A Novel Fast Marching Approach for COLREGS Compliant Dynamic Obstacle Avoidance for Unmanned Surface Vehicles

R. W. Oude Grote Bevelsborg

A Novel Fast Marching Approach for COLREGS Compliant Dynamic Obstacle Avoidance for Unmanned Surface Vehicles

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

by

R. W. Oude Grote Bevelsborg

March 6, 2019

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
DELFT CENTER FOR SYSTEMS AND CONTROL (DCSC)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

A NOVEL FAST MARCHING APPROACH FOR COLREGS COMPLIANT DYNAMIC OBSTACLE AVOIDANCE FOR UNMANNED SURFACE VEHICLES

by

R.W. OUDE GROTE BEVELSBORG

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE SYSTEMS AND CONTROL

Dated: March 6, 2019

Supervisor(s):

dr. G. Giordano

Reader(s):

dr. ir. A.J.J. van den Boom

dr. M. Kok

ABSTRACT

Successful implementation of fully autonomous vehicles is a much desired objective. Within the area of unmanned marine craft or unmanned surface vehicles (USV), the implementation of truly autonomous navigation is still a challenge. During times in which humans are still actively involved in the process of maritime navigation, regulations are needed to prevent accidents from happening. It is therefore essential for the transition towards more autonomy that USVs are able to obey the regulations as well and clearly exhibit the right behavior to enable other ships to anticipate the behavior of USVs and react appropriately and timely. In maritime navigation, ships have to obey the International Regulations for preventing collisions at sea, also known as *COLREGS* (COLlision REGulationS).

This thesis is focused on the guidance of USVs, and is specifically aimed at COLREGS compliant path re-planning to avoid potential collisions. A first method has been conceived based upon the Saturated Fast Marching Square method where trailing points are used to correctly guide the USV past dynamic vessels according to COLREGS. A second novel method builds upon the previous one, by combining the Saturated Fast Marching Square method with a path generating Genetic Algorithm. The first method performs as supposed in simple environments such as when encountering a single dynamic obstacle. However, the first method starts to under-perform as complexity increases through the presence of both static and dynamic obstacles. The additional second method performs accurately in complex scenarios as well. Both methods have been tested in single case simulations and in extensive random simulations to assess their COLREGS compliant path re-planning accuracy.

Contents

List of Figures

List of Tables

Glossary

USV	Unmanned Autonomous Vehicle
COLREGS	COLlision REGulationS
FMM	Fast Marching Method
SFMS	Saturated Fast Marching Square
EA	Evolutionary Algorithm
GA	Genetic Algorithm
AI	Artificial Intelligence
MPC	Model Predictive Control

1 Introduction

The last few decades have seen a considerable spike in the interest of automating processes, supporting human activity with the use of robots or even substituting human involvement entirely. A major area within the broader context that has gained a great amount of attention deals with autonomous robots. Autonomous robots can be seen as intelligent agents that have both sensors and actuators to interact with the physical world around them in combination with the processing power to determine the right course of action. To operate in dynamic environments, autonomous robots must be able to effectively utilize and coordinate their limited physical and computational resources [50]. In a world that is subject to constant change, developing fully autonomous robots is an enormously difficult but popular challenge yet to be fully tackled. One of the fields of endeavor addresses autonomous Unmanned Surface Vehicles (USVs) or autonomous marine craft that can operate on oceans, seas, rivers and in shallow waters. Operations that can be executed by USVs range from shallow water surveying [37], environmental data gathering [39], surveillance [11] to concepts such as an autonomous river cleaning robot [51].

An autonomous USV needs to deal with a lot of issues to truly become fully autonomous in practice. Important subjects are the determination of an appropriate path for the USV to follow, the detection of obstacles, both static ones such as buoys and dynamic ones like surrounding vessels, and the actual avoidance of vessels during encounters. There may come a time when all the vessels operate autonomously and can communicate with one another easily. In this scenario, it is a logical choice to ensure that all USVs pursue optimal paths. However, when humans are still involved in the process of maritime navigation, regulations are needed to prevent accidents from happening. It is therefore essential for the transition towards more autonomy that USVs are able to comply with all the existing regulations.

In maritime navigation, ships should obey the International Regulations for preventing collisions at sea, also known as *COLREGS* (COLlision REGulationS). It is of importance for autonomous USVs to obey the COLREGS in order to operate safely in real environments and let other ships anticipate their behaviour and react appropriately. Section 3.3 will further elaborate on the specifics of the COLREGS regulations which are a central theme for this thesis.

To realize a fully autonomous USV, it is necessary to deal with three fundamental elements, namely: navigation, guidance and control. These three elements can also be viewed as subsystems that need to work in interaction with one another in order to manoeuvre safely. Imperfections in any of the subsystems may degrade the performance of the entire system [43]. The interaction between these 3 pillars is illustrated in Figure 1.1.

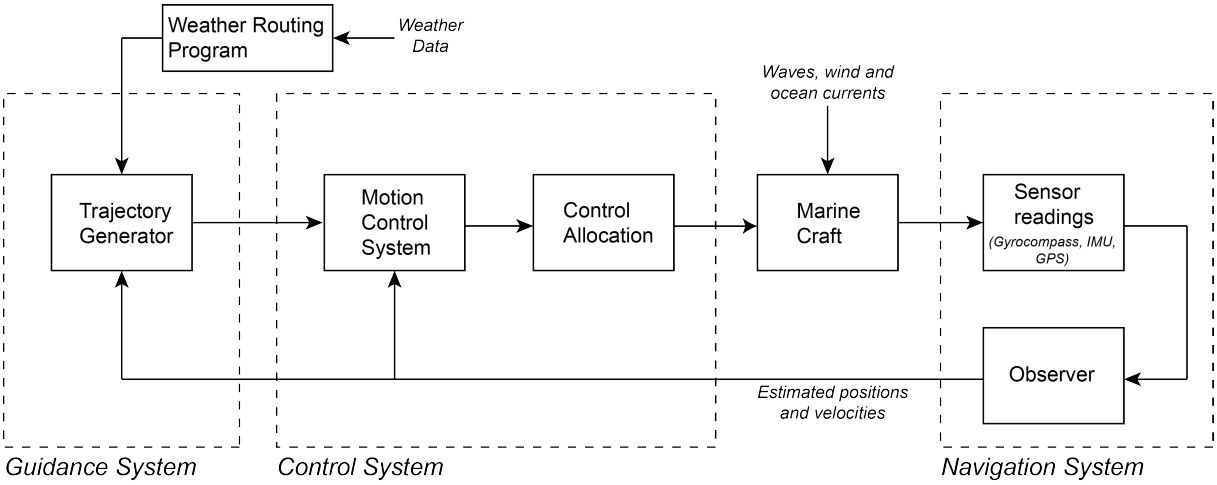


Figure 1.1: The Guidance, Navigation, Control (GNC) workflow [18]

1. *The Navigation System* is responsible for the identification of the current and future states of the USV (such as the position, orientation, velocity and acceleration). Furthermore, it has to detect its surrounding environment based on past and current states in combination with other environmental information obtained from on-board sensors (such as currents and wind speeds).

2. *The Guidance System* focuses on continuously generating and updating optimal, feasible and smooth trajectory commands that can be implemented by the control system based on the information provided by the navigation system, objectives, vehicle capabilities and environmental conditions.

3. *The Control System* concentrates on determining proper control forces and moments to be generated in accordance with the orders provided by both the navigation and guidance subsystems whilst also satisfying control objectives.

Significant developments in all three subsystems, both individually and as a whole, have been accomplished. However, in the autonomous surface vehicle development, the combined effort of path planning and correct collision avoidance remains a largely unsolved challenge. A viable guidance system that is compliant with the regulations is essential for increasing the autonomy of USVs.

1.1. Contribution

In this thesis, a novel guidance system is proposed that incorporates the COLREGS regulations into the path determination. Two methods have been conceived, based on the principles of the global path planning algorithm of the Fast Marching Method (FMM). The FMM has been chosen as starting foundation due to its positive performance in global path planning. The first method is entirely based upon the principles of the Fast Marching Method whereas the second method uses a promising combination of the Fast Marching Method and a Genetic Algorithm to determine the appropriate COLREGS compliant path.

1.2. Thesis Overview

In this chapter the problem of COLREGS-compliant path re-planning was introduced and motivated. The remaining report is organized as follows.

An extensive overview of the recent literature and state-of-the-art methods concerning the development of the autonomous USV is discussed in Chapter 2.

The theoretical background forming the basis of this thesis is scrutinized in Chapter 3, dealing with the USV model, the control system used and a detailed explanation of the Fast Marching Method.

Chapter 4 presents the first method for path re-planning such that the newly generated path is COLREGS compliant. It handles collision detection, COLREGS situation classification and the path re-planning based upon the Saturated Fast Marching Square method with trailing points as newly adopted source points.

Chapter 5 continues with a newly proposed method to improve on the previous one. The second method combines the Saturated Fast Marching Square method with a path-generating Genetic Algorithm. The two techniques are suitably integrated to complement each other.

Both methods are extensively studied through testing them in several single case studies in Chapter 6, based on potential collisions with one or more dynamic obstacles. Furthermore, plain environments are used as well as environments filled with static obstacles.

Lastly, the performance of the two methods is statistically examined in Chapter 7 by simulating a lot of randomly picked collision situations. The analysis and comparison of the two methods reveals that both methods act accordingly in simple situations. Nonetheless, as complexity increases the second method outperforms the first.

2

Literature Review

Extensive research has already been performed within the area of autonomous maritime navigation, focusing on the earlier mentioned subareas of Navigation, Guidance and Control. This chapter reviews several of the main contributions to the subareas of autonomous maritime navigation in recent times, with a focus on the Guidance subsystem. Specifically, the highlighted topics are the effective path re-planning in the case of a potential collision with dynamic obstacles in compliance to the regulations.

2.1. Navigation

Most of the research that has been done in the past focuses on further development of one specific subsystem. Within the area of navigation, detection of the surrounding environment is done using sensors such as vision based cameras [44] or stereo vision [21, 52] (creating a depth map by using images produced from cameras at different view points). But environment perception can also be achieved through the use of more active sensors such as LIDAR [19], radar [25, 57] or sonar [8, 13].

Information about the position and orientation is conventionally received through the Global Positioning System (GPS) and Inertial Measurement Unit (IMU). Sensor data can be imprecise and therefore corrective actions are needed. Several potential disadvantages are high noise, low precision, inadequate information, accumulated errors from inherent drift, time-varying model uncertainties and sensor faults [43, 63]. The Kalman Filter is an often used and effective way to estimate terms such as velocity and to eliminate noise and accumulative error. The Kalman Filter can even be used for predicting trajectories of moving ships in real time [31].

2.2. Guidance

A viable guidance system is essential for increasing the autonomy of USVs. Path planning can be seen as the fundamental aspect of an USV guidance system. One can generally distinguish between global and local path planning. The Global Path Planner identifies a free path from the current position to the destination whereas the Local Path Planner has the objective to avoid dynamic obstacles and re-route so as to avoid a potential collision [43]. There used to be a clear distinction between algorithms that function well as global path planners and as local path planners. The main reason was that the global path planning was computationally intense optimization to find the optimal path whereas the local planner was used in a more reactive fashion. However, increasingly, more algorithms can handle both at the same time.

More reactive local path planners have also been studied for USV guidance. A frequently used technique is the so-called method of Artificial Potential Fields, an elegant but simple method to generate a path. Basically, the robot is modeled as a particle under the influence of an artificial potential field which superimposes both repulsive forces from obstacles and attractive forces from the goal. However, one of the main risk is that the traditional approach can quite easily get stuck in local minima, rendering it ineffective as global planner [33]. But in the case of nearby obstacles, rerouting is easy to implement through the method of Artificial Potential Fields. There have been various modifications to the traditional method leading to the new modified Artificial Potential Fields, focusing on avoiding local minima or so-called deadlock situations [60, 61] and circumventing dynamic obstacles [34, 60].

Advanced optimization methods, used for global path planning, can produce optimal trajectories that might even incorporate danger levels, weather routing and energy saving. Evolutionary Algorithms (EA) represent a class of Artificial

Intelligence (AI) increasingly employed for global path planning, where the subclass of Genetic Algorithms (GA) is the most widely adopted method [3, 22]. Other popular heuristic search methods that have been used for USVs include algorithms such as the Dijkstra algorithm [42] and the A* algorithm [62]. These are however focused on implementing a good global path to avoid static obstacles and are not concerned with the avoidance of dynamic obstacles.

The Fast Marching Method is an advanced global path planner that has become popular due to its ability to always find the global optimal path, whenever it exists. The algorithm is similar to Dijkstra's algorithm. The thesis partially builds upon the Fast Marching Method and the in-depth theory behind it is explained in Section 3.4. The Traditional Fast Marching Method (FMM) has the problem that the generated optimal path is usually too close to obstacles. This drawback makes it impractical for guidance of USVs as the immediate surroundings of these obstacles can be dangerous (e.g. shallow waters) [32]. To solve this issue, Liu et al. [32] proposed the Fast Marching Square Method (FMS), generating a path with an increasing safety margin, but at the sacrifice of total distance cost. The method was however only used for static obstacles. Furthermore, it provides a real case study on the novel method of the Angle Guidance Fast Marching (AGFM) method [30]. The AGFM generates a path compliant with USV dynamics and orientation restrictions as a USV is often under-actuated and restricted by various motion constraints. It mentions the importance of implementing COLREGS constraints for path planning but it is not developed.

On the contrary, Sun et al. [53] successfully implemented it for guidance on USVs such that it also steers clear of any encountering dynamic obstacle. Liu et al. [31] introduced a new predictive path planning algorithm, integrating the Fast Marching Method (FMM) and a Kalman Filter (KF) algorithm. The FMM generates an optimized path and the KF algorithm is used to predict the future locations of the USV and dynamic obstacles. The KF algorithm can detect collision risks and the joint algorithm can then reactively generate a new path. A variable two-dimensional Gaussian distribution is employed to model the dynamic safety area. Collision avoidance is achieved but not by obeying COLREGS. Sun et al. [53] and Liu et al. [31] recommend that incorporation of COLREGS is an important objective to fulfill for practicability. The Fast Marching Method has significant potential as a global planner and it has potential to be used for COLREGS compliant re-planning as well.

2.2.1. COLREGS compliancy

Campbell et al. [10] highlight that most of the guidance systems do not implement the COLREGS regulations satisfactorily in complex scenarios. Polvara et al. [43] mention that most path-planning algorithms ignore the COLREGS despite its mandatory nature in open waters.

Several methods have been studied to also implement path re-planning such that the USV obeys COLREGS regulations in case of a potential collision. Deterministic algorithms such as the Artificial Potential Field [34–36, 40, 59], the Virtual Force Field [58] and tree search methods such as A* searching [1, 9, 16] and Rapidly Exploring Random Trees [14] have been used in order to avoid dynamic obstacles in a COLREGS compliant manner. The modified versions of Artificial Potential Fields and Virtual Force Field framework actually try to incorporate COLREGS compliant collision avoidance by using repulsive regions around the dynamic vessels shaped to guide the USV around it accordingly. However, the reactive nature of the Artificial Potential Fields method and the Virtual Force Field method often results in jagged or exaggerated behavior, making it less attractive as a practical guidance algorithm. Furthermore, the methods can have significant trouble in finding an appropriate path in more cluttered environments.

Lee et al. [27] combined the Virtual Force Field framework with Fuzzy Logic to make sure it can respond according to COLREGS regulations. The complete system involved around two hundred fuzzy rules with four linguistic variables posing quite a computational challenge.

The A* algorithm performs a so called best-first search of the most probable paths towards the goal through a cost function. Campbell and Naeem [9] developed a heuristic Rule-based Repairing A* (R-RA*) algorithm. The decisions are based upon the relative states of the approaching vessels, where specific regions on the map are heuristically punished, such as the starboard side of a upcoming vessel in a head-on collision. Agrawal and Dolan [1] proposed a grid-based map incorporating A* path planning. The COLREGS compliancy is enforced through using Artificial Terrain Costs which similarly resembles a cost for crossing distinct areas.

Naeem et al. [41] proposed a method based on a Line-Of-Sight guidance technique coupled with a simple manual biasing scheme applied to the dynamic model of an unmanned surface vehicle for obeying COLREGS. The manual biasing scheme is based on the notion that the current heading angle of the USV is altered towards the starboard side in case the relative distance is too small. Furthermore, the USV can never enter the safe zone of other dynamic obstacles. A PID autopilot is used for following the generated paths.

The Velocity Obstacle (VO) method is widely used as a local path planning algorithm as well. The method constructs a

velocity space based on a set of relative velocities between the USV itself and the target vessels. A velocity that avoids collision is chosen guaranteeing a collision-free path. An additional cost function to implement COLREGS constraints is adopted to impose compliancy for the Velocity Obstacle (VO) method [6, 24, 48]. The major issue with the method is that continuously changing the velocity variable is undesirable.

The class of Evolutionary Algorithms have also been used for obeying COLREGS as well. Hu et al. [20] developed a multi-objective Particle Swarm Optimization and Lazarowska [26] a multi-criteria Ant Colony Optimization. Both studies analyse potential paths by criteria such as the path length and smoothness. Paths that go starboard are punished to try to make the USV obey the COLREGS regulations. A Genetic Algorithm is proposed by Tsou et al. [55] as assistant to provide evasive maneuvers to the human operator in case of a potential collision. The paths are tested according to path length and from an economic viewpoint. It states that the method adheres to COLREGS compliancy, but details for this are not provided.

Lastly, the Fast Marching Method has once been modified to abid to COLREGS. Beser and Yildirim [7] presented an alternative strategy in case the LIDAR and radar sensors fail. The strategy is based on collision detection by checking bearing through visual sensors (a camera). In case a collision is detected, the Fast Marching Square method is used with a temporary target point on the starboard side of the obstacle. The details of the method, its variables and explicit results are not very clear though. Only one test with a single dynamic obstacle is presented.

2.3. Control

The kinematics and dynamics of the USV are explained in depth in Section 3.1. The parameters vary per vessel and need to be determined through various tests. Most control strategies focus on the issue of course-keeping control, to maintain the reference course of the vehicle.

Although a lot of effort has already been devoted to the development of more advanced control methods, the classical PID control still dominates control system design as the industry still largely favours it, particularly for autopilot design [5]. The advanced methods that have been investigated for the application of USVs are Optimal control [17, 28, 45, 64], backstepping control [6, 23], Intelligent Control, based on fuzzy functions [12, 13] or on Neural Networks [29, 49], Robust Control [46, 56] and Sliding Mode Control [4, 54].

Optimal control strategies are concerned with finding a control law that achieves an optimality criterion for a specific system. Lefeber et al. [28] studied the complete state-tracking problem for the underactuated ship, with the inputs of the surge control force and the yaw control moments. An LQR controller was developed for the problem and it can be concluded that the tracking error was quite significant. One can also add an integrator in order to get rid of the steady-state error of the system and drive the error to zero in time. Feng et al. [17] showed how to compute a proper LQI controller for an USV. Campbell et al. [10] mentioned that an LQR controller is not very practical as it assumes that all states are known which is not realistic. A method such as the Linear Quadratic Gaussian (LQG) is preferable as it uses a Kalman Filter in cascade for estimating the unknown states. Naeem et al. [39] developed a control strategy based on a Linear Quadratic Gaussian controller. The study validated the suitability of using LQG as an autopilot, although the missions were mainly based on following a specified heading angle.

The essence of Model Predictive Control is to optimize the manipulable input variables that are used to forecast the process behavior. The forecasting is accomplished through the utilization of an explicit dynamic process model. The model is an essential element and the heavy reliance on models for MPC can also result in poor performance due to the fact that models are not necessarily good forecasters and the impact of assumptions can weigh on the accuracy. Luckily, feedback can overcome some of the effects of poor models [45]. Zheng et al. [64] developed the MPC framework and the cost function for the problem of USV course-keeping control. Annamalai and Motwani [2] simulated both the LQG and MPC controllers in order to be able to compare their relative performance as autopilots for USVs. In these simulations, it was found that the MPC clearly outperforms the LQG in accuracy. However, the MPC still has a relatively large tracking error as well, so optimal control still lacks accuracy for the USV course-keeping.

