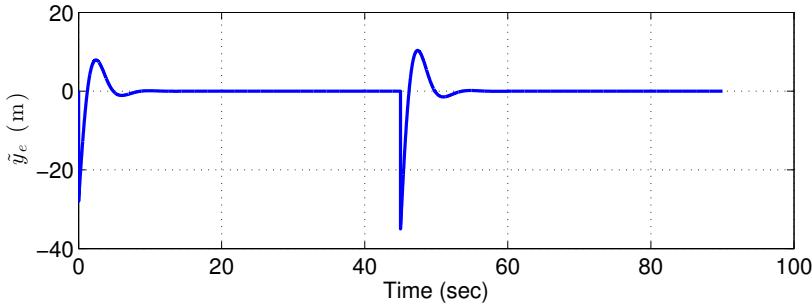
Figure 9.6: Cross-track estimation error for the adaptive observer.
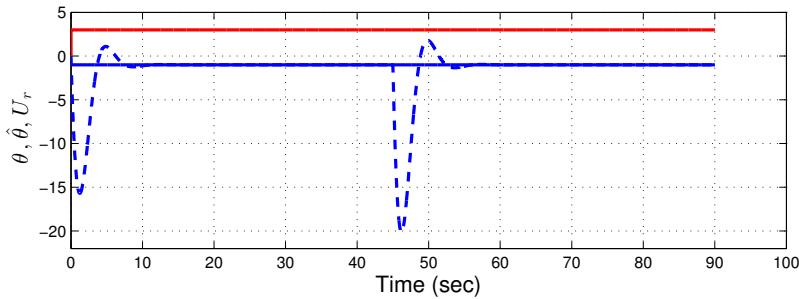
Figure 9.7: Comparison between the true θ (blue solid line) and the estimate $\hat{\theta}$ (blue dashed line) for the adaptive observer. The red solid line depicts the total relative speed U_r.
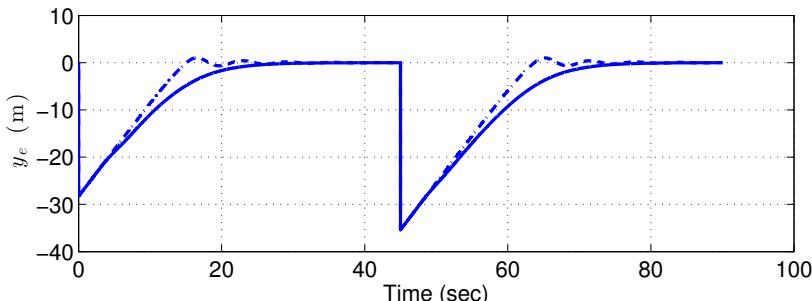
Figure 9.8: Cross-track error plots for the adaptive observer (solid line) and the adaptive LOS controller (dashed line).
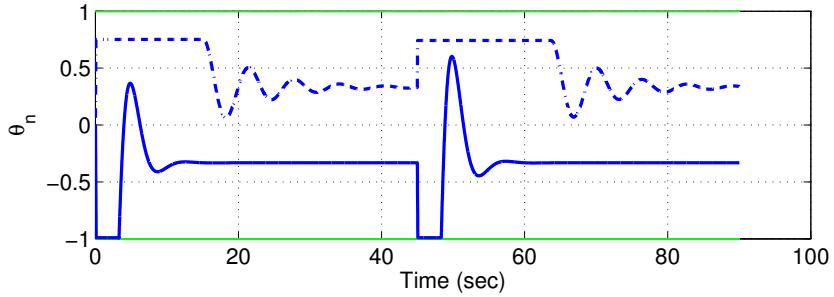
Figure 9.9: Comparison of θ_n for the adaptive observer (solid blue line) and the adaptive LOS controller (dashed blue line).

Case study 2: Adaptive integral LOS controller

The main simulation parameters are the same as in Case study 1. The adaptive gain is chosen as: $\gamma = 1.0$. As expected from Theorem 2, the cross-track error y_e converges to zero, see the blue dashed line in Figure 9.8. Compared to the adaptive observer, the adaptive LOS controller leads the vehicle to reach the desired path faster, but with with s small overshoot as shown in Figure 9.8. The variables θ_n and α for the adaptive LOS controller are plotted as blue dashed lines in Figures 9.9 and 9.10, respectively.

Case study 3: Adaptive disturbance observer with vehicle model

This case study is almost the same as the first one, with the major difference that now the dynamic model is also included. The purpose of presenting both case studies 1 and 3 is to simply visualize some of the differences that occur when the model is included (which is, in any case, much closer to reality) and justify the need for studying the cascade structure between the guidance system and the control system, instead of taking into account just the guidance algorithm for inferring upon the system stability. The dynamics is described by a simple mass-spring-damper model.

The results can be seen in Figs 9.11–9.15. As expected, the system needs some extra time to converge now, therefore the total simulation run for 180 sec instead of 90 sec. Moreover, there is small oscillations around the path, contrary to the first study case. This has to do with the tuning of the system, which is now less nontrivial to do. The tuning gains were chosen as $K_1 = 10$ and $K_2 = 0.8$.
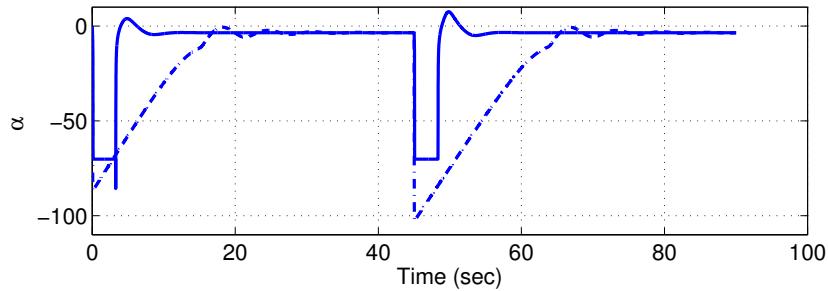
Figure 9.10: Comparison of α between the adaptive observer (solid blue line) and the adaptive LOS controller (dashed blue line).
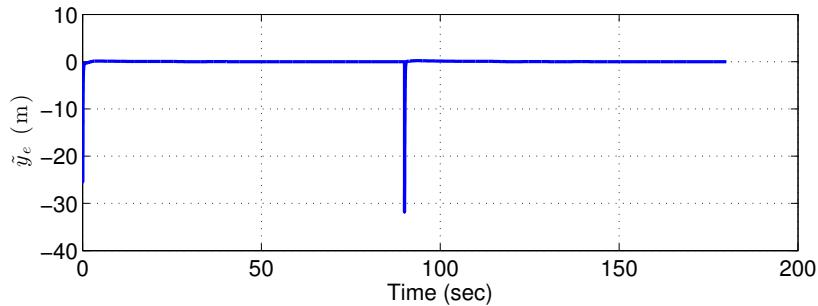
Figure 9.11: Estimation error of the cross-track error for the adaptive observer including vehicle model.
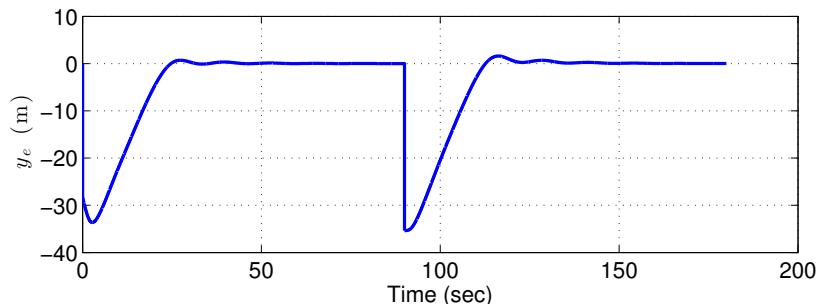
Figure 9.12: Cross-track error for the adaptive observer including vehicle model.
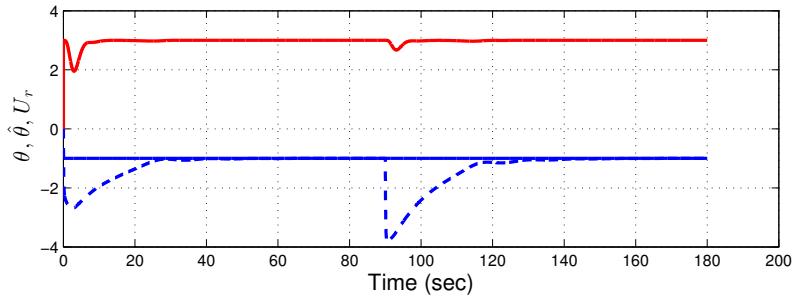
Figure 9.13: Comparison between the true θ (blue solid line) and the estimate $\hat{\theta}$ (blue dashed line) for the adaptive observer including vehicle model. The red solid line depicts the total relative speed U_r.
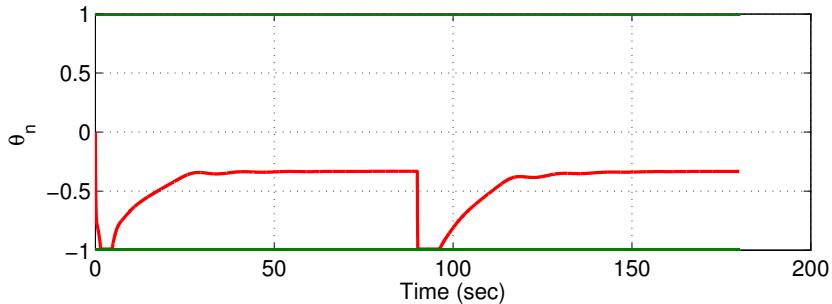
Figure 9.14: Plot of θ_n for the adaptive observer including vehicle model.
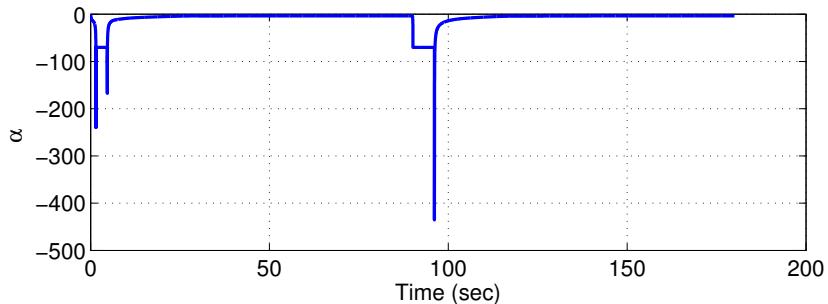
Figure 9.15: Plot of α for the adaptive observer including vehicle model.

9.5.6   Conclusions

Two non-linear adaptive path-following algorithms for estimation and compensation of ocean currents have been presented: 1) a globally κ-exponentially stable adaptive disturbance observer intended for an indirect adaptive control approach, and 2) a globally convergent direct adaptive control law.

The algorithms are based on a classical LOS guidance principle for marine craft and *integral action* is obtained by parameter adaptation. This resulted in a conceptual new integral LOS guidance law based, which effectively compensate for drift forces due to waves, wind and ocean currents. The structure of the adaptive integral LOS guidance law is different from the well-established integral LOS controller of Børhaug et al. [31]. Both curved and straight-line path following are considered in the theoretical analysis. The results have been verified by three case studies.

Chapter 10

Path Tracking for Underactuated Vehicles

10.1 Introduction

In Chapters 7–9, we developed guidance algorithms only for path-following applications, which means that the guidance system generated reference trajectories only for the heading angle. In general, depending on the motion control objectives involved, the following motion control scenarios can be considered: a) target tracking, b) path following, c) path tracking, and d) path maneuvering. For more details on the motion control scenarios the reader is referred to [208], [35] and [42].

Path tracking refers to the case where the vehicle is assigned to track an object that moves along a predefined path. This implies that the mission involves both temporal and spatial constraints that have to be satisfied in order for the mission to be accomplished. Naturally, in this scenario an underactuated vehicle should be able to control both its heading angle (or course angle, in the case where environmental forces are present) and its surge velocity in order to satisfy the corresponding constraints. This is different compared to the path following scenario of Chapters 7–9, where it can be assumed that the vehicle has a constant total speed and it is necessary to control only its heading angle in order to achieve the desired result. However, the path tracking scenario is not so far from the target tracking case and it can be implemented as such [42].

Therefore, for an underactuated vehicle the guidance system should generate reference trajectories for both the heading angle and the surge velocity such that the vehicle will manage to track an object. For marine vehicles, the most popular guidance techniques in the literature are the following: a) Pure Pursuit (PP), b) Line-of Sight (LOS) and c) Constant Bearing (CB). These are presented in detail in [41]. The combination of the guidance system (which generates the reference trajectories) with the velocity and heading autopilots (which are assigned to track these reference trajectories) form a system which can often be difficult to analyze and infer upon its stability. An integrated approach for accurate trajectory tracking was presented in [130]. In [6] the authors reported asymptotic trajectory tracking by employing a cascade control strategy consisting of a kinematic and a dynamic

control law. A solution where the primary task is to steer the vehicle on the desired path and the secondary task is to assign the speed so as to track the target was given in [76]. A robust methodology for the maneuvering problem, which consists of a geometric task and a dynamic task, in the presence of bounded disturbances was developed in [210]. A generalization of the maneuvering problem was presented in [211] where the LOS guidance law was employed in order to demonstrate the stabilization of more general manifolds. Following the same distinction between the tasks, the authors in [36] implemented a LOS guidance scheme for the geometric task and an appropriate speed assignment for the dynamic task, assuming the vehicle is a particle. Many other solutions have appeared in the literature, including the combined problem of path planning and trajectory tracking (see [196], [7]) and cases where there is large modeling parametric uncertainty [2].

