# Online Path Planning for Surface Vehicles Exposed to Unknown Ocean Currents Using Pseudospectral Optimal Control

**Anastasios M. Lekkas** * **Ann L. Roald** ** **Morten Breivik** **

* *Department of Marine Technology (e-mail: anastasios.lekkas@ntnu.no)*
** *Department of Engineering Cybernetics (e-mail: annlouise1991@gmail.com, morten.breivik@ieee.org)*
*Norwegian University of Science and Technology (NTNU), NO-7491, Trondheim, Norway*

**Abstract:** This paper investigates the feasibility of using pseudospectral (PS) optimal control in real-time path planning for marine surface vehicles in environments where both obstacles and unknown disturbances are present. In particular, the simplified kinematic equations of an underactuated marine surface vehicle exposed to unknown ocean currents are considered, and the software package DIDO is used to compute the optimal path via PS optimization, initially assuming the ocean current is zero. In that case, the resulting path is minimum-length (similar to Dubins path) but not minimum-time. The main contribution concerns the addition of a nonlinear observer, which estimates online the effects of the ocean current on the vehicle, and that of a guidance system which generates appropriate reference trajectories in order to minimize the position error and track the optimal trajectory successfully. It is shown that through occasional replanning, according to the information about the ocean current parameters coming from the observer, the updated path converges to the minimum-time path. Two different implementations of the approach are presented and illustrated through numerical simulations.

*Keywords:* path planning, pseudospectral optimal control, collision avoidance, marine surface vehicles, DIDO, integral LOS guidance, ocean current observer

## 1. INTRODUCTION

Autonomous vehicles depend on their ability to exploit the available knowledge of their environment and plan their actions accordingly. In this context, for an underactuated marine surface vehicle an important challenge is to plan paths such that the vehicle accomplishes its mission in a safe and efficient way. Among other things this involves: a) considering the vehicle's kinematic and dynamic constraints, b) keeping a minimum distance (clearance) from static and dynamic obstacles, and c) compensating for unknown factors that are always present in real-life applications and can be the cause for large position errors. A lot of research has been dedicated to these three aspects and several notable solutions are available in the existing literature.

Often the first step is to determine a set of successive waypoints on a given map, which are subsequently connected resulting in a final path comprised of piecewise linear and/or curved segments. Designing paths that satisfy kinematic and dynamic constraints is related to properties such as geometric continuity, parametric continuity, minimum turning radius, and maximum parametric speed. Second-order geometric continuity (denoted as $G^2$) is a prerequisite for curvature continuity, which ensures continuous lateral acceleration, see also Tsourdos et al. (2011). Second-order parametric continuity (denoted as $C^2$) ensures smooth propagation of the path parameter along the final path, a necessary property for operations that involve temporal constraints and path/trajectory tracking. Interestingly, it was shown by Barsky and DeRose (1989) that two curves meet with $n$th-degree geometric continuity $G^n$ if and only if their arc length parametrizations meet with $C^n$ continuity. The minimum turning radius, or maximum curvature, sets a limit on how small the turning circle of the vehicle can be, whereas the maximum parametric speed is related to the maximum speed with which the vehicle can perform path tracking. In addition, quantities such as path length and computational time required to generate the path are very important in real-time applications. An overview of additional path properties and evaluation criteria can be found in (Lekkas, 2014).

Most of the time, a path cannot satisfy all desired properties and the designer has to compromise and select the one most fitting for a given task. For instance, Dubins (1957) showed that the shortest path for a car-like vehicle with prescribed minimum turning radius consists of straight lines and circular arcs, however this path includes curvature discontinuities. Clothoids avoid the curvature discontinuities but the computational cost is much higher (Tsourdos et al., 2011). Fermat's spiral segments are curvature-continuous and very fast to compute (Lekkas et al., 2013) but paths consisting of straight lines and curved segments are not always the desired option, especially when environmental forces act on the vehicle and a minimum-time approach is preferred. Natural cubic splines are $C^2$ but can introduce wiggling and zigzagging, hence making the path longer and also more difficult to guarantee collision avoidance.

There are several approaches to collision avoidance and the works by LaValle (2006); Thrun et al. (2005); Choset et al. (2005) are standard references in the field. Two main approaches can be observed:

1) The aforementioned *waypoint-based approach* is to place the waypoints in locations such that, when connected by piecewise segments, the final path is obstacle-free. Many methods have been used for choosing the locations of the waypoints, such as probabilistic methods (rapidly-exploring random trees, probabilistic roadmaps, etc.), deterministic roadmap methods (such as the Voronoi diagram) and others.
2) *Optimization methods*, where waypoints are not involved at all. The equation describing the motion control of the system is employed and the obstacles are taken into account by using suitable constraints that do not allow the system's states (for instance, the vehicle's position) to reach the values that would allow for a collision to occur.