Backstepping is a recursive design technique that can be used to develop stabilizing controls for certain non-linear systems. Klinger et al. [23] and Bertaska et al. [6] developed a backstepping method for position and heading control of USVs. The backstepping control approach can incorporate practical control laws and thereby offer some robustness to disturbances associated with the actual conditions on water [23]. Furthermore, it can incorporate nonlinear terms associated with the vessel dynamics.

However, control methods such as optimal control or backstepping control are designed for a specific vessel, so their performance depends on the precise model. The parametrization of such as model is difficult to achieve and the behavior of ships deviate tremendously from one another. A simplified model such as the bicycle model is often used when the

focus is on either the Navigation and/or Guidance subsystems. A PurePursuit controller [15] can be utilized as a path-following controller, as discussed in Section 3.2.

3

Theoretical Background

3.1. The USV Model: Kinematics and Dynamics

One can identify three special axes in any ship, called the vertical, transverse and longitudinal axes. In maneuvering, a marine craft experiences 6 degrees of Freedom (DOF), consisting of three rotational movements around these aforementioned axes, known as roll, pitch and yaw and three translational movements, known as heave, sway and surge. The motions are visualized in Figure 3.1.

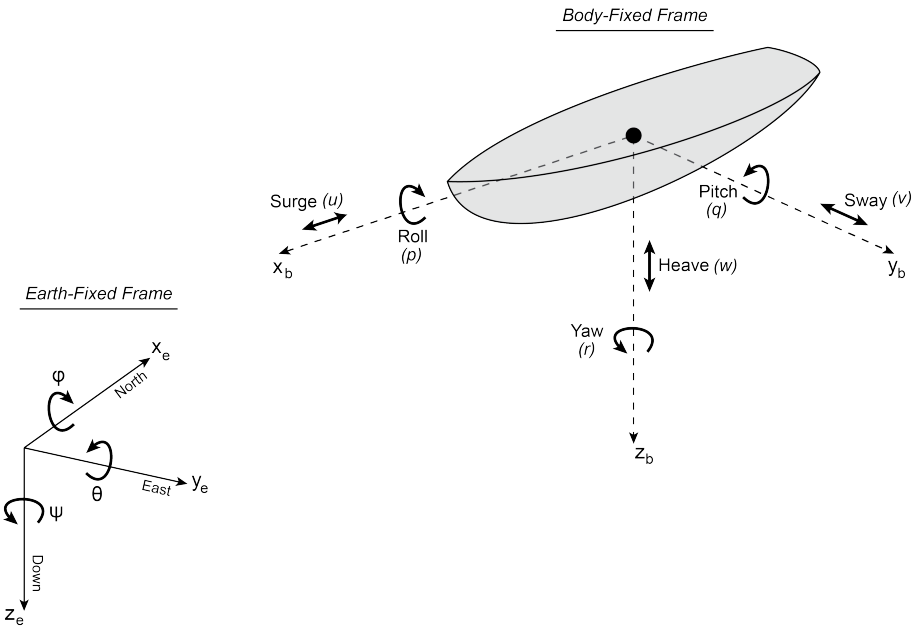


Figure 3.1: The 6-DOF motions of a ship

Two different frames of reference are used to describe the model. First, an Earth-Fixed reference frame $\{x_e, y_e, z_e\}$ based on the North East Down (NED) coordinate system. Second, a body-fixed reference frame $\{x_b, y_b, z_b\}$ is used, which is fixed to the craft. The body axes coincide with the principle axes. The vector for the positions and Euler angles is formulated in the earth-fixed frame as $\boldsymbol{\eta} = [x, y, z, \phi, \theta, \psi]^T$ and the vector of velocities is formulated in the body-fixed frame as $\mathbf{v} = [u, v, w, p, q, r]^T$. Fossen [18] formulated a kinematic and dynamic framework for marine craft. The complete 6 DOF model for Marine craft can be represented in the form of:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \tag{3.1}$$

where $\boldsymbol{\tau}$ is a vector of forces and moments on the USV and the model matrices $\mathbf{M}$, $\mathbf{C}(\mathbf{v})$ and $\mathbf{D}(\mathbf{v})$ denote respectively inertia, Coriolis and damping. The vector $\mathbf{g}(\boldsymbol{\eta})$ represents the generalized gravitational and buoyancy forces.

The model surface vessels can be reduced to a 3 DOF representation for simplification, based on the assumption that θ and ϕ are small, which is valid for most conventional ships. The motions of interest are of surge, sway and heading or yaw

while the roll, pitch and heave can be neglected.

The position coordinates and heading in the local earth-fixed frame are represented by $\boldsymbol{\eta} = [x, y, \psi]^T$ where x is the North position, y is the East position and ψ is the heading angle. The system states are $\mathbf{v} = [u, v, r]^T$ denoting respectively the surge velocity, the sway velocity and the yaw rate of the USV in the body-fixed frame. The rotation matrix from body-fixed to earth-fixed frame captured by:

$$\mathbf{R}(\boldsymbol{\psi}) = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.2}$$

The velocity can be rewritten to the local earth-fixed frame:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\boldsymbol{\psi})\mathbf{v} \tag{3.3}$$

The maneuvering model consisting of 3 DOF is based on:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \tag{3.4}$$

where $\mathbf{C}(\mathbf{v}) = \mathbf{C}_{RB}(\mathbf{v}) + \mathbf{C}_A(\mathbf{v})$ is the summation of the rigid body and added mass Coriolis centripetal terms. $\mathbf{D}(\mathbf{v}) = \mathbf{D}_L + \mathbf{D}_n(\mathbf{v})$ is a summation of the linear and nonlinear damping matrix which corresponds to the drag on the vessel.
$\mathbf{M}$ denotes the combined matrix of vessel rigid-body inertia and added mass:

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \tag{3.5}$$

It is assumed that the USV has a homogeneous mass distribution and xz-plane symmetry so

$$I_{xy} = I_{yz} = 0 \tag{3.6}$$

Furthermore, Coriolis and centripetal terms are covered by the following matrices:

$$\mathbf{C}_{RB}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & -m(y_g r - u) \\ m(x_g r + v) & m(y_g r - u) & 0 \end{bmatrix} \tag{3.7}$$

$$\mathbf{C}_A(\mathbf{v}) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}} v + \frac{Y_{\dot{r}} + N_{\dot{\psi}}}{2} r \\ 0 & 0 & -X_{\dot{u}} u \\ -(Y_{\dot{v}} v + \frac{Y_{\dot{r}} + N_{\dot{\psi}}}{2} r) & X_{\dot{u}} u_r & 0 \end{bmatrix} \tag{3.8}$$

The damping matrix $\mathbf{D} = \mathbf{D}_L + \mathbf{D}_n(\mathbf{v})$ corresponds to the drag on the USV, subdivided in linear

$$\mathbf{D}_L = \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix} \tag{3.9}$$

and non-linear terms

$$\mathbf{D}_n(\mathbf{v}) = \begin{bmatrix} -X_{u|u}|u| & 0 & 0 \\ 0 & -Y_{v|v}|v_r| - Y_{r|v}|r| & -Y_{v|r}|v_r| - Y_{r|r}|r| \\ 0 & -N_{v|v}|v_r| - N_{r|v}|r| & -N_{v|r}|v_r| - N_{r|r}|r| \end{bmatrix} \tag{3.10}$$

The generalized force and moments vector $\boldsymbol{\tau} = [\tau_x \tau_y \tau_z]^T$ denotes the actuator forces. These are usually the propeller force and the rudder angle, making the USV underactuated.

3.1.1. State-Space representation

The dynamics of a USV are non-linear but there are several ways to reach a linear state-space representation. Several assumptions are made such as that the speeds are low, the relative velocity $\mathbf{v}_r$ can be ignored as ocean currents are treated as a minor disturbance and can be compensated for by integral action. The states are described by $\mathbf{x} = [\boldsymbol{\eta}^T, \mathbf{v}^T]^T$, the input is $\mathbf{u} = \boldsymbol{\tau}$, which can consist of the thrust of the motors and the rudder angle and the disturbances (e.g. waves and wind) are captured by $\boldsymbol{\tau}_{disturb}$. The output is measured in the form of the position and heading angle.

The state-space system now becomes:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\boldsymbol{\tau}_{disturb} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \tag{3.11}$$

with the matrices:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{0} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{E} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix} \tag{3.12}$$

The different parameters have to be obtained through system identification and the system (3.11) has to be linearized around certain velocities.

The focus of the thesis is on correct path re-planning in the event of a potential collision. The dynamic models of vessels vary per vessel and the variables have to be measured or estimated. Therefore, a simpler path tracking controller, the Pure Pursuit Controller, is adopted for path tracking.

3.2. Pure Pursuit Control

The pure pursuit method [15] is a popular method, serving as a path tracking scheme by following waypoints along the path. The essential aspect of the algorithm is to choose a goal point some distance ahead of the USV on the path. The pure pursuit approach was originally designed to serve as a method to geometrically determine the curvature that drives the vehicle towards a chosen path point. The pose of the USV is known and consists of its coordinates (x, y) and its current orientation or heading ψ. A constant look-ahead distance is utilized to determine the look-ahead point (x_{LA}, y_{LA}), which acts as a tracking point for the USV. An illustration of the geometry is shown in Figure 3.2.

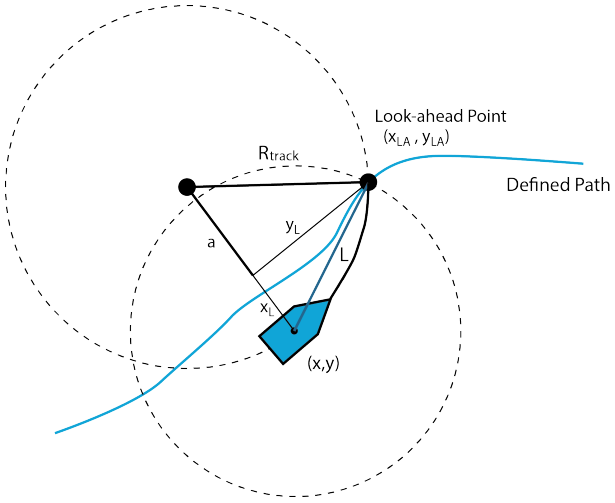


Figure 3.2: The geometric overview for the PurePursuit algorithm

Based on the illustration, the following equations hold:

$$\begin{aligned} x_L^2 + y_L^2 &= L^2 \\ a^2 + y_L^2 &= R_{track}^2 \\ x_L + a &= R_{track} \end{aligned} \tag{3.13}$$

Now the equations can be rewritten into

$$\begin{aligned} (R_{track} - x_L)^2 + y_L^2 &= R_{track}^2 \\ R_{track}^2 - 2R_{track}x_L + L^2 &= R_{track}^2 \end{aligned} \tag{3.14}$$

Now R_{track} can be rewritten into

$$R_{track} = \frac{L^2}{2x_L} \tag{3.15}$$

and the curvature becomes

$$\rho = \frac{1}{R_{track}} \tag{3.16}$$

The algorithm is iterated at every cycle so the actual USV pose is updated, a new look-ahead point is computed and the new arc radius is calculated again. Because of the circular nature, the angular velocity can now be determined through the curvature in combination with the constant linear velocity v

$$\dot{\psi} = \rho v \tag{3.17}$$

The above values are iteratively used to update the pose of the USV as follows

$$\begin{aligned} x(k+1) &= x(k) + v \cdot \cos\psi(k) \\ y(k+1) &= y(k) + v \cdot \sin\psi(k) \\ \psi(k+1) &= \psi(k) + \dot{\psi}(k) \end{aligned} \tag{3.18}$$

The PurePursuit algorithm does not incorporate USV dynamics. It is merely adopted to follow the paths as the focus is on appropriate path re-planning.

3.3. COLREGS

These rules can be encoded within the motion control system as constraints to regulate the behavior. There are three main situations that are explicitly considered in this thesis as these are of high significance for collision avoidance: crossing, overtaking and head-on situations. The responses are shown in Figure 3.3.

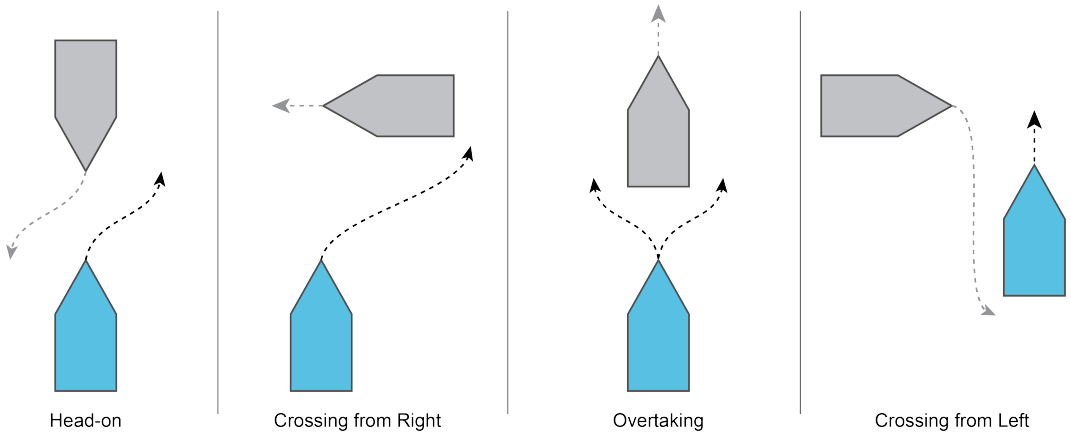


Figure 3.3: The international rules of COLREGS in specific situations

In the case of a *Head-on* situation, both vehicles should change course so that they will pass with the other vessel on their port side.

Crossing from right implies that the USV has to give-way in order to let the other vessel pass.

Overtaking is allowed on both sides of the obstacle although the USV should keep sufficient distance.

Crossing from left gives the USV the right of way and the other vessel has to give-way. However, when the other vessel does not abide the rules, collision should be avoided at all cost.

3.4. The Fast Marching Method

Sethian [47] introduced the Fast Marching Method (FMM) which can be used as a grid searching algorithm for path planning. The algorithm is similar to Dijkstra's algorithm and information only flows outward from the starting area. It is based on the idea of simulating wave propagation through iteratively solving the Eikonal equation

$$|\nabla T(x,y)| \cdot V(x,y) = 1 \tag{3.19}$$

where $(x, y) \in \Omega$ are coordinates within the map $\Omega \subset \mathbb{R}^2$. The time required for the wave to reach the point $(x, y) \in \Omega$ is denoted by $T(x, y)$ and the speed of the wavefront at each point is $V(x, y)$. The objective of the FMM is to calculate the time needed for the propagating wave to reach the individual points on the map. The source point from which the wave originates is conditioned as $T(x_0, y_0) = 0$.

The Fast Marching Method is a special case of the Hamilton Jacobi PDE which can not be solved analytically. The problem is discretized by means of a grid map. An intuitive method of approximating the solution is to compute the time values $T(x, y)$ at the individual grid nodes in the sequence in which the wavefront passes through these grid nodes. The values at each particular grid node are based on the values at the neighboring grid nodes similar to the time dependency of a point as the wavefront crosses to previously crossed points in the direction from which the wavefront emanated. The discretized neighboring points are illustrated in Figure 3.4.

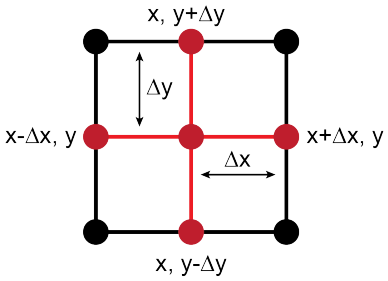


Figure 3.4: The discretized neighboring points

The Eikonal equation (3.19) is discretized using an ordered upwind difference scheme to approximate partial derivatives. The discretized one-sided partial difference operators are computed by

$$\begin{aligned} D^{-x}_{(x,y)} &= \frac{T_{(x,y)} - T_{(x-\Delta x,y)}}{\Delta x} \\ D^{+x}_{(x,y)} &= \frac{T_{(x+\Delta x,y)} - T_{(x,y)}}{\Delta x} \\ D^{-y}_{(x,y)} &= \frac{T_{(x,y)} - T_{(x,y-\Delta y)}}{\Delta y} \\ D^{+y}_{(x,y)} &= \frac{T_{(x,y+\Delta y)} - T_{(x,y)}}{\Delta y} \end{aligned} \tag{3.20}$$

The discretization of the gradient $\nabla T(x, y)$ can be rewritten into

$$\max\left(D^{-x}_{(x,y)} T, 0\right)^2 + \min\left(D^{+x}_{(x,y)} T, 0\right)^2 + \max\left(D^{-y}_{(x,y)} T, 0\right)^2 + \min\left(D^{+y}_{(x,y)} T, 0\right)^2 = \frac{1}{V^2_{(x,y)}} \tag{3.21}$$

A simpler but less accurate solution is

$$\max\left(D^{-x}_{(x,y)} T, -D^{+x}_{(x,y)} T, 0\right)^2 + \max\left(D^{-y}_{(x,y)} T, -D^{+y}_{(x,y)} T, 0\right)^2 = \frac{1}{V^2_{(x,y)}} \tag{3.22}$$

The minimum time it takes to reach the neighbouring discretized points as shown in Figure 3.4 can be captured by

$$\begin{aligned} T &= T_{(x,y)} \\ T_1 &= \min\left(T_{(x-\Delta x,y)}, T_{(x+\Delta x,y)}\right) \\ T_2 &= \min\left(T_{(x,y-\Delta y)}, T_{(x,y+\Delta y)}\right) \end{aligned} \tag{3.23}$$

By substituting (3.20) into (3.22) and using (3.23), the Eikonal equation can be rewritten for discrete 2D spaces as

$$\max\left(\frac{T - T_1}{\Delta x}, 0\right)^2 + \max\left(\frac{T - T_2}{\Delta y}, 0\right)^2 = \frac{1}{V^2_{(x,y)}} \tag{3.24}$$

Equation (3.24) can consequently be solved as follows:

$$\frac{1}{V_{(x,y)}} = \begin{cases} \sqrt{\frac{T-T_1}{\Delta x}^2 + \frac{T-T_2}{\Delta y}^2} & T \leq \max(T_1, T_2) \\ \frac{T-T_1}{\Delta x} & T_2 \leq T \leq T_1 \\ \frac{T-T_2}{\Delta y} & T_1 \leq T \leq T_2 \end{cases} \tag{3.25}$$

The wavefront propagates iteratively from the sourcepoint through all the grid nodes, hence providing the solution of Equation (3.25) for every single node. There are three labels to categorize the state of each node:

- *Far* - the nodes that the wave has not yet reached so the value of T is not known yet
- *Narrow Band* - consists of candidate nodes to be part of the front wave in the next iteration. These are assigned a value T that can still change in future iterations of the algorithm
- *Frozen* - Nodes that have already been processed by the wavefront and thus their value of T is fixed

Initially, all nodes in the grid-map belong to the *Far* set with infinite arrival time. The initial starting points or the wave sources are assigned a value $T(x_0, y_0) = 0$ and labeled *Frozen*. The non-frozen neighbors are adopted in the *Narrow Band*. The main FMM loop starts by choosing the node from the *Narrow Band* with minimum arrival time. All its non-frozen neighbors are evaluated: the Eikonal is solved and the new arrival time value is kept if it is improved. In case the node is in *Far*, it is transferred to the *Narrow Band*. Finally, the previously chosen point from *Narrow Band* is transferred to *Frozen* and the process will iterate until the *Narrow Band* is empty.

Figure 3.5 visualizes the Fast Marching Method process where the black nodes have been processed so *Frozen*, the orange nodes are in the *Narrow Band* and the white nodes are the *Far Values* that are not yet known.