This chapter builds upon and extends the methodology presented in [36]. One major difference is that, instead of a particle, an underactuated vehicle model is used for the analysis. Moreover, contrary to [36], we include a more complete proof regarding the minimization of both the along-track and cross-track errors. First, assuming perfect heading angle and surge velocity tracking, we employ the LOS guidance for minimizing the cross-track error and develop a methodology for obtaining a velocity assignment that minimizes the along-track error. Next, we consider stability of the total system in cascade by taking into account the convergence of the heading angle and surge velocity controllers. By using well-known results from nonlinear cascade systems theory we show that the total system is globally κ-exponentially stable, a stability concept which was introduced by [213]. Finally, we extend the method even further in order to account for the influence of ocean currents.

The rest of the chapter is organized as follows: Section 10.2 presents the vehicle model and the controllers as well as the virtual vehicle model that is used for tracking. Section 10.3 gives a brief overview of the LOS guidance law for cross-track error minimization. In Section 10.4, a new surge velocity guidance technique is developed in order to minimize the along-track error. Section 10.5 deals with the stability of the total system. In Section 10.6 the theoretical analysis of the guidance system in absolute velocities is supported by simulation results. Section 10.7 extends the method to the case where ocean currents influence the vehicle's motion and, finally, Section 10.8 concludes the chapter.

10.2 Vehicle and Virtual Vehicle Models

10.2.1 Vehicle Model

The ship equations of motion are usually represented in three DOFs by neglecting *heave, roll* and *pitch* [88]:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{10.1}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{wave}} \tag{10.2}$$

where $\boldsymbol{\eta} := [x, y, \psi]^\top$, $\boldsymbol{\nu} := [u, v, r]^\top$ and

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \in SO(2) \tag{10.3}$$

is the rotation matrix in yaw. It is assumed that wind and wave-induced forces $\boldsymbol{\tau}_{\text{wind}}$ and $\boldsymbol{\tau}_{\text{wave}}$ can be linearly superpositioned. The system matrices $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$ and $\mathbf{C}(\boldsymbol{\nu}) = \mathbf{C}_{RB}(\boldsymbol{\nu}) + \mathbf{C}_A(\boldsymbol{\nu})$ are usually derived under the assumption of port-starboard symmetry and that surge can be decoupled from the sway and yaw motions [88]. Moreover,

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix} \tag{10.4}$$

$$\mathbf{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & -mr & -mx_g r \\ mr & 0 & 0 \\ mx_g r & 0 & 0 \end{bmatrix} \tag{10.5}$$

$$\mathbf{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u \\ -Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u & 0 \end{bmatrix} \tag{10.6}$$

Hydrodynamic damping will in its simplest form be linear:

$$\mathbf{D} = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} \tag{10.7}$$

while a nonlinear expression based on second-order modulus functions describing quadratic drag and cross-flow drag is:

$$\mathbf{D}(\boldsymbol{\nu}) = \begin{bmatrix} -X_{|u|u}|u| & 0 \\ 0 & -Y_{|v|v}|v| - Y_{|r|v}|r| \\ 0 & -N_{|v|v}|v| - N_{|r|v}|r| \end{bmatrix}$$

$$\begin{matrix} 0 \\ -Y_{|v|r}|v| - Y_{|r|r}|r| \\ -N_{|v|r}|v| - N_{|r|r}|r| \end{matrix} \tag{10.8}$$

Other nonlinear representations are found in [88], [87].

10.2.2 Surge Velocity and Heading Angle Controllers

The maneuvering model used in guidance and control systems only needs to capture the most important hydrodynamic effects. Such a model can be based on the following assumptions [88]:

A1: Surge can be decoupled from the sway and yaw motions.

A2: The yaw dynamics can be accurately described by a nonlinear Nomoto model and stabilized by a heading autopilot. This implies that the sway velocity $v(t)$ is bounded for all $t \geq 0$.

A3: The drift forces due to ocean currents, wind and waves can be neglected in the model since the guidance and control systems are designed to include integral action.

Based on Assumptions 1–3 an underactuated ship with two controls, thrust T and rudder δ, can be modeled as:

$$\dot{x} = u\cos(\psi) + \sigma_1 \tag{10.9}$$

$$\dot{y} = u\sin(\psi) + \sigma_2 \tag{10.10}$$

$$\dot{\psi} = r \tag{10.11}$$

where $\sigma_1 = -\sin(\psi)v$ and $\sigma_2 = \cos(\psi)v$ are known time-varying signals that can be measured, while the *surge* and *yaw* dynamics are modeled as:

$$(m - X_{\dot{u}})\dot{u} - X_{|u|u}|u|u = (1-t)T \tag{10.12}$$

$$T\dot{r} + n_1 r + n_2 r^3 = K\delta \tag{10.13}$$

where $t > 0$ is the thrust deduction number, $-X_{\dot{u}} > 0$ is the added mass in surge, and $-X_{|u|u} > 0$ is the resistance or quadratic damping in surge. The Nomoto gain and time constants are recognized as $K > 0$ and $T > 0$, respectively, while according to Norrbin (1963) $n_1 = 1$ for course-stable ships, $n_1 = -1$ for course-unstable ships, and $n_2 > 0$. The model of [176] is a first-order model, which can be used to describe the yaw dynamics of most commercial ships.

Consequently, it is trivial to design a feedback linearizing controller that will result in GES surge velocity error dynamics:

$$
\begin{aligned}
T = &\frac{1}{1-t}[(m - X_{\dot{u}})(\dot{u}_d - K_{p_u}\tilde{u} - K_{i_u}\int_0^t \tilde{u}d\tau) \\
&- X_{|u|u}|u|u],
\end{aligned}
\tag{10.14}
$$

with $(\tilde{\cdot}) = (\cdot) - (\cdot)_d$, and similarly for the heading angle error dynamics:

$$\delta = \frac{1}{K}(n_2 r^3 + n_1 r - K_{p_r}\tilde{\psi} - K_{i_r}\int_0^t \tilde{\psi}d\tau - K_{d_r}\tilde{r}). \tag{10.15}$$

The controllers (10.14)–(10.15) result in GES equilibrium points at $\tilde{u} = \tilde{r} = 0$. In reality this is hardly the case due to the saturation of the actuators. However, this result is useful in order to obtain a proof of concept in Section 10.5 where the stability of the overall path-tracking system is studied and the convergence time of the controllers can affect performance.

10.2.3 Virtual Vehicle Kinematics and Tracking Error

We consider a 2-D continuous path $(x_p(\theta), y_p(\theta))$, where $\theta \geq 0$ denotes the path variable, that goes through a set of successive waypoints (x_k, y_k) for $k = 1, ..., N$. The path-tangential angle at the point $(x_p(\theta^*), y_p(\theta^*))$ (defined by $\theta = \theta^*$) on the path is:

$$\gamma_p = \text{atan2}\left(y_p'(\theta^*), x_p'(\theta^*)\right) \tag{10.16}$$

In the case of a straight line $\gamma_p = \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k)$ is constant between the waypoints, while for a curved path γ_p will be time varying as defined by (10.16).

For the path-tracking scenario it is reasonable to assume that a virtual particle is navigating with a total speed U_t on the desired path, therefore its position $p_t^n = (x_t, y_t)$ is computed by integrating the inertial velocities:

$$\dot{x}_t = U_t \cos(\gamma_p), \tag{10.17}$$

$$\dot{y}_t = U_t \sin(\gamma_p). \tag{10.18}$$

Then the position error for a given vehicle position (x, y) is given by:

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} = \mathbf{R}^\top(\gamma_p) \begin{bmatrix} x - x_t \\ y - y_t \end{bmatrix}, \tag{10.19}$$

therefore, the along-track and the cross-track error can be rewritten:

$$x_e = (x - x_t)\cos(\gamma_p) + (y - y_t)\sin(\gamma_p), \tag{10.20}$$

$$y_e = -(x - x_t)\sin(\gamma_p) + (y - y_t)\cos(\gamma_p). \tag{10.21}$$

The objective of the vehicle in this case is to track the virtual particle, that is $p - p_t \to 0$. For this study we will assume that the virtual vehicle moves on a straight line and that no unknown external disturbances act on the vessel.

10.3   Cross-track Error Minimization

In order to minimize the cross-track error we employ the LOS guidance law. This problem has been studied extensively in the literature and in this chapter we will use the formulation presented for the horizontal plane in [156], briefly revised here. The time-derivative of (10.21) can be written in phase-amplitude form as:

$$\dot{y}_e = U\sin(\psi - \gamma_p + \beta), \tag{10.22}$$

where $U = \sqrt{u^2 + v^2}$ is the total speed of the vehicle and the sideslip angle is defined as:

$$\beta = \text{atan2}(v, u). \tag{10.23}$$

Remark 10.1. It should be noted that in a tracking scenario (10.22) holds only when the virtual vehicle moves on a straight line. On the other hand, for a path-following scenario (10.22) holds for curved paths as well.

Assuming perfect heading tracking ($\psi = \psi_d$) the LOS guidance generates the following heading angle reference trajectories:

$$\psi_d = \gamma_p + \text{atan}\left(\frac{-y_e}{\Delta}\right) - \beta, \tag{10.24}$$

where $\Delta > 0$ is the lookahead distance. Combining (10.22) and (10.24) yields:

$$\dot{y}_e = U\frac{-y_e}{\sqrt{\Delta^2 + y_e^2}}. \tag{10.25}$$

Proposition 3:   For $U, \Delta > 0$ the system (10.25) has a globally κ-exponentially stable equilibrium point at $y_e = 0$.

Proof. The proof can be found in [156]. This was first proven by [193]. $\square$

10.4 Along-track Error Minimization

Contrary to the path-following task, the path-tracking scenario requires the minimization of both (10.20) and (10.21). As it was shown in Section 10.3, the cross-track error can be minimized by generating heading commands according to the LOS guidance law. The along-track error, on the other hand, will be minimized by generating appropriate surge velocity reference trajectories.

Before proceeding we assume the following:

A4: The heading commands are perfectly tracked ($\psi = \psi_d$).

A5: The surge velocity commands are perfectly tracked ($u = u_d$).

The time-derivative of (10.20) in combination with (10.1) gives:

$$
\begin{aligned}
\dot{x}_e &= (\dot{x} - \dot{x}_t)\cos(\gamma_p) + (\dot{y} - \dot{y}_t)\sin(\gamma_p), \\
&= (u\cos(\psi_d) - v\sin(\psi_d) - U_t\cos(\gamma_p))\cos(\gamma_p) + \\
&\quad + (u\sin(\psi_d) + v\cos(\psi_d) - U_t\sin(\gamma_p))\sin(\gamma_p), \\
&= u\cos(\gamma_p - \psi_d) + v\sin(\gamma_p - \psi_d) - U_t.
\end{aligned}
\tag{10.26}
$$

Combining (10.26) with the heading reference trajectories (10.24) yields:

$$
\begin{aligned}
\dot{x}_e &= u\cos(\underbrace{\beta + \operatorname{atan}(y_e/\Delta)}_{\zeta}) + \\
&\quad + v\sin(\underbrace{\beta + \operatorname{atan}(y_e/\Delta)}_{\zeta}) - U_t.
\end{aligned}
\tag{10.27}
$$

In order to transform (10.27) in a more practical form, we exploit the following property:

$$
\operatorname{atan}(a) \pm \arctan(b) = \operatorname{atan}\left(\frac{a \pm b}{1 \mp ab}\right)
\tag{10.28}
$$

Combining (10.23), (10.27) and (10.28) gives:

$$
a = \frac{v}{u}, \quad b = \frac{y_e}{\Delta}
\tag{10.29}
$$

$$
\zeta = \operatorname{atan}\left(\frac{v\Delta + uy_e}{u\Delta - vy_e}\right)
\tag{10.30}
$$

where $\zeta \neq (\pi/2) \pm \kappa\pi$ for $\kappa = 1, 2, ..., n$, for more details on this see Remark 10.2 at the end of this section. For the sake of simplicity we define:

$$
\xi := \frac{v\Delta + uy_e}{u\Delta - vy_e}
\tag{10.31}
$$

From (10.27), (10.30), (10.31), we have:

$$
\begin{aligned}
\dot{x}_e &= u_d\cos(\operatorname{atan}(\xi)) + v\sin(\operatorname{atan}(\xi)) - U_t, \\
&= \frac{u_d}{\sqrt{1 + \xi^2}} + \frac{v\xi}{\sqrt{1 + \xi^2}} - U_t.
\end{aligned}
\tag{10.32}
$$

Consequently, we choose the desired surge velocity as:

$$u_d = -v\xi + \left(\sqrt{1 + \xi^2}\right)(-k_x x_e + U_t),\qquad(10.33)$$

with $k_x > 0$.

Proposition 4: Assuming perfect heading and surge velocity tracking, the system (10.32) has a GES equilibrium point at $x_e = 0$ if the surge velocity assignment is given by (10.33).

Proof. This follows from (10.32)–(10.33) which gives $\dot{x}_e = -k_x x_e$. □

Remark 10.2. Regarding Eq. (10.26), for underactuated vehicles only the surge velocity and the heading angle are available for control. This means that it is not possible to find surge velocity commands capable of minimizing the along-track error (10.26) if the vehicle is moving at a direction normal to the straight line, that is $\cos(\gamma_p - \psi_d) = 0 \Rightarrow \gamma_p - \psi_d = (\pi/2) \pm \kappa\pi$ for $\kappa = 1, 2, ..., n$. However, such a value for the heading angle will be generated by the LOS guidance law only if the cross-track error is infinite, since in that case the most effective way to approach the path is to move on a direction normal to the path: $|y_e| \to \infty \Rightarrow \psi_d \to \gamma_p - (\pi/2)$. However, due to the stability result established in Section 10.3, the cross-track error will indeed be bounded.

10.5 Stability of the Total System

The stability analysis presented in Sections 10.3–10.4 assumed perfect heading and surge velocity tracking. In reality, however, this is never the case since the heading and speed controllers will always need time before converging to the desired values. Consequently, the stability of the overall system should be considered where the heading and surge velocity controllers act as the driving systems and the guidance system, which minimizes the along-track error and the cross-track error, is the driven system. For the sake of clarity, we rewrite first the along-track error/speed controller subsystem in cascade form, followed by the cross-track error/speed controller/heading controller subsystem.

