

Learning Optimal Guidance Schemes with Safety Guarantees for Underactuated Marine Vehicles^{*}

Eirik L. Foseid^{*} Erlend A. Basso^{*}
Henrik M. Schmidt-Didlaukies^{*} Mathias Marley^{*}
Kristin Y. Pettersen^{*} Jan Tommy Gravdahl^{*}

^{*} Norwegian University of Science and Technology, Trondheim, Norway
eirik.l.foseid@ntnu.no

Abstract: In this paper, we propose a guidance scheme for underactuated marine vehicles with collision-avoidance guarantees. The guidance scheme is based on a line-of-sight approach and ensures path-following when safety allows it, but deviates from the desired path if safety mandates it. We provide formal safety guarantees by proving forward invariance of the collision-free safe set using a barrier function. However, while barrier-function-based control schemes provide formal safety guarantees, the real-life performance is often lacking due to the challenge of manually tuning such controllers. We demonstrate that the performance of the control law can be improved by replacing a part of the control law by a neural network, while retaining the same formal safety guarantees.

Copyright © 2024 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Line-of-sight guidance, barrier functions, safety-critical control, neural networks, differentiable programming

1. INTRODUCTION

Collision avoidance is a critical design aspect when implementing safe autonomous systems. Autonomous systems are increasingly being used in diverse applications such as transportation, robotics, and manufacturing. As the complexity of these systems grows, ensuring safety and reliability becomes more challenging. Methods for collision avoidance have traditionally relied on rule-based algorithms or trajectory optimization techniques. However, these approaches often suffer from limitations such as lack of adaptability to dynamic environments, and difficulty in handling constraints on system inputs and states.

Control barrier functions (CBFs) (Ames et al., 2019) offer a solution that provides a rigorous mathematical framework for ensuring the safety of autonomous systems. CBFs are Lyapunov-like functions that can be employed to construct control laws that guarantee safety, such as ensuring that system states remains within a desired safe set at all times. CBF designs for collision avoidance for fully actuated marine surface vehicles were proposed in Thyri et al. (2020) and Basso et al. (2020). The former assumes that the dynamics are fully known, while the latter includes unknown ocean currents in the dynamics. To overcome this unknown disturbance, the path following problem is solved by utilizing integral action in the nominal controller, while the robust CBF formulation of Emam et al. (2019) is employed to ensure collision avoidance. However, since inte-

gral action is included at the same level as the CBFs, integral wind-up leads to large transient tracking errors following an evasive maneuver. To mitigate this issue, a CBF design for collision avoidance for underactuated marine surface vehicles was proposed in Marley et al. (2021). In that paper, CBFs are employed at the guidance level as opposed to the actuator level, which enables us to add integral action at the actuator level controllers without encountering integral wind-up complications. CBFs are also employed for collision avoidance of underactuated marine vehicles in Haraldsen et al. (2023), in which the safety criteria are derived from the so-called velocity obstacle principle.

Other algorithms for collision avoidance for marine vehicles include model predictive control approaches such as Johansen et al. (2016), Eriksen et al. (2019) and Tengesdal et al. (2022). However, none of these approaches have provable safety guarantees.

The main contributions of this paper are twofold. First, we introduce a novel barrier-function-based line-of-sight (LOS) guidance scheme with strict safety guarantees in terms of collision avoidance for underactuated marine surface vehicles. As opposed to most CBF-based methods, we do not synthesize the control law by solving a quadratic program online. Instead, we provide a closed-form expression of the safeguarding control law, and prove forward invariance of the resulting closed-loop system using a barrier function. Second, we demonstrate that the performance of the control law can be improved by replacing a part of the control law by a neural network, while retaining the same strict safety guarantees. Specifically, the proposed control law includes a saturation function, where the shape of the function is a design choice influencing the closed-loop be-

^{*} This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, through the ERC Advanced Grant 101017697-CRÈME, and by the Research Council of Norway through project No. 304667 and the Centres of Excellence funding scheme, project 223254, NTNU AMOS.

havior/performance. The choice of saturation function has a major impact on the aggressiveness of the evasive maneuver. Moreover, it is not trivial to select the saturation function to obtain a satisfactory trade-off between minimizing the tracking error while avoiding overly aggressive turning maneuvers. To overcome this problem, we propose a neural-network-based saturation function and demonstrate how this improves performance compared to an approach based on manual selection.

This paper is organized as follows. Section 2 presents a unicycle model that we use to model a marine surface vehicle in transit, as well as a uniformly globally asymptotically stabilizing LOS guidance law for this model. Then, Section 3 proposes a novel barrier-function-based control law which ensures safety by guaranteeing forward invariance of the safe set. In Section 4, we present a learning-based extension of the safeguarding control law by employing neural networks, differentiable programming and gradient-based optimization to improve the performance while still satisfying the same strict safety guarantees. Section 5 presents our results from a simulation study where we compare the performance of the closed-loop system using the learning-based method to the nominal control law. Finally, Section 6 presents our conclusions.