In this paper, which is an outcome of the work described in Roald (2015), we implement and extend a methodology that belongs to the second category above. More specifically, we investigate the use of *pseudospectral optimal control*, developed by Ross and Fahroo (2003), for real-time path planning and tracking of underactuated surface vehicles in environments where both obstacles and unknown ocean currents are present. Pseudospectral (PS) optimal control is an attractive option because it has been reported to reduce computational time considerably, an important practical aspect where optimal control methods are usually least competitive. In Bollino et al. (2007), the authors gave an overview of the path-planning problem within an optimal control framework, with focus on pseudospectral control. We use the DIDO software package to implement the optimal path-planning computation. DIDO has been used in a recent work by Hurni and Kiriakidis (2015) for similar purposes, that is, minimum-time path planning for an AUV in uncertain current. In that work, the authors' approach was to exploit knowledge of extreme variations of current in order to achieve worst-case planning.

In this paper, we consider a different implementation according to which the information from the vehicle's onboard sensors (measuring position and velocity) can be used in order to estimate the ocean current online and to replan the desired minimum-time path. The main contribution involves augmenting DIDO in order to incorporate online estimation and rejection of unknown ocean currents. This is achieved via the indirect adaptive integral Line-of-Sight (LOS) guidance law developed by Fossen and Lekkas (2015).

## 2. THEORETICAL BACKGROUND

This section gives an overview of the theoretical background behind pseudospectral methods, based on which DIDO operates. The presentation follows mainly those of Ross and Fahroo (2003) and Gong et al. (2008).

### *2.1 Problem Formulation*

We consider a system with the following continuous dynamics:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \qquad t_0 < t < t_f$$
$$\mathbf{x}(t_0) = \mathbf{x}_0,$$

where $\mathbf{x}$ is the state of the system, $t_0$ and $t_f$ the initial and final time respectively, $\mathbf{u} : [t_0, t_f] \to A \subseteq \mathbf{R}^m$ is the control function, and $\mathbf{f} : \mathbf{R} \times (t_0, t_f) \times A \to \mathbf{R}^d$ is the controlled dynamics (Khalil, 2002; Falcone and Ferretti, 2013). When the admissible controls are defined as $\mathcal{A} := \{\mathbf{u} : (t_0, t_f) \to A, \text{measurable}\}$, then for any control $\mathbf{u} \in \mathcal{A}$ the solution is well defined (in a weak sense). In order to find the optimal solution, the following cost function is defined:

$$\mathbf{J}[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f] := \mathbf{E}(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathbf{F}(\mathbf{x}(t), \mathbf{u}(t), t)\; dt, \tag{1}$$

where the term $\mathbf{E}(\mathbf{x}(t_f), t_f)$ represents the terminal cost and $\mathbf{F}(\mathbf{x}(t), \mathbf{u}(t), t)$ the running cost. The terminal cost depends only on the terminal time and pose, while the running cost varies with state, control action and time, see Falcone and Ferretti (2013).

### *2.2 Legendre Pseudospectral Approximations*

The Legendre PS method uses the basic discretization principles of PS methods. In simple terms, the Legendre PS method is just another way of discretizing the problem to cast a smooth nonlinear optimization problem. It accomplishes this using Legendre polynomials and Gauss-Lobatto quadratures. Assume $B$ is the problem to be solved. The following briefly summarize how a PS method attempts to find a solution (Gong et al., 2008):

- Application of Pontryagin's minimum principle to the initial problem $B$ gives a set of necessary conditions. These conditions can be expressed in terms of a boundary-value problem, which is a problem of solving a generalized equation. This process is referred to as dualization and it results in the problem $B^{\lambda}$.
- The goal of PS methods is to solve problem $B$ by discretizing it to problem $B^N$. An optimal solution of the discretized problem $B^N$ must automatically satisfy the discretized necessary conditions $B^{\lambda N}$. This transformation requirement is also called the *covector mapping principle* and results in an approach simpler than tackling $B^{\lambda}$, which would involve developing and solving for the necessary conditions.
- Based on the aforementioned points, it can be said that the Legendre pseudospectral method belongs to the category of *direct optimization methods*, for which discretization takes place first.

The initial problem B is described as follows:

Minimize (1) subject to

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \tag{2}$$
$$\mathbf{g}(\mathbf{x}, \mathbf{u}, t) \leq 0, \tag{3}$$
$$\mathbf{h}(\mathbf{x}, \mathbf{u}, t) = 0, \tag{4}$$

where it is assumed that all the nonlinear functions $(\mathbf{f}, \mathbf{g}, \mathbf{h})$ are continuously differentiable w.r.t. their arguments and their gradients are Lipschitz-continuous over the domain. By using the approximation and techniques from (Ross and Fahroo, 2003) described below, the initial problem B is discretized into problem $B^N$. The first step is to select $N+1$ cardinal functions $\phi_l$ $(l = 0, 1, \dots, N)$ over the interval $[t_0, t_f]$ such that they satisfy the Kronecker delta function:

$$\phi_l(t_k) = \delta_{lk} = \begin{cases} 1 \text{ if l = k} \\ 0 \text{ if } l \neq k \end{cases}, \quad k = 0, 1, \dots, N, \tag{5}$$

where the grid points are called nodes.