10.5.1 Formulating the Along-track Error/Speed Controller/ Heading Controller Cascade System

The next step is to rewrite the along-track error subsystem as a function of $\tilde{u}$, u, $\tilde{\psi}$ and ψ. The assumptions of perfect heading angle and surge velocity tracking do not hold anymore and since $\tilde{\psi} = \psi - \psi_d$, $\tilde{u} = u - u_d$, we rewrite (10.26) as:

$$\dot{x}_e = (\tilde{u} + u_d)\cos(\gamma_p - \psi_d - \tilde{\psi}) + v\sin(\gamma_p - \psi_d - \tilde{\psi}) - U_t,\qquad(10.34)$$
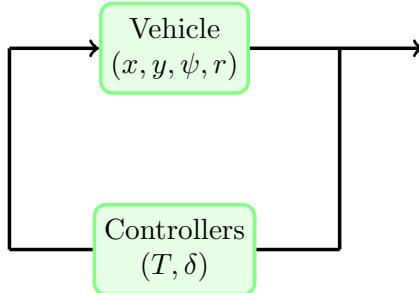
Figure 10.1: The heading and surge velocity controllers along with the vehicle form the driving system of the cascade. The tracking errors $\tilde{u}, \tilde{\psi}$ result in an interconnecting term which can affect the overall performance significantly. The output of this system is $\tilde{\psi}, \tilde{u}$.
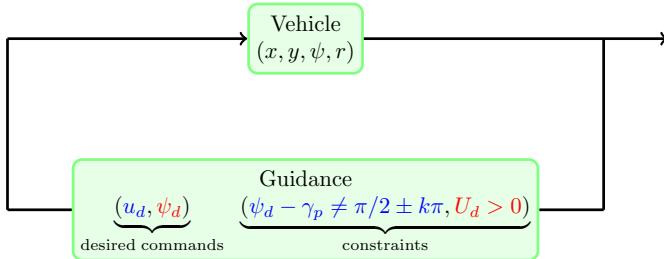
Figure 10.2: The guidance system along with the vehicle form the driven system because its stability depends on the tracking dynamics of the heading controller and the surge velocity controller. The blue command and constraint correspond to the along-track error whereas the red ones correspond to the cross-track error. The output of this system is x_e, y_e.

which, after application of several trigonometric properties and adding/subtracting terms, can be written as:

$$
\begin{aligned}
\dot{x}_e =\,& u_d \cos\left(\gamma_p - \psi_d\right) + v \sin\left(\gamma_p - \psi_d\right) + \\
& + \left(\cos\left(\tilde{\psi}\right) - 1\right)\left(u_d \cos\left(\gamma_p - \psi_d\right) + v \sin\left(\gamma_p - \psi_d\right)\right) + \\
& + \sin\left(\tilde{\psi}\right)\left(u_d \sin\left(\gamma_p - \psi_d\right) - v \cos\left(\gamma_p - \psi_d\right)\right) + \\
& + \tilde{u}\left(\frac{\cos\left(\tilde{\psi}\right)}{\sqrt{1 + \xi^2}} + \frac{\xi \sin\left(\tilde{\psi}\right)}{\sqrt{1 + \xi^2}}\right) - U_t.
\end{aligned} \tag{10.35}
$$

Inspired by [30], we transform (10.35) in the form:

$$
\dot{x}_e = u_d \cos\left(\gamma_p - \psi_d\right) + v \sin\left(\gamma_p - \psi_d\right) - U_t + \boldsymbol{\omega}^{\mathrm{T}} \tilde{\mathbf{x}}_2, \tag{10.36}
$$

where $\tilde{\mathbf{x}}_2 = [\tilde{u} \ \ \tilde{\psi}]$ ($\tilde{\mathbf{x}}_1$ will be defined in Section 10.5.3) and $\boldsymbol{\omega}^{\mathrm{T}} = [\omega_1 \ \ \omega_2]$ with:

$$\omega_1 = \sin(\tilde{\psi} + \phi_{1\xi}), \quad \phi_{1\xi} = \mathrm{atan2}(1, \xi), \tag{10.37}$$

$$\omega_2 = \frac{\cos(\tilde{\psi}) - 1}{\tilde{\psi}}(u_d \sin(\gamma_p - \psi_d) + v \cos(\gamma_p - \psi_d)) +$$

$$+ \frac{\sin(\tilde{\psi})}{\tilde{\psi}}(u_d \sin(\gamma_p - \psi_d) - v \cos(\gamma_p - \psi_d)). \tag{10.38}$$

Combining (10.33) and (10.36) yields:

$$\dot{x}_e = -k_x x_e + \boldsymbol{\omega}^{\mathrm{T}} \tilde{\mathbf{x}}_2. \tag{10.39}$$

10.5.2   Formulating the Cross-track Error/Speed Controller/ Heading Controller Cascade System

Since $\tilde{u} = u - u_d$ and $\tilde{\psi} = \psi - \psi_d$, we rewrite the cross-track error system (10.22) as:

$$\dot{y}_e = (\tilde{u} + u_d)\sin(\tilde{\psi} + \psi_d - \gamma_p) + v \cos(\tilde{\psi} + \psi_d - \gamma_p), \tag{10.40}$$

which, again, after several trigonometric transformations and addition/subtraction of terms, can be written as:

$$\dot{y}_e = u_d \sin(\psi_d - \gamma_p) + v \cos(\psi_d - \gamma_p) +$$

$$+ (\cos(\tilde{\psi}) - 1)(u_d \sin(\psi_d - \gamma_p) + v \cos(\psi_d - \gamma_p)) +$$

$$+ \sin(\tilde{\psi})(u_d \sin(\psi_d - \gamma_p) + v \cos(\psi_d - \gamma_p)) +$$

$$+ \tilde{u}\left(\frac{\sin(\tilde{\psi})}{\sqrt{1 + \xi^2}} + \frac{\xi \cos(\tilde{\psi})}{\sqrt{1 + \xi^2}}\right). \tag{10.41}$$

Inspired by [30], we transform (10.41) in the form:

$$\dot{y}_e = u_d \sin(\psi_d - \gamma_p) + v \cos(\psi_d - \gamma_p) + \boldsymbol{\chi}^{\mathrm{T}} \tilde{\mathbf{x}}_2, \tag{10.42}$$

where $\tilde{\mathbf{x}}_2 = [\tilde{u} \ \ \tilde{\psi}]$ and $\boldsymbol{\chi}^{\mathrm{T}} = [\chi_1 \ \ \chi_2]$ with:

$$\chi_1 = \sin(\tilde{\psi} + \phi_{2\xi}), \quad \phi_{2\xi} = \mathrm{atan}(\xi), \tag{10.43}$$

$$\chi_2 = \frac{\cos(\tilde{\psi}) - 1}{\tilde{\psi}}(u_d \sin(\psi_d - \gamma_p) + v \cos(\psi_d - \gamma_p)) +$$

$$+ \frac{\sin(\tilde{\psi})}{\tilde{\psi}}(u_d \sin(\psi_d - \gamma_p) + v \cos(\psi_d - \gamma_p)). \tag{10.44}$$

Combining (10.24) and (10.42) yields:

$$\dot{y}_e = -\frac{U_d y_e}{\sqrt{\Delta^2 + y_e^2}} + \boldsymbol{\chi}^{\mathrm{T}} \tilde{\mathbf{x}}_2. \tag{10.45}$$

10.5.3 Stability of the Cascaded System

Equations (10.39) and (10.45) indicate that the total system has the same structure as (8.57)–(8.58). We define:

$$\mathbf{x}_1 := [x_e, y_e]^\mathrm{T}, \quad \mathbf{x}_2 := [\tilde{u}, \tilde{\psi}]^\mathrm{T}, \tag{10.46}$$

and compare with (8.57)–(8.58), hence getting:

$$\mathbf{f}(t, \mathbf{x}_1) = -kx_e - \frac{U_d y_e}{\sqrt{\Delta^2 + y_e^2}}, \tag{10.47}$$

$$\mathbf{g}(t, \mathbf{x})\mathbf{x}_2 = \boldsymbol{\omega}^\mathrm{T}\tilde{\mathbf{x}}_2 + \boldsymbol{\chi}^\mathrm{T}\tilde{\mathbf{x}}_2. \tag{10.48}$$

$$\mathbf{f}(t, \mathbf{x}_2) = \left[\left(\frac{1}{m - X_{\dot{u}}}\right) \left(K_{p_u}\tilde{u} + K_{i_u}\int_0^t \tilde{u}d\tau\right), \right.$$
$$\left. \frac{1}{T}(K_{d_r}\tilde{r} + K_{p_r}\tilde{\psi} + K_{i_r}\int_0^t \tilde{\psi}d\tau) \right] \tag{10.49}$$

Therefore we can proceed with the following theorem:

Theorem 10.1. *The total cascade system* (10.47)–(10.49) *has a globally κ-exponentially stable equilibrium point at* $\mathbf{x} = \mathbf{0}$ *if the control laws are given by* (10.14)-(10.15) *and the desired yaw angle and surge velocity are given by* (10.24) *and* (10.33), *respectively.*

Proof. This follows from satisfying Assumptions A1–A3 in Section 2.1.

Assumption A1: From Propositions 1 and 2 we already know that the equilibrium point $y_e = x_e = 0 \Rightarrow \mathbf{x_1} = \mathbf{0}$ is globally κ-globally exponentially stable when the heading angle and the surge velocity are perfectly tracked. By choosing the LFC $V_T = (1/2)(x_e^2 + y_e^2)$, and:

$$\left\|\frac{\partial V_T}{\partial \mathbf{x}_1}\right\| \|\mathbf{x}_1\| = \|\mathbf{x}_1\|\|\mathbf{x}_1\| \Rightarrow$$
$$\left\|\frac{\partial V_T}{\partial \mathbf{x}_1}\right\| \|\mathbf{x}_1\| \leq c_1 V_T(\mathbf{x}_1) \text{ for } c_1 \geq 2 \text{ and } \forall \|\mathbf{x}_1\| \geq 0. \tag{10.50}$$

The condition

$$\left\|\frac{\partial V_T}{\partial \mathbf{x}_1}\right\| \leq c_2 \quad \forall \|\mathbf{x}_1\| \leq \mu \tag{10.51}$$

is also satisfied $\forall \|\mathbf{x}_1\| \leq \mu, \quad \mu > 0$.

Assumption A2: This condition is related to the interconnecting terms, namely $\boldsymbol{\omega}^\mathrm{T}\tilde{\mathbf{x}}$ and $\boldsymbol{\chi}^\mathrm{T}\tilde{\mathbf{x}}$. It can be shown that this constraint is satisfied. Regarding (10.43), we have that $\chi_1 \leq |\tilde{u}|$. Eq. (10.44) is bounded since the terms $(\cos(\tilde{\psi}) - 1)/\tilde{\psi}$ and $\sin(\tilde{\psi})/\tilde{\psi}$ are bounded and well-defined at $\tilde{\psi} = 0$. Similarly, from (10.37) we have that $\omega_1 \leq |\tilde{u}|$ and ω_2 can be shown to be bounded in a similar way as χ_2.

Assumption A3: It has already been proved that the equilibrium points $\tilde{u}, \tilde{\psi} = 0$, $\rightarrow \mathbf{x}_2 = \mathbf{0}$ is GES. This means that the solutions satisfy:

$$|\tilde{u}(t)| \leq \lambda_{u_o}|\tilde{u}(t_o)|e^{-2(t-t_o)} \tag{10.52}$$

$$|\tilde{\psi}(t)| \leq \lambda_{\psi_o}|\tilde{\psi}(t_o)|e^{-2(t-t_o)} \tag{10.53}$$

and therefore by choosing $\nu_1(|\tilde{u}(t_o)|) = (\lambda_{u_o})|\tilde{u}(t_o)|$ and $\nu_2(|\tilde{\psi}(t_o)|) = (\lambda_{\psi_o})|\tilde{\psi}(t_o)|$ the integrability condition is satisfied. Since all three assumptions are satisfied and, in addition to this, the nominal system Σ_1 has a globally κ-exponentially stable equilibrium and the system Σ_2 has a GES equilibrium, we conclude that the cascade system has a globally κ-exponentially stable equilibrium at $\mathbf{x} = \mathbf{0}$. $\qquad\square$