Notation

The Euclidean norm is denoted $|x| = (x^\top x)^{1/2}$. The standard basis vectors in \mathbb{R}^n are denoted e_1, \dots, e_n . We denote the boundary of a set $K \subset \mathbb{R}^n$ by ∂K . The unit circle is defined by $\mathbb{S} := \{x \in \mathbb{R}^2 : |x| = 1\}$, and the group of planar rotations by $\text{SO}(2) := \{R \in \mathbb{R}^{2 \times 2} : R^\top R = I, \det R = 1\}$. A unit vector $z \in \mathbb{S}$ maps to a rotation matrix through the map $R : \mathbb{S} \rightarrow \text{SO}(2)$ defined by $R(z) := (z \ S z)$, where $S := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$. We denote the signum function by $\text{sgn} : \mathbb{R} \rightarrow \{-1, 1\}$, where $\text{sgn}(0) = 1$. Furthermore, $\text{atan2} : \mathbb{R}^2 \setminus \{0\} \rightarrow (-\pi, \pi]$ denotes the four-quadrant inverse of \tan , such that $\text{atan2}(y, x)$ gives the angle between the x -axis and the ray from the origin to the point (x, y) .

2. PROBLEM FORMULATION

For a marine surface vehicle in transit, we approximate the kinematics by the unicycle model

$$\dot{p} = vz, \quad (1)$$

where $p \in \mathbb{R}^2$ denotes the position, $v > 0$ is the speed-over-ground, and $z \in \mathbb{S}$ is the course, which is considered a control input.

The reference path is a continuously differentiable mapping $p_r : \mathbb{R} \rightarrow \mathbb{R}^2$ parameterized by the path parameter $\theta \in \mathbb{R}$. We additionally assume that the path is regular, that is $|p'_r(\theta)| \neq 0$ for all $\theta \in \mathbb{R}$, such that we can define a unit tangent vector field $\tau : \mathbb{R} \rightarrow \mathbb{S}$ by

$$\tau(\theta) := \frac{p'_r(\theta)}{|p'_r(\theta)|}. \quad (2)$$

We formulate the nominal control objective as a maneuvering problem

Maneuvering problem (Skjetne et al., 2004)

- **Geometric Task:** Force the position of the ship to converge to the desired path,

$$\lim_{t \rightarrow \infty} |p(t) - p_r(\theta(t))| = 0. \quad (3)$$

- **Dynamic Task:** Force the path speed to converge to the desired speed assignment:

$$\lim_{t \rightarrow \infty} |\dot{\theta}(t) - v_s(\theta)| = 0. \quad (4)$$

The nominal control problem is solved by employing a line-of-sight guidance law together with an appropriate update law for θ . To this end, we define the along-track and cross-track errors according to the first and second components of the vector

$$\varepsilon(p, \theta) := R(\tau(\theta))^\top (p - p_r(\theta)), \quad (5)$$

respectively. The LOS guidance law will ensure that the cross-track error ε_2 tends to zero, while the update law for θ will ensure that the along-track error ε_1 tends to zero. The LOS guidance law is defined by (Marley et al., 2021)

$$\bar{\kappa}(p, \theta) := R(z_\Delta(p, \theta))\tau(\theta) \quad (6)$$

where

$$z_\Delta(p, \theta) = \frac{1}{\sqrt{\Delta^2 + \varepsilon_2(p, \theta)^2}} \begin{pmatrix} \Delta \\ -\varepsilon_2(p, \theta) \end{pmatrix}, \quad (7)$$

denotes the line-of-sight vector and $\Delta > 0$ is the lookahead-distance (Fossen, 2020). Note that the course angle representing $\bar{\kappa}$ on the interval $(-\pi, \pi]$ is equivalent to the sum of the angles corresponding to the unit vectors z_Δ and τ , mapped to the interval $(-\pi, \pi]$.

The speed assignment

$$v_s(\theta) := \frac{v}{|p'_r(\theta)|}, \quad (8)$$

ensures that $|p'_r(\theta)|v_s(\theta) = v$, implying that $|\dot{p}_r(\theta(t))| \rightarrow v$ if $|\dot{\theta}(t)| \rightarrow v_s(\theta(t))$. The update law for θ is defined as

$$\dot{\theta} := e_1^\top z_\Delta(p, \theta)v_s(\theta) + \mu(p - p_r(\theta))^\top p'_r(\theta), \quad (9)$$

where $\mu > 0$. The first term represents the along-track velocity, while the latter term is a so-called ‘‘gradient feedback’’ term (Skjetne et al., 2011), ensuring that the along-track error tends to zero.