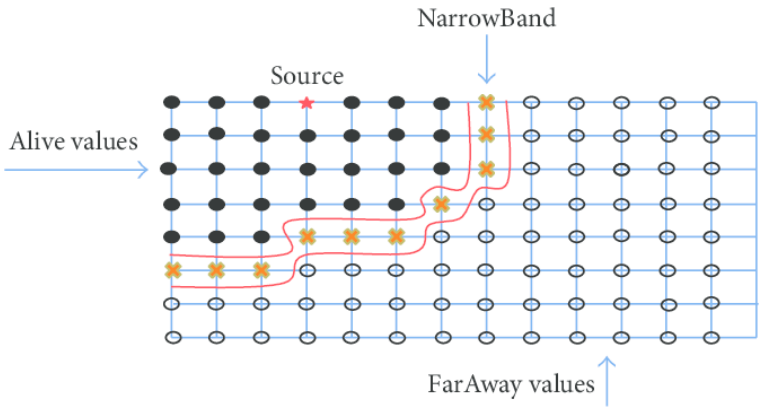


Figure 3.5: Visualization of the Fast Marching Method [38]

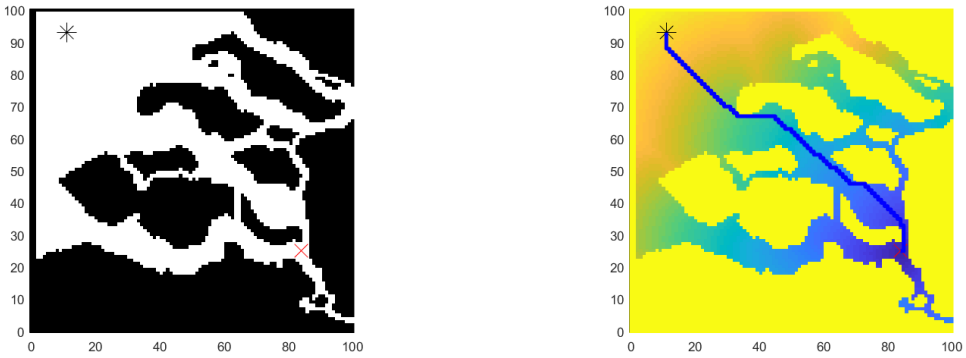
After the wave has propagated through the entire grid-map, a time-of-arrival map $\mathcal{T}$ is obtained. The optimal path from any goal-point on the grid-map to the starting-point of the wavefront can be determined through gradient descent, given that the path exists. Moreover, the Fast Marching Method guarantees that the path computed by gradient descent is optimal as every node has the lowest possible value of T assigned.

3.5. Saturated Fast Marching Square

Liu et al. [32] mention that using the Fast Marching Method directly for path planning generates an optimal path that is too close to obstacles. This drawback makes it impractical for guidance of USVs as the immediate surroundings of these obstacles can be dangerous (e.g. shallow waters).

The Saturated Fast Marching Square method (SFMS) solves this issue. Basically, the concept behind saturated Fast Marching Square method is to execute the traditional Fast Marching Method algorithm (FMM) twice but with different purposes. First, the FMM is applied to propagate waves from all nodes representing obstacle areas. This generates a time-of-arrival map $\mathcal{T}_1$ where nodes further away from obstacles take longer to reach. The increasing distance between free nodes and obstacle nodes leads to higher time-of-arrival values $T(x, y)$, indicating safer locations. The saturation limit is used to indicate an accepted safety bound around obstacles in time-of-arrival map $\mathcal{T}_1$. The generated time-of-arrival map $\mathcal{T}_1$ is used as the input speed map $\mathcal{V}$ for the second FMM with its starting point being the goal point. The path obtained

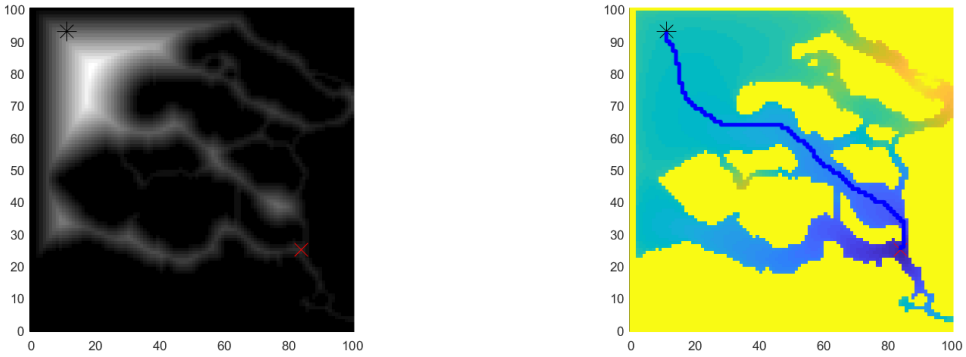
through gradient descent now has an increasing safety margin, but at the sacrifice of a longer traveling distance. The Saturated Fast Marching Square method is performed with two different saturation values $sat = 0.01$ and $sat = 0.99$. The speed map with $sat = 0.01$, shown in Figure 3.6a is equal to the normal map where black patches indicate obstacles or land and white indicates free space or water. However, with a higher saturation value such as in Figure 3.7a, the values of the speed are higher in the middle of the rivers, away from the obstacles.



(a) The saturated speed map $\mathcal{V}$ ($sat = 0.01$)

(b) The time-of-arrival map $\mathcal{T}$ ($sat = 0.01$)

Figure 3.6: The Saturated Fast Marching Square Method with saturation 0.01



(a) The saturated speed map $\mathcal{V}$ ($sat = 0.99$)

(b) The time-of-arrival map $\mathcal{T}$ ($sat = 0.99$)

Figure 3.7: The Saturated Fast Marching Square Method with saturation 0.99

The path from the USV towards the goal point (or the wavefront sourcepoint) is found iteratively as illustrated in Algorithm 1.

Algorithm 1 Compute FMM path

$(x, y) \leftarrow$ Start Point of USV
$j \leftarrow 1$
$path(1) = (x, y)$
while $(x, y) \neq SourcePoint$ **do**
  Analyse $\mathcal{T}$ of neighbours: $T(x_{new}, y_{new}) = \min\Big(T(x \pm 1, y), T(x, y \pm 1)\Big)$
  if $T(x_{new}, y_{new}) < T(x, y)$ **then**
    $(x, y) = (x_{new}, y_{new})$
    $path(j + 1) = (x_{new}, y_{new})$
    $j = j + 1$
  else
    break
  end if
end while

The Saturated Fast Marching Square method is valuable to determine a safe path in a static environment. However, incorporating accurate avoidance of dynamic obstacles such as moving ships results in a significantly more complex situation.

4

Method 1: Trailing Points

The goal of this chapter is to construct a path for the USV that reaches the goal whilst avoiding collisions with any dynamic obstacles in a COLREGS compliant manner. The algorithm can be split in three parts: 1) detecting potential collisions, 2) identifying the corresponding COLREGS scenario and 3) finding a new path which obeys the COLREGS regulations. First, the method for collision detection is clarified, followed by COLREGS classification and finally the path re-planning method is covered.

Several key assumptions have been made regarding the dynamic obstacles:

- Both the exact location and the heading direction of the dynamic obstacles is known (based on a suitable estimation process)
- In reality altering the speed of vessels is not preferable. Therefore, the velocity of both the dynamic obstacles and the USV is required to stay constant.

4.1. Collision Detection

It is safer for USVs on collision course to change course early in the process in comparison to late responses as ship rerouting is based upon anticipation. Therefore, a model has been conceived that incorporates these preferences.

A collision detection area in the form of an ellipse is adopted around both the USV and the dynamic obstacles. The collision detection area varies in shape based on the direction and velocity of the object. Figure 4.1 shows a representation of the collision detection area based upon two different speeds.

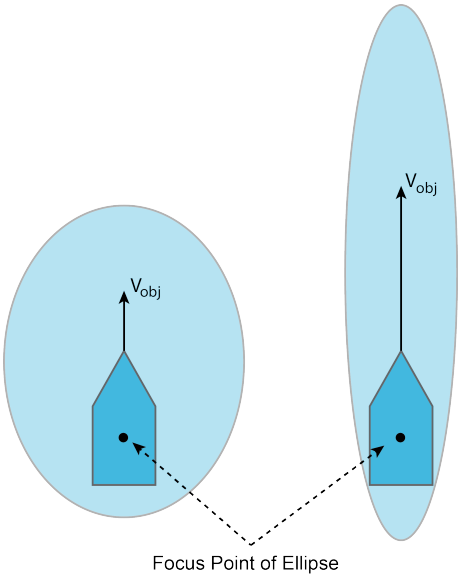


Figure 4.1: The collision detection area with varying speeds

The geometry of the ellipse is shown in Figure 4.2. The distance from the center to both foci c can be obtained by

$$c = \sqrt{a^2 - b^2} \tag{4.1}$$

The length and area of the ellipse is varied by relating a to the speed of the object

$$a = p \cdot v \tag{4.2}$$

where p is a parameter that can be determined based on the size of the speed map $\mathcal{V}$.

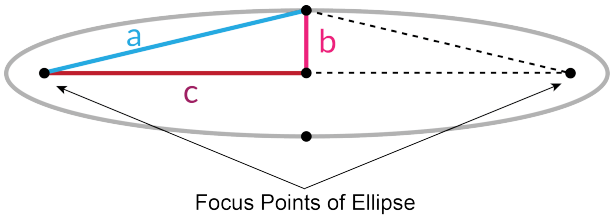


Figure 4.2: Geometric overview of the ellipse

The ellipse A_{obj} is positioned accordingly by first rotating it to coincide with the heading of the object ψ.

$$P = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} A_{obj} \tag{4.3}$$

It is then translated such that the center of the dynamic object μ_{obj} coincides with the focus point in the direction of its heading as shown in Figure 4.1.

The ellipse is transformed into a normalized distribution $\mathscr{A}$ based on the Euclidean distance. Firstly, the Euclidean distance to every point within the ellipse is computed

$$d_{A_{obj}}(\mathbf{x} \in A_{obj}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu}_{obj})^2} \tag{4.4}$$

The normalized distribution $\mathscr{A}$ can be obtained by taking the infinity norm of the Euclidean distance within the ellipse.

$$\mathscr{A}_{obj} = \frac{d_{A_{obj}}(\mathbf{x} \in A_{obj})}{\left\| d_{A_{obj}}(\mathbf{x} \in A_{obj}) \right\|_{\infty}} \tag{4.5}$$

The normalized distribution can be visualized as shown in Figure 4.3 where the values increase from 0 at the center of the dynamic object to a maximum of 1 in the red region furthest away from the center.

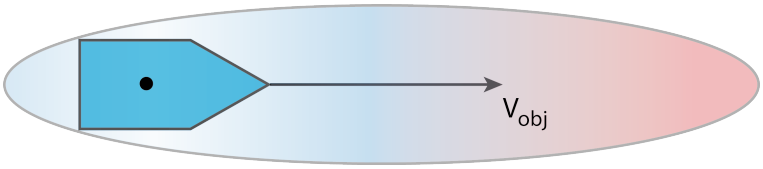


Figure 4.3: The collision detection area with varying speeds

A collision detection area is computed for all dynamic obstacles and the USV itself. To timely detect a collision between the USV and other dynamic obstacles, the product of the intersecting collision detection areas is used.

$$\mathscr{A}_{product} = \mathscr{A}_{USV}(\mathbf{x}) \cdot \mathscr{A}_{obstacle}(\mathbf{x}) \quad \text{where} \quad (\mathbf{x} \in A_{USV} \cap A_{obj}) \tag{4.6}$$

The maximum value within intersecting collision detection areas is computed and measured against a threshold.

$$\mathscr{T} = max(\mathscr{A}_{product}) \tag{4.7}$$

If $\mathscr{T}$ exceeds the threshold, the situation is marked as a collision hazard. Figure 4.4 illustrates two situations of crossing objects.

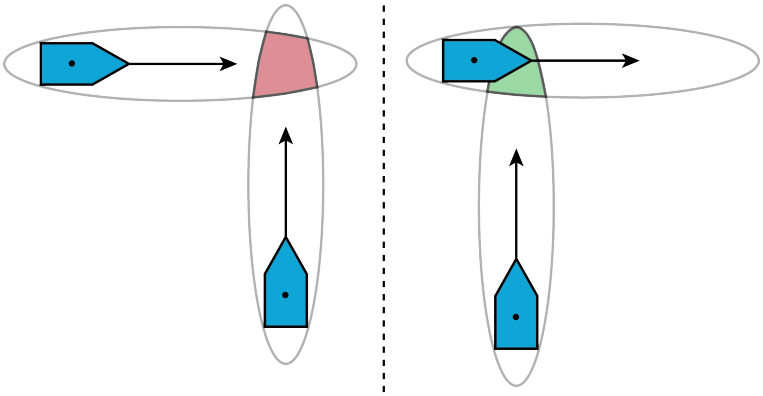


Figure 4.4: The product of intersecting collision detection areas in two different situations

The coloured zones are the product of both collision detection areas, where the red zone has a maximum value above the threshold whereas the green zone does not exceed the threshold. The proposed method has the main advantage that vessels can still cross each other when there is enough space enabling a safe crossing.

The parameters concerning the collision detection method are expressed in Table 4.1. Furthermore, various values are analyzed to test the performance.

Table 4.1: The parameters for Method 1: Collision Detection

	Parameters	value	Unit	Description
Method 1	a	$(2 \cdot v)$, $(3 \cdot v)$, $(6 \cdot v)$	m	Geometric parameter for collision detection ellipse
	b	2, 3	m	Geometric parameter for collision detection ellipse
	$\mathcal{T}$	0.5	-	An adequate threshold for collision detection

It is found that a value of $(2 \cdot v, 2)$ for the geometric representation of the ellipse works relatively well for the simulations with low speeds as described in Chapter 6.

The Fast Marching Method is based upon a discrete map. Therefore, the collision detection areas need to be discretized as well. Figure 4.5 shows a situation in which the USV could pass in front of the dynamic obstacle without high risk of collision.

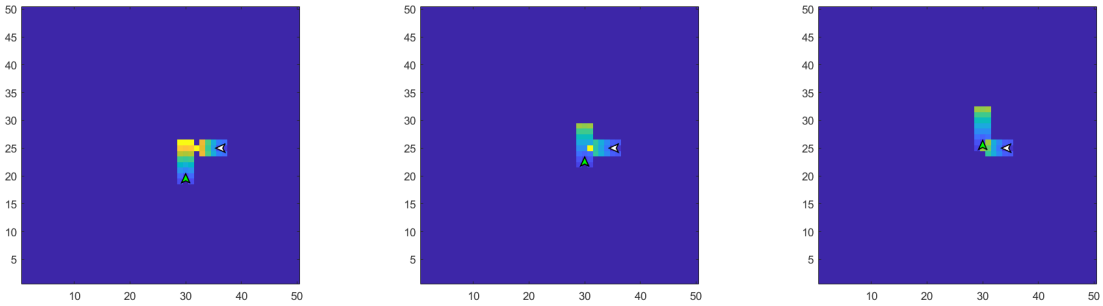


Figure 4.5: The collision detection areas in right crossing single encounter

However, when the collision detection areas cross a threshold as illustrated in Figure 4.6, the USV needs to determine the corresponding COLREGS situation and the appropriate action.

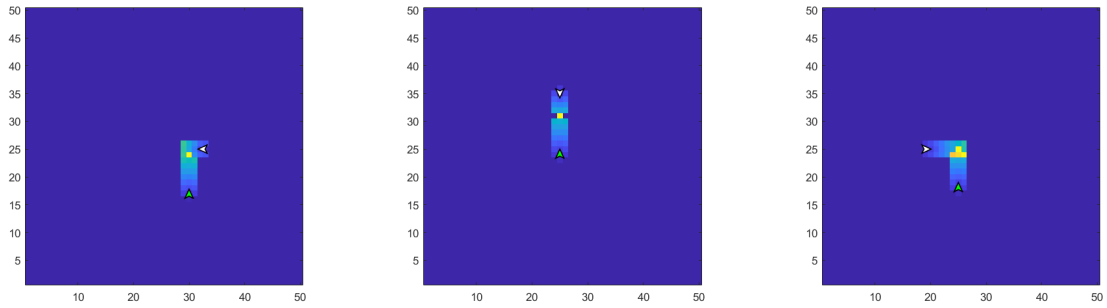


Figure 4.6: Exceeding threshold in single encounters

4.2. COLREGS Classification

The COLREGS specify different actions given specific situations. These situations as specified in Section 3.3 have to be correctly determined.

A mathematical approach is adopted to discriminate between different scenarios. As illustrated in Figure 4.7, the point (x, y) can be on the left or right side of the line between points (x_1, y_1) and (x_2, y_2).

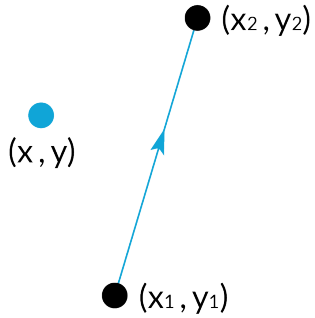


Figure 4.7: Determine if point is on left or right side of line segment

To find the correct side of the point (x, y), the cross-product can be used as follows

$$s = \frac{(x - x_1)(y_2 - y_1) - (y - y_1)(x_2 - x_1)}{\left\|(x - x_1)(y_2 - y_1) - (y - y_1)(x_2 - x_1)\right\|} \tag{4.8}$$

If the point is situated on the left side of the line $s = -1$ whereas if the point is on the right side $s = 1$.

A perimeter around the USV is used to restrict the area that needs to be observed. All the dynamic objects that enter the observable circle around the USV are detected. Furthermore, the two points: 1) center of the USV and the 2) crossing point between the planned path and the perimeter are used to form the line needed to correctly implement Equation (4.8) for collision situations, as illustrated in Figure 4.8.

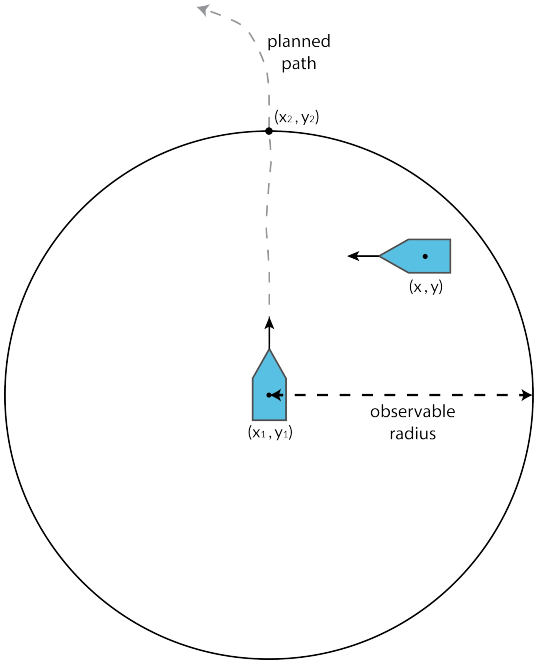


Figure 4.8: Dynamic obstacle on left or right side of initial path

The USV can now distinguish between dynamic obstacles that are present either on its right or its left side. However, the USV also needs to clearly distinguish between the specific situations of: right crossing, left crossing, head-on collision and overtake. The heading of both the USV and the approaching obstacle can be utilized to distinguish between these scenarios.

The relative heading is calculated for any approaching obstacle

$$\psi_{rel} = \left\| \psi_{USV} - \psi_{obst} \right\| \tag{4.9}$$

Moreover, the relative heading ψ_{rel} can never be more than 180°. When the relative heading is within the specific range, the collision situation is classified as a Head-On collision. The range is chosen to be between

$$160^\circ \leq \psi_{rel} \leq 180^\circ \tag{4.10}$$

as this quite accurately represents the risk zone.

For situations in which the USV has to overtake another obstacle, the range of the relative heading is set to be between

$$\psi_{rel} \leq 20^\circ \tag{4.11}$$

Figure 4.9 illustrates the distinctions for relative heading.

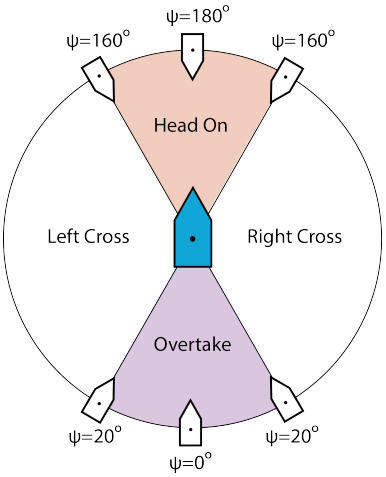


Figure 4.9: The segmentation for the relative heading

4.3. Path Re-planning

4.3.1. Obstacle Safety Region

To ensure that the USV avoids being within the vicinity of any approaching dynamic obstacle, a method has been conceived that integrates the concept behind Artificial Potential Fields within the Fast Marching Method.