10.6   Simulations

The simulations were implemented with the model for Cybership II, a $1:70$ replica of a supply ship. The model parameters and more details regarding the vessel can be found in [208]. The virtual vehicle's velocity was $U_t = 3$ m/sec. The controllers' gains were chosen as: $K_{p_r} = 100.34$, $K_{d_r} = 44.6$, $K_{i_r} = 10.38$, $K_{p_u} = 47.47$, $K_{i_u} = 34.12$. Moreover, $\Delta = 9$ m and $k_x = 0.3$. When the simulation starts the vessel has an initial cross-track error of approximately 2.2 m and the along-track error starts to increase because the vessel is not moving on the desired path, where the virtual vehicle moves, yet, see Fig. 10.3. As a result, the desired surge velocity increases fast in order for the vessel to catch up with the virtual vehicle, Fig. 10.4. The total position error converges to zero when both $x_e = y_e = 0$, this occurs after 25 sec approximately. From then on, the vessel keeps moving at a constant speed of $u_d = U_t = 3$ m/sec and the steady state heading angle is $\psi = \gamma_p$.
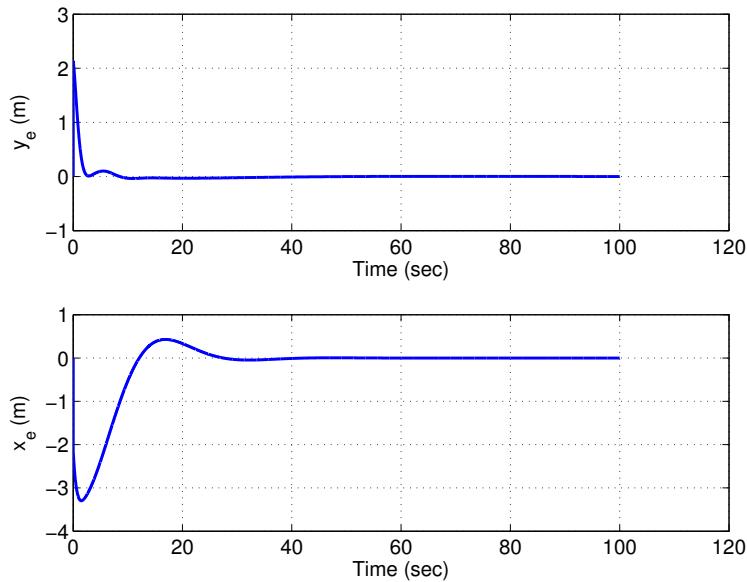
Figure 10.3: The cross-track and the along-track error converge to zero.
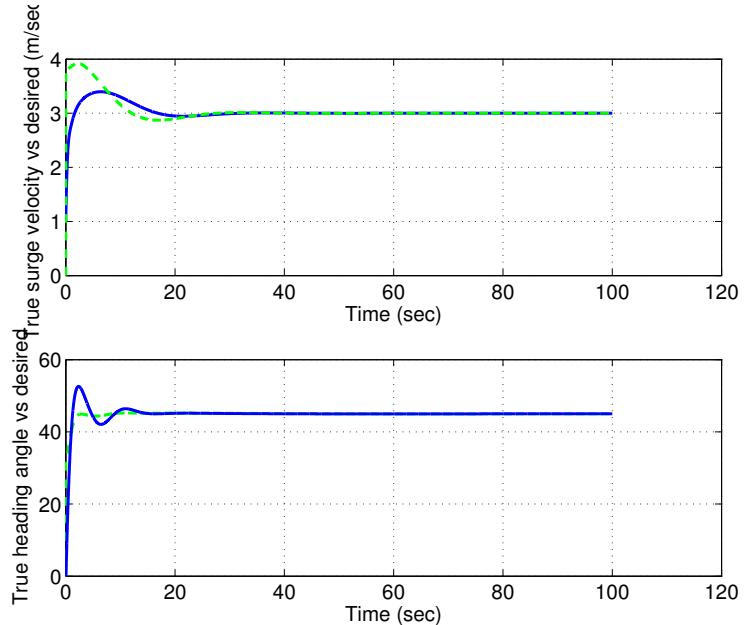
Figure 10.4: True surge velocity u (continuous line) vs desired surge velocity u_d (dashed line) and true heading ψ (continuous line) vs desired heading ψ_d (dashed line).

10.7 Path Tracking under the Influence of Ocean Currents

10.7.1 Introduction

In this section we combine and extend the results of Sections 9.5.3 and 10.3–10.6 in a very straightforward way. First of all, a relative-velocity kinematic setting is considered. In Section 9.5.3, we designed a nonlinear observer for estimating the ocean current effect in the direction normal to the path. Following the same line of reasoning, we now design a nonlinear observer capable of estimating the ocean current effect in the direction tangent to the path. In this way, it is possible to generate suitable trajectories for the speed controller which can compensate for the ocean current effect in that direction, hence making the guidance method developed in the previous sections of the present chapter suitable for path-tracking applications under the influence of ocean currents. It will be shown that the estimates from the two observers (the one from Section 9.5.3 for the cross-track error and the one we develop here for the along-track error) make it possible to compute the ocean current parameters w.r.t. the inertial frame without requiring absolute velocity measurements. The results in this section are unpublished and presented here without proof for the total cascade system. However, finding a proof for this guidance scheme is feasible if an approach similar to that of Section 10.5 is followed.

10.7.2 Guidance System Development

In this case we will work again with the relative velocity-based kinematics. We consider again the along-track error equation:

$$x_e = (x - x_t) \cos(\gamma_p) + (y - y_t) \sin(\gamma_p), \tag{10.54}$$

and, similarly to the absolute velocities case, differentiate x_e w.r.t. time:

$$
\begin{aligned}
\dot{x}_e &= u_r \cos(\psi_d) \cos(\gamma_p) - v_r \sin(\psi_d) \cos(\gamma_p) + V_x \cos(\gamma_p) \\
&= +u_r \sin(\psi_d) \sin(\gamma_p) + v_r \cos(\psi_d) \sin(\gamma_p) + V_y \sin(\gamma_p) - U_t, \tag{10.55} \\
&= u_r \cos(\psi_d - \gamma_p) + v_r \sin(\gamma_p - \psi_d) + \underbrace{U_c \sin(\gamma_p + \beta_{cx})}_{\theta_x} - U_t, \tag{10.56}
\end{aligned}
$$

where $\beta_{cx} = \text{atan2}(V_x, V_y)$. It should be noted that in previous sections (9.5, for instance), we defined $\beta_c = \text{atan2}(V_y, V_x)$. Consequently, we observe that $\beta_c + \beta_{cx} = \pi/2$. We rewrite (10.56) in a more convenient form:

$$\dot{x}_e = u_r \cos(\gamma_p - \psi) + v_r \sin(\gamma_p - \psi) + \theta_x - U_t. \tag{10.57}$$

From the solution of the cross-track error minimization problem (see Section 9.5) we already have:

$$\psi_d = \gamma_p - \beta_r + \arctan\left(-\frac{1}{\Delta}(y_e + \alpha_y)\right), \tag{10.58}$$

$$\alpha_y = \Delta \frac{\hat{\theta}_y / U_r}{\sqrt{1 - (\hat{\theta}_y / U_r)^2}}, \tag{10.59}$$

$$\hat{\theta}_y = k_{2y}(y_e - \hat{y}_e). \tag{10.60}$$

Note that, compared to Section 9.5.3, the notation in (10.58)–(10.60) has been modified slightly in order to avoid confusions with the observer we are designing in this section. It is interesting to observe that, when it comes to the path-tracking problem as a whole, the LOS algorithm (10.58) acts as a constraint to the part of the guidance algorithm which generates the speed reference trajectories. Combining (10.57) and (10.58) yields:

$$
\dot{x}_e = u_r \cos\left(-\beta_r + \arctan\left(-\frac{1}{\Delta}(y_e + \alpha_y)\right)\right)
$$

$$
+ u_r \sin\left(\underbrace{-\beta_r + \arctan\left(-\frac{1}{\Delta}(y_e + \alpha_y)\right)}_{\zeta_t}\right) + \theta_x - U_t, \tag{10.61}
$$

where ζ_t is computed as follows:

$$
\zeta_t = \arctan\left(\frac{v_r \Delta + u_r(y_e + \alpha_y)}{v_r(y_e + \alpha_y) - u_r \Delta}\right). \tag{10.62}
$$

For the sake of notational brevity we define $\xi_t := \arctan(\zeta_t)$ and rewrite once again the along-track error propagation equation:

$$
\dot{x}_e = \frac{u_{rd}}{\sqrt{1 + \xi_t^2}} + \frac{v_r}{\sqrt{1 + \xi_t^2}} - U_t + \theta_x. \tag{10.63}
$$

Therefore, a suitable relative velocity reference trajectory u_{rd} must be computed in order to minimize x_e. We observe that choosing:

$$
u_{rd} = \sqrt{1 + \xi_t^2}\left(-v_r \frac{\xi_t}{\sqrt{1 + \xi_t^2}} + U_t + \alpha_x - k_x x_e\right), \tag{10.64}
$$

where $\alpha_x = -\hat{\theta}_x$, we get the expression:

$$
\dot{x}_e = -k_x x_e + \alpha_x + \theta_x. \tag{10.65}
$$

We proceed with proposing the following observer:

$$
\dot{\hat{x}}_e = -k_x \hat{x}_e + \hat{\theta}_x + \alpha_x + k_{1x}(x_e - \hat{x}_e), \tag{10.66}
$$

$$
\dot{\hat{\theta}}_x = k_{2x}(x_e - \hat{x}_e), \tag{10.67}
$$

which results in the error dynamics:

$$
\dot{\tilde{x}}_e = -(k_{x_e} + k_{1x})\tilde{x}_e + \tilde{\theta}_x, \tag{10.68}
$$

$$
\dot{\tilde{\theta}}_x = -k_{2x}\tilde{x}_e. \tag{10.69}
$$

Equations (10.68)–(10.69) describe a linear system with a globally exponentially stable equilibrium point at $(\tilde{x}_e, \tilde{\theta}_x) = (0,0)$, if the gains involved are chosen so that the matrix

$$
\mathbf{K} = \begin{bmatrix} -(k_{x_e} + k_{1x}) & 1 \\ -k_{2x} & 0 \end{bmatrix} \tag{10.70}
$$

has two negative real eigenvalues.

The methodology presented here for the path-tracking problem utilizes two adaptive controllers in order to compensate for the current effects. One main improvement in this case is that both θ_y and θ_x are estimated and, therefore, it is now possible to estimate all the parameters of the current. Starting from:

$$\theta_y = U_c \sin(\gamma_p - \beta_c), \tag{10.71}$$

$$\theta_x = U_c \cos(\gamma_p - \beta_c), \tag{10.72}$$

first we divide (10.71) by (10.72), which gives the expression for the current angle estimate:

$$\hat{\beta}_c = \gamma_p + \text{atan2}(\hat{\theta}_y, \hat{\theta}_x), \tag{10.73}$$

and then we compute the current velocity:

$$\hat{U}_c = \sqrt{(\hat{\theta}_y)^2 + (\hat{\theta}_x)^2}. \tag{10.74}$$

10.7.3 Simulations

The relative velocity-based path-tracking guidance system developed in this section was simulated in order to test its efficiency. A simple 3-DOF model was used in order to demonstrate the efficiency of the algorithm. The dynamics are similar to (8.3)–(8.5) and the relative kinematics as is (9.26)–(9.28). The vehicle mass was $m = 1$ kg, its speed controlled by a proportional controller and the yaw by a feedback linearizing controller of the same structure as (8.52) with $\alpha_1 = \alpha_2 = \lambda = b = 1$. The virtual vehicle is moving with speed $U_t = 5$ m/s along the line connecting the waypoints $(0,0) - (60, 200)$, which gives the path-tangential angle $\gamma_p = 73.3$ deg. The ocean current velocity vector magnitude is $U_c = 1$ m/s and its orientation w.r.t. the inertial frame is $\beta_c = -40$ deg. The initial position and heading of the marine vessel is $(x_0, y_0, \psi_0) = (-20, 10, 0)$ and the lookahead distance for the LOS algorithm $\Delta = 50$ m. The gain values are $k_x = 0.5$, $k_{1y} = 10$, $k_{2y} = 0.8$, $k_{1x} = 10$, and $k_{2x} = 1$. The observers' initial conditions are $(\hat{x}_{e0}, \hat{\theta}_{x0}, \hat{y}_{e0}, \hat{\theta}_{y0}) = (0, 0, 0, 0)$. Finally, the simulation duration is 250 seconds, with a sampling time of 0.01 sec. Figures 10.5–10.6 show that the observers are successful in estimating both θ_x and θ_y. In addition, it can be seen that the guidance system is using this information in order to minimize both the cross-track and along-track errors, which converge to zero. The control inputs of the guidance system, α_x and α_y, as well as θ_n are plotted in Fig. 10.8. Fig. 10.7 shows the estimates of both the current angle and current velocity w.r.t. the inertial frame converging to the real values. Finally, Fig. 10.9 plots the heading, relative sideslip, and absolute sideslip angles. It is observed that in the steady state β is nonzero, because the heading is not aligned with the course angle, whereas β_r is zero because the vehicle is following a straight line and is not turning.
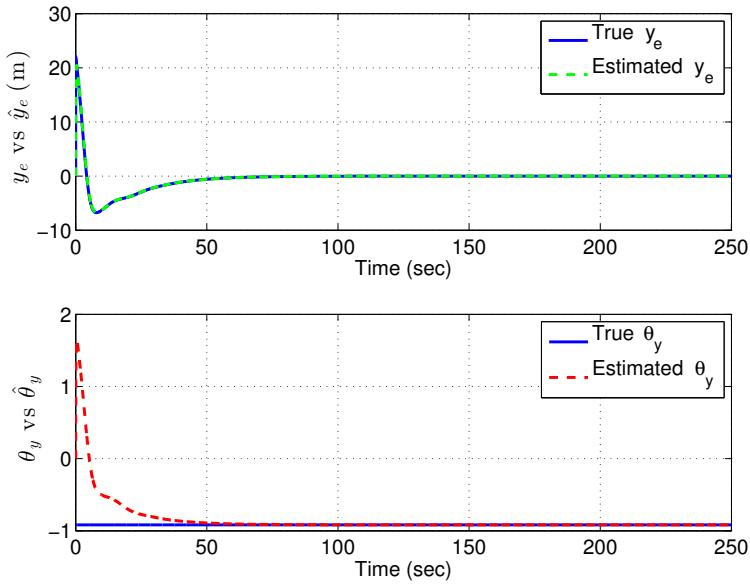
Figure 10.5: Estimates from the nonlinear observer related to the cross-track error.
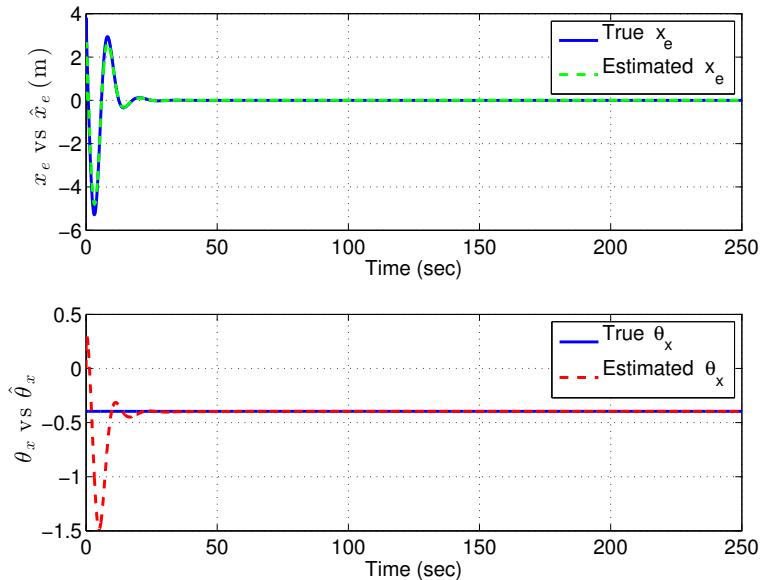
Figure 10.6: Estimates from the nonlinear observer related to the along-track error.
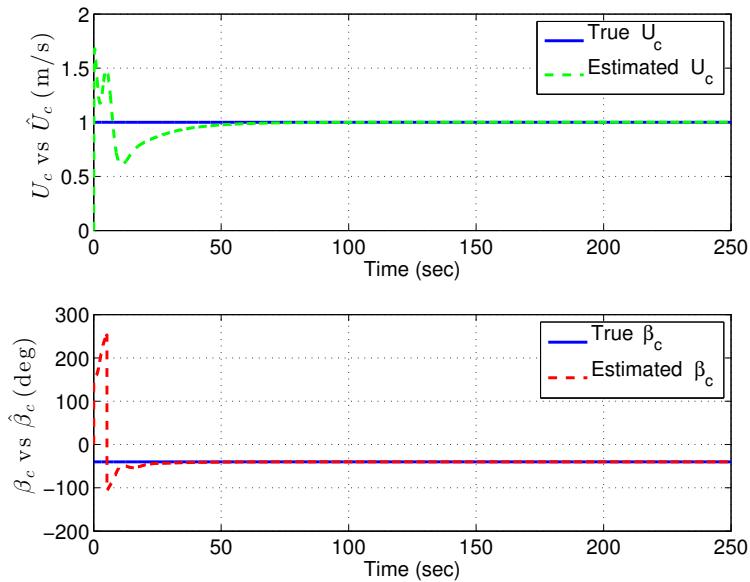
Figure 10.7: The path-tracking case with the two adaptive observers results in the estimation of both the current velocity and orientation in the NED frame.
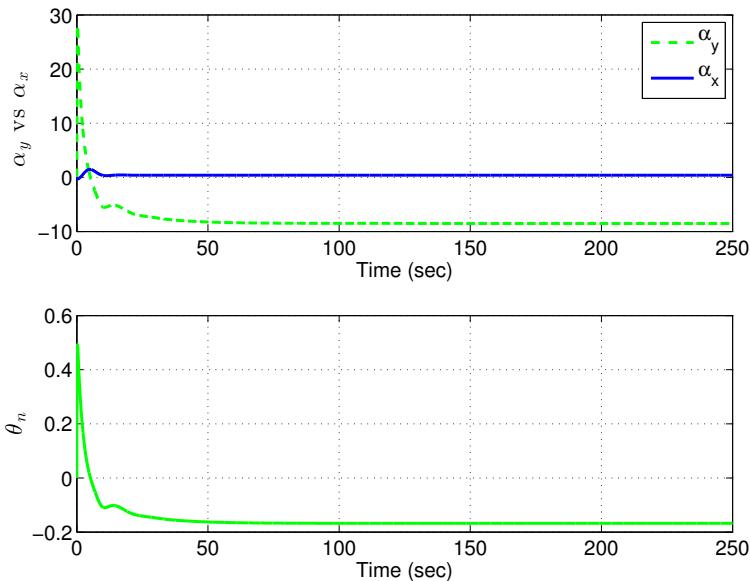
Figure 10.8: Plots of the control inputs α_y, α_x, and θ_n.
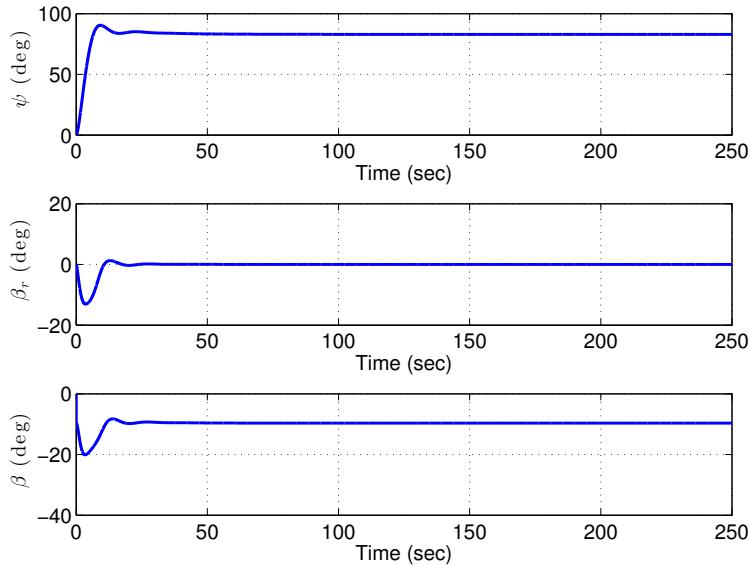
Figure 10.9: Plots of the heading, relative sideslip, and absolute sideslip angles.

10.8 Conclusions

This chapter dealt with the development of a guidance technique that can generate appropriate reference trajectories for the surge velocity of an underactuated system in order to minimize the along-track error in a path-tracking scenario. These velocity reference trajectories are fed into the surge velocity controller. This solution works in cooperation with the, already well-known, LOS guidance which generates the corresponding reference trajectories for the heading angle so as to minimize the cross-track error in the same scenario. The heading angle reference trajectories are fed into the heading autopilot. The total stability of the cascade system consisting of the vehicle, the guidance system and the controllers was studied and it was shown to be globally κ-exponentially stable, a result which was supported by computer simulations.

Then, the approach was extended to the case where a constant current force acts on the vehicle. Using relative-velocity kinematics, an observer for the along-track error and the current effect in the direction tangent to the path was added to the design. The outcome was a successful estimation of the two parameters of the ocean current, something which was not possible in the path-following case. An interesting extension is to solve the path-tracking problem for 3-D curves paths under the influence of ocean currents, by taking into account actuator constraints.

Part IV

Closing Remarks

Chapter 11

Conclusions & Future Work

The results presented in this thesis can be extended in many ways, some more obvious than others.

Path-Evaluation Criteria. Regarding the path-evaluation criteria, the most straightforward step is to quantify tractability. Quantification is probably one of the key elements to creating algorithms capable of making decisions in real time. All the criteria presented in Chapter 3, with the exception of tractability, can be quantified without problems. Then it is mostly a matter of assigning weighing factors to each one of them, which is a procedure totally dependent on the problem in hand, and the final decision can be made by comparing the overall scores among paths. Parameters that could be used to quantify tractability are the rate of curvature, or the derivative of the path-tangential angle expression. In this way, even if a path has a very low allowance, it will be penalized if course changes occur unnecessarily often. The shape control can be assessed by developing a standard test where the effect of displacing a number of waypoints of a pre-specified path is measured in terms of allowance, for instance. It might also be worth looking into techniques stemming from fuzzy logic, so as to allow for some tolerance in assessing tractability. Moreover, it is recommendable to include additional criteria if a specific application would benefit from it. A very good example of this is the ability of a path-generation method to create offset paths, a property which could be of great use in applications related to formation control.

Fermat's Spiral Paths. Their simplicity, low demand for computational power, and nice properties when it comes to the smoothness, tractability etc, indicate that FS-based paths should be extended in order to cover a larger number of applications. The most obvious way to do this is to extend the FS paths in three dimensions, a goal which presupposes the definition of torsion. A similar procedure as the one in [116] could be followed for that purpose. Such a task would also require straight lines with torsion characteristics, so that a vehicle can move along them with different assigned roll angles. The existence of such paths, given specific problem constraints, could entail theoretical challenges, similar to those confronted when designing Dubins paths. The methodology of [59] could be a good starting point. In addition, extending to parallel paths for formation control is of interest.

The geometry of the FS paths is expected to be beneficial for formation control because it is easy to create offset curves, not to mention the benefit from the low computational cost, which would be even more evident for more than one paths being generated in real time. Naturally, the collision-avoidance method of Chapter 6 can be modified so as to assist in achieving such an implementation. In that case, the clearance should also take into account the minimum desired distance between paths for different vehicles. Defining this minimum distance is an interesting part of the problem, because the maximum cross-track error induced by unknown (albeit necessarily bounded) disturbances should be incorporated in the design procedure.

Monotone Cubic Hermite Spline Interpolation. It is the author's opinion that paths consisting of straight lines and arc segments (combinations of circular and spiral arcs, in particular) should be preferred over curved paths, whenever the motion control scenario and the geometry of the problem permit it, of course. Several arguments justifying this were given in Chapter 5. However, in cases where a curved path is the best way to go, it is worth investigating which methods give superior results. Curvature continuity is often a desired property by designers, but these approaches are usually less flexible. Maybe it is worth sacrificing curvature continuity on the waypoints and obtain a more practical path in return. In any case, there might be methodologies in the existing literature which can result in curved paths with nice shape properties and sufficient smoothness both in 2D and 3D [64, 65, 146, 167]. These should be investigated and compared with CHSI.

Voronoi Diagrams. This very powerful tool can certainly be further modified so as to include more parameters which are of interest when addressing the collision-avoidance problem for UAVs, AUVs etc. Regarding marine surface vessels, a useful extension is to add one parameter to the problem constraints, and this would be the sea depth. To this end, [53] should be designed so as to construct a Voronoi Diagram (VD) which will take into account two parameters: a) the clearance (as before), and b) the water depth. Then the VD will generate a roadmap which will keep the ship at a safe distance from both the obstacles in the plane and shallow waters. Additional interesting extensions are the inclusion of a dynamic obstacle by modifying the VD in real time. Also, a three-dimensional VD-based methodology could be of great use for increasing the autonomy level and performance of underwater vehicles. To this end, complementary sensing-based techniques would be necessary so as to make the approach safer and more robust.

Line-of-Sight Guidance and Path Tracking. Despite the fact that the LOS guidance law has been studied extensively by the marine control community, especially over the last ten years, there might be some room for substantial improvement. This would be to incorporate a derivative term for avoiding oscillations around the desired path, an approach which is expected to give better results compared to the LOS guidance with time-varying lookahead distance. This improvement would come at a cost, that of adding one extra state to the problem, not to mention the more complicated analysis in order to infer upon the stability of the system. The method must be able to eliminate oscillations when taking actuator constraints into account. Otherwise the whole endeavor might be pointless,

since the time-varying Δ can improve performance in normal cases. Finally, for the PID-LOS guidance, it would be very useful to establish a methodology for tuning the gains in a structured and efficient way. Furthermore, the quaternion version of the LOS can be modified so as to consider the full 6-DOF kinematics. In a similar way, the direct and indirect integral LOS path-following controllers can be easily extended to three dimensions. The same is true about the guidance law for path tracking developed in Chapter 10.

The overall goal is to combine all these solutions and build an efficient and unified approach for general path-tracking applications. General, in this case, pertains to tracking a curved path in 3-D space under the influence of unknown forces, with actuator constraints and modeling uncertainties. These uncertainties was one extra motivation for developing the several methods of this thesis at a kinematic level. A feedback-based approach is by nature suitable for dealing with such problems, contrary to what might be the case with predictive model-based methodologies, as discussed in [212].

Appendices

Appendix A

Dijkstra-Yen Algorithm

A.1 Dijkstra-Yen Algorithm

Dijkstra's algorithm (see [72]) is an optimization algorithm widely used in operational research. It is a (non-oriented) graph-based algorithm where usually one node represents the starting point and another node represents the ending point of the path we want to build up. A table should collect the properties of the links that connects the various nodes, where usually the element $T(i, j) > 0$ of the table represents the *connection penalty* that covering the (i, j) link will add to the final *total path penalty*, obviously $T(i, j) = \{\emptyset\}$ means that the nodes (i, j) cannot be connected. The goal of the algorithm is to chose the path that assures the minimum penalty between all the possible feasible paths.

```
function Dijkstra(Graph, initNode, endNode)
    for each vertex v in Graph
        dist[v] := infinity
        previousVertex[v]=undefined
    end for
    dist[source]=0
    Q=the set of all nodes in Graph
    while Q is not empty
        u=vertex in Q with smallest distance
        remove u from Q
        for each neighbor v of u
            alt=dist[u]+dist_between(u, v)
            if alt < dist[v]
                dist[v]=alt
                previous[v]=u
            end if
        end for
    end while
end function
```

Yen added a variation to this algorithm that allows to find the *k-shortest path* (238), that means the second shortest path, the third one and so on. In this work

this algorithm has been taken as inspiration to find the second optimal path if the first one does not match the clearance constraints. The idea of the algorithm can be seen from the following pseudo-code:

```
function pseudo-Yen(Graph,initNode,endNode,minClearance)
  while checkClearance==1
    for each waypoint i of optimal path
      if distance (waypoint(i)-obstacles)<minClearance
        rootPath=optimalPath(0:i);
        spurNode=i;
        update Graph to eliminate all connection to i;
        endPath=dikstra(Graph, spurNode, endNode);
        newPath=[rootPath, endPath];
        checkClearance=1;
      else
          check Clearance=0;
      end if
    end for
  end while
end function
```

The algorithm is described by few steps, the logic is to add step by step a new node to the path that is connected to one of the nodes already composing the path. The selection of the new node always consider the fact that we want the minimum penalty links included in the final path.

References

[1] M. Abramowitz and I. Stegun. *Handbook of mathematical functions*. Dover Publishing Inc. New York, 1970.

[2] A. P. Aguiar and J. P. Hespanha. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE Transactions on Automatic Control*, 52(8):1362–1379, 2007.