Theorem 1. (Skjetne et al. (2011)). The control law defined by (6) and (9) renders the set

$$\mathcal{A} := \{(p, \theta) \in \mathbb{R}^3 : p = p_r(\theta)\}, \quad (10)$$

uniformly globally asymptotically stable for the closed-loop system

$$\begin{aligned} \dot{p} &= vR(z_\Delta(p, \theta))\tau(\theta) \\ \dot{\theta} &= e_1^\top z_\Delta(p, \theta)v_s(\theta) + \mu(p - p_r(\theta))^\top p'_r(\theta). \end{aligned} \quad (11)$$

Problem statement: Modify the guidance law (6) such that the marine vehicle deviates from the desired path p_r whenever a safety objective demands it. We express this problem through the following two objectives:

- (1) Safety objective: Given some unsafe domain $K_u \subset \mathbb{R}^2$, render $K_s := \mathbb{R}^2 \setminus K_u$ forward invariant.
- (2) Nominal objective: The maneuvering problem.

3. SAFE GUIDANCE LAW

Define the barrier function candidate $B : \mathbb{R}^2 \rightarrow \mathbb{R}$ by

$$B(p) := r - |p - p_o|, \quad (12)$$

where $r > 0$ is the radius of the obstacle and p_o is the center of the obstacle. Our goal is to design a safeguarding control law ensuring forward invariance of the safe set $K_s \subset \mathbb{R}^2$ defined by

$$K_s := \{p \in \mathbb{R}^2 : B(p) \leq 0\}, \quad (13)$$

for the system (1). To this end, consider the mapping $\kappa : \mathbb{S} \times \mathbb{S} \times [-1, 1] \rightarrow \mathbb{S}$ defined by

$$\kappa(z, \zeta, s) := \begin{cases} z & z^\top \zeta \geq s, \\ s\zeta + \text{sgn}(z^\top S\zeta)\sqrt{1-s^2}S\zeta & z^\top \zeta < s. \end{cases} \quad (14)$$

The mapping κ is continuous at all points in the set $\{(z, \zeta, s) \in \mathbb{S} \times \mathbb{S} \times [-1, 1] : z^\top \zeta \neq -1\}$. We prove this fact in Lemma 3, which can be found in the appendix.

Now, let $\sigma : \mathbb{R} \rightarrow [-1, 1]$ be a continuous function satisfying $x\sigma(x) > 0$ for all $x \in \mathbb{R} \setminus \{0\}$. The resulting guidance law

$$\begin{aligned} \dot{\theta} &= e_1^\top z_\Delta(p, \theta)v_s(\theta) + \mu(p - p_r(\theta))^\top p_r'(\theta) \\ z &= \kappa(\bar{\kappa}(p, \theta), z_o(p), \sigma(B(p))), \end{aligned} \quad (15)$$

where κ is given by (14), and

$$z_o(p) := \frac{p - p_o}{|p - p_o|}, \quad (16)$$

leads to the closed-loop system

$$\left. \begin{aligned} \dot{p} &= v\kappa(R(z_\Delta(p, \theta))\tau(\theta), z_o(p), \sigma(B(p))) \\ \dot{\theta} &= e_1^\top z_\Delta(p, \theta)v_s(\theta) + \mu(p - p_r(\theta))^\top p_r'(\theta) \end{aligned} \right\} (p, \theta) \in C, \quad (17)$$

where the flow set is defined by

$$C := \{(p, \theta) \in \mathbb{R}^3 : p \neq p_o\}. \quad (18)$$

The guidance law (15) is similar to (Marley et al., 2020, Eq. (14)) modulo the saturation function σ and the state θ . The guidance scheme (15) is identical to the nominal guidance law given by (6) and (9) when safety permits it, that is, when (p, θ) is in the set

$$\mathcal{N} := \{(p, \theta) \in \mathbb{R}^3 : z_o(p)^\top \bar{\kappa}(p, \theta) \geq \sigma(B(p))\}. \quad (19)$$

However, when the nominal guidance law is not safe, that is, when $(p, \theta) \notin \mathcal{N}$, then evasive action is being taken until (p, θ) is again in \mathcal{N} .

Membership in the set \mathcal{N} at a point (p, θ) depends on the nominal course of the vehicle relative to the obstacle, $z_o(p)^\top \bar{\kappa}(p, \theta)$, and the distance of the vehicle to the obstacle $B(p)$. The quantity $z_o(p)^\top \bar{\kappa}(p, \theta)$ can be understood as the cosine of the course relative to the obstacle. Moreover, the function σ controls how large the set of allowable nominal courses \mathcal{N} is, as seen in (19) and visualized in Figure 1 by the solid green circle segments. The function σ can therefore be considered a tuning function.

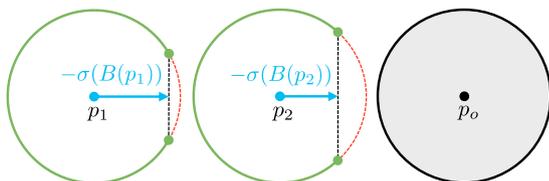


Fig. 1. Safe courses for the vehicle at two positions p_1 and p_2 shown in green. Unsafe courses are shown in red. The figure is inspired by (Marley et al., 2020, Fig. 4).