Whenever a dynamic obstacle enters the observable perimeter around the USV, a circular safety region around the entering obstacle is initiated. The safety region is represented by a circular area, with a radius r_{safety}. The constructed circle represents an area of very low speed values, which is projected onto the static speed map $\mathcal{V}$. Figure 4.10 illustrates the projection with the green area representing high speed values and the red circle characterizing a low speed area.

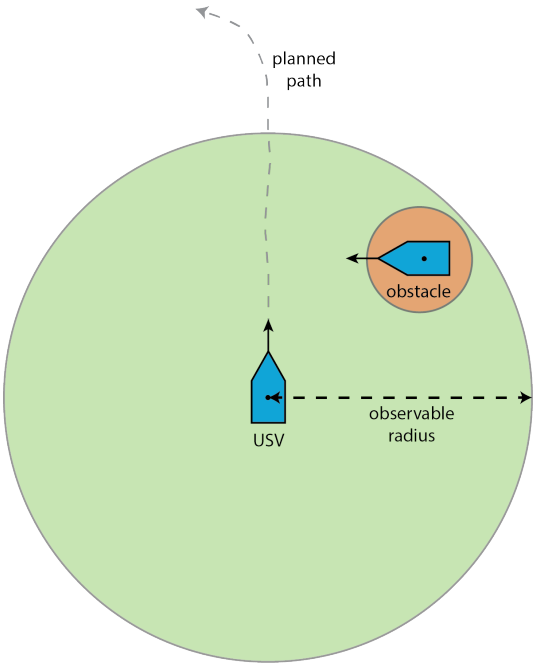


Figure 4.10: The safety region around approaching dynamic obstacles

The speed map will continuously be updated based on approaching obstacles and their corresponding ellipses being projected onto the static speed map of the static environment. This ensures that the USV will not venture too close to any dynamic obstacle.

4.3.2. COLREGS compliant Dynamic Obstacle avoidance

The major challenge is to bring it all together and let the USV respond according to COLREGS regulations in collision hazard situations. It is essential that the COLREGS compliant avoidance strategies are only applied in case of a truly potential collision.

As elaborated in Section 3.4, the Fast Marching Method finds the globally optimal path from its starting point to any goal point if the specific path is reachable. That makes the method effective to use in any path planning situation. Trailing points have been utilized to make the USV respond COLREGS compliant in the different situations. The various strategies to tackle the different COLREGS situations are explained separately.

Right Crossing

When a dynamic obstacle is crossing the USV from the right in a potential collision, the USV needs to alter its course to give way for the encountering obstacle. In such situations, a trailing point is adopted to generate the path. Figure 4.11 shows the location of the trailing point which follows the dynamic obstacle. The Fast Marching Method is now applied with the trailing point as source point. The optimal path from both the USV center point and the goal point towards the trailing point can now be computed.

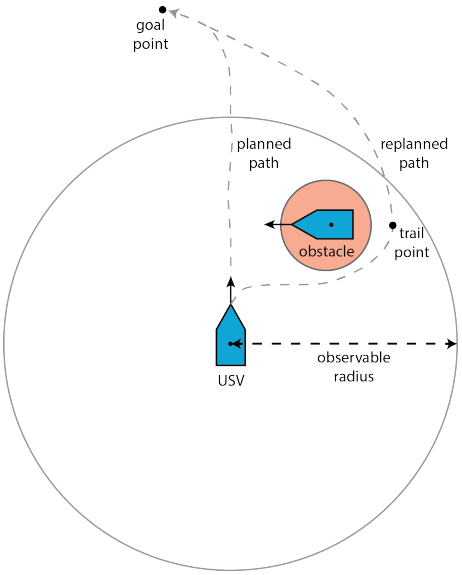


Figure 4.11: The COLREGS compliant path for right crossings

Left Crossing

When a dynamic obstacle crosses from the left, the dynamic obstacle has to give way for the USV to pass in front in case of a potential collision. However, a collision has to be avoided at all cost, so whenever a dynamic obstacle does not abide according to COLREGS, the USV has to steer clear of any potential crash.

Therefore, the Fast Marching Method will just be executed with the source point being equal to goal point. The safety area around the approaching obstacle will make sure that collision is avoided at all cost. Figure 4.12 shows the particular set-up.

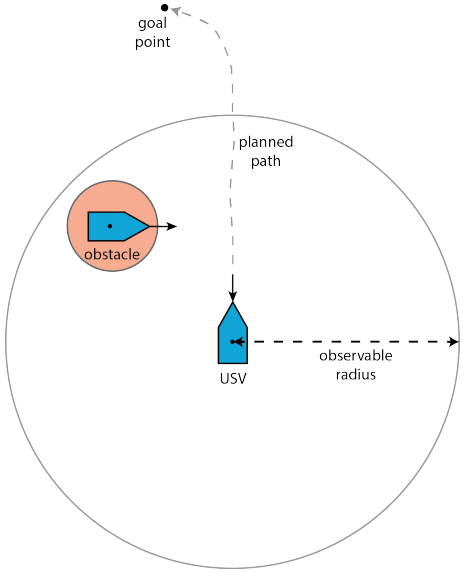


Figure 4.12: The COLREGS compliant path for left crossings

Head-on

When the USV nears another dynamic obstacle head-on, both ships have to pass each other on the right side. Therefore, a trailing point is again used on the left side of the approaching obstacle, as illustrated in Figure 4.13. The trailing point is the starting point of the Fast Marching Method and the path is computed from both the USV center point and the goal point towards the trailing point.

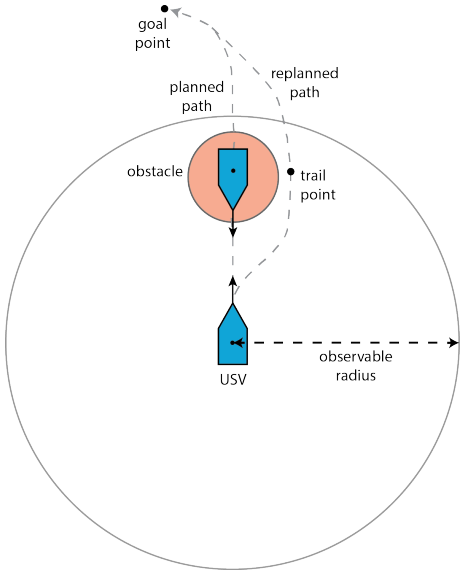


Figure 4.13: The COLREGS compliant path for head-on traverses

Overtaking

In the situation when the USV has to overtake another dynamic obstacle, it can do so on both sides of the obstacle. Therefore, the regular Fast Marching Method is applied with the goal point as source point.

4.3.3. Resetting Conditions

To successfully adopt the strategies that provide COLREGS compliant responses, one needs to implement accurate resetting conditions. These are needed to return to the regular Fast Marching Method with the source point being equal to the actual goal point instead of the trailing point whenever the risk of collision has diminished enough.

The resetting conditions for both the right crossing and the head-on situation rely upon geometric localization. Figure 4.14 illustrates the geometric triangle used for resetting the source point. If the dynamic obstacle providing the trailing point crosses the area spanned by the triangle, the source point changes from the trailing point to the goal point.

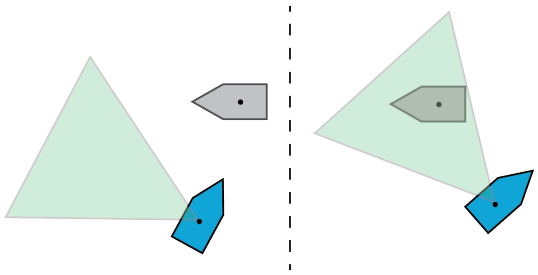


Figure 4.14: Resetting the source point for the FMM from trailing point to goal point

Furthermore, a measure is adopted to make sure that the USV will not start to follow a trail point instead of reaching its goal. If the distance towards the dynamic obstacle exceeds the distance towards its trailing point, the source point will reset as well. The condition is also expressed through the equation:

$$if \quad d_{obst} \leq d_{trail} \quad \rightarrow \quad reset \tag{4.12}$$

The parameters for COLREGS classification and path re-planning of method 1 are presented in Table 4.2.

Table 4.2: The tested parameters

	Parameters	value	Unit	Description
FMM Trailing Points	r_{safety}	4 & 8	m	The safety radius around the obstacles
	r_{detect}	30	m	The radius within which dynamic obstacles are detected

The trailing points are situated on the edge of the safety radius. It is observed that a larger safety radius results in a safer situation with the USV responding relatively early with path re-planning. However, it forces the USV to change path in case of a left cross-over as well.

4.4. An overview: A Flowchart

The process for the entire method is represented by the flowchart in Figure 4.15.

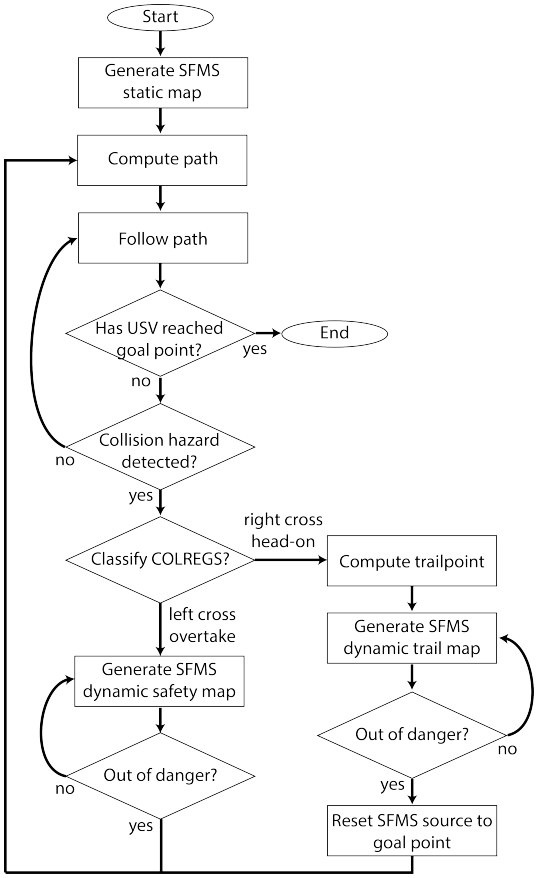


Figure 4.15: The flowchart for method 1

4.5. Evaluation

Method 1 based on trailing points has the potential to enforce COLREGS compliant behavior. However, there are several key features that make the method less appropriate for use. Firstly, the collision detection method has the issue that the size of the ellipse has to be determined. If the area of the ellipse is large, a lot of situations are classified as collision hazardous, which is unwanted. However, too small an ellipse results in accidents.

Secondly, the response is quite reactive as the path is re-planned quite closely around the intersecting dynamic obstacle, making anticipation for other vessels difficult. Moreover, the constant change of the position of both the dynamic obstacles and their safety perimeters enforce the algorithm to continuously update the time-to-arrival map through performing the Fast Marching Method again, which is a relatively large computational effort.

Thirdly, in case the safety radius is small, the newly generated paths through the use of trailing points are relatively late in actually changing course and avoiding collision. However, when the safety radius is large, the paths become excessively longer. Therefore, an optimal safety radius has to be determined, which can be varied according to the speed of the dynamic obstacles and/or USV. Furthermore, the resetting conditions fail in some situations with more dynamic obstacles in the vicinity. Therefore, another method has been developed that better handles these issues, as elaborated in Chapter 5.

5

Method 2: A Path Generating Genetic Algorithm

The split between collision detection, collision classification and path re-planning is used again to systematically explain the method.

5.1. Collision Detection

The collision detection method covered in Chapter 4 is based upon intersecting zones. However, the method lacks prediction power. Therefore, a predictive collision detection algorithm has been adopted.

The key assumptions for method 1 in Chapter 4 also apply for the following method:

- Both the exact location and the heading direction of the dynamic obstacles is known (based on a suitable estimation process)
- In reality altering the speed of vessels is not preferable. Therefore, the velocity of both the dynamic obstacles and the USV is required to stay constant.

To detect a potential collision, a finite time-horizon is constructed for each dynamic obstacle, based upon its current heading and velocity. The finite time-horizon consists of the extrapolated future positions related to specific time stamps. The time-horizon is taken to be equal to $t_{horizon} = 10\ sec$. All the computed future positions within the time-horizon are combined in a matrix

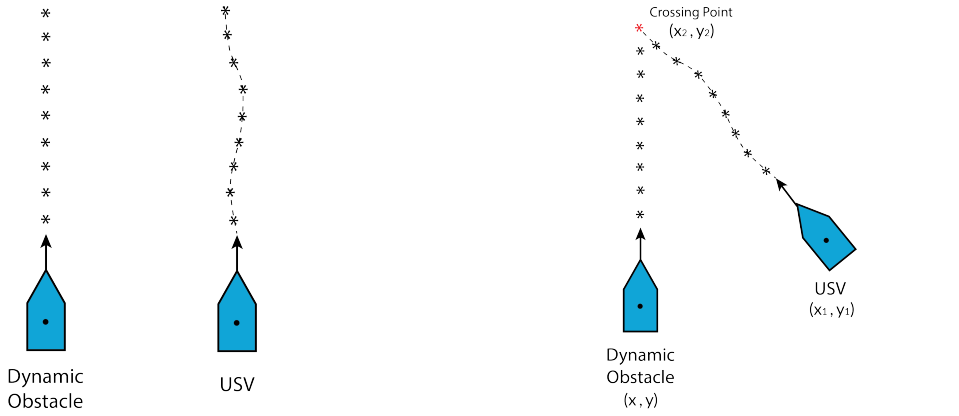
$$horizon_{obstacle} = \begin{bmatrix} p_{t_1} & p_{t_2} & \cdots & p_{t_{horizon}} \end{bmatrix}^T \tag{5.1}$$

The same procedure is used for the USV itself. However, we know the exact path that the USV takes. Therefore, the time-horizon is computed along the path. Figure 5.1a illustrates the time horizon for both the USV and a dynamic obstacle.

To detect a potential collision, the algorithm continuously computes the difference between equivalent points for the time horizons of dynamic obstacles and the USV.

$$\Delta horizon = \begin{bmatrix} \| p_{usv,t_1} - p_{obstacle,t_1} \| & \cdots & \left\| p_{usv,t_{horizon}} - p_{obstacle,t_{horizon}} \right\| \end{bmatrix}^T \tag{5.2}$$

Whenever the difference between similar time-stamp points is small, a future collision is likely when the paths are maintained at the same speed. The time horizon collision point is stored as the crossing point, as shown in Figure 5.1b.



(a) Time-horizons for dynamic obstacle and the USV

(b) The crossing point for time-stamp matching positions

Figure 5.1: The time-horizon

5.2. COLREGS Classification

In case a potential collision occurs, the situation needs to be classified correctly. The classification methodology as described in Section 4.2 is used with a slight modification.

To determine whether the dynamic obstacle is on the left or right side, Equation (4.8) is used again where the crossing point as shown in Figure 5.1b is applied instead. The reasoning behind adapting the crossing point is that it clearly indicates the location of collision and therefore distinguishing between the left and right situation is improved. The relative heading is utilized in similar fashion as shown in Section 4.2, to differentiate between the COLREGS scenarios.

5.3. The Genetic Algorithm

The method of the Genetic Algorithm (GA) belongs to the class of Evolutionary Algorithms, inspired by the process of biological evolution. Genetic Algorithms have proved to be particularly well-suited for the optimization problems dealing with multiple objectives. The main reason is that there is usually not a single optimal solution in multi-objective problems but rather a set of alternative solutions [65].

The Genetic Algorithm is a population-based search algorithm, where the individuals within the population represent samples from the entire set of possibilities. Often the individuals represent single points dealing with function optimization, but optimization of things such as game strategies or classifier rules are also possible. The individuals adjust and evolve over time to generate increasingly better individuals by mixing and sharing their information. Specifically through the processes of *mutation*, altering the information of a single sample within the current population and *cross-over*, recombining of samples within the current population. Figure 5.2 illustrates the process of the Genetic Algorithm.

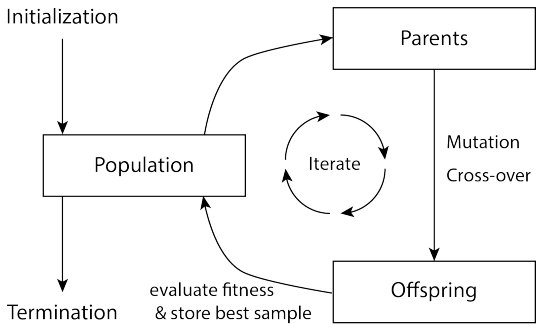


Figure 5.2: The principles of the Genetic Algorithm

5.3.1. Fast Marching Semi-Random Path Generation

The problem is finding an appropriate COLREGS-compliant path to the target that avoids a potential collision. The Fast Marching Square Method is first used to generate a time-of-arrival map $\mathcal{T}$. The source point, from which the wave front

starts, equals the goal point. With the computed time-of-arrival map, a path for avoiding static obstacles can be determined from any point on the map. The USV uses the generated path to get to the goal. However, in case of a potential collision with a dynamic obstacle, a new COLREGS compliant path towards the goal needs to be generated.

The time-of-arrival map $\mathcal{T}$ is used to produce multiple random paths from the current location of the USV to the goal point that are used as the initial population for the Genetic Algorithm, as seen in Figure 5.3.

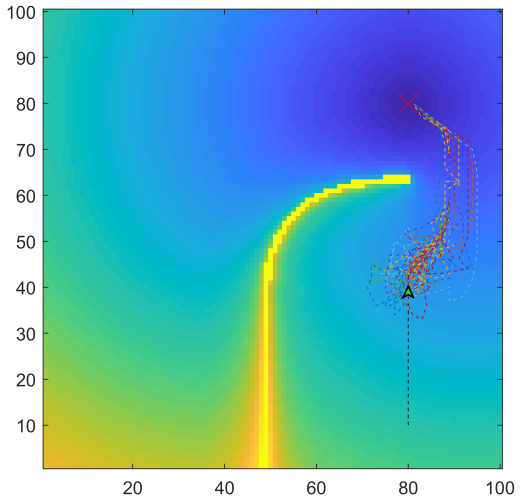


Figure 5.3: The generation of semi-random paths

A modified steepest descent is used to compute the new possible paths. Basically, every k iterations a random step to a neighbouring point is opted instead of the optimal one. The number of generated semi-random paths is defined by n_{paths}. The randomizing step to a neighboring point is only performed up until a percentage n_{random} of the current path. Therefore, the generated paths all converge towards the goal point and are especially different in the first n_{random} percent of the course. The complete functioning is explained in Algorithm 2. One can also opt to skip multiple neighbours and thus increase the influence of randomness on the path generation.

Algorithm 2 Compute multiple FMM semi-random paths

$path_{init} \leftarrow$ the initial path from current position towards the goal
$L_{path_{init}} \leftarrow$ the length of initial path
$n_{paths} \leftarrow$ number of computed semi-random paths
$n_{random} \leftarrow$ percentage of LengthInitPath at which randomization stops
$pop \leftarrow$ store population of generated paths
$k \leftarrow$ which iteration to jump to random neighbour
$j \leftarrow 1$
for $i = 1 : n_{paths}$ **do**
  $(x, y) \leftarrow$ Start Point of USV
  $path(1) \leftarrow (x, y)$
  while $(x, y) \neq SourcePoint$ **do**
    if $modulo(j, k) == 0 \cap \frac{L_{path}}{L_{path_{init}}} < n_{random} \cap \Big(T(x \pm 1, y), T(x, y \pm 1)\Big) \neq \infty$ **then**
      $(x_{new}, y_{new}) \leftarrow$ random neighbour $\Big((x \pm 1, y), (x, y \pm 1)\Big)$
      $path(j + 1) = (x_{new}, y_{new})$
    else
      Analyse $\mathcal{T}$ of neighbours: $T(x_{new}, y_{new}) = \min\Big(T(x \pm 1, y), T(x, y \pm 1)\Big)$
      if $T(x_{new}, y_{new}) < T(x, y)$ **then**
        $(x, y) = (x_{new}, y_{new})$
        $path(j + 1) = (x_{new}, y_{new})$
        $j = j + 1$
      else
        break
      end if
    end if
  end while
  $pop(i) \leftarrow path$
end for

5.3.2. The Fitness Function

The initial population is used as input to the Genetic Algorithm in order to find a well-suited path. A fitness function needs to be well-defined to find appropriate paths. For the problem defined in this thesis, several aspects are key and are discussed individually. Figure 5.4 is an example of how the random paths compare to the current path. Here, the current path equals the initial path from the USV position to the goal point. The random generated paths consist of an equivalent number of waypoints compared to the current path.