[3] G. Albers and T. Roos. Voronoi diagrams of moving points in higher dimensional spaces. In *Algorithm Theory—SWAT'92*, pages 399–409. Springer, 1992.

[4] G. Albers, L. J. Guibas, J. S. Mitchell, and T. Roos. Voronoi diagrams of moving points. *International Journal of Computational Geometry & Applications*, 8(03):365–379, 1998.

[5] V. Alberti and E. Babbel. Improved driving stability by active braking of the individual wheels. In *Proc. of the International Symposium on Advanced Vehicle Control*, pages 717–732, 1996.

[6] F. Alonge, F. D'Ippolito, and F. Raimondi. Trajectory tracking of underactuated underwater vehicles. In *Proceedings of the 40th IEEE Conference on Decision and Control*, volume 5, pages 4421–4426, 2001.

[7] G. Ambrosino, M. Ariola, U. Ciniglio, F. Corraro, A. Pironti, and M. Virgilio. Algorithms for 3D UAV path generation and tracking. In *45th IEEE Conference on Decision and Control*, pages 5275–5280. IEEE, 2006.

[8] G. Ambrosino, M. Ariola, U. Ciniglio, F. Corraro, E. De Lellis, and A. Pironti. Path generation and tracking in 3-d for uavs. *IEEE Transactions on Control Systems Technology*, 17(4):980–988, 2009.

[9] E. P. Anderson, R. W. Beard, and T. W. McLain. Real-time dynamic trajectory smoothing for unmanned air vehicles. *IEEE Transactions on Control Systems Technology*, 13(3):471–477, 2005.

[10] D. Angeli and E. D. Sontag. Forward completeness, unboundedness observability, and their Lyapunov characterizations. *Systems & Control Letters*, 38 (4):209–217, 1999.

[11] G. Antonelli. On the use of adaptive/integral actions for six-degrees-of-freedom control of autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 32(2):300–312, 2007.

[12] G. Antonelli, F. Caccavle, S. Chiaverini, and G. Fusco. A novel adaptive control law for autonomous underwater vehicles. In *Proceedings of the International Conference on Robotics and Automation*, volume 1, pages 447–452. IEEE, 2001.

[13] G. Antonelli, F. Caccavale, S. Chiaverini, and G. Fusco. A novel adaptive control law for underwater vehicles. *IEEE Transactions on Control Systems Technology*, 11(2):221–232, 2003.

[14] G. Antonelli, S. Chiaverini, and G. Fusco. A fuzzy-logic-based approach for mobile robot path tracking. *IEEE Transactions on Fuzzy Systems*, 15(2):211–221, 2007.

[15] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.

[16] E. Bakolas and P. Tsiotras. Optimal pursuit of moving targets using dynamic Voronoi diagrams. In *49th IEEE Conference on Decision and Control*, pages 7431–7436, 2010.

[17] E. Bakolas and P. Tsiotras. Time-optimal synthesis for the Zermelo-Markov-Dubins problem: The constant wind case. In *American Control Conference*, pages 6163–6168. IEEE, 2010.

[18] E. Bakolas and P. Tsiotras. The Zermelo–Voronoi diagram: A dynamic partition problem. *Automatica*, 46(12):2059–2067, 2010.

[19] E. Bakolas and P. Tsiotras. Optimal synthesis of the Zermelo–Markov–Dubins problem in a constant drift field. *Journal of Optimization Theory and Applications*, 156(2):469–492, 2013.

[20] B. A. Barsky and A. D. DeRose. Geometric continuity of parametric curves. Technical report, University of California at Berkeley, 1984.

[21] B. A. Barsky and T. D. DeRose. Geometric continuity of parametric curves: three equivalent characterizations. *Computer Graphics and Applications, IEEE*, 9(6):60–69, 1989.

[22] D. M. Bevly, R. Sheridan, and J. C. Gerdes. Integrating INS sensors with GPS velocity measurements for continuous estimation of vehicle sideslip and tire cornering stiffness. In *Proceedings of the American Control Conference*, pages 25–30. IEEE, 2001.

[23] P. Bhattacharya and M. L. Gavrilova. Voronoi diagram in optimal path planning. In *4th International Symposium on Voronoi Diagrams in Science and Engineering*, pages 38–47. IEEE, 2007.

[24] P. Bhattacharya and M. L. Gavrilova. Roadmap-based path planning-using the Voronoi diagram for a clearance-based shortest path. *Robotics & Automation Magazine*, 15(2):58–66, 2008.

[25] M. Bibuli, M. Caccia, and L. Lapierre. Path-following algorithms and experiments for an autonomous surface vehicle. In *Proceedings of IFAC Conference onControl Applications in Marine Systems*, Bol, Croatia, 2007.

[26] M. Bibuli, G. Bruzzone, M. Caccia, G. Indiveri, and A. A. Zizzari. Line following guidance control: Application to the Charlie unmanned surface vehicle. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 2008.

[27] M. Bibuli, G. Bruzzone, M. Caccia, and L. Lapierre. Path-following algorithms and experiments for an unmanned surface vehicle. *Journal of Field Robotics*, 26(8):669–688, 2009.

[28] J. D. Boissonnat, A. Cerezo, and J. Leblond. A note on shortest paths in the plane subject to a constraint on the derivative of the curvature. Technical report, INRIA, 1994.

[29] J.-D. Boissonnat, C. Wormser, and M. Yvinec. Curved voronoi diagrams. In *Effective Computational Geometry for Curves and Surfaces*, pages 67–116. Springer, 2006.

[30] E. Børhaug and K. Y. Pettersen. Cross-track control for underactuated autonomous vehicles. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, pages 602–608, Seville, Spain, 2005.

[31] E. Børhaug, A. Pavlov, and K. Y. Pettersen. Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents. In *47th IEEE Conference on Decision and Control*, pages 4984–4991, Cancun, Mexico, 2008.

[32] E. Børhaug, A. Pavlov, E. Panteley, and K. Y. Pettersen. Straight line path following for formations of underactuated marine surface vessels. *IEEE Transactions on Control Systems Technology*, 19(3):493–506, 2011.

[33] S. A. Bortoff. Path planning for uavs. In *Proceedings of the American Control Conference*, volume 1, pages 364–368. IEEE, 2000.

[34] C. L. Bottasso, D. Leonello, and B. Savini. Path planning for autonomous vehicles by trajectory smoothing using motion primitives. *IEEE Transactions on Control Systems Technology*, 16(6):1152–1168, 2008.

[35] M. Breivik. *Topics in guided motion control of marine vehicles*. PhD thesis, Norwegian University of Science and Technology, 2010.

[36] M. Breivik and T. I. Fossen. Path following for marine surface vessels. In *Proceedings of the OTO'04*, Kobe, Japan, 2004.

[37] M. Breivik and T. I. Fossen. Path following of straight lines and circles for marine surface vessels. In *Proc. of the IFAC CAMS*, Ancona, Italy, 2004.

[38] M. Breivik and T. I. Fossen. Principles of guidance-based path following in 2D and 3D. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, pages 627–634, Seville, Spain, 2005.

[39] M. Breivik and T. I. Fossen. Applying missile guidance concepts to motion control of marine craft. In *Proceedings of the 7th IFAC Conference on Control Applications in Marine Systems*, Bol, Croatia, 2007.

[40] M. Breivik and T. I. Fossen. Guidance laws for planar motion control. In *47th IEEE Conference on Decision and Control*, pages 570–577. IEEE, 2008.

[41] M. Breivik and T. I. Fossen. *Guidance laws for autonomous underwater vehicles*, chapter 4, pages 51–76. INTECH Education and Publishing, 2009.

[42] M. Breivik, V. E. Hovstein, and T. I. Fossen. Straight-line target tracking for unmanned surface vehicles. *Modeling, Identification and Control*, 29(4): 131–149, 2008.

[43] R. W. Brockett. Asymptotic stability and feedback stabilization. *Differential Geometric Control Theory*, pages 181–191, 1983.

[44] R. A. Brooks. Elephants don't play chess. *Robotics and autonomous systems*, 6(1):3–15, 1990.

[45] R. A. Brooks. Intelligence without representation. *Artificial intelligence*, 47 (1):139–159, 1991.

[46] H. Bruyninckx and D. Reynaerts. Path planning for mobile and hyper-redundant robots using Pythagorean hodograph curves. In *8th International Conference on Advanced Robotics*, pages 595–600. IEEE, 1997.

[47] M. Burger. *Disturbance Rejection using Conditional Integrators*. PhD thesis, Norwegian University of Science and Technology, 2011.

[48] M. Burger, A. Pavlov, and K. Y. Pettersen. Conditional integrators for path following and formation control of marine vessels under constant disturbances. In *The 8th IFAC International Conference on Manoeuvring and Control of Marine Craft*, 2009.

[49] W. Caharija, M. Candeloro, K. Y. Pettersen, and A. J. Sørensen. Relative velocity control and integral LOS for path following of underactuated surface vessels. In *9th IFAC Conference on Manoeuvring and Control of Marine Craft*, Arenzano, Italy, 2012.

[50] W. Caharija, K. Y. Pettersen, A. J. Sørensen, M. Candeloro, and J. T. Gravdahl. Relative velocity control and integral line of sight for path following of autonomous surface vessels: Merging intuition with theory. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, pages 1–12, 2013.

[51] A. Caiti, V. Calabro, F. Di Corato, D. Meucci, and A. Munafo. Cooperative distributed algorithm for AUV teams: a minimum entropy approach. In *OCEANS*, Bergen, Norway, 2013. IEEE.

[52] J. Candela, R. C. Beardsley, and R. Limeburner. Separation of tidal and subtidal currents in ship-mounted acoustic Doppler current profiler observations. *Journal of geophysical research*, 97(C1):769–788, 1992.

[53] M. Candeloro, A. M. Lekkas, A. J. Sørensen, and T. I. Fossen. Continuous curvature path planning using Voronoi diagrams and Fermat's spirals. In *9th IFAC Conference on Control Applications in Marine Systems*, Osaka, Japan, 2013.

[54] J. Canny. *The complexity of robot motion planning*. The MIT press, 1988.

[55] J. Canny and B. Donald. Simplified Voronoi diagrams. *Discrete & Computational Geometry*, 3(1):219–236, 1988.

[56] T. J. Chalko. High accuracy speed measurement using GPS (Global Positioning System). *Scientific Engineering Research P/L*, 2002.

[57] P. Chandler, S. Rasmussen, and M. Pachter. UAV cooperative path planning. In *AIAA Guidance, Navigation, and Control Conference*, pages 1255–1265, 2001.

[58] A. J. Chang, M. Brazil, J. H. Rubinstein, and D. A. Thomas. Curvature-constrained directional-cost paths in the plane. *Journal of Global Optimization*, 53(4):663–681, 2012.

[59] H. Chitsaz and S. M. LaValle. Time-optimal paths for a Dubins airplane. In *46th IEEE Conference on Decision and Control*, 2007.

[60] H. Choset. *Sensor based motion planning: The hierarchical generalized Voronoi graph*. PhD thesis, California Institute of technology, 1996.

[61] H. Choset, I. Konukseven, and J. Burdick. Mobile robot navigation: issues in implementating the generalized voronoi graph in the plane. In *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 241–248. IEEE, 1996.

[62] H. Choset, I. Konukseven, and A. Rizzi. Sensor based planning: A control law for generating the generalized voronoi graph. In *8th International Conference on Advanced Robotics*, pages 333–338. IEEE, 1997.

[63] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. The MIT Press, 2005.

[64] P. Costantini and C. Manni. Shape-preserving C^3 interpolation: the curve case. *Advances in Computational Mathematics*, 18(1):41–63, 2003.

[65] P. Costantini, T. N. Goodman, and C. Manni. Constructing C^3 shape preserving interpolating space curves. *Advances in Computational Mathematics*, 14(2):103–127, 2001.

[66] R. Curry, M. Lizarraga, B. Mairs, and G. H. Elkaim. L_2^+, an improved line of sight guidance law for UAVs. In *American Control Conference*. IEEE, 2013.

[67] A. R. Dahl. Overview and evaluation of path planning and guidance algorithms. Technical report, Department of Engineering Cybernetics, NTNU, 2012.

[68] A. R. Dahl. Path planning and guidance for marine surface vessels. Master's thesis, Norwegian University of Science and Technology, 2013.

[69] R. Daily and D. M. Bevly. The use of GPS for vehicle stability control systems. *IEEE Transactions on Industrial Electronics*, 51(2):270–277, 2004.

[70] M. Davidson, V. Bahl, and K. L. Moore. Spatial integration for a nonlinear path tracking control law. In *Proceedings of the 2002 American Control Conference.*, pages 4291–4296. IEEE, 2002.

[71] R. A. DeCarlo, S. H. Zak, and G. P. Matthews. Variable structure control of nonlinear multivariable systems: A tutorial. *Proceedings of the IEEE*, 76(3): 212–232, 1988.

[72] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[73] G. L. Dirichlet. Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *J. reine angew. Math.*, 40:209–227, 1850.

[74] F. Dougherty and G. Woolweaver. At-sea testing of an unmanned underwater vehicle flight control system. In *Proceedings of the (1990) Symposium on Autonomous Underwater Vehicle Technology*, pages 65–73. IEEE, 1990.

[75] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516, 1957.

[76] P. Encarnaçao and A. Pascoal. Combined trajectory tracking and path following: an application to the coordinated control of autonomous marine craft. In *Proceedings of the 40th IEEE Conference on Decision and Control*, pages 964–969, 2001.

[77] D. Eppstein. Finding the k shortest paths. *SIAM Journal on computing*, 28 (2):652–673, 1998.

[78] S. Eriksson-Bique, D. Kirkpatrick, and V. Polishchuk. Discrete Dubins paths. *arXiv preprint arXiv:1211.2365*, 2012.

[79] G. Farin and N. Sapidis. Curvature and the fairness of curves and surfaces. *Computer Graphics and Applications*, 9(2):52–57, 1989.