Theorem 2. The safe set $K := K_s \times \mathbb{R}$ is forward invariant for the system (17).

Proof. Let \mathcal{U} be a neighborhood of ∂K . The derivative of B along the solutions of (17) for $(p, \theta) \in \mathcal{U} \setminus K$ is given by

$$\begin{aligned} \dot{B}(p, \theta) &= -vz_o(p)^\top \left(\sigma(B(p))z_o(p) \right. \\ &\quad \left. + \text{sgn}(-z_o^\top SR(z_\Delta(p, \theta))\tau(\theta))\sqrt{1-\sigma(B(p))^2}Sz_o(p) \right) \\ &= -v\sigma(B(p)) \\ &\leq 0, \end{aligned}$$

where the last inequality follows from the fact that $(p, \theta) \in \mathcal{U} \setminus K$ implies $B(p) > 0$ and hence $\sigma(B(p)) > 0$. It follows that K is forward invariant for the system (17). \square

4. LEARNING SATURATION FUNCTIONS

The saturation function σ is a design parameter since there are many functions that satisfy the required properties. Different saturation functions impact the behaviour of the closed-loop system in different ways. For our application it mainly impacts the turning radius of the vehicle and how close the system gets to the barrier. Therefore, the choice of saturation function depends on the desired behaviour of the closed-loop system.

Instead of manually selecting the saturation function, we propose a neural-network-based saturation function satisfying the required properties. We then use differentiable programming and gradient-based optimization to find an optimal saturation function.

4.1 Neural-network-based saturation function

One valid choice of saturation function is

$$\sigma(x) := \tanh\left(\frac{x}{a}\right) \quad (20)$$

where a is a parameter adjusting the slope of the function, as shown in Figure 2.

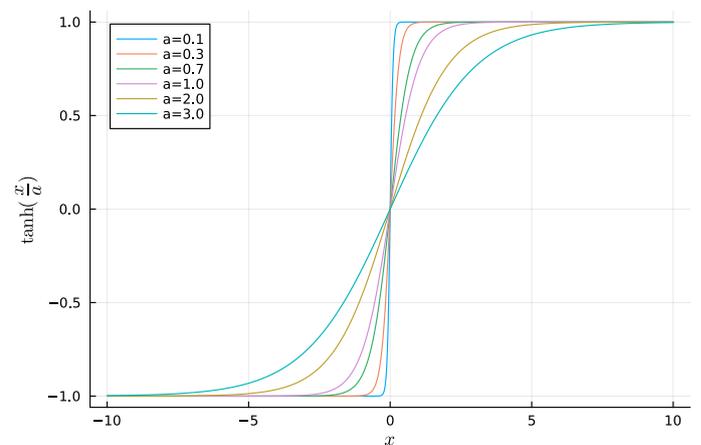


Fig. 2. Saturation function (20) for different values of a .

We propose the saturation function $\sigma_\phi : \mathbb{R} \times \mathbb{R}^l \rightarrow (-1, 1)$

$$\sigma_\phi(x, y) := \tanh\left(\frac{x}{a_\phi(x, y)}\right), \quad (21)$$

where $a_\phi : \mathbb{R} \times \mathbb{R}^l \rightarrow (0, a_{\max})$ now is a neural network model with parameters ϕ controlling the overall shape of the saturation function. We have here let a_ϕ depend on an

additional input $y \in \mathbb{R}^l$, which can be selected on a case-by-case basis. This allows a much richer set of avoidance behaviors to be learned by the network. We remark that since a_ϕ maps to $(0, a_{\max})$ and \tanh is strictly increasing, it follows that

$$\sigma_\phi(x, y)x > \tanh\left(\frac{x}{a_{\max}}\right)x > 0, \quad (22)$$

for all $(x, y) \in (\mathbb{R} \setminus \{0\}) \times \mathbb{R}^l$. The design parameter a_{\max} therefore provides convenient control of the “minimal aggressiveness” of the learned saturation function. Furthermore, a_ϕ is continuous as long as the activation functions employed in the neural network are chosen to be continuous. This is because a_ϕ then becomes a composition of a large but finite number of continuous functions. As $a_\phi(x, y) \neq 0$, it follows that σ_ϕ is also continuous in this case. Since the bound (22) holds uniformly in y and σ_ϕ is continuous, the conclusions of Theorem 2 hold, no matter how the additional input y is chosen.