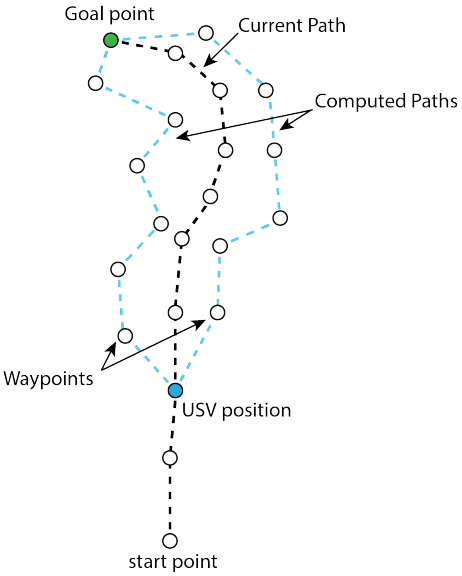


Figure 5.4: The generation of a path population

The Length of the Path

First of all, the length of the path is calculated. The fraction of the current Path from the USV location towards the goal is used as part of the total fitness function

$$f_{length} = \frac{L_{path}}{L_{currentPath}} \tag{5.3}$$

Paths that are significantly longer in comparison to the current path, are therefore less fit to become a solution.

The Jaggedness of the Path

There is a preference for smooth paths that do not very abruptly change course. To determine the jaggedness of a generated path, the difference between angles of consequential waypoints is calculated by

$$\Delta\psi = \left\| \psi_{wp,2} - \psi_{wp,1} \right\| \tag{5.4}$$

This is done for all the successive waypoints

$$\psi_{total} = \sum_{i=1}^{n_{wp}-1} \left\| \psi_{wp,i+1} - \psi_{wp,i} \right\| \tag{5.5}$$

and can then be transformed into

$$f_{\psi_{total}} = \frac{\psi_{total}}{2\pi} \tag{5.6}$$

which can be used for the fitness function.
Figure 5.5 demonstrates the principle.
As the angles ψ of path 1 are all equal, there is no jaggedness. However, path 2 does show differences and the aggregate differences between successive waypoints amounts to the total jaggedness of the path.

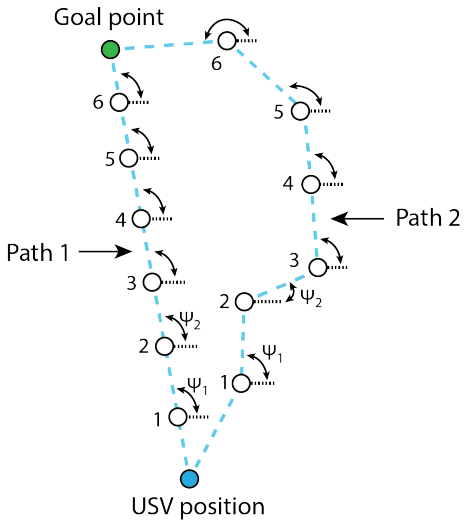


Figure 5.5: Calculate the jaggedness of a generated path

A Collision Free Path

The new path has to be collision free as well. The tool to detect if a path is collision free has already been implemented through the time horizon from Section 5.1. So for every newly generated path, the corresponding time horizon is also quantified. For every obstacle within the vicinity, the difference between the time horizon of the newly generated path and that of dynamic obstacles in calculated

$$\Delta horizon = \left[\left\| p_{path,t_1} - p_{obstacle,t_1} \right\| \quad \cdots \quad \left\| p_{path,t_{horizon}} - p_{obstacle,t_{horizon}} \right\| \right]^T \tag{5.7}$$

Whenever the distance of any of the points in the matrix is too small (eg. $d_{col,ref} \leq 3$), a collision is likely when continuing according to the path. Therefore, the path is punished accordingly

$$\begin{aligned} if \quad & any(\Delta horizon) \leq d_{col,ref} \\ & f_{collision} = 1 \end{aligned} \tag{5.8}$$

A higher threshold makes an accident more unlikely as the re-planned path is more distant from the obstacle paths.

COLREGS compliance

One of the big challenges is how to ensure that COLREGS compliant paths are favored. The COLREGS classification method is used to distinguish between the different scenarios enabling the choice for the proper constraint. The scenarios are discussed individually below.

For every generated path a corresponding time-horizon is calculated, as mentioned previously. The individual points of the time-horizon are analyzed to determine if a path is likely to obey the COLREGS regulations.

Right Crossing

In the case of a Right Crossing, a corresponding Crossing Point has been established. All the individual time-horizon points of a generated random path are examined. It is determined if the individual points are either on the left side (-1) or the right side (1) of the line between the USV and the crossing point

$$s_1 = \frac{(p_{horizon,1} - p_{USV}) \times (p_{crosspoint} - p_{USV})}{\left\| (p_{horizon,1} - p_{USV}) \times (p_{crosspoint} - p_{USV}) \right\|} \tag{5.9}$$

This is done for all the points within the time-horizon. Figure 5.6 demonstrates the separation, where time-horizon points on the left side are marked by the colour *red*, whereas points on the right side are characterized by the colour *green*.
To obey the COLREGS regulations when re-planning the path for a right crossing, time-horizon points on the right side are preferred. Therefore, a threshold t_{crit} of a minimal number of right-sided points is required for a generated path. If the threshold is not reached, the path is be punished as follows

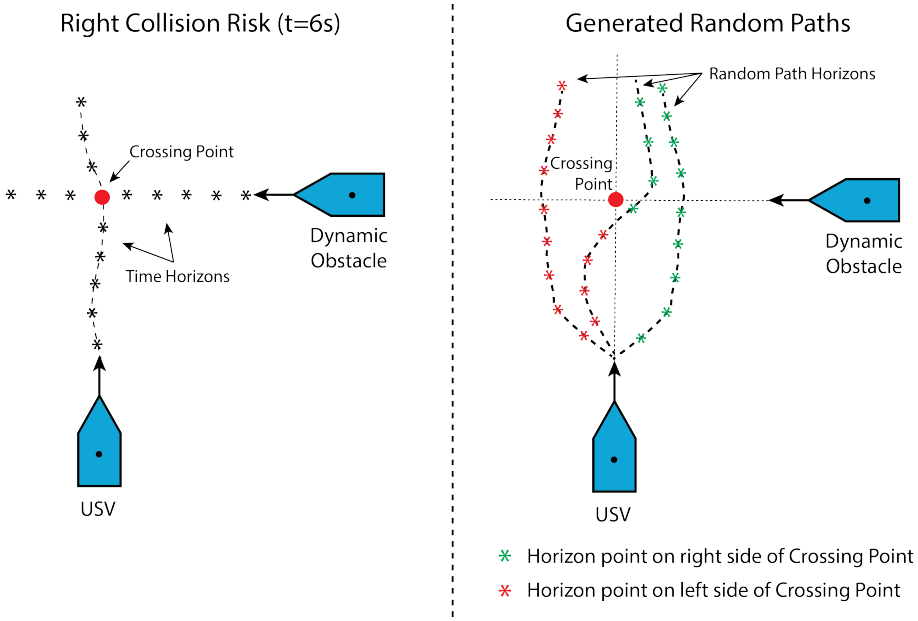


Figure 5.6: The marked time-horizons of generated paths for right crossings

$$
\begin{aligned}
&if \quad \sum_{i=1}^{n_{horizon}} s_i \leq t_{crit} \\
&\quad f_{COLREGS} = 1 \\
&else \\
&\quad f_{COLREGS} = 0
\end{aligned}
\tag{5.10}
$$

Head-On

For head-on collisions, the same method applies as for the *right crossing*, as the USV needs to pass on the right side. Figure 5.7 demonstrates the separation of the time-horizon into right and left side points.
Again, the path is punished if the minimum threshold is not hit.

$$
\begin{aligned}
&if \quad \sum_{i=1}^{n_{horizon}} s_i \leq t_{crit} \\
&\quad f_{COLREGS} = 1 \\
&else \\
&\quad f_{COLREGS} = 0
\end{aligned}
\tag{5.11}
$$

Left Crossing

For the scenario of a left crossing it is possible to not punish any side. If the dynamic obstacle is adjusting either its speed or its course, the risk of collision disappears. However, if the dynamic obstacles continues, the genetic algorithm generates a path either on the left or right side to avoid collision.

However, to stimulate the USV to move in front of the approaching dynamic obstacle, a punishment could be added to time horizon points on the left of the crossing point.

Overtaking

For situations in which the USV is overtaking another dynamic obstacle, the path can be adjusted on either side of the dynamic obstacle. thus, there is no need to force the USV to any side by constraints.

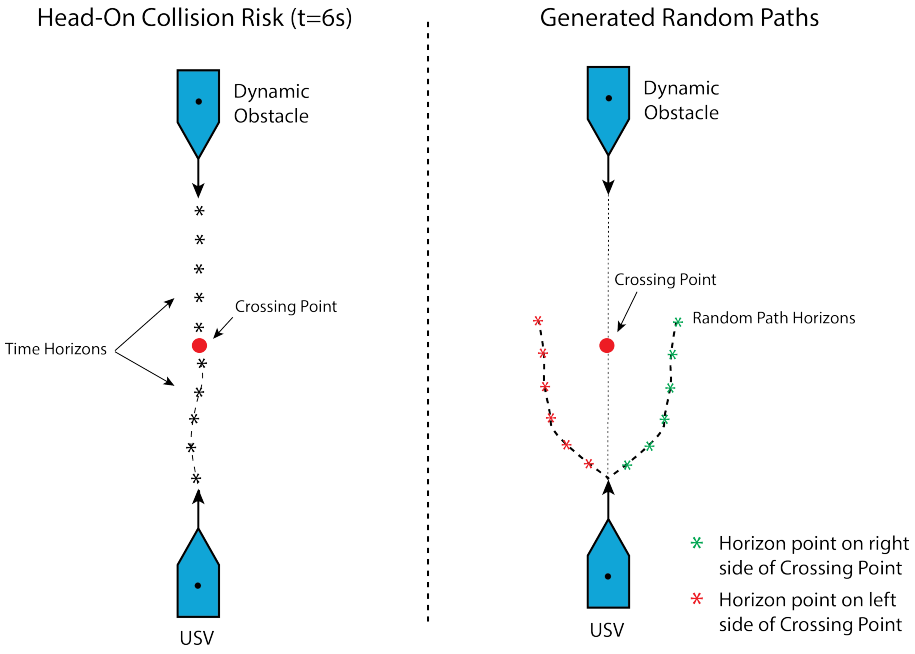


Figure 5.7: The marked time-horizons of generated paths for head-on collision

The Fitness Function

The Fitness of all the generated paths can now be assessed by using the following weighted fitness function

$$F = \omega_{length} \cdot f_{length} + \omega_{jaggedness} \cdot f_{\psi_{total}} + \omega_{collision} \cdot f_{collision} + \omega_{COLREGS} \cdot f_{COLREGS} \tag{5.12}$$

The weights can be tweaked according to the preferences as some constraints can be contradictory. For example, always being COLREGS compliant usually implies that the best path is not optimal considering its length.

The multi-objective and to some extent contradictory nature of the fitness function makes it difficult to specify accurate stopping criteria. Moreover, different paths can achieve similar fitness values due to the different components. Therefore, the quality of the resulting paths is analyzed with different settings for the number of generations. A low number of generations acting as a cap on the amount of iterations, suffices as a good stopping criterion.

5.3.3. Optimizing the Path

The initial population of paths and the corresponding fitness values are used as input for the Genetic Algorithm. First of all, the best path according to the determined fitness function is stored. The processes of Cross-Over and Mutation, explained below, are iterated n_{gen} times to get new and improved generations.

Cross-Over

The probability of a cross-over is represented by $p_{crossover}$.The probability of cross-overs is kept relatively small as high percentages may lead to premature convergence. A cross-over is the combination of two parent paths. Figure 5.8 illustrates the process.

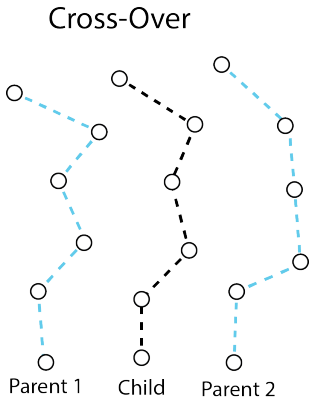


Figure 5.8: Genetic Algorithm Cross-Over for generated paths

Mutation
The probability of a mutation to a single parent is represented by $p_{mutation}$. For the process of mutation, the individual waypoints of a parent path are changed slightly to generate a new Child path as shown in Figure 5.9

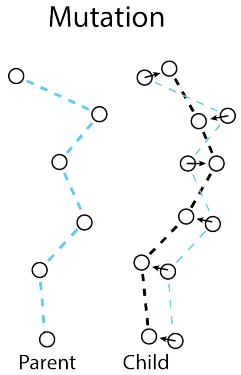


Figure 5.9: Genetic Algorithm Mutation for generated paths

A portion of the new child paths is mutated randomly from the parent population and moreover, a portion of the new children are also obtained through mutation of the fittest parent paths.
The parameters that have been manipulated for method 2 are presented in Table 5.1.

Table 5.1: The tested parameters

	Parameters	value	Unit	Description
Method 2	pop	50, 100	-	Total population of random paths
	n_{paths}	20, 40	-	The number of fittest path used for Genetic Algorithm
	n_{gen}	2, 5, 10	-	The number of generations of parent to child
	$p_{combine}$	0.1	-	The probability of combining two parent paths
	p_{mutate_1}	0.7	-	The probability of mutation for a random parent path
	p_{mutate_2}	0.2	-	The probability of mutation for a very fit parent path
	$t_{horizon}$	10	s	The maximum time-horizon
	$d_{col,ref}$	3, 5	-	The threshold distance for collision free fitness
	t_{crit}	4, 6, 8	-	The threshold for COLREGS compliant fitness

First of all, large values for the parameters pop, n_{paths} and n_{gen} lead to a relatively high computational effort. It is found that higher values for the total population of paths (pop) are beneficial as it increases the variability of the paths and therefore finding an appropriate path gets easier. The number of fittest paths n_{paths} does not need to be really high because parent paths with low fitness have a high probability of not contributing to a fitter path. Furthermore, some of the generated initial paths already have high fitness, a high proportion of generations is unnecessary or even unwanted due

to the additional computational effort.

5.4. An overview: A Flowchart

The process for the entire method is represented by the flowchart in Figure 5.10.

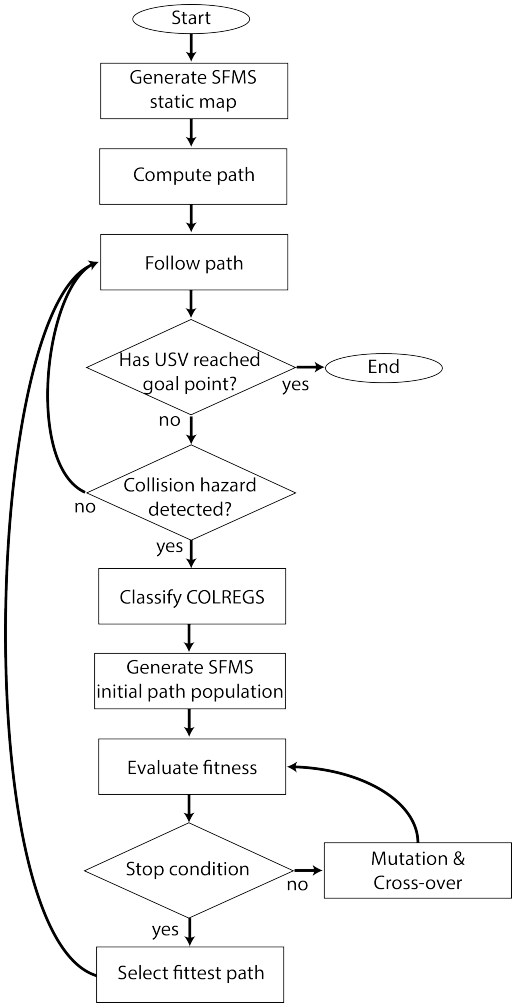


Figure 5.10: The flowchart for method 2

5.5. Evaluation

The combination of the Saturated Fast Marching Square Method (SFMS) and the Genetic Algorithm have significant benefits. The SFMS generates a time-to-arrival map that can be used to find the path from any point on the map to the goal point. The map only needs to be computed once, so it is not computationally heavy. The new paths are generated by combining the time-to-arrival map with a path-generating Genetic Algorithm (GA). The paths can be analyzed due to the ability to test the discrete paths against a fitness function. The combination of SFMS and GA are computationally advantageous and the re-planned paths are in general superior. There is a high level of uniqueness in the generated paths as the generated paths never completely match in similar situations. This can be both a strength, enabling the algorithm to deal with a lot of situations, and a weakness in not fully knowing the algorithm response.

5.5.1. Emergency Script

Method 2 also has the advantage that it lends itself to possible implementation of an emergency script in case the collision risk is high. So when the collision is about to occur within a short time horizon, the emergency script could incorporate speed commands such as an emergency break or increasing the speed to move away so that collision is avoided. This would be a new topic of research.

6

Simulations: Case studies

Several individual case studies have been designed to be able to study the performance of both algorithms and any corresponding issues. First, simple single encounter situations are scrutinized to understand the performance in the most straightforward of situations. Second, more complex situations with multiple dynamic obstacles are examined to determine how the algorithms cope with increasingly difficult situations. Lastly, the performance is tested in the presence of both static obstacles and dynamic obstacles.

First several parameters are tested on the simple single encounter situations to obtain a clear picture of the influence of individual parameters. It has to be noted that at first, low speeds have been used for both the USV and the dynamic obstacles in order to properly assess performance.

6.1. Single Encounters

For the case studies on single encounters, an empty map without any obstacles is used with a width of 50×50 meters. The dynamic obstacle has a speed of 2 m/s, whereas the USV has a speed of 3 m/s to ensure that overtaking is possible. The green triangle shape represents the USV whereas the white triangle shape is a dynamic obstacle. Table 6.1 illustrates the parameters that are utilized and the range of values that has been used.

Table 6.1: The parameters used in the single case studies

	Parameters	value	Unit	Description
Method 1	r_{safety}	4	m	The safety radius around the obstacles
	r_{detect}	30	m	The radius within which dynamic obstacles are detected
Method 2	pop	100	-	Total population of random paths
	n_{paths}	20	-	The number of fittest path used for Genetic Algorithm
	n_{gen}	2	-	The number of generations of parent to child
	$p_{combine}$	0.1	-	The probability of combining two parent paths
	p_{mutate_1}	0.7	-	The probability of mutation for a random parent path
	p_{mutate_2}	0.2	-	The probability of mutation for a very fit parent path
	$t_{horizon}$	10	s	The maximum time-horizon
	$d_{col,ref}$	3	-	The threshold distance for collision free fitness
	t_{crit}	4	-	The threshold for COLREGS compliant fitness

6.1.1. Right Crossing Single Encounter

When a potential collision with a right crossing occurs, the USV has to give way to the approaching obstacle and pass behind it. The parameters for the right crossing single encounter are reported in Table 6.2.
Upon initiation, a first path is generated from the initial starting point of the USV to the goal point, as can be observed in Figure 6.1. The path is the obtained through the discrete steepest descent of the generated time-of-arrival map from the Saturated Fast Marching Square method ($saturation = 0.2$).