[80] R. T. Farouki. Pythagorean-hodograph quintic transition curves of monotone curvature. *Computer-Aided Design*, 29(9):601–606, 1997.

[81] R. T. Farouki. *Pythagorean—hodograph Curves.* Springer, 2008.

[82] R. T. Farouki and T. Sakkalis. Pythagorean hodographs. *IBM Journal of Research and Development*, 34(5):736–752, 1990.

[83] R. T. Farouki and T. Sakkalis. Real rational curves are not 'unit speed'. *Computer Aided Geometric Design*, 8(2):151–157, 1991.

[84] P. Fermat. Ad locos planos et solidos isagoge. *Varia Opera Mathematica*, 1679.

[85] O.-E. Fjellstad and T. I. Fossen. Position and attitude tracking of AUV's: a quaternion feedback approach. *IEEE Journal of Oceanic Engineering*, 19 (4):512–518, 1994.

[86] O.-E. Fjellstad and T. I. Fossen. Singularity-free tracking of unmanned underwater vehicles in 6 DOF. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, volume 2, pages 1128–1133. IEEE, 1994.

[87] T. I. Fossen. *Marine control systems: Guidance, navigation and control of ships, rigs and underwater vehicles.* Marine Cybernetics Trondheim, 2002.

[88] T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control.* John Wiley and Sons Ltd., 2011.

[89] T. I. Fossen. How to incorporate wind, waves and ocean currents in the marine craft equations of motion. In *9th IFAC Conference on Manoeuvring and Control of Marine Craft*, Arenzano, Italy, 2012.

[90] T. I. Fossen and J. G. Balchen. The NEROV autonomous underwater vehicle. *Proceedings of the OCEANS conference*, 90:1414–1420, 1991.

[91] T. I. Fossen and A. M. Lekkas. Direct and indirect adaptive integral line-of-sight path-following controllers for marine craft exposed to ocean currents. *International Journal of Adaptive Control and Signal Processing*, (submitted), 2014.

[92] T. I. Fossen, A. Loría, and A. Teel. A theorem for UGAS and ULES of (passive) nonautonomous systems: Robust control of mechanical systems and ships. *International Journal of Robust and Nonlinear Control*, 11(2):95–108, 2001.

[93] T. I. Fossen, M. Breivik, and R. Skjetne. Line-of-sight path following of underactuated marine craft. *Proceedings of the 6th IFAC MCMC, Girona, Spain*, pages 244–249, 2003.

[94] T. Fraichard and A. Scheuer. From Reeds and Shepp's to continuous-curvature paths. *IEEE Transactions on Robotics*, 20(6):1025–1035, 2004.

[95] E. Frazzoli, M. A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1): 116–129, 2002.

[96] E. Fredriksen and K. Y. Pettersen. Global κ-exponential way-point maneuvering of ships: Theory and experiments. *Automatica*, 42(4):677–687, 2006.

[97] F. N. Fritsch and R. E. Carlson. Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, 17(2):238–246, 1980.

[98] P. J. From and J. T. Gravdahl. Representing attitudes as sets of frames. In *American Control Conference*, pages 2465–2472. IEEE, 2007.

[99] D. Fryxell, P. Oliveira, A. Pascoal, and C. Silvestre. Integrated design of navigation, guidance and control systems for unmanned underwater vehicles. In *Proceedings OCEANS '94*, volume 3, pages 111–105. IEEE, 1994.

[100] J. M. Fulford, K. G. Thibodeaux, and W. R. Kaehrle. Comparison of current meters used for stream gaging. In ASCE, editor, *Proceedings of the ASCE Symposium*, pages 376–385, Buffalo, New York, 1994.

[101] L. Gajny, R. Bearee, E. Nyiri, and O. Gibaru. Path planning with PH G^2 splines in R^2. In *46th IEEE Conference on Systems and Computer Science*, pages 1–6, 2012.

[102] M. Garber and M. Lin. Constraint-based motion planning using Voronoi diagrams. *Algorithmic Foundations of Robotics V*, pages 541–558, 2004.

[103] S. Garrido, L. Moreno, and D. Blanco. Voronoi diagram and fast marching applied to path planning. In *International Conference on Robotics and Automation*, pages 3049–3054. IEEE, 2006.

[104] S. Garrido, L. Moreno, D. Blanco, and M. L. Munoz. Sensor-based global planning for mobile robot navigation. *Robotica*, 25(2):189–199, 2007.

[105] S. Garrido, L. Moreno, D. Blanco, and P. Jurewicz. Path planning for mobile robot navigation using Voronoi diagram and fast marching. *International Journal of Robotics and Automation*, 2(1):42–64, 2011.

[106] D. J. Gates. Nonlinear path following method. *Journal of guidance, control, and dynamics*, 33(2):321–332, 2010.

[107] R. Geraerts. Planning short paths with clearance using explicit corridors. In *IEEE International Conference on Robotics and Automation*, pages 1997–2004, 2010.

[108] R. Geraerts and M. H. Overmars. Clearance based path optimization for motion planning. In *IEEE International Conference on Robotics and Automation, 2004*, pages 2386–2392, 2004.

[109] C. M. Gold. The use of the dynamic Voronoi data structure in autonomous marine navigation. *Advanced Robotics: Beyond 2000*, pages 217–220, 1998.

[110] R. Goldman. Understanding quaternions. *Graphical models*, 73(2):21–49, 2011.

[111] I. R. Goralski and C. M. Gold. Maintaining the spatial relationships of marine vessels using the kinetic Voronoi diagram. In *4th International Symposium on Voronoi Diagrams in Science and Engineering*, pages 84–90. IEEE, 2007.

[112] R. Goralski, C. Gold, and M. Dakowicz. Application of the kinetic voronoi diagram to the real-time navigation of marine vessels. In *6th International Conference on Computer Information Systems and Industrial Management Applications*, pages 129–134. IEEE, 2007.

[113] H. F. Grip, L. Imsland, T. A. Johansen, J. C. Kalkkuhl, and A. Suissa. Vehicle sideslip estimation. *IEEE Control Systems Magazine*, 29(5):36–52, 2009.

[114] L. J. Guibas, J. S. Mitchell, and T. Roos. Voronoi diagrams of moving points in the plane. In *Graph-Theoretic Concepts in Computer Science*, pages 113–125. Springer, 1992.

[115] A. Hac and M. D. Simpson. Estimation of vehicle side slip angle and yaw rate. In *Proceedings of the SAE World Congress*, pages 1032–1038. SAE International, 2000.

[116] G. Harary and A. Tal. 3D Euler spirals for 3D curve completion. *Computational Geometry*, 45(3):115–126, 2012.

[117] J. Haugen. Guidance algorithms for planar path-based motion control scenarios. Master's thesis, Norwegian University of Science and Technology, 2010.

[118] A. J. Healey and D. Lienard. Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE Journal of Oceanic Engineering*, 18(3):327–339, 1993.

[119] S. J. Hills and D. R. Yoerger. A nonlinear sliding mode autopilot for unmanned undersea vehicles. In *Proceedings OCEANS'94.*, volume 3, pages III–93. IEEE, 1994.

[120] A. Hladio, C. Nielsen, and D. Wang. Path following for a class of mechanical systems. *IEEE Transactions on Control Systems Technology*, 21(6):2380–2390, 2013.

[121] Y.-J. Ho and J.-S. Liu. Collision-free curvature-bounded smooth path planning using composite Bezier curve based on Voronoi diagram. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 463–468, 2009.

[122] S. Hota and D. Ghose. Optimal trajectory generation for convergence to a rectilinear path. *Journal of Intelligent & Robotic Systems*, pages 1–20, 2013.

[123] I.-A. Ihle, J. Jouffroy, and T. I. Fossen. Formation control of marine surface craft: A Lagrangian approach. *IEEE Journal of Oceanic Engineering*, 31(4): 922–934, 2006.

[124] I.-A. F. Ihle, M. Arcak, and T. I. Fossen. Passivity-based designs for synchronized path-following. *Automatica*, 43(9):1508–1518, 2007.

[125] G. Indiveri, A. A. Zizzari, and V. G. Mazzotta. Linear path following guidance control for underactuated ocean vehicles. In *Proceedings of the IFAC Conference on Control Applications in Marine Systems*, Bol, Croatia, 2007.

[126] G. Indiveri, S. Cretí, and A. A. Zizzari. A proof of concept for the guidance of 3D underactuated vehicles subject to constant unknown disturbances. In *9th IFAC Conference on Manoeuvring and Control of Marine Craft*, Arenzano, Italy, 2012.

[127] B. Jalving. The NDRE-AUV flight control system. *IEEE Journal of Oceanic Engineering*, 19(4):497–501, 1994.

[128] K. G. Jolly, R. Sreerama Kumar, and R. Vijayakumar. A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits. *Robotics and Autonomous Systems*, 57(1):23–33, 2009.

[129] U. Jørgensen and R. Skjetne. Generating safe and equally long trajectories for multiple unmanned agents. In *20th Mediterranean Conference on Control & Automation (MED)*, pages 1566–1571. IEEE, 2012.

[130] I. Kaminer, A. Pascoal, E. Hallberg, and C. Silvestre. Trajectory tracking for autonomous vehicles: An integrated approach to guidance and control. *Journal of Guidance, Control, and Dynamics*, 21(1):29–38, 1998.

[131] E. D. Kaplan and C. J. Hegarty. *Understanding GPS: Principles and applications.* Artech House Publishers, 2006.

[132] L. E. Kavraki and S. M. LaValle. Motion planning. In *Springer Handbook of Robotics*, chapter 5, pages 109–131. Springer Berlin Heidelberg, 2008.

[133] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[134] H. Khalil. *Nonlinear systems.* Upper Saddle River, NJ: Prentice Hall, 2002. ISBN 0130673897.

[135] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.

[136] J. Kim, F. Zhang, and M. Egerstedt. An exploration strategy by constructing voronoi diagrams with provable completeness. In *Proceedings of the 48th Conference on Decision and Control, and the 28th Chinese Control Conference*, pages 7024–7029. IEEE, 2009.

[137] J. Kim, F. Zhang, and M. Egerstedt. A provably complete exploration strategy by constructing Voronoi diagrams. *Autonomous Robots*, 29(3-4):367–380, 2010.

[138] D. Kingston, R. Beard, A. Beard, T. McLain, M. Larsen, and W. Ren. Autonomous vehicle technologies for small fixed wing uavs. In *AIAA Journal of Aerospace Computing, Information, and Communication*. Citeseer, 2003.

[139] D. E. Knuth. *Digital typography*, volume 78 of CSLI lecture notes. CSLI Publications, Stanford, California, 1999.

[140] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1398–1404, 1991.

[141] M. Kothari, I. Postlethwaite, and D.-W. Gu. UAV path following in windy urban environments. *Journal of Intelligent & Robotic Systems*, pages 1–16, 2013.

[142] M. Krstic, I. Kanellakopoulos, P. V. Kokotovic, et al. *Nonlinear and adaptive control design*, volume 8. John Wiley & Sons New York, 1995.

[143] B. Kuipers and Y.-T. Byun. A robust, qualitative method for robot spatial learning. In *AAAI*, volume 88, pages 774–779, 1988.

[144] B. Kuipers and Y.-T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and autonomous systems*, 8(1):47–63, 1991.

[145] J. B. Kuipers. *Quaternions and rotation sequences*. Princeton university press, 1999.

[146] P. Lamberti and C. Manni. Shape-preserving C^2 functional interpolation via parametric cubics. *Numerical Algorithms*, 28(1-4):229–254, 2001.

[147] L. Lapierre and B. Jouvencel. Robust nonlinear path-following control of an AUV. *IEEE Journal of Oceanic Engineering*, 33(2):89–102, 2008.

[148] L. Lapierre and D. Soetanto. Nonlinear path-following control of an AUV. *Ocean engineering*, 34(11):1734–1744, 2007.

[149] L. Lapierre, D. Soetanto, and A. Pascoal. Nonlinear path following with applications to the control of autonomous underwater vehicles. In *42nd IEEE Conference on Decision and Control*, volume 2, pages 1256–1261, 2003.

[150] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.

[151] S. M. LaValle. *Planning algorithms*. Cambridge university press, 2006.

[152] D. A. Lawrence, E. W. Frew, and W. J. Pisano. Lyapunov vector fields for autonomous unmanned aircraft flight control. *Journal of Guidance, Control, and Dynamics*, 31(5):1220–1229, 2008.

[153] R. K. Lea, R. Allen, and S. L. Merry. A comparative study of control techniques for an underwater flight vehicle. *International Journal of Systems Science*, 30(9):947–964, 1999.

[154] E. Lefeber. *Tracking control of nonlinear mechanical systems*. PhD thesis, University of Twente, 2000.

[155] A. M. Lekkas and T. I. Fossen. A time-varying lookahead distance guidance law for path following. In *9th IFAC Conference on Manoeuvring and Control of Marine Craft*, Arenzano, Italy, 2012.

[156] A. M. Lekkas and T. I. Fossen. *Line-of-Sight Guidance for Path Following of Marine Vehicles*, chapter 5, In Advanced in Marine Robotics, pages 63–92. LAP LAMBERT Academic Publishing (O. Gal, Ed.), 2013.

[157] A. M. Lekkas and T. I. Fossen. A quaternion-based LOS guidance scheme for path following of AUVs. In *9th IFAC Conference on Control Applications in Marine Systems*, Osaka, Japan, 2013.

[158] A. M. Lekkas and T. I. Fossen. Integral LOS path following for curved paths based on a monotone cubic Hermite spline parametrization. *IEEE Transactions on Control Systems Technology*, 2014.

[159] A. M. Lekkas and T. I. Fossen. Minimization of cross-track and along-track errors for path tracking of marine underactuated vehicles. In *European Control Conference (submitted)*, Strasbourg, France, 2014.

[160] A. M. Lekkas, A. R. Dahl, M. Breivik, and T. I. Fossen. Continuous-curvature path generation using Fermat's spiral. *Modeling, Identification and Control*, 34(4):183–198, 2013.