4.2 Vehicle simulation model

In order to obtain more realistic results, we opt to use a higher-fidelity six degree-of-freedom model for training and evaluation. Specifically, the mathematical model of the marine surface vehicle is the Otter from Maritime Robotics and is available in the MSS toolbox (Fossen and Perez, 2004). Let $\eta := (p, p_z, q) \in \mathbb{R}^2 \times \mathbb{R} \times \mathbb{S}^3$ denote the configuration of the marine vehicle, where p_z denotes the z or down position, and $q \in \mathbb{S}^3$ is a unit quaternion representing the orientation. Moreover, let $\nu \in \mathbb{R}^6$ denote the linear and angular velocities of the marine vehicle. For readability, we let $\omega = \nu_6$, that is, the yaw rate of the vehicle. The equations of motion are given by (Fossen, 2020, Chapter 6)

$$\begin{aligned} \dot{\eta} &= J(\eta)\nu, \\ M\dot{\nu} + C(\nu)\nu + d(\nu) + g(\eta) &= u, \end{aligned} \quad (23)$$

where $M \in \mathbb{R}^{6 \times 6}$ is the inertia matrix consisting of rigid-body and hydrodynamic inertia, $C(\nu) \in \mathbb{R}^{6 \times 6}$ is the matrix of Coriolis and centripetal forces, $d(\nu) \in \mathbb{R}^6$ contains dissipative hydrodynamic forces, and $g(\eta) \in \mathbb{R}^6$ contains gravitational and buoyancy forces. Finally, $u = (u_1, 0, 0, 0, 0, u_6) \in \mathbb{R}^6$ represents control forces.

4.3 Training of the neural network

We want the marine vehicle to avoid the obstacle while minimizing the tracking error as much as possible. Naïvely, this would lead to aggressive turning in some cases, so additionally we want to maintain a turning rate below a certain threshold value.

For the neural network $(x, y) \mapsto a_\phi(x, y)$, we use a dense neural network with inputs $x = B(p)$ and $y = (z_o(p), \nu_1, \nu_2) \in \mathbb{R}^4$ and three hidden layers. The first hidden layer has 30 outputs and we use the swish activation function introduced in Ramachandran et al. (2017), which is defined by $\text{swish}(x) := x\varsigma(x)$, where ς is the logistic function

$$\varsigma(x) := \frac{1}{1 + e^{-x}}. \quad (24)$$

Moreover, the second and third hidden layers have a width 30 and 10 with swish and $\log(\cosh(\cdot))$ activation functions,

respectively. For the output layer we use a scaled logistic activation function $x \mapsto a_{\max}\varsigma(x)$ to ensure that the network maps from $\mathbb{R} \times \mathbb{R}^4$ to $(0, a_{\max})$.

We implement the differential equations (23) in conjunction with the control system in Julia (Bezanson et al., 2017), where they are solved using a differentiable ordinary differential equation (ODE) solver provided by DifferentialEquations.jl and DiffEqFlux.jl (Rackauckas and Nie, 2017; Rackauckas et al., 2019).

The loss function measures the performance of the closed-loop system for several different scenarios where we simulate the system starting in different initial conditions $x_0 = (\eta_0, \nu_0, \theta_0)$. Every simulation produces a numerical approximation of a solution to (23) on the domain $[0, T]$ represented by its values at N evenly spaced points in the domain. The final simulation time $T > 0$ and the number of points $N > 1$ in the discrete time domain are parameters chosen by the designer. We denote such a discretized solution by $\bar{x} = (\bar{\eta}, \bar{\nu}, \bar{\theta}) = (\eta_k, \nu_k, \theta_k)_{k=0}^{N-1}$. The loss incurred over the course of a simulation, for one choice of initial conditions, is now defined by

$$L(\bar{x}) := \sum_{k=1}^{N-1} l_1(p_k, \theta_k) + \sum_{k=1}^{N-1} l_2(p_k, \theta_k, \omega_k), \quad (25)$$

with the constituent functions

$$l_1(p, \theta) := \begin{cases} 0 & (p, \theta) \in \mathcal{N}, \\ c_1 \varepsilon_2(p, \theta)^2 & (p, \theta) \notin \mathcal{N}, \end{cases} \quad (26)$$

$$l_2(p, \theta, \omega) := \begin{cases} 0 & (p, \theta) \in \mathcal{N}, \\ 0 & (p, \theta) \notin \mathcal{N}, |\omega| < c_3, \\ c_2(|\omega| - c_3)^2 & (p, \theta) \notin \mathcal{N}, |\omega| \geq c_3, \end{cases} \quad (27)$$

where \mathcal{N} denotes the set where the nominal controller is active, as defined in (19). The parameter $c_1 > 0$ controls the penalization of deviations from the reference path when the vehicle is avoiding an obstacle, while the parameter $c_2 > 0$ controls the penalization of the turning rate when its magnitude exceeds $c_3 \geq 0$.

The total loss function over the set of initial conditions \mathcal{I} is given by

$$\mathcal{L}(\phi) := \sum_{x_0 \in \mathcal{I}} L(\bar{p}(x_0, \phi), \bar{\theta}(x_0, \phi), \bar{\omega}(x_0, \phi)). \quad (28)$$

Since we employ a differentiable ODE solver (Rackauckas and Nie, 2017), we are able to calculate the derivative of the loss function (28) with respect to the parameters ϕ of the neural network as a function of the entire solution of the ODE. We can then apply gradient-based optimization to search for optimal parameters ϕ used in the saturation function σ_ϕ .