Table 6.2: The parameters for Single Right Crossing

	Parameters	value	Unit	Description
USV	startpoint	(25,5)	-	-
	goalpoint	(25,45)	-	-
	velocity	3	m/s	-
Obstacle 1	startpoint	(45,35)	-	-
	goalpoint	(5,35)	-	-
	velocity	2	m/s	-

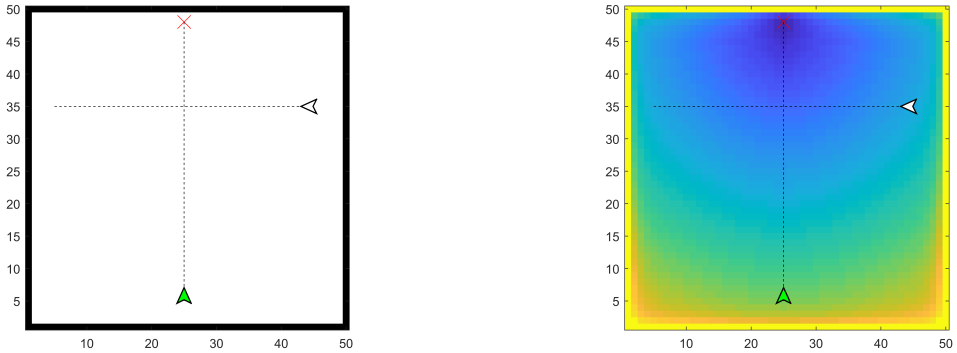


Figure 6.1: Initial path before right crossing single encounter

Method 1: Trailing Points

When a potential collision is detected and classified as right crossing, the USV needs give-way to the approaching obstacle. The path is now re-planned to pass behind the dynamic obstacle by starting the Fast Marching Method from a trailing point behind the obstacle forcing the path to go behind it towards the goal point, illustrated in Figure 6.2.

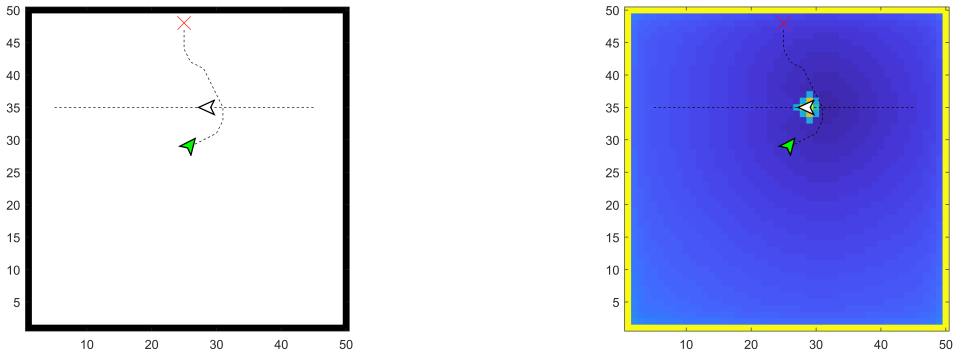
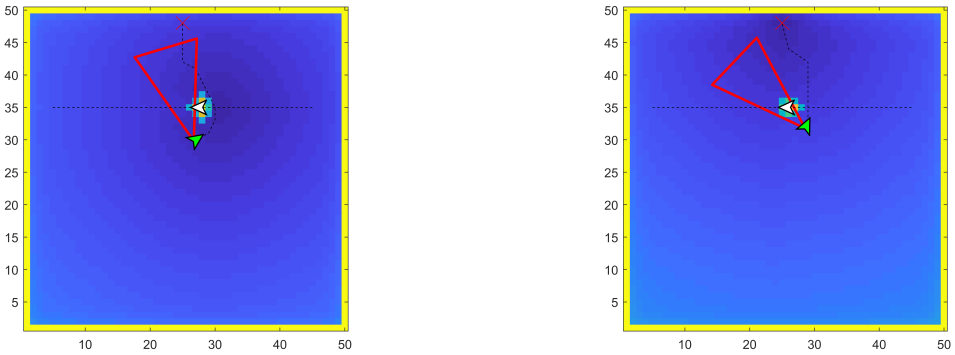


Figure 6.2: Re-planning the path through a trail point for a right cross encounter

The trailing point is alleviated once the obstacle is within the reset area of the USV and the danger of collision is over. The switch between the Fast Marching Method based upon the trailing point and the goal point is clearly demonstrated in Figure 6.3. One can clearly observe that the dynamic obstacle has not yet entered the reset area in Figure 6.3a, whereas it has crossed the reset area in Figure 6.3b, thereby re-instating goal point as the FMM source point.



(a) Reset not yet activated and FMM source in trail point

(b) The reset activated and FMM source in goal point

Figure 6.3: Reset trail point for the goal point when USV has given way for the obstacle

Method 2: Path Generating Genetic Algorithm

Method 1 based on trailing points generates the same result with equivalent parameters. However, method 2 based on the path generating Genetic Algorithm varies every single iteration. Moreover, the deduced path can differ significantly based on the weight preferences of the weighted fitness function.

The Genetic Algorithm generates a first random population of $n = 100$. These are analyzed for their fitness and the best 25 paths ($n_{path} = 25$) are used as the initial population for the Genetic Algorithm.
Figure 6.4 demonstrates two generated best paths based on a high preference for a short travelling distance. As can be observed, the USV passes the dynamic obstacle relatively closely. The green stars represent the time horizons.

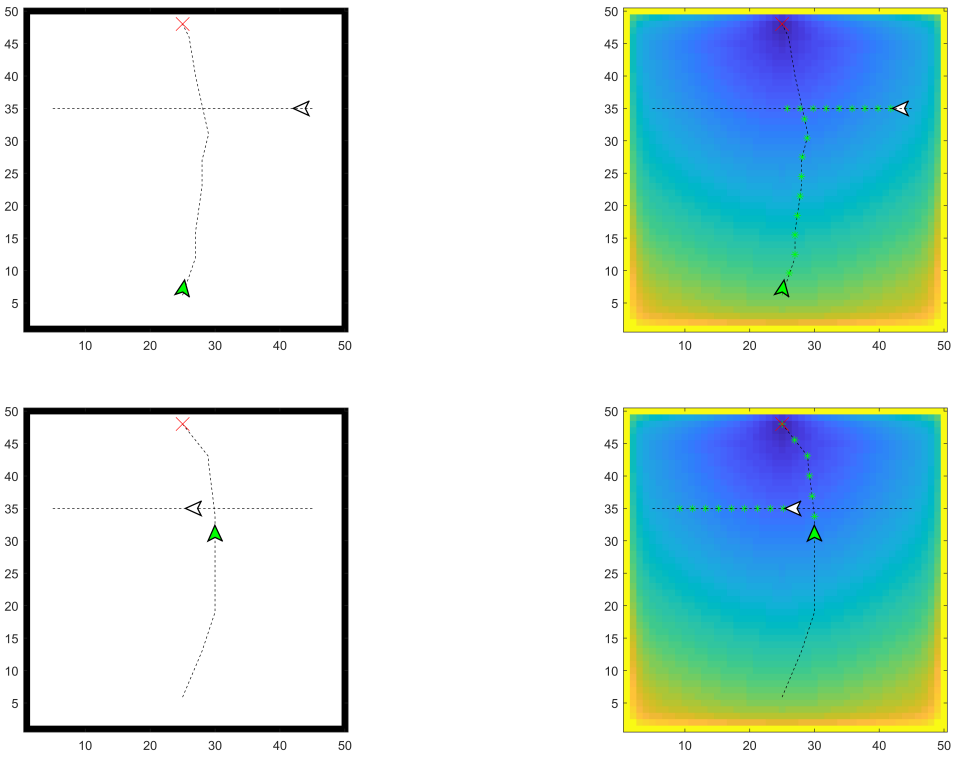


Figure 6.4: The Genetic Algorithm generated best path with preference for short travelling distance

When a high preference is given to complying with the COLREGS and avoiding collisions, paths become significantly longer as can be observed in Figure 6.5. An appropriate trade-off has to be made, but when the weights are similar in order of magnitude, a good path is usually obtained.

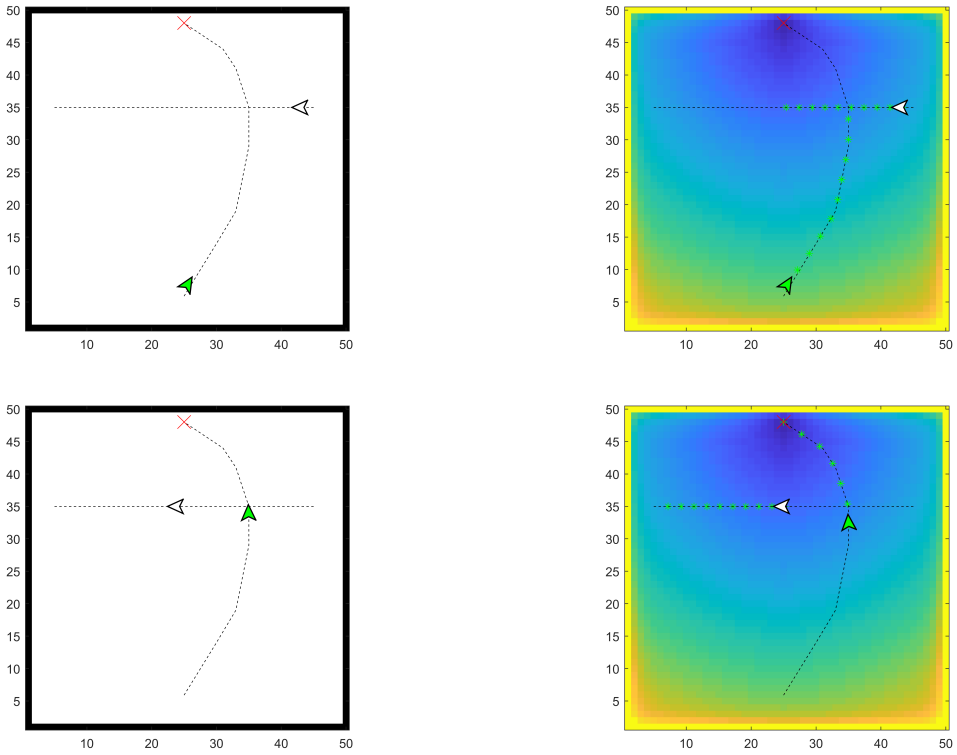


Figure 6.5: The Genetic Algorithm generated best path with large preference for complying to COLREGS

6.1.2. Left Crossing Single Encounters

When a left crossing occurs, the dynamic obstacle has to give way for the USV to pass. However, when the right-of-way is not given, one should always avoid collisions. The parameters for the left crossing single encounter are:

Table 6.3: The parameters for Single Left Crossing

	Parameters	value	Unit	Description
USV	startpoint	(25,5)	-	-
	goalpoint	(25,45)	-	-
	velocity	3	m/s	-
Obstacle 1	startpoint	(45,35)	-	-
	goalpoint	(5,35)	-	-
	velocity	2	m/s	-

The initial situation is shown in Figure 6.6.

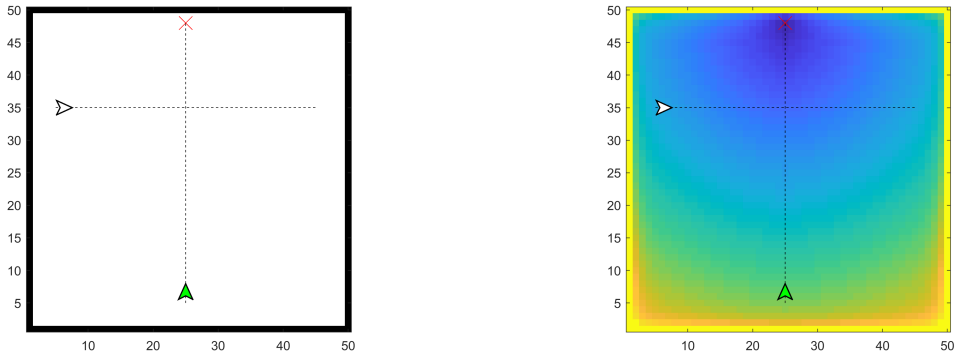


Figure 6.6: Re-planning the path through a trail point for a right cross encounter

Method 1: Trailing Points

The Left cross situation is tricky as the USV has in general got the right the pass in front. Therefore, the safety perimeter around the dynamic obstacle is expanded to try to ensure that collisions do not occur. However, when an approaching dynamic obstacle still continues its path, the distance between both is minimal, as can be seen in Figure 6.7

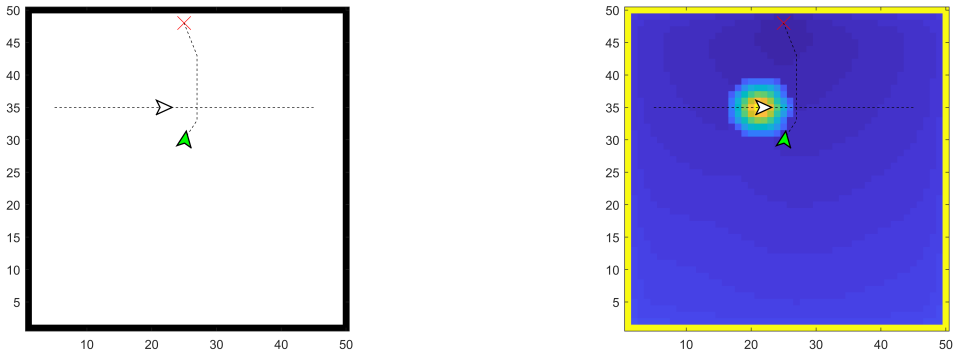


Figure 6.7: Divert the course in a left crossing situation with FMM

Method 2: Path Generating Genetic Algorithm

With the Genetic Algorithm, anticipation is quicker and a safe path is chosen early on in the process that both avoids collision and results in a relatively optimal new path, which can be observed in Figure 6.8.

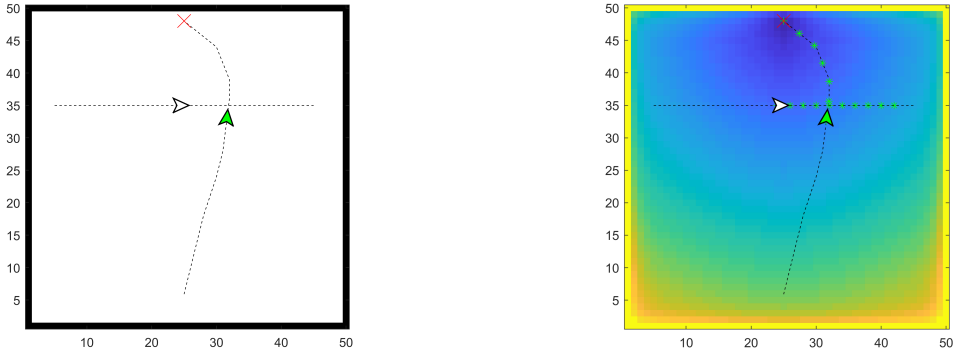


Figure 6.8: Divert the course in a left crossing situation with Genetic Algorithm

6.1.3. Head-on Single Encounters

When the USV encounters Head-on situation, the path needs to be re-planned such that the USV passes the approaching obstacle to its left side. The parameters for the specific situation are:

Table 6.4: The parameters for Single Head-on

	Parameters	value	Unit	Description
USV	startpoint	(25,5)	-	-
	goalpoint	(25,45)	-	-
	velocity	3	m/s	-
Obstacle 1	startpoint	(25,45)	-	-
	goalpoint	(25,5)	-	-
	velocity	2	m/s	-

The situation is illustrated in Figure 6.9.

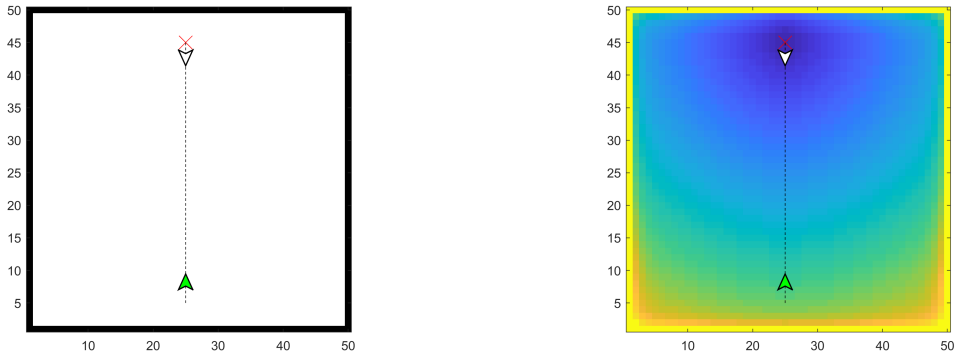


Figure 6.9: Initial path before head-on single encounter

Method 1: Trailing Points

When the situation is classified as a head-on collision, the newly adopted trailing point functioning as the source point for the Fast Marching Method is situated on the left side of the approaching obstacle. Therefore, the path forces the USV to pass on the appropriate side. The response is shown in Figure 6.10.

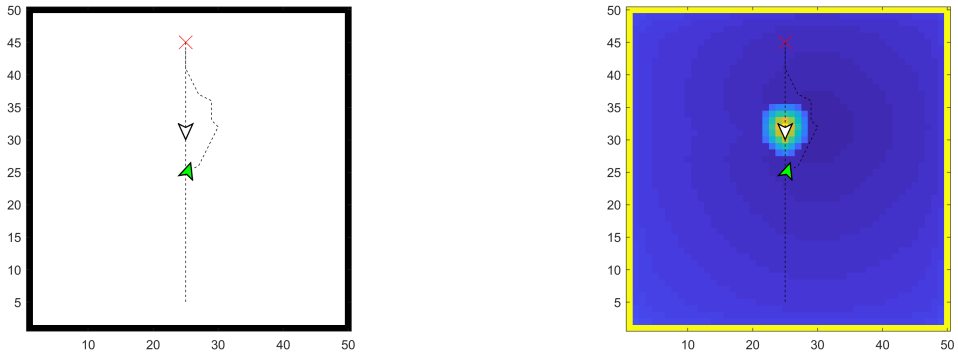


Figure 6.10: Re-planned path for trailing point FMM with head-on single encounter

The Fast Marching method based on the trailpoint is reset to the regular FMM when the dynamic obstacle crossed the reset area, as shown in Figure 6.11.

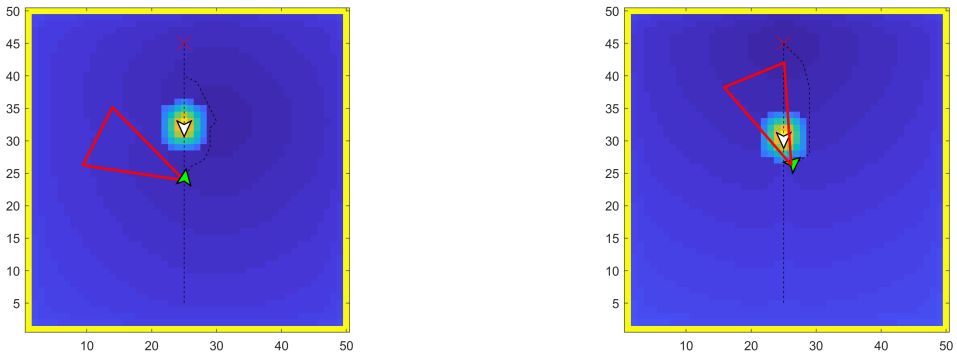


Figure 6.11: Resetting the trailing point to goal point with head-on single encounter

Method 2: Path Generating Genetic Algorithm

The genetic algorithm generates a safer path adapting the course of the USV early on, as can be seen in Figure 6.12.

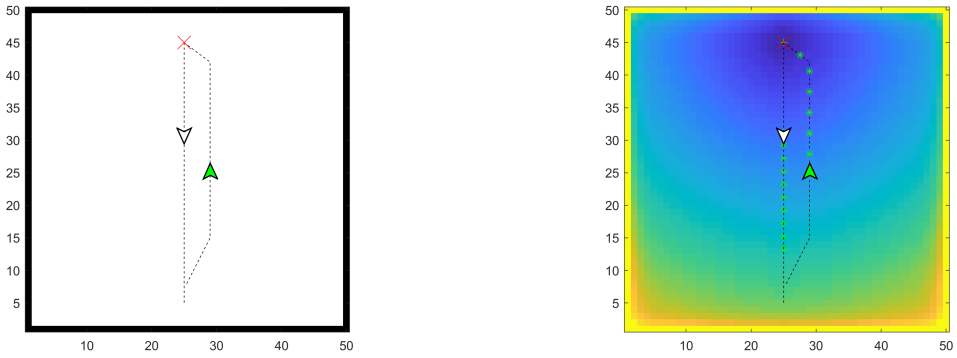


Figure 6.12: The re-planned path through the Genetic Algorithm in a head-on single encounter

As can be observed, it correctly manoeuvres alongside the approaching obstacle while maintaining a safe distance.