After a choice of Gauss-Lobatto points has been made, the state and control functions are approximated by $N$th degree polynomials (Ross and Fahroo, 2003):

$$\boldsymbol{x}(t) \approx \boldsymbol{x}^N(t) := \sum_{l=0}^{N} \boldsymbol{x}_l \phi_l(t), \tag{6}$$

$$\boldsymbol{u}(t) \approx \boldsymbol{u}^N(t) := \sum_{l=0}^{N} \boldsymbol{u}_l \phi_l(t), \tag{7}$$

where $\phi_l$ are the Lagrange interpolating polynomials of order $N$ and (5) holds. In the Legendre PS method, the grid points are the shifted Legendre-Gauss-Lobatto (LGL) points. The shift is achieved by mapping the physical domain $t \in [t_0, t_f]$ to a computational domain $\tau \in [-1, 1]$ using the affine transformation:

$$\tau(t) = \frac{2t - (\tau_f + \tau_0)}{(\tau_f - \tau_0)} \tag{8}$$

where $\tau$ denotes both the transformation and the transformed variable. The Lagrange interpolating polynomials in this case are defined as:

$$\phi_l(t) = \frac{1}{N(N+1)L_N(t_l)} \frac{(t^2-1)\dot{L}_N(t)}{t - t_l} \quad l = 0, 1, ..., N, \tag{9}$$

where $L_N$ is the Legendre polynomial of degree $N$, and $t_l$ are the zeros of $\dot{L}_N$. Moreover, it can be verified that (5) holds.

Therefore $\boldsymbol{x}_l = \boldsymbol{x}^N(\tau_l)$, $\boldsymbol{u}_l = \boldsymbol{u}^N(\tau_l)$ where $\tau_l = \tau(t_l)$ so that $\tau^N \equiv \tau_f$. By differentiating (6) and evaluating it at the node points $t_k$ the result is:

$$\dot{\boldsymbol{x}}^n(\tau_k) = \frac{d\boldsymbol{x}^N}{d\tau}\Big|_{\tau=\tau_k} = \frac{d\boldsymbol{x}^N}{dt}\frac{dt}{d\tau}\Big|_{t_k} \tag{10}$$

$$= \frac{2}{\tau_f - \tau_0} \sum_{l=0}^{N} D_{kl} \boldsymbol{x}_{kl} \equiv \frac{2}{\tau_f - \tau_0} \boldsymbol{d}_k \tag{11}$$

where $D_{kl} = \dot{\phi}_l(t_k)$ are entries of the $(N+1) \times (N+1)$ differentiation matrix called D. This differentiation matrix is represented as:

$$\boldsymbol{D} := [D_{kl}] = \begin{cases} \dfrac{L_N(t_k)}{L_N(t_l)} \dfrac{1}{t_k - t_l} & l \neq k \\ -\dfrac{N(N+1)}{4} & \text{l = k = 0} \\ \dfrac{N(N+1)}{4} & \text{l = k = N} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

This causes the approximation of the state dynamics to become:

$$\frac{\tau_f - \tau_0}{2} \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) - \sum_{l=0}^{N} D_{kl} \boldsymbol{x}_l = 0, \quad k = 0, 1..., N. \tag{13}$$

Then the cost function $\boldsymbol{J}$ is approximated using the Gauss-Lobatto integration rule:

$$\boldsymbol{J}[\mathbf{X}^N, \mathbf{U}^N, \tau_0, \tau_f] = E(\boldsymbol{x}_o, \boldsymbol{x}_N, \tau_0, \tau_f) + \frac{\tau_f - \tau_0}{2} \sum_{l=0}^{N} F(\boldsymbol{x}_k, \boldsymbol{u}_k) w_k, \tag{14}$$

where

$$\mathbf{X}^N = [\boldsymbol{x}_0; \boldsymbol{x}_1; ...; \boldsymbol{x}_N], \qquad \mathbf{U}^N = [\boldsymbol{u}_0; \boldsymbol{u}_1; ...; \boldsymbol{u}_N] \tag{15}$$

and $w_k$ are the LGL weights given by

$$w_k := \frac{2}{N(N+1)} \frac{1}{L_N(t_k)^2}, \qquad k = 0, 1, ..., N. \tag{16}$$

Combining the equations above gives the discretized version of problem $B$, i.e. problem $B^N$, which can be described as follows:

Find the $(N+1)(N_x + N_u) + 2$ vector

$$\mathbf{X}_{NP} = (\mathbf{X}^N; \mathbf{U}^N; \tau_0; \tau_f) \tag{17}$$

that minimizes (14) subject to

$$\frac{\tau_f - \tau_0}{2} \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) - \sum_{l=0}^{n} D_{kl} \boldsymbol{x}_l = 0, \tag{18}$$

$$\mathbf{g}(\mathbf{x}, \mathbf{u}, \tau) \leq 0, \tag{19}$$

$$\mathbf{h}(\mathbf{x}, \mathbf{u}, \tau) = 0. \tag{20}$$

Although the PS methods have many advantages like high accuracy and computational savings, they also have some drawbacks. One of the drawbacks is that constraints are only enforced at the LGL node points and at the specific event times. As a result, the constraints are not always satisfied between these node points. This may cause problems if there are too few nodes. One example is that the calculated path may cut the corner of an obstacle because the nodes on either side of it are in the clear space, thus resulting in infeasible paths.