For training, we used 9 initial positions located at 1 m North equally spaced between ± 8 m East. Each initial position was given a unique straight line reference path going in the North-direction starting at the initial position. The initial positions and reference paths are shown in Figure 3.

During training, the surge force u_1 is set to eight values from 100 N to 300 N with increments of 25 N. The yaw moment u_6 is computed by a PID controller in order to track the desired course as output by the LOS guidance algorithm (15) together with the neural-network-based satura-

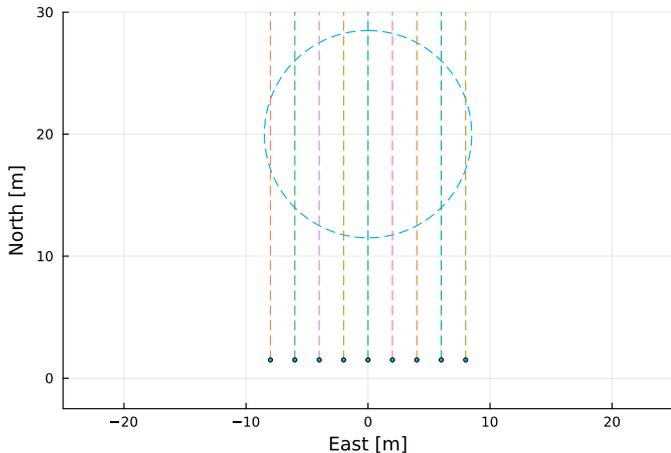


Fig. 3. Initial positions and reference trajectories used for training.

tion function. Moreover, the obstacle has a radius of 8.5 m and is centered at (20 m, 0 m). Finally, the parameters of the loss function were chosen as $c_1 = 1$, $c_2 = 1000$ and $c_3 = 100$.

The training was done by using the Adam optimizer (Kingma and Ba, 2015) for 30 iterations with a learning rate of 0.05. We then continued the optimization for 100 iterations using a learning rate of 0.01.

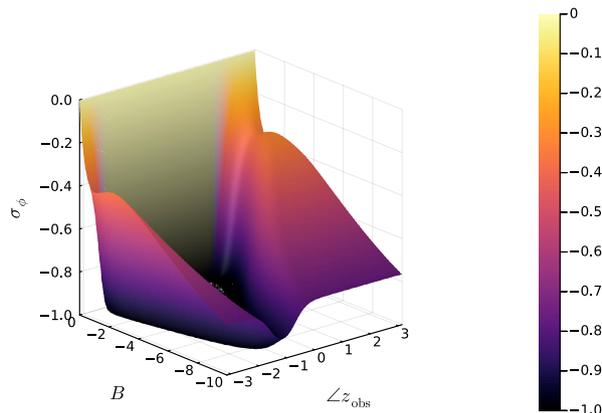


Fig. 4. The learned saturation function for a range of values of B and $z_{\text{obs}} := \text{atan2}(e_2^T z_{\text{obs}}, e_1^T z_{\text{obs}})$ with $\nu_1 = 3.2 \text{ m s}^{-1}$ and $\nu_2 = 0$.

A surface plot of the learned saturation function as a function of B and z_{obs} for a constant horizontal speed of $\nu_1 = 3.2 \text{ m s}^{-1}$ and $\nu_2 = 0 \text{ m s}^{-1}$ can be seen in Figure 4. From Figure 4, it is clear that when the angle representation of z_{obs} approaches π , the saturation function σ_ϕ is more aggressive. This is expected since a relative orientation of π radians represents a head-on collision scenario. On the other extreme, a relative orientation of 0 radians represents moving away from the obstacle. This added information about the relative orientation allows for less conservative collision avoidance. For instance, the ship can potentially pass closer to the obstacle while following the nominal course, as the value of the saturation function can be kept close to -1 at certain relative orientations. This is in contrast to the nominal saturation function (20) which has

no information about how the ship is approaching the obstacle, relying only on the value of the barrier function B .

5. EVALUATION RESULTS

Figure 5 and Figure 6 show trajectories of the system at speeds of 3.5 m s^{-1} and 2.2 m s^{-1} , respectively, comparing the learned saturation function and the nominal saturation function (20) for different values of the tuning parameter a . Each plot shows the nominal trajectories for 10 values of a , where the lowest value is chosen such that the vehicle is very close to the obstacle region while not entering it.

The trajectories resulting from use of the learned saturation function have a very distinctive shape. While the initial avoiding action for the learned cases is undertaken in a similar manner as the nominal cases, the second turn back towards the nominal path is initiated much earlier. This leads to a tidier and more efficient avoiding maneuver, allowing the vehicle to return to the path in a more timely manner. Due to the turn rate penalty defined in (27), the learned trajectory can attain a larger curvature at lower speeds, as is seen by comparing Figure 5 and Figure 6. Since it is easier to accommodate the turn rate constraint at lower speeds, we observe in Figure 6 that the learned trajectory comes much closer to the obstacle.