6.1.4. Overtaking Single Encounters

When the USV overtakes any moving obstacle, it can overtake on both sides. The parameters for the specific situation are:

Table 6.5: The parameters for Single Overtaking

	Parameters	value	Unit	Description
USV	startpoint	(25,5)	-	-
	goalpoint	(25,45)	-	-
	velocity	3	m/s	-
Obstacle 1	startpoint	(25,15)	-	-
	goalpoint	(25,45)	-	-
	velocity	1.5	m/s	-

Figure 6.13 shows an overtake situation.

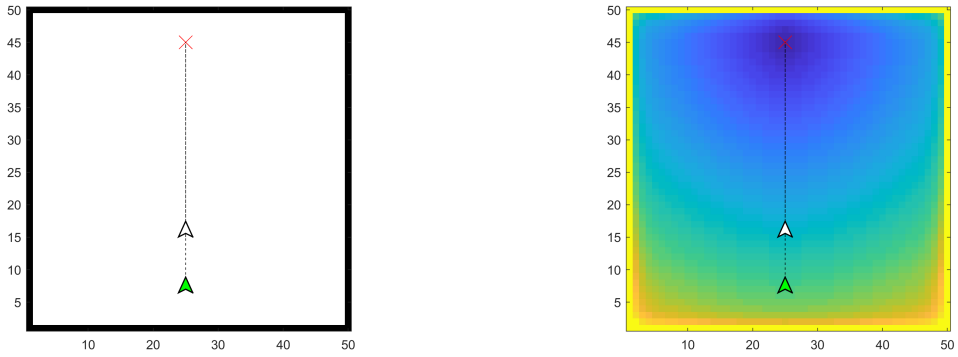


Figure 6.13: Initial path before overtaking single encounter

Method 1: Trailing Points

The Fast Marching Method can simply be applied from the center of the USV and it will automatically guide it past the safety zone of the obstacle in front. The execution of the Fast Marching method can be observed in Figure 6.14. The larger the obstacle safety perimeter, the greater the distance between both.

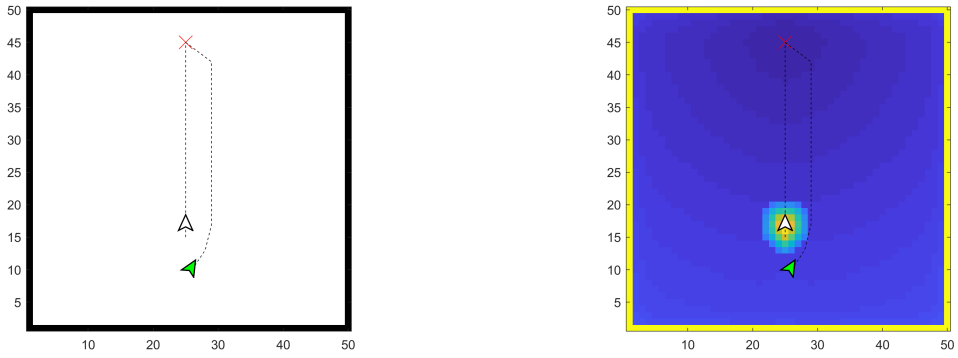


Figure 6.14: Re-planned path for trailing point FMM with overtaking single encounter

Method 2: Path Generating Genetic Algorithm

The path generated through the Genetic Algorithm is shown in Figure 6.15. The path can form on either the right or left side.

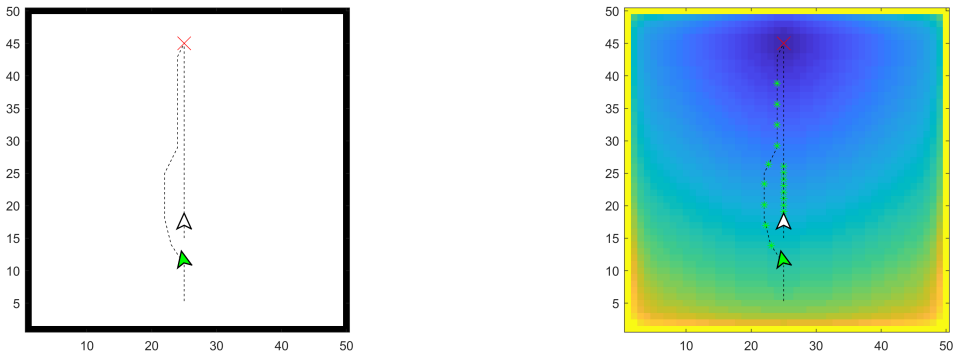


Figure 6.15: The path generated by Genetic Algorithm for overtaking single encounter

6.2. Multiple Dynamic Obstacles

The case studies on single encounters show that the COLREGS regulations are obeyed. However, the conceived methods also need to work when multiple dynamic obstacles are present in close proximity. Therefore, a few key situations are studied in more detail.

6.2.1. 3 dynamic obstacles in close proximity

When multiple dynamic vessel are present in close proximity of each other, obeying COLREGS becomes even more important, as anticipation for all surrounding vessels is key. The parameters for this encounter are:

Table 6.6: The parameters for multiple vessel encounter

	Parameters	value	Unit	Description
USV	startpoint	(25,5)	-	-
	goalpoint	(25,45)	-	-
	velocity	3	m/s	-
Obstacle 1	startpoint	(35,20)	-	-
	goalpoint	(5,20)	-	-
	velocity	2	m/s	-
Obstacle 2	startpoint	(45,30)	-	-
	goalpoint	(5,30)	-	-
	velocity	2	m/s	-
Obstacle 3	startpoint	(35,48)	-	-
	goalpoint	(30,5)	-	-
	velocity	2	m/s	-

There are two potential right crossings and a possible head-on collision. The starting situation is illustrated in Figure 6.16.

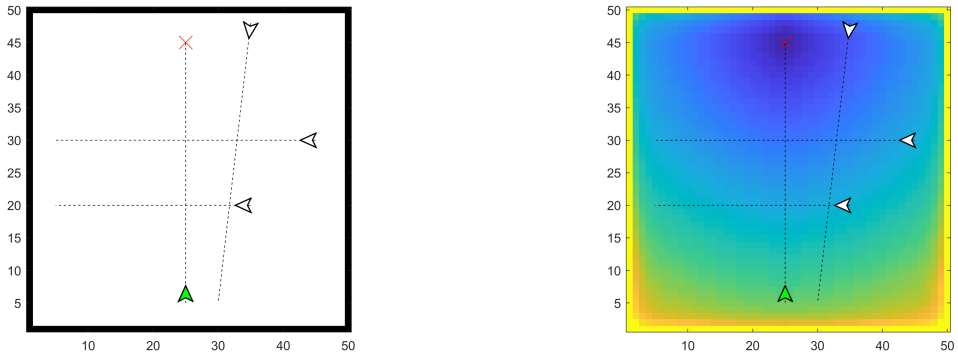


Figure 6.16: The initial path with 3 dynamic obstacles

Method 1: Trailing Points

The trailing point method works well when there is enough space without congestion. However, in the case of multiple obstacles in close proximity, issues can arise as other obstacles might cross trailing points of obstacles. Figure 6.17 shows the generated path through time and how it collides with obstacle 3 due to adapting its course for obstacle 2. A larger safety distance already improves the situation although unexpected reactions can still occur.

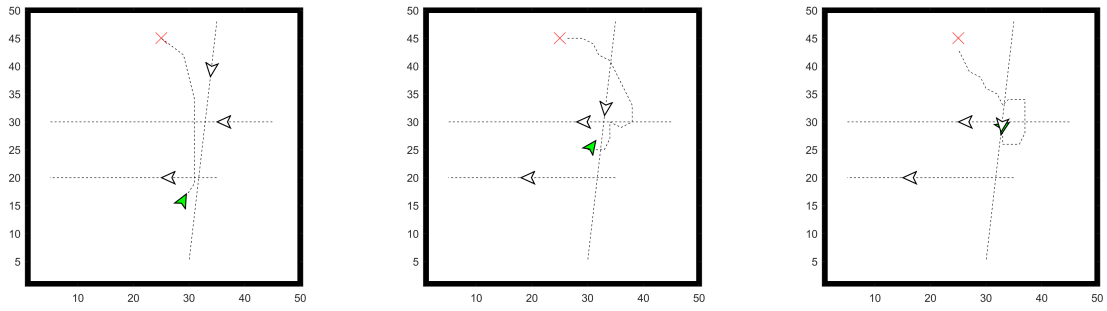


Figure 6.17: The re-planned path through trailing points with 3 dynamic obstacles

Method 2: Path Generating Genetic Algorithm
For the path generated by the Genetic Algorithm, it is different. it takes into account all the potential collision in the future and computes a path that both obeys the regulations and avoids future collisions. The resulting path for the above scenario is shown in Figure 6.18.

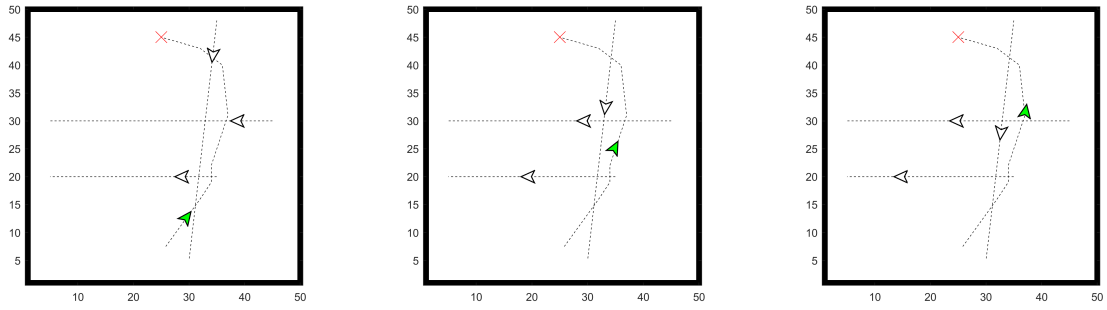


Figure 6.18: The re-planned path through trailing points with 3 dynamic obstacles

6.3. Static and Dynamic Obstacles

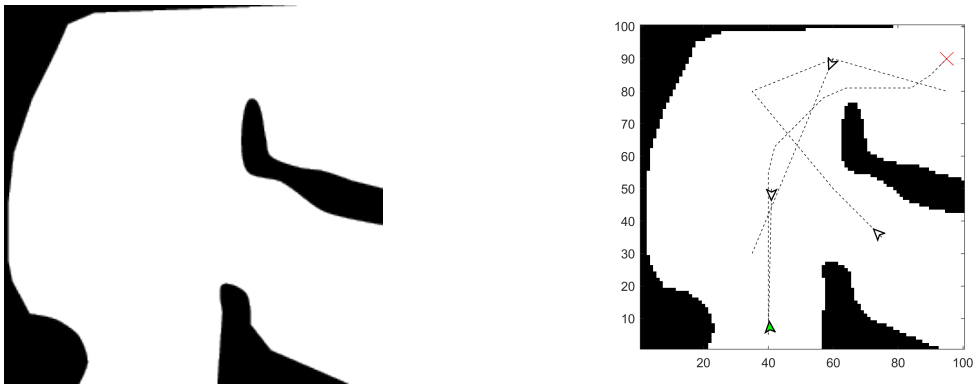
The main advantage of using the Fast Marching Method has to do with the ease of path-planning through a static map. Therefore, the performance of both methods has to be observed in a map with both static and dynamic obstacles. Consequently, simulations are performed in both a river landscape (100×100) and a map with multiple static obstacles (100×100). Moreover, the paths of the dynamic obstacles are curved to more accurately reproduce real scenarios. Some parameters are changed slightly to ensure safer path re-planning, covered in Table 6.7. Furthermore, the speeds of both the USV and the dynamic obstacles is increased.

Table 6.7: The parameters used in the single case studies

	Parameters	value	Unit	Description
Method 1	r_{safety}	8	m	The safety radius around the obstacles
Method 2	$d_{col,ref}$	5	-	The threshold distance for collision free fitness

6.3.1. Simulating a river

The map of the river is illustrated in Figure 6.19a, where the black area resembles land and the white area represents the joining rivers. The individual paths of the dynamic obstacles and the initial path for the USV are shown in Figure 6.19b.



(a) The river landscape

(b) The varying dynamic obstacle paths

Figure 6.19: River Simulation

The parameters for the river simulation are presented in Table 6.8.

Table 6.8: The parameters for Single Overtaking

	Parameters	value	Unit	Description
USV	startpoint	$(40,5)$	-	-
	goalpoint	$(95,90)$	-	-
	velocity	8	m/s	-
Obstacle 1	path	$\begin{bmatrix} 41 & 50 \\ 40 & 5 \end{bmatrix}$	-	The path is described by waypoints $[x_i \quad y_i]$
	velocity	5	m/s	-
Obstacle 2	path	$\begin{bmatrix} 60 & 90 \\ 35 & 30 \end{bmatrix}$	-	The path is described by waypoints $[x_i \quad y_i]$
	velocity	7	m/s	-
Obstacle 3	path	$\begin{bmatrix} 75 & 35 \\ 60 & 50 \\ 35 & 80 \\ 60 & 90 \\ 95 & 80 \end{bmatrix}$	-	The path is described by waypoints $[x_i \quad y_i]$
	velocity	6	m/s	-

Method 1: Trailing Points

It is noticeable that method 1 starts to have trouble due to the increased speed of both the USV and the dynamic obstacles. Figure 6.20 illustrates the path of the USV and how it responds too late with higher speeds.

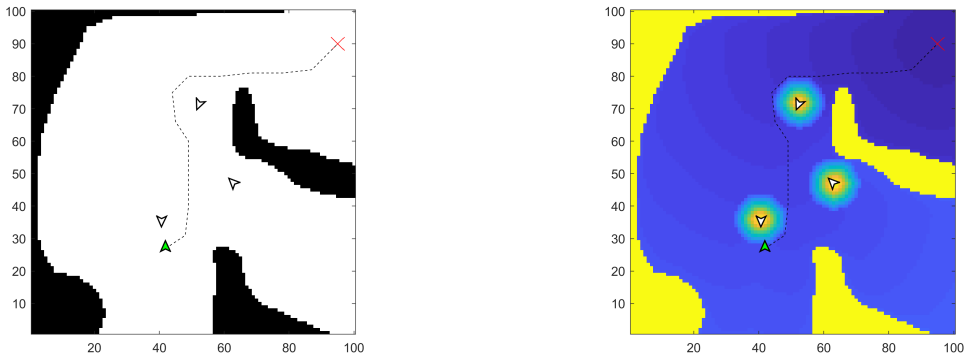


Figure 6.20: Method 1 River Simulation

Method 2: Path Generating Genetic Algorithm

It is interesting to observe that Method 2 still responds well in-time and accurately with increased velocities as well. Figure 6.21 shows the newly generated path without any collision risks.

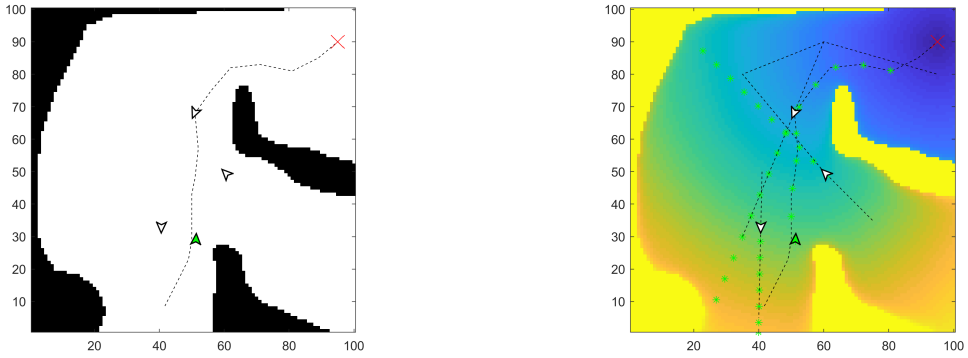
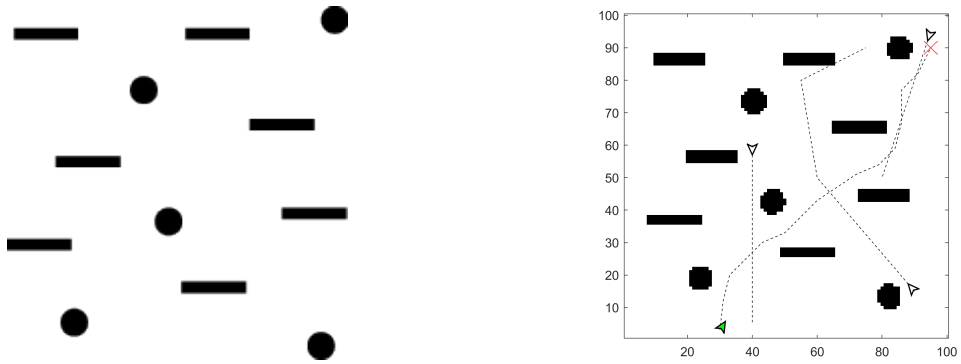


Figure 6.21: Method 2 River Simulation

6.3.2. Random Static Obstacles

Lastly, the methods are tested in an environment with both individual static obstacles and moving dynamic obstacles.



(a) The river landscape

(b) The varying dynamic obstacle paths

Figure 6.22: Static Obstacle Simulation

The parameters for the static obstacles simulation are presented in Table 6.9.

Table 6.9: The parameters for Single Overtaking

	Parameters	value	Unit	Description
USV	startpoint	$(30,3)$	-	-
	goalpoint	$(95,90)$	-	-
	velocity	4	m/s	-
Obstacle 1	path	$\begin{bmatrix} 40 & 60 \\ 40 & 5 \end{bmatrix}$	-	The path is described by waypoints $[x_i \quad y_i]$
	velocity	3	m/s	-
Obstacle 2	path	$\begin{bmatrix} 90 & 95 \\ 80 & 50 \end{bmatrix}$	-	The path is described by waypoints $[x_i \quad y_i]$
	velocity	3	m/s	-
Obstacle 3	path	$\begin{bmatrix} 90 & 15 \\ 60 & 50 \\ 55 & 80 \\ 75 & 90 \end{bmatrix}$	-	The path is described by waypoints $[x_i \quad y_i]$
	velocity	3	m/s	-

The environment is now quite challenging due to the presence of both dynamic and static obstacles. method 1 runs into problems as the trailpoints are located precisely on the static obstacles, disabling the Fast Marching Method to be properly executed. However, method 2 is able to adapt the path. Figure 6.23 illustrates such a re-planned path.

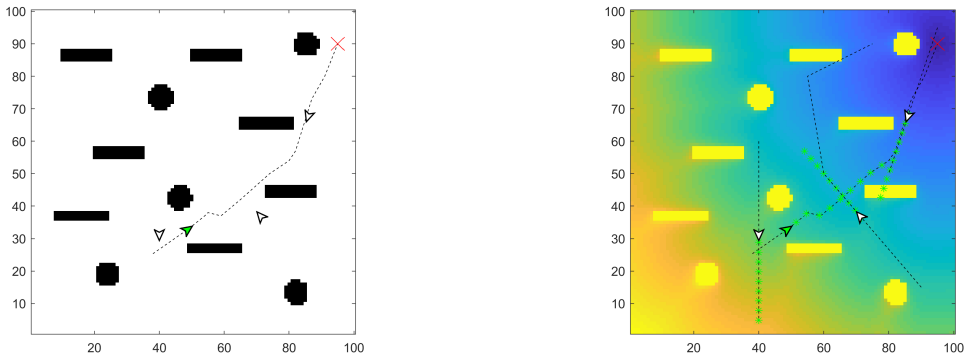


Figure 6.23: Method 2 River Simulation

However, it should be noted that method 2 does not always perform perfectly in such as a difficult environment.