### *2.3 DIDO: A MATLAB Application Package*

DIDO is a MATLAB application package used to solve optimal control problems based on pseudospectral optimal control theory. Although the pseudospectral method can be used to discretize the Hamilton-Jacobi-Bellman equation, DIDO solves the optimal control problem by using the Legendre pseudospectral discretization method to transform the problem into an equivalent discretized optimization problem, and then solves this using a well-developed Nonlinear Program (NLP) solver. The NLP solver used is based on Sequential Quadratic Programming (SQP) and called SNOPT, which is described in detail by Gill et al. (2005).

## 3. VEHICLE MODEL AND TRACKING ERROR

### *3.1 Kinematic Marine Surface Vehicle Model*

The vehicle kinematic equations for horizontal plane motion are originally expressed in terms of the relative *surge*

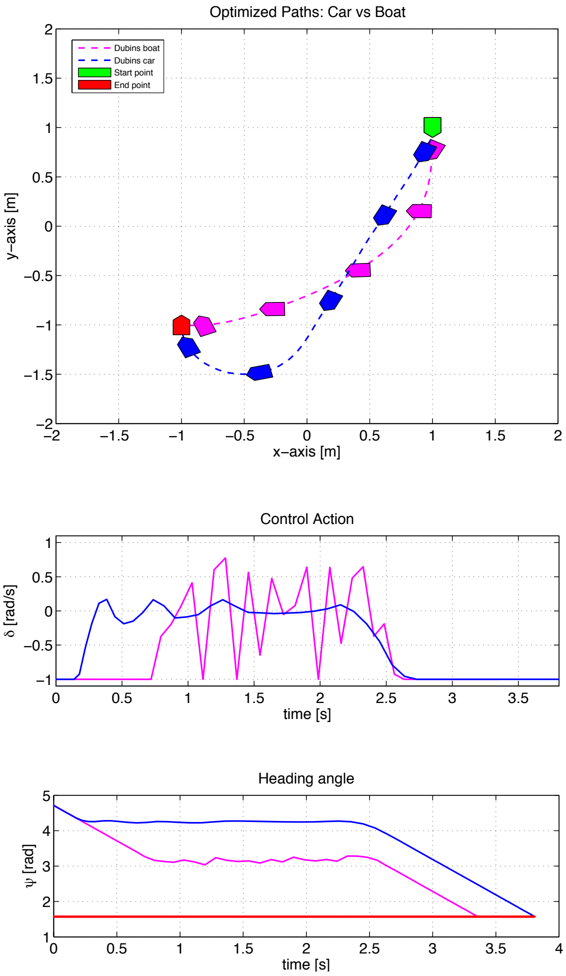


Fig. 1. Optimized path: Dubins car (blue) vs Dubins boat (magenta). A large sideslip angle can alter the result significantly. The red line depicts the final desired heading angle $\psi_d$.

and *sway* velocities $u_r = u - u_c$ and $v_r = v - v_c$ according to Fossen (2011):

$$\dot{x} = u_r \cos(\psi) - v_r \sin(\psi) + V_x \tag{21}$$

$$\dot{y} = u_r \sin(\psi) + v_r \cos(\psi) + V_y \tag{22}$$

$$\dot{\psi} = r \tag{23}$$

where $\psi$ and $r$ are the yaw angle and rate, respectively. The body-fixed ocean current velocities $(u_c, v_c)$ and North-East current velocities $(V_x, V_y)$ satisfy:

$$[u_c, v_c]^\top = \mathbf{R}^\top(\psi)[V_x, V_y]^\top, \tag{24}$$

where $\mathbf{R}^\top(\psi)$ denotes the rotation matrix from the inertial frame to the body-fixed reference frame. When computing the optimal path in DIDO, in order to reduce the computational cost as much as possible, we assume negligible sideslip angle *due to turning*:

$$\beta_r = \text{atan2}(v_r, u_r) = 0, \tag{25}$$

and consider a simplified model of a marine surface vehicle. For the sake of accuracy, it should be mentioned that the vehicle should have absolute total velocity $U = \sqrt{u^2 + v^2}$ higher than that of the ocean current, in order for it to be able to move against the current. This was discussed in detail in (Fossen and Lekkas, 2015) and can be described as follows:

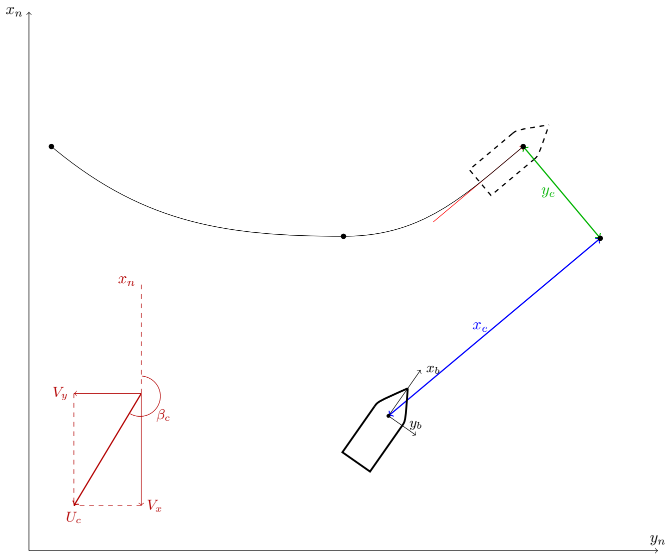


Fig. 2. Illustration of the path-tracking problem solved in this paper. The vehicle (solid line), which is under the influence of an unknown constant disturbance of magnitude $U_c$ and direction $\beta_c$, is tracking a virtual vehicle (dashed line) moving on a curved path.

$$U_r \geq U - U_c > 0. \tag{26}$$

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 $\psi$. Because of (25), however, we get $v_r = 0$. It is also worth noting that the absolute sway velocity $v$ is *not* zero in the presence of ocean currents because $v = v_c$. This agrees with intuition because for the underactuated vehicle to stay on the path while an ocean current component is pushing it away from it, a heading correction is necessary to counteract that component. Fig. 1 illustrates the differences in optimal path planning between a boat-like vehicle with sideslip angle and a car-like vehicle without sideslip angle.

### *3.2 Virtual Vehicle Kinematics and Tracking Error*

We consider a 2-D continuous parametrized curved path $(x_p(\theta), y_p(\theta))$ that connects the successive waypoints $(x_k, y_k)$ for $k = 1, 2, ..., \zeta$. In this case, the path-tangential angle is varying and can be computed as:

$$\gamma_p = \text{atan2}(y_p'(\theta), x_p'(\theta)), \tag{27}$$

where $(\cdot)'(\theta) = \partial(\cdot)/\partial\theta$. For the path-tracking scenario, it is reasonable to assume that a virtual vehicle is navigating with a speed $U_t > 0$ on the desired path, therefore its position $\mathbf{p}_t^n = (x_t, y_t)$ is computed by:

$$\dot{x}_t = U_t \cos(\gamma_p), \tag{28}$$

$$\dot{y}_t = U_t \sin(\gamma_p). \tag{29}$$

The objective of the surface vehicle in this case is to track the virtual particle, that is $\mathbf{p} - \mathbf{p_t} \to 0$. An illustration of the problem can be seen in Fig. 2. Consequently, the position error for a given vehicle position $(x, y)$ is given by:

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} = \mathbf{R}^\top(\gamma_p) \begin{bmatrix} x - x_t \\ y - y_t \end{bmatrix}, \tag{30}$$

and therefore the along-track and the cross-track errors become (see also Fig. 2):

$$x_e = (x - x_t)\cos(\gamma_p) + (y - y_t)\sin(\gamma_p), \quad (31)$$
$$y_e = -(x - x_t)\sin(\gamma_p) + (y - y_t)\cos(\gamma_p). \quad (32)$$

Note that using the virtual vehicle approach not only makes it easier and more intuitive to generate desired position references but also helps to avoid singularities when defining the cross-track error, more details are given in Lapierre et al. (2003).

## 4. OCEAN CURRENT COMPENSATION

It was shown in (Lekkas and Fossen, 2014) that the derivatives of (31)–(32) are computed as follows:

$$\dot{y}_e = U_r \sin(\psi + \beta_r - \gamma_p) + \underbrace{U_c \sin(\beta_c - \gamma_p)}_{\theta_y} \quad (33)$$

$$\dot{x}_e = u_r \cos(\psi_d - \gamma_p) + v_r \sin(\gamma_p - \psi_d) + \underbrace{U_c \sin(\gamma_p + \beta_{cx})}_{\theta_x} - U_t, \quad (34)$$

where $\beta_{cx} = \text{atan2}(V_x, V_y)$ and $\theta_x$, $\theta_y$ denote the varying components of the ocean current force in the body-fixed frame. The following adaptive observers for the estimates $\hat{y}_e, \hat{x}_e, \hat{\theta}_y$ and $\hat{\theta}_x$ were designed in (Lekkas and Fossen, 2014):

$$\dot{\hat{y}}_e = -\frac{U_r(\hat{y}_e + \alpha_y)}{\sqrt{\Delta^2 + (y_e + \alpha_y)^2}} + \hat{\theta}_y + k_1(y_e - \hat{y}_e), \quad (35)$$
$$\dot{\hat{\theta}}_y = k_2(y_e - \hat{y}_e), \quad (36)$$
$$\dot{\hat{x}}_e = -k_x \hat{x}_e + \hat{\theta}_x + \alpha_x + k_3(x_e - \hat{x}_e), \quad (37)$$
$$\dot{\hat{\theta}}_x = k_4(x_e - \hat{x}_e), \quad (38)$$

where $k_i > 0$ represent tuning constants, $U_r$ is the relative speed and $\Delta$ is the user-specified lookahead distance of the Line-of-Sight (LOS) guidance scheme (Fossen et al., 2003). Here, $\alpha_x$ and $\alpha_y$ are control inputs computed as follows:

$$\alpha_x = -\hat{\theta}_x, \quad (39)$$
$$\alpha_y = \Delta \frac{(\hat{\theta}_y / U_r)}{\sqrt{1 - (\hat{\theta}_y / U_r)^2}}. \quad (40)$$