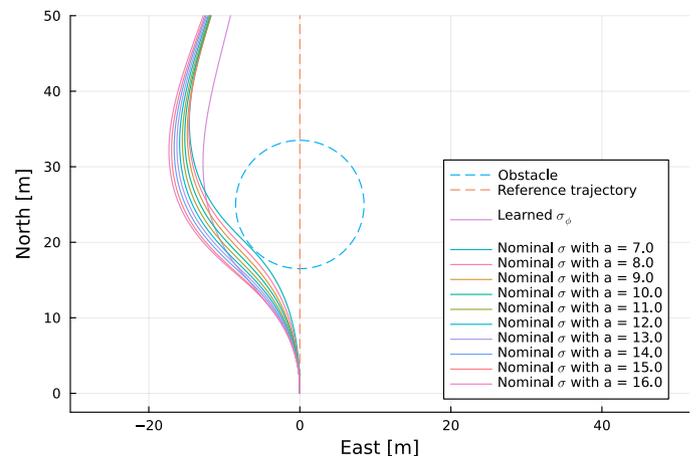


Fig. 5. Trajectories when the ship has a surge speed of 3.5 m s^{-1} .

6. CONCLUSION

In this paper, we have proposed a novel barrier-function-based control law which ensures safety by guaranteeing forward invariance of a collision-free safe set. We also present a learning-based extension of the safeguarding control law, employing neural networks, differentiable programming and gradient-based optimization, that improves the performance while still satisfying the same strict safety guarantees. We present results from a simulation study where we validate the performance of the closed-loop system using the learning-based method, and show that it improves the performance of the nominal control law.

Appendix A. CONTINUITY OF SAFEGUARDING CONTROL LAW

Lemma 3. The control law (14) restricted to the set $\{(z, \zeta, s) \in \mathbb{S} \times \mathbb{S} \times [-1, 1] : z^T \zeta \neq -1\}$ is continuous.

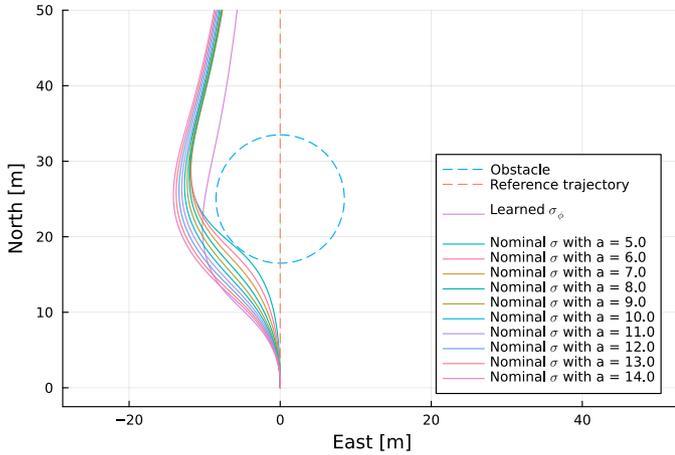


Fig. 6. Trajectories when the ship has a surge speed of 2.2 ms^{-1} .

Proof. To prove continuity of κ , we first show continuity of $\xi : (-\pi, \pi) \times [-1, 1] \rightarrow (-\pi, \pi)$ defined by

$$\xi(\varphi, s) := \begin{cases} \varphi & \cos(\varphi) \geq s \\ \text{sgn}(\varphi) \arccos(s) & \cos(\varphi) < s \end{cases} \quad (\text{A.1})$$

It is clear that ξ is continuous at all (φ, s) such that $\cos(\varphi) > s$ and at all (φ, s) such that $\cos(\varphi) < s$ (note that $\varphi \neq 0$ in this region). We want to show that ξ is continuous also at all (φ, s) such that $\cos(\varphi) = s$. Consider fixed (φ_0, s_0) satisfying $\cos(\varphi_0) = s_0$ and $\varphi_0 \neq 0$. From continuity of \arccos , it follows that for every $\epsilon > 0$, there exists $\delta_a > 0$ such that $s \in [-1, 1]$ and $|s - s_0| \leq \delta_a$ implies $|\arccos(s) - \arccos(s_0)| \leq \epsilon$. Let $\delta := \min\{\epsilon, \delta_a, \frac{|\varphi_0|}{2}\}$, which ensures that $\text{sgn}(\varphi) = \text{sgn}(\varphi_0)$. Then, for all $(\varphi, s) \in (-\pi, \pi) \times [-1, 1]$ such that $|(\varphi, s) - (\varphi_0, s_0)| \leq \delta$, we have that

$$|\xi(\varphi, s) - \xi(\varphi_0, s_0)| = |\varphi - \varphi_0| \leq \epsilon$$