[161] R. Levien and C. H. Séquin. Interpolating splines: which is the fairest of them all. *Computer Aided Design and Applications*, 4:91–102, 2009.

[162] J.-H. Li and P.-M. Lee. Design of an adaptive nonlinear controller for depth control of an autonomous underwater vehicle. *Ocean engineering*, 32(17): 2165–2181, 2005.

[163] E. Liceaga-Castro and G. M. van der Molen. Submarine H_∞ depth control under wave disturbances. *IEEE Transactions on Control Systems Technology*, 3(3):338–346, 1995.

[164] M. Lindhé, P. Ogren, and K. H. Johansson. Flocking with obstacle avoidance: A new distributed coordination algorithm based on voronoi partitions. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1785–1790. IEEE, 2005.

[165] M. M. Lipschutz. *Differential Geometry*. McGraw-Hill, 1969.

[166] A. Loría, T. I. Fossen, and E. Panteley. A separation principle for dynamic positioning of ships: Theoretical and experimental results. *IEEE Transactions on Control Systems Technology*, 8(2):332–343, 2000.

[167] C. Manni. On shape preserving C^2 hermite interpolation. *BIT Numerical Mathematics*, 41(1):127–148, 2001.

[168] Y. Matsumoto, K. Imai, and H. Suzuki. Curved voronoi diagrams consisting of influence areas with differentiable boundaries. In *4th International Symposium on Voronoi Diagrams in Science and Engineering*, pages 270–275. IEEE, 2007.

[169] M. Mattei and L. Blasi. Smooth flight trajectory planning in the presence of no-fly zones and obstacles. *Journal of guidance, control, and dynamics*, 33 (2):454–462, 2010.

[170] R. McEwen and K. Streitlien. Modeling and control of a variable-length AUV. *Proc 12th UUST*, 2001.

[171] C. B. Moler. *Numerical computing with MATLAB*. Society for Industrial Mathematics, 2004.

[172] H. P. Moreton. *Minimum curvature variation curves, networks, and surfaces for fair free-form shape design*. PhD thesis, University of California, 1983.

[173] K. Nagatani, Y. Iwai, and Y. Tanaka. Sensor based navigation for car-like mobile robots using generalized voronoi graph. In *International Conference on Intelligent Robots and Systems*, volume 2, pages 1017–1022. IEEE, 2001.

[174] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard. Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*, 23 (3):519–529, 2007.

[175] W. Nelson. Continuous-curvature paths for autonomous vehicles. In *Proceedings of the International Conference on Robotics and Automation*, pages 1260–1264. IEEE, 1989.

[176] N. Norrbin. On the design and analysis of the zig-zag test on base of quasi-linear frequency response. Technical Report B 104-3, The Swedish State Shipbuilding Experimental Tank (SSPA), Gothenburg, Sweden, 1963.

[177] C. Ó'Dúnlaing and C. K. Yap. A "retraction" method for planning the motion of a disc. *Journal of Algorithms*, 6(1):104–111, 1985.

[178] P. Ogren and N. E. Leonard. Obstacle avoidance in formation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003.

[179] S. R. Oh and J. Sun. Path following of underactuated marine surface vessels using line-of-sight based model predictive control. *Ocean Engineering*, 37 (2-3):289–295, 2010.

[180] J. Osborne and R. Rysdyk. Waypoint guidance for small uavs in wind. *AIAA Infotech@ Aerospace*, 193(1-4):1–12, 2005.

[181] E. Panteley and A. Loría. On global uniform asymptotic stability of nonlinear time-varying systems in cascade. *Systems & Control Letters*, 33(2):131–138, 1998.

[182] E. Panteley, E. Lefeber, A. Loría, and H. Nijmeijer. Exponential tracking control of a mobile car using a cascaded approach. In *Proceedings of the IFAC Workshop on Motion Control*, pages 221–226, 1998.

[183] E. Panteley, A. Loría, and A. Teel. Relaxed persistency of excitation for uniform asymptotic stability. *IEEE Transactions on Automatic Control*, 46 (12):1874–1886, 2001.

[184] F. A. Papoulias. Bifurcation analysis of line of sight vehicle guidance using sliding modes. *International Journal of Bifurcation and Chaos*, 1(4):849–865, 1991.

[185] S. Park, J. Deyst, and J. P. How. A new nonlinear guidance logic for trajectory tracking. In *In Proceedings of the AIAA Guidance, Navigation and Control Conference*. Citeseer, 2004.

[186] S. Park, J. Deyst, and J. P. How. Performance and lyapunov stability of a nonlinear path following guidance method. *Journal of Guidance, Control, and Dynamics*, 30(6):1718–1728, 2007.

[187] A. Pavlov, H. Nordahl, and M. Breivik. MPC-based optimal path following for underactuated vessels. In *8th IFAC International Conference on Manoeuvring and Control of Marine Craft*, pages 340–345, Guaruja, Brazil, 2009.

[188] J. W. Pearson. Computation of hypergeometric functions. Master's thesis, University of Oxford, 2009.

[189] M. D. Pedersen and T. I. Fossen. Marine vessel path planning and guidance using potential flow. In *9th IFAC Conference on Manoeuvring and Control of Marine Craft*, 2012.

[190] T. Perez, Ø. N. Smogeli, T. I. Fossen, and A. J. Sørensen. An overview of the marine systems simulator (MSS): A Simulink toolbox for marine control systems. *Modeling, Identification and Control*, 27(4):259–275, 2006.

[191] J. Peters. Geometric continuity. *Handbook of Computer Aided Geometric Design*, pages 193–229, 2002.

[192] J. Peters. Changing variables. *IEEE Computer Graphics and Applications*, 32(3):88–93, May-June 2012.

[193] K. Y. Pettersen and E. Lefeber. Way-point tracking control of ships. In *Proceedings of the 40th IEEE Conference on Decision and Control*, volume 1, pages 940–945. IEEE, 2001.

[194] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.

[195] J. H. Reif. Complexity of the mover's problem and generalizations extended abstract. In *Proceedings of the 20th Annual IEEE Conference on Foundations of Computer Science*, pages 421–427, 1979.

[196] F. Repoulias and E. Papadopoulos. Planar trajectory planning and tracking control design for underactuated AUVs. *Ocean Engineering*, 34(11):1650–1667, 2007.

[197] T. Roos. Voronoi diagrams over dynamic scenes. *Discrete Applied Mathematics*, 43(3):243–259, 1993.

[198] T. Roos and H. Noltemeier. Dynamic Voronoi diagrams in motion planning. In *Computational Geometry-Methods, Algorithms and Applications*, pages 227–236. Springer, 1991.

[199] C.-K. Ryoo, H. Cho, and M.-J. Tahk. Optimal guidance laws with terminal impact angle constraint. *Journal of Guidance, Control, and Dynamics*, 28 (4):724–732, 2005.

[200] C.-K. Ryoo, H.-S. Shin, and M.-J. Tahk. Energy optimal waypoint guidance synthesis for antiship missiles. *IEEE Transactions on Aerospace and Electronic Systems*, 46(1):80–95, 2010.

[201] R. Rysdyk. Course and heading changes in significant wind. *Journal of guidance, control, and dynamics*, 30(4):1168–1171, 2007.

[202] M. Shanmugavel, A. Tsourdos, R. Zbikowski, and B. White. Path planning of multiple UAVs with clothoid curves in two dimensions. In *Automatic Control in Aerospace*, volume 17, pages 461–466, 2007.

[203] M. Shanmugavel, A. Tsourdos, R. Zbikowski, and B. A. White. 3D path planning for multiple UAVs using Pythagorean hodograph curves. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Hilton Head, South Carolina*, pages 20–23, 2007.

[204] M. Shanmugavel, A. Tsourdos, B. White, and R. Żbikowski. Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs. *Control Engineering Practice*, 18(9):1084–1092, 2010.

[205] N. A. Shneydor. *Missile guidance and pursuit: kinematics, dynamics and control.* Elsevier, 1998.

[206] C. Silvestre, A. Pascoal, and A. J. Healey. AUV control under wave disturbances. In *International Symposium on Unmanned Untethered Submersible Technology*, pages 228–239. University of New Hampshire-Marine Systems, 1997.

[207] R. Skejic, M. Breivik, and T. E. Berg. Investigating ship maneuvers around a floating structure under the influence of a uniform current in deep and calm water. In *Second International Conference on Ship Manoeuvring in Shallow and Confined Water: Ship to Ship Interaction*, pages 339 – 350, 2011.

[208] R. Skjetne. *The maneuvering problem.* PhD thesis, Norwegian University of Science and Technology, 2005.

[209] R. Skjetne, S. Moi, and T. I. Fossen. Nonlinear formation control of marine craft. In *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002.

[210] R. Skjetne, T. I. Fossen, and P. V. Kokotović. Robust output maneuvering for a class of nonlinear systems. *Automatica*, 40(3):373–383, 2004.

[211] R. Skjetne, U. Jørgensen, and A. Teel. Line-of-sight path-following along regularly parametrized curves solved as a generic maneuvering problem. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 2467–2474. IEEE, 2011.

[212] S. Skogestad. Feedback: Still the simplest and best solution. *Modeling, Identification and Control*, 30(3):149–155, 2009.

[213] O. J. Sørdalen and O. Egeland. Exponential stabilization of nonholonomic chained systems. *IEEE Transactions on Automatic Control*, 40(1):35–49, 1995.

[214] L. V. Steenson, A. B. Phillips, S. R. Turnock, M. E. Furlong, and E. Rogers. Effect of measurement noise on the performance of a depth and pitch controller using the model predictive control method. In *Autonomous Underwater Vehicles (AUV), 2012 IEEE/OES*, pages 1–8. IEEE, 2012.

[215] J. Stephant, A. Charara, and D. Meizel. Virtual sensor: Application to vehicle sideslip angle and transversal forces. *IEEE Transactions on Industrial Electronics*, 51(2):278–289, 2004.

[216] A. Sud, E. Andersen, S. Curtis, M. Lin, and D. Manocha. Real-time path planning for virtual agents in dynamic environments. In *ACM SIGGRAPH*, 2008.

[217] A. Sud, E. Andersen, S. Curtis, M. C. Lin, and D. Manocha. Real-time path planning in dynamic virtual environments using multiagent navigation graphs. *IEEE Transactions on Visualization and Computer Graphics*, 14(3): 526–538, 2008.

[218] Y. Sun and C. Cheah. Adaptive setpoint control for autonomous underwater vehicles. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 2, pages 1262–1267, 2003.

[219] L. Techy and C. A. Woolsey. Minimum-time path planning for unmanned aerial vehicles in steady uniform winds. *Journal of guidance, control, and dynamics*, 32(6):1736–1746, 2009.

[220] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[221] S. Thrun and A. Bücken. Integrating grid-based and topological maps for mobile robot navigation. In *Proceedings of the National Conference on Artificial Intelligence*, pages 944–951, 1996.

[222] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press Cambridge, 2005.

[223] G. J. Toussaint, T. Basar, and F. Bullo. Motion planning for nonlinear underactuated vehicles using H_∞ techniques. In *Proceedings of the American Control Conference*, volume 5, pages 4097–4102. IEEE, 2001.

[224] A. Tsourdos, B. White, and M. Shanmugavel. *Cooperative path planning of unmanned aerial vehicles*. Wiley Online Library, 2011.

[225] V. Utkin. Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control*, 22(2):212–222, 1977.

[226] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les parallélloèdres primitifs. *Journal für die reine und angewandte Mathematik*, 134:198–287, 1908.

[227] D. Walton and D. Meek. Curvature extrema of planar parametric polynomial cubic curves. *Journal of computational and applied mathematics*, 134(1):69–83, 2001.

[228] E. W. Weisstein. Archimedean spiral. From MathWorld–A Wolfram Web Resource (accessed 03/12/2013)., 2013. URL http://mathworld.wolfram.com/ArchimedeanSpiral.html.

[229] E. W. Weisstein. Fermat's Spiral. From MathWorld–A Wolfram Web Resource (accessed 19/12/2013)., 2013. URL http://mathworld.wolfram.com/FermatsSpiral.html.

[230] E. W. Weisstein. Voronoi diagram. From MathWorld–A Wolfram Web Resource (accessed 20/01/2014), 2014. URL http://mathworld.wolfram.com/VoronoiDiagram.html.

[231] S. B. Williams, P. Newman, G. Dissanayake, J. Rosenblatt, and H. Durrant-Whyte. A decoupled, distributed AUV control architecture. In *Proceedings of the International Symposium on Robotics*, 2000.

[232] K. Yang and S. Sukkarieh. An analytical continuous-curvature path-smoothing algorithm. *IEEE Transactions on Robotics*, 26(3):561–568, 2010.

[233] R. Yanushevsky. *Guidance of Unmanned Aerial Vehicles*. CRC Press, 2011.

[234] R. Yanushevsky and W. Boord. Lyapunov approach to guidance laws design. *Nonlinear Analysis: Theory, Methods & Applications*, 63(5):743–749, 2005.

[235] R. Yanushevsky and W. Boord. New approach to guidance law design. *Journal of Guidance, Control, and Dynamics*, 28(1):162–166, 2005.

[236] R. T. Yanushevsky. Concerning Lyapunov-based guidance. *Journal of guidance, control, and dynamics*, 29(2):509–511, 2006.

[237] J. Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.

[238] J. Y. Yen. Finding the lengths of all shortest paths in n-node nonnegative-distance complete networks using 1 2 n 3 additions and n 3 comparisons. *Journal of the ACM (JACM)*, 19(3):423–424, 1972.

[239] J. Yuh, J. Nie, and C. G. Lee. Experimental study on adaptive control of underwater robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 393–398. IEEE, 1999.

[240] F. Zhou, B. Song, and G. Tian. Bézier curve based smooth path planning for mobile robot. *Journal of Information & Computational Science*, 8(12): 2441–24450, 2011.