In order for the surface vehicle to converge to the path and track the virtual vehicle, the indirect adaptive LOS guidance law generates the following heading angle and relative surge speed reference trajectories (Lekkas and Fossen, 2014):

$$\psi_d = \gamma_p - \beta_r + \arctan\left(-\frac{1}{\Delta}(y_e + \alpha_y)\right), \quad (41)$$
$$u_{r_d} = \sqrt{1 + \xi_t^2}\left(-v_r \frac{\xi_t}{\sqrt{1 + \xi_t^2}} + U_t + \alpha_x - k_x x_e,\right), \quad (42)$$

where $\xi_t := f(u_r, v_r, \Delta, y_e, \alpha_y)$ and $k_x > 0$ is a tuning constant. Eqs.(41)–(42) constitute the *guidance system*, which feeds reference trajectories to the control system in order to minimize the position error. The current components $\theta_y$ and $\theta_x$, which are estimated by the observers, can also be used to compute the ocean current vector (Lekkas and Fossen, 2014). This is done by the following equations:

$$\theta_y = U_c \sin(\beta_c - \gamma_p) \quad (43)$$
$$\theta_x = U_c \cos(\beta_c - \gamma_p) \quad (44)$$
$$\Downarrow$$
$$\hat{U}_c = \sqrt{\hat{\theta}_y^2 + \hat{\theta}_x^2} \quad (45)$$
$$\hat{\beta}_c = \gamma_p + \arctan\left(\frac{\hat{\theta}_y}{\hat{\theta}_x}\right) \quad (46)$$

where $\hat{(\cdot)}$ represents the estimates given by the observers. These estimates are used to produce new optimized paths based on more accurate information about the ocean current.

## 5. PATH-PLANNING IMPLEMENTATION AND SIMULATIONS

### *5.1 Implementation Descriptions*

In order to test the feasibility of using PS optimal control for path planning of marine surface vehicles in environments where obstacles and unknown currents are present, two versions of the problem were implemented. Both implementations start in the following way: The current is unknown and thus initially assumed to be zero. The algorithm computes the optimal path and the vehicle begins to track the virtual vehicle. As the surface vehicle moves, the nonlinear observer estimates the ocean current components and corrects the heading angle and surge speed in order to minimize the tracking error. From then on, the differences are the following:

**V1:** The path *does not* change, i.e. in this version the vehicle converges to the path computed initially as optimal. In this case, the solution is the same as a Dubins path, but approximated through PS optimal control.

**V2:** Based on the new information, the algorithm *recomputes* the optimal path starting from the vehicle's pose at that time instant. The vehicle is assigned to track the recomputed path each time the replanning process takes place.

If the computational cost was trivial, the ideal case would be to implement V2 and recompute the path at each time step. However this is hardly the case in real applications, especially when optimization-based solutions are concerned. Therefore, V1 represents the solution which is optimal w.r.t. to the path length, i.e. *minimum length.* V2 initially coincides with V1 but attempts to incorporate all incoming ocean current information in order to replay and finally approximate the *minimum-time* solution. The larger the number of times the path is replanned, the more refined the approximation becomes, but there are limits to how frequent it is feasible or even meaningful to do the replanning. One obvious reason is the computational cost required, while another reason is that the environmental conditions might vary continuously and fast, hence making it impractical to change the target path all the time. A third reason is that minimum-time paths also set specific requirements on the vehicle's speed, which might not always be possible or practical to satisfy. A mixed solution is to replan at given time intervals, hence computing a path that lies between minimum length and minimum time,