when $\cos(\varphi) \geq s$, and

$$\begin{aligned} |\xi(\varphi, s) - \xi(\varphi_0, s_0)| &= |\text{sgn}(\varphi) \arccos(s) - \varphi_0| \\ &= |\text{sgn}(\varphi) \arccos(s) - \text{sgn}(\varphi_0) \arccos(s_0)| \\ &= |\arccos(s) - \arccos(s_0)| \leq \epsilon \end{aligned}$$

when $\cos(\varphi) < s$. For the case $\varphi_0 = 0$, it suffices to pick $\delta := \min\{\epsilon, \delta_a\}$. It now holds that

$$|\xi(\varphi, s) - \xi(\varphi_0, s_0)| = |\varphi| \leq \epsilon$$

when $\cos(\varphi) \geq s$, and

$$|\xi(\varphi, s) - \xi(\varphi_0, s_0)| = |\arccos(s)| \leq \epsilon.$$

when $\cos(\varphi) < s$. This establishes continuity of ξ . Continuity of κ now follows by noting that

$$\kappa(z, \zeta, s) = \cos(\xi(f(z, \zeta), s))\zeta + \sin(\xi(f(z, \zeta), s))S\zeta \quad (\text{A.2})$$

where $f(z, \zeta) := \text{atan2}(z^T S\zeta, z^T \zeta)$ is continuous since $z^T \zeta \neq -1$ in the domain in question. \square

REFERENCES

Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. (2019). Control Barrier Functions: Theory and Applications. In *Proc. 2019 European Control Conference*. Naples, Italy.

Basso, E.A., Thyri, E.H., Pettersen, K.Y., Breivik, M., and Skjetne, R. (2020). Safety-Critical Control of Autonomous Surface Vehicles in the Presence of Ocean

Currents. In *Proc. 2020 IEEE Conference on Control Technology and Applications (CCTA)*, 396–403.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V.B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1), 65–98.

Emam, Y., Glotfelter, P., and Egerstedt, M. (2019). Robust Barrier Functions for a Fully Autonomous, Remotely Accessible Swarm-Robotics Testbed. In *Proc. 58th IEEE Conf. Decision and Control*. Nice, France.

Eriksen, B.H., Breivik, M., Wilthil, E.F., Flåten, A.L., and Brekke, E.F. (2019). The branching-course model predictive control algorithm for maritime collision avoidance. *Journal of Field Robotics*, 36(7), 1222–1249.

Fossen, T.I. and Perez, T. (2004). Marine systems simulator (MSS). URL github.com/cybergalactic/MSS.

Fossen, T.I. (2020). *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2nd edition.

Haraldsen, A., Wiig, M.S., Ames, A.D., and Pettersen, K.Y. (2023). Safety-Critical Control of Nonholonomic Vehicles in Dynamic Environments using Velocity Obstacles. ArXiv:2310.00713 [cs, eess].

Johansen, T.A., Cristofaro, A., and Perez, T. (2016). Ship Collision Avoidance Using Scenario-Based Model Predictive Control. *IFAC-PapersOnLine*, 49(23), 14–21.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*. San Diego, CA, USA.

Marley, M., Skjetne, R., Basso, E., and Teel, A.R. (2021). Maneuvering with safety guarantees using control barrier functions. In *Proc. IFAC Conf. Contr. Appl. Marine Systems*, volume 54. Oldenburg, Germany.

Marley, M., Skjetne, R., Breivik, M., and Fleischer, C. (2020). A hybrid kinematic controller for resilient obstacle avoidance of autonomous ships. *IOP Conference Series: Materials Science and Engineering*, 929(1), 012022.

Rackauckas, C., Innes, M., Ma, Y., Bettencourt, J., White, L., and Dixit, V. (2019). Diffeqflux.jl - A julia library for neural differential equations. *CoRR*, abs/1902.02376.

Rackauckas, C. and Nie, Q. (2017). DifferentialEquations.jl - A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia. *Journal of Open Research Software*, 5(1), 15.

Ramachandran, P., Zoph, B., and Le, Q.V. (2017). Searching for Activation Functions. ArXiv:1710.05941 [cs].

Skjetne, R., Fossen, T.I., and Kokotović, P.V. (2004). Robust output maneuvering for a class of nonlinear systems. *Automatica*.

Skjetne, R., Jørgensen, U., and Teel, A.R. (2011). Line-of-sight path-following along regularly parametrized curves solved as a generic maneuvering problem. In *IEEE Conference on Decision and Control and European Control Conference*, 2467–2474. IEEE, Orlando, FL, USA.

Tengesdal, T., Johansen, T.A., and Brekke, E.F. (2022). Ship Collision Avoidance Utilizing the Cross-Entropy Method for Collision Risk Assessment. *IEEE Transactions on Intelligent Transportation Systems*.

Thyri, E.H., Basso, E.A., Breivik, M., Pettersen, K.Y., Skjetne, R., and Lekkas, A.M. (2020). Reactive collision avoidance for ASVs based on control barrier functions. In *Proc. 4th IEEE Conf. Control Technol. and Applications*. Montreal, Canada.