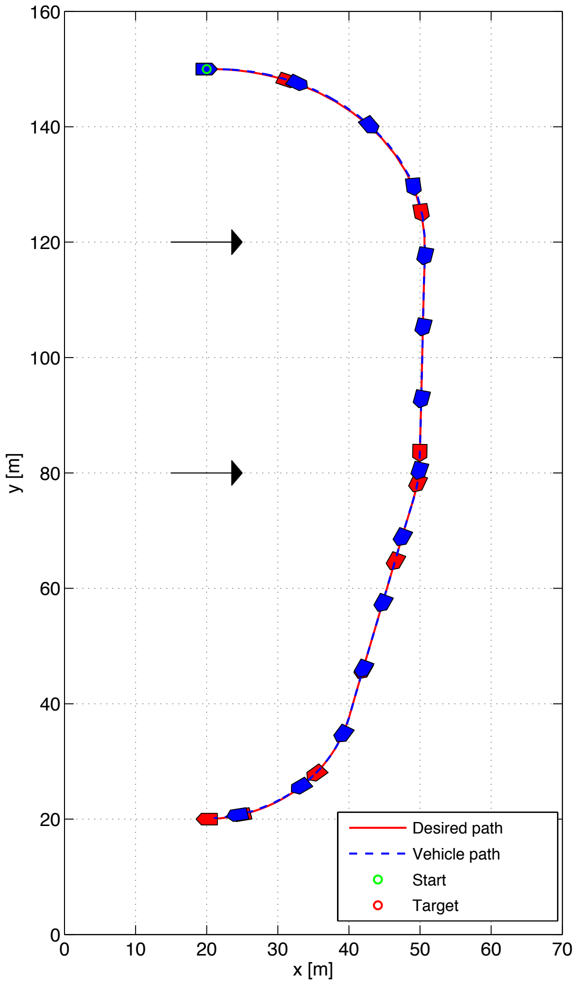


Fig. 3. Following the path in the case where the ocean current is estimated online. The replanning occurs when the vehicle is approx. at position $\mathbf{p}_r = (50, 80)$.

which is how V2 has been implemented in this paper.

### 5.2 Simulation Results

Fig. 3 shows the vehicle following the replanned path. In this case, initially the ocean current is unknown and therefore DIDO is given the values $V_x = V_y = 0$ m/sec. Note that due to the simplified dynamics, the first part of the path is identical to Dubins' path, that is, the *minimum-length* path. The replanning occurs when the surface vehicle is approximately at position $\mathbf{p}_r = (50\text{ m}, 80\text{ m})$. By that time the ocean current has been estimated and the new *minimum-time* path still resembles a Dubins path, which is now rotated w.r.t the minimum-length segment. In Fig. 4, the minimum-time path (solid red curve, it assumes the ocean current is known from the beginning) is compared with the minimum-length path (solid blue curve, assumes the ocean current remains unknown) described in V1 and the path resulting from the mixed approach (dashed green curve), where occasional replanning takes place as described in V2 and shown in Fig. 3. It is worth noting that the occasionally replanned path of V2 initially, when no information on the current is available, overlaps with the minimum-length path, whereas later

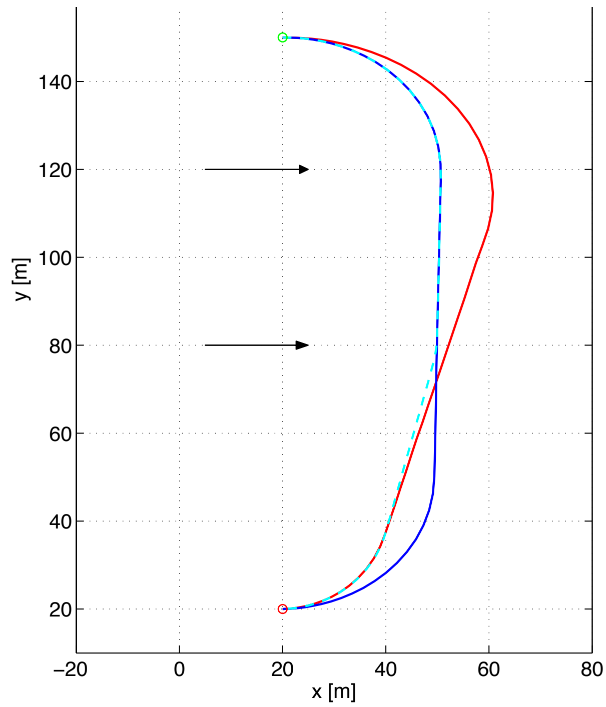


Fig. 4. Path shape for the three cases where the ocean current is respectively known (solid red curve), completely unknown (solid blue curve) and occasional replanning takes place (dashed green curve).

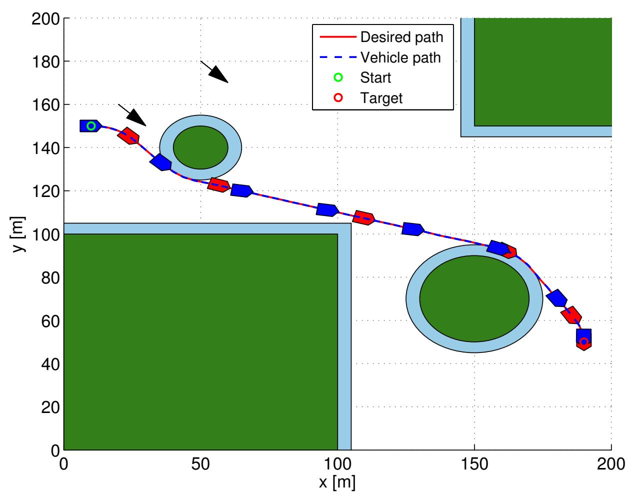


Fig. 5. Path planning and following in the presence of obstacles (green areas). The light blue areas illustrate the clearance constraints. The ocean current flows from North-West to South-East.

on it changes and converges to the minimum-time path. Fig. 5 demonstrates the efficiency of the presented path planning and guidance systems in the presence of both ocean currents and obstacles. It is possible to introduce clearance constraints by expanding the obstacles virtually, which is illustrated by the light blue areas around the obstacles. Fig. 6, on the other hand, shows the catastrophic errors that could occur if the external disturbances are not taken into account, i.e. if LOS guidance *without* integral action is implemented.

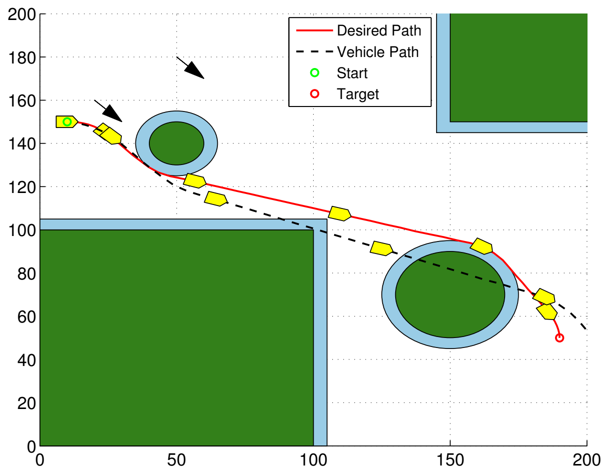


Fig. 6. During navigation under the influence of an unknown ocean current the vehicle will fail to follow the desired path if conventional LOS without integral term is used for guidance.

## 6. CONCLUSIONS

This paper dealt with the implementation of a pseudospectral (PS) optimal control approach for path planning and tracking for marine surface vehicles in environments where both obstacles and unknown ocean currents are present. The software package DIDO was used for computing the optimized path.

The main contribution was the addition of a nonlinear observer in order to estimate the unknown ocean currents and compute progressively the minimum-time (contrary to minimum-length) path through occasional replanning. In addition to recomputing the path via PS optimization, a suitable guidance system was employed to ensure the underactuated vehicle was successful in tracking the path without errors. Two different implementations were presented and tested through numerical simulations. The results so far indicate that PS optimal control is a promising method for such applications due to its accuracy and reduced computational cost.


## 7. ACKNOWLEDGEMENTS

This work was supported by the Research Council of Norway through the project "Next Generation Subsea IMR Operations" (project number 234108) and the Centres of Excellence funding scheme (project number 223254).


## REFERENCES


Barsky, B.A. and DeRose, T.D. (1989). Geometric continuity of parametric curves: Three equivalent characterizations. *IEEE Computer Graphics and Applications*, 9(6), 60–69.

Bollino, K.P., Lewis, L.R., Sekhavat, P., and Ross, I.M. (2007). Pseudospectral optimal control: A clear road for autonomous intelligent path planning. In *Proceedings of the AIAA InfoTech at Aerospace Conference and Exhibit*, 1228–1241. Rohnert Park, US.

Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L.E., and Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*. The MIT Press.

Dubins, L.E. (1957). 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.

Falcone, M. and Ferretti, R. (2013). *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations*. SIAM.

Fossen, T.I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley and Sons Ltd.

Fossen, T.I., Breivik, M., and Skjetne, R. (2003). Line-of-sight path following of underactuated marine craft. *Proceedings of the 6th IFAC MCMC, Girona, Spain*, 244–249.

Fossen, T.I. and Lekkas, A.M. (2015). 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*, http://dx.doi.org/10.1002/acs.2550.

Gill, P.E., Murray, W., and Saunders, M.A. (2005). SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1), 99–131.

Gong, Q., Fahroo, F., and Ross, I.M. (2008). Spectral algorithm for pseudospectral methods in optimal control. *Journal of Guidance, Control, and Dynamics*, 31(3), 460–471.

Hurni, M.A. and Kiriakidis, K. (2015). Planning the minimum time and optimal survey trajectory for autonomous underwater vehicles in uncertain current. *Robotics*, 4(4), 516–528.

Khalil, H. (2002). *Nonlinear Systems*. Prentice Hall.

Lapierre, L., Soetanto, D., and Pascoal, A. (2003). Nonlinear path following with applications to the control of autonomous underwater vehicles. In *42nd IEEE Conference on Decision and Control*. Maui, US.

LaValle, S.M. (2006). *Planning Algorithms*. Cambridge university press.

Lekkas, A.M. (2014). *Guidance and Path-Planning Systems for Autonomous Vehicles*. Ph.D. thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Lekkas, A.M., Dahl, A.R., Breivik, M., and Fossen, T.I. (2013). Continuous-curvature path generation using Fermat's spiral. *Modeling, Identification and Control*, 34(4), 183–198.

Lekkas, A.M. and Fossen, T.I. (2014). Trajectory tracking and ocean current estimation for marine underactuated vehicles. In *In Proceedings of the IEEE Conference on Control Applications*, 905–910. Antibes, France.

Roald, A.L. (2015). *Path Planning for Vehicle Motion Control Using Numerical Optimization Methods*. Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Ross, I.M. and Fahroo, F. (2003). Legendre pseudospectral approximations of optimal control problems. In *New Trends in Nonlinear Dynamics and Control and their Applications*, 327–342. Springer.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. The MIT Press.

Tsourdos, A., White, B., and Shanmugavel, M. (2011). *Cooperative Path Planning of Unmanned Aerial Vehicles*. Wiley Online Library.