7

Simulations: Performance

To study the performance of both algorithms in more depth, simulations within the environment are performed with random start and goal points for the dynamic obstacles and the USV. The performance is measured by analyzing and comparing the following items:

- The initial length of the path
- The specific COLREGS classification when an encounter occurs
- Whether the path is planned compliant to COLREGS
- The additional length of the generated path compared to the initial optimal path
- Run possible emergency script when collision risk is really high. Additional measures can be adopted such as increasing speed

The performance for the following scenarios is investigated:

- 1 dynamic obstacle in open environment
- 2 dynamic obstacles in open environment
- 3 dynamic obstacles in open environment

A method similar to the reset condition of Section 4.3.3 has been developed to observe if the USV acts according to the regulations in case of a potential collision. It is based on the crossing of areas by the USV around dynamic obstacles. Figure 7.1 illustrates the active areas depending on the collision scenario.

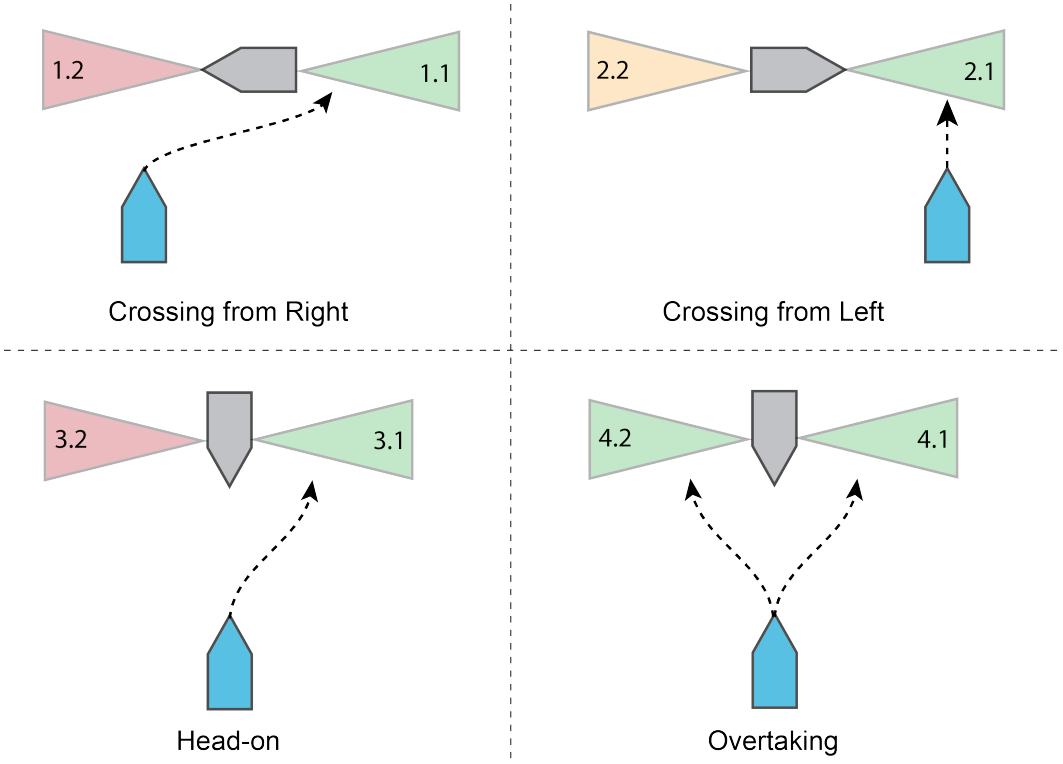


Figure 7.1: Detect USV response to Collision scenarios

So when a potential collision is detected, the situation is classified as one of the four COLREGS class and stored. Next, the above method is utilized to detect how the USV passes the dynamic obstacle. The numbers as indicated in Figure 7.1 are used again in later bar-charts to indicate the response.

A gain in the length weight ω_{length} leads to shorter more optimal paths. However, the response is less impelled to comply with COLREGS regulations and it is more prone to collisions. A gain in $\omega_{COLREGS}$ leads to an increase in the number of correct COLREGS compliant paths. Lastly, it seems that a threshold $t_{crit} > 4$ is high enough to achieve required re-planning performance.

So the parameters that are used for both methods are:

Table 7.1: The parameters used in the single case studies

	Parameters	value	Unit	Description
Method 1	r_{safety}	4	m	The safety radius around the obstacles
	r_{detect}	30	m	The radius within which dynamic obstacles are detected
Method 2	pop	100	-	Total population of random paths
	n_{paths}	40	-	The number of fittest path used for Genetic Algorithm
	n_{gen}	2	-	The number of generations of parent to child
	$p_{combine}$	0.1	-	The probability of combining two parent paths
	p_{mutate_1}	0.7	-	The probability of mutation for a random parent path
	p_{mutate_2}	0.2	-	The probability of mutation for a very fit parent path
	$t_{horizon}$	10	s	The maximum time-horizon
	$d_{col,ref}$	5	-	The threshold distance for collision free fitness
	t_{crit}	4	-	The threshold for COLREGS compliant fitness

Both methods are compared in performance by iterating through 1000 simulations with different randomized start and goal points.

7.1. Encounter with Single Dynamic Obstacle

Firstly, the performance for both methods is tested. It is found that Method 1 based on trailing points has 276 situations with a potential collision and 1 actual scenario in which additional measures need to be adopted. One can further develop a script to correctly implement velocity changes for collision avoidance.

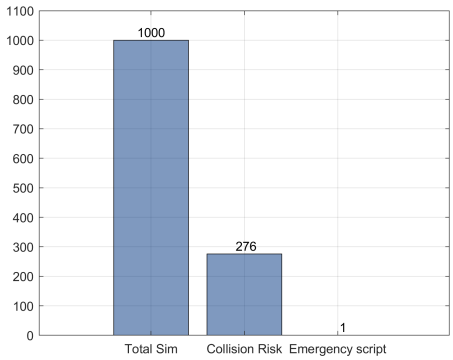


Figure 7.2: Method 1: Trailing Point Collision Data

The COLREGS performance with path re-planning is as required, as illustrated in Figure 7.3. The values are based on the the COLREGS detection algorithm expressed in Figure 7.1. It can be observed that for the re-planned paths for both the right cross and head-on situations the regulations are almost always obeyed.

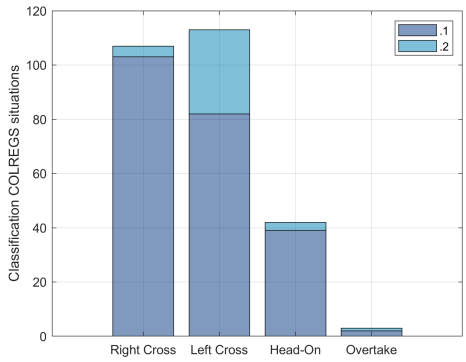


Figure 7.3: Method 1: Trailing Point COLREGS compliancy (.1 & .2 → Figure 7.1)

Lastly, the optimality of the method is analyzed by determining the additional path length compared to the initial path, as shown in Figure 7.4.

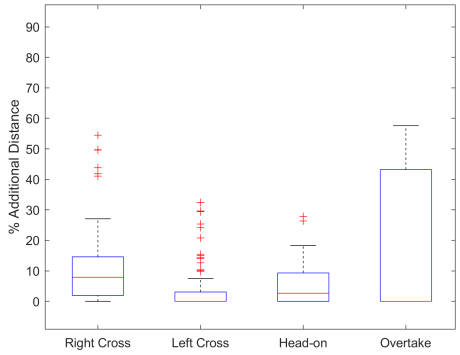


Figure 7.4: Method 1: Trailing Point Rerouting Optimality

The same is done for method 2: the Path Generating Genetic Algorithm and no emergency scenario is needed.

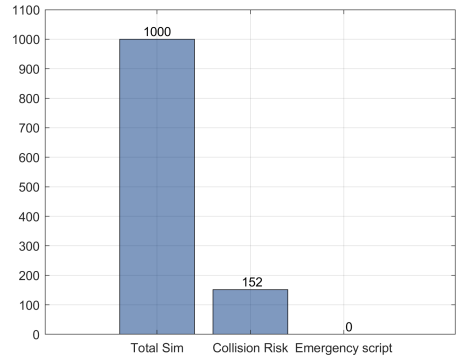


Figure 7.5: Method 2: Genetic Algorithm Collision Data

Figure 7.6 illustrates the actual performance regarding the obedience towards the regulations. It is observed that the percentage of actual correct re-planning is still really high.

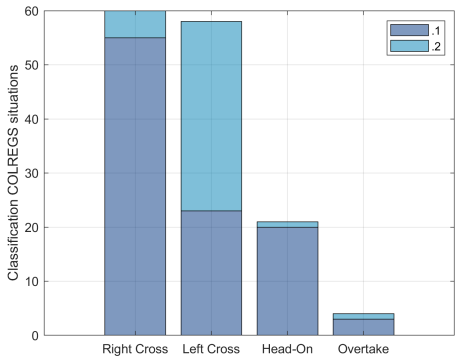


Figure 7.6: Method 2: Genetic Algorithm COLREGS compliancy (.1 & .2 → Figure 7.1)

Furthermore, the optimality of the generated paths comparable for both methods. However, method 2 generates less outlier paths.

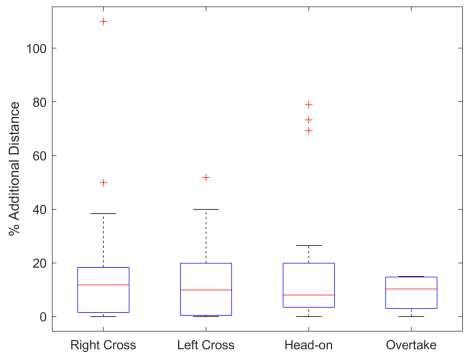


Figure 7.7: Method 2: Genetic Algorithm Rerouting Optimality

7.2. Encountering 2 Dynamic Obstacles

The same procedure is used for analyzing the performance with more dynamic obstacles present in the environment. Now the behavior for 2 dynamic obstacles or 2000 potential collisions is assessed. With multiple obstacles in the vicinity the performance of method 1 decreases. The performance indications for method 1 can be found in Figure 7.8.

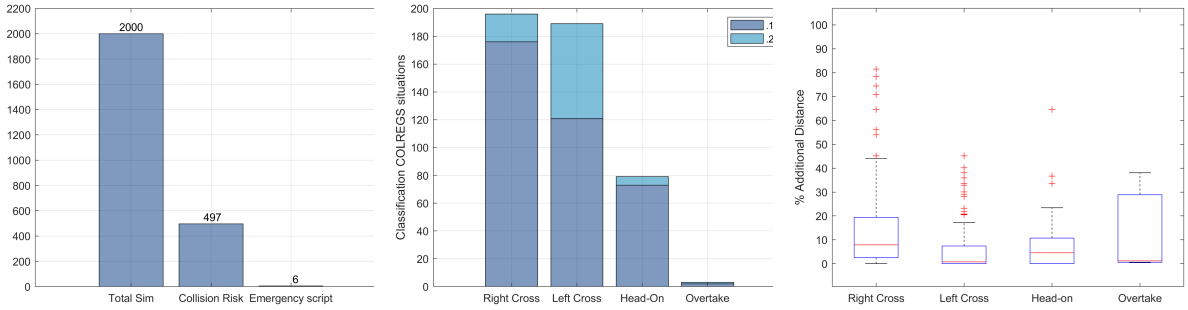


Figure 7.8: The performance of method 1 (trail points) with 2 dynamic obstacles

Moreover, both the path distance of outlier paths and the shear number of outliers have risen sharply.

Method 2 starts to improve in performance compared to method 1 as more dynamic obstacles are in close proximity, because it maintains the same level of correct path re-planning whilst the additional distance for path re-planning remains relatively low. However, the spread of the additional path distance for the overtake situation in Figure 7.8 is quite large. This is due the relatively small number of occurrences and a high outlier. The specific outlier situation (85%) is the result of an overtake where the dynamic obstacle passed right next to the goal point. The USV therefore needs to adapt its path quite drastically as the USV maintains a constant speed.

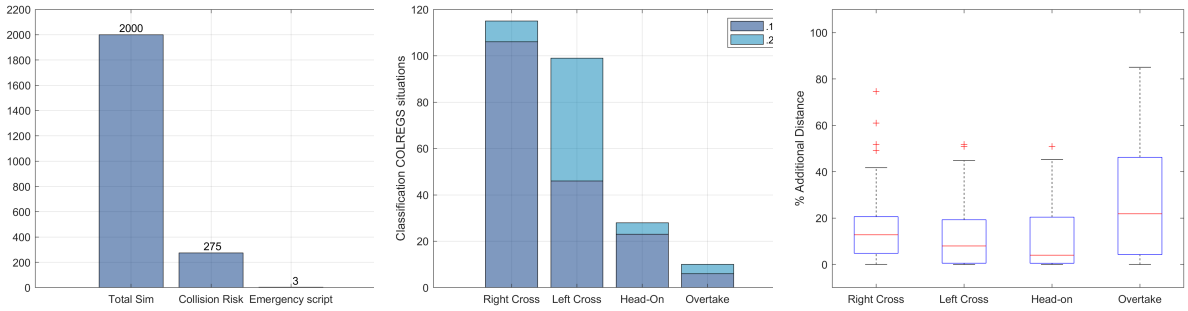


Figure 7.9: The performance of method 2 (Genetic Algorithm) with 2 dynamic obstacles

7.3. Encountering 3 Dynamic Obstacles

For 3 dynamic obstacles surrounding the USV, simulating 3000 potentially colliding paths, the difference in behavior is even more apparent. Method 1 now results in a larger proportion of high risk situations with 29 emergency script executions, whereas method 2 contributes significantly less with only 8 emergency script executions.

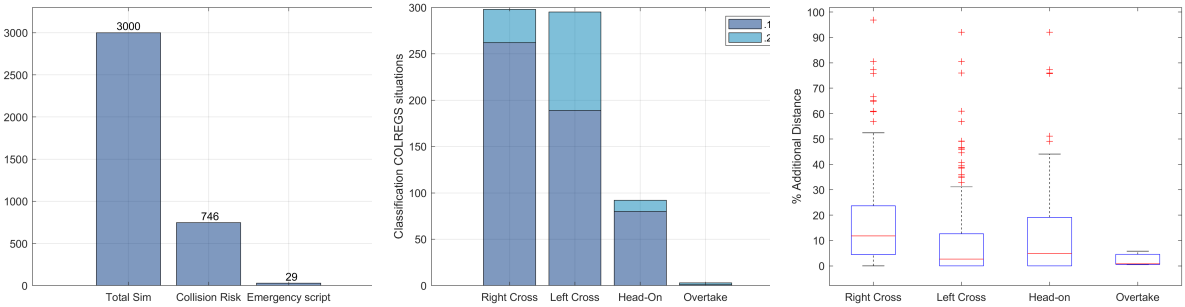


Figure 7.10: The performance of method 1 (trail points) with 3 dynamic obstacles

The compliancy towards the COLREGS regulations reduces for method 1 as well, while it maintains the same percentage for the method 2. Lastly, the additional distance travelled by path re-planning, becomes significantly more with method 1 whereas it maintains relatively the same with method 2.

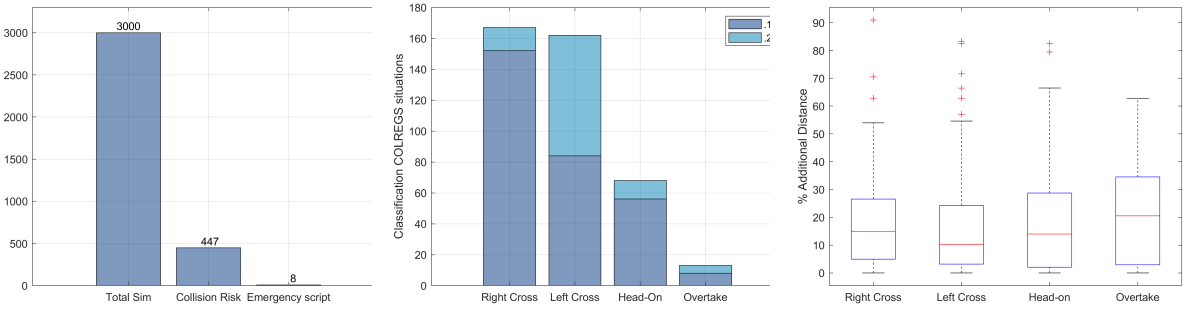


Figure 7.11: The performance of method 2 (Genetic Algorithm) with 3 dynamic obstacles

For method 1 this may be due to incorrect resetting of the conditions to move from trailpoint FMM towards regular FMM, resulting in following dynamic obstacles instead of moving towards the goal point.

Tables 7.2 and 7.3 give an overview of the key performance indicators.

Table 7.2: The simulation results for Method 1

# of simulations	# of obstacles	Potential collision	COLREGS compliant	Emergency script
1000	1	276	97.46 %	1
1000	2	497	94.77 %	6
1000	3	746	93.57 %	29

Table 7.3: The simulation results for Method 2

# of simulations	# of obstacles	Potential collision	COLREGS compliant	Emergency Script
1000	1	152	96.05 %	0
1000	2	275	92.17 %	3
1000	3	447	94.08 %	8

It must be noted that the environment is small and the dynamic obstacles do not respond at all to any potential collision. As mentioned, an additional emergency script for further improving the performance of method 2 based on the path generating Genetic Algorithm can be used to incorporate different levels of velocities along the path such that the USV can adjust its velocity along the path to further help with collision avoidance.

8

Conclusion and Recommendations

The main goal of this thesis was to investigate the potential of the Fast Marching Method to be implemented as a path planner that incorporates the COLREGS regulations during dynamic obstacle avoidance. The motivation to pursue research in this direction was explained in Chapter 1 supported by a review of the current literature in Chapter 2. The results are summarized first, followed by future recommendations based on this thesis research.

8.1. Summary

In general, the autonomous Unmanned Surface Vehicle has three collaborating subsystems: the Navigation, Guidance and Control subsystems. The Navigation System deals with the detection of the surrounding environment through sensors and the identification of its states. The Guidance System focuses on continuously generating and updating optimal, feasible and smooth trajectory commands, whereas the Control System is busy with the actual implementation such that the proper actions are executed. These three systems have to closely cooperate in order to attain a high level of autonomy within the USV, but they can be studied individually for improvement.

The Guidance System deals with the path planning and this subsystem is crucial to ensure that USVs become COLREGS compliant. The path has to be re-planned in such a way that the USV avoids static obstacles as well as dynamic obstacles according to the International Regulations for preventing collisions at sea, or *COLREGS*, explained in Section 3.3.

A PurePursuit controller was used to follow the generated paths and simulate the USV behavior. A simple model was used to focus on the actual path re-planning algorithm instead of the USV dynamics.

The Fast Marching Method has been used as a foundation for tackling the problem of COLREGS compliant path re-planning. The first method is based upon the use of Trail Points when a specific COLREGS situation is identified. For the specific situations of a right crossing or head-on collision, temporary trail points guide the USV past the dynamic obstacles so that the regulations are obeyed. For a detailed description see Chapter 4. The main issue with the conceived first method is the reactive nature and the relatively late response.

To counter the issues related to the first method, a new way was developed through the combined use of the Fast Marching Method and the Genetic Algorithm. In case of an identified COLREGS situation, the Saturated Fast Marching Square Method is used to develop an initial population of semi-random paths towards the goal point. Next the Genetic Algorithm is used to optimize the paths and find a suitable path that satisfies the COLREGS conditions. For a detailed description see Chapter 5.

Both methods have been studied through single case simulations in different environments (plain and with static obstacles present) and a statistical simulation in a plain environment with one up to three dynamic obstacles. The performance of Method 1 based on the Trailing Points reduces with complexity whereas Method 2 based on the Genetic Algorithm is relatively constant. Furthermore, Method 2 can be used in all environments and it is computationally easier.

A flowchart overview of both method 1 and method 2 is given in respectively Figure 4.15 and Figure 5.10. Method 2 is more robust in its path re-planning capabilities compared to method 1. Moreover, the decision space of method 2 can be extended to incorporate different velocities at the different waypoints on the path in the case of a high risk of collision because it already optimizes the paths according to a fitness function. The velocity will become an extra term within the optimization and the controller needs to adjust correspondingly. However, this is more difficult to do with method 1

as the paths are not optimized at all. Therefore, a completely different optimization model needs to be constructed for velocity terms.

8.2. Future Recommendations

There are several additions that could further improve the performance of the guidance subsystem.

- The current methods in the thesis are based upon the USV maintaining a constant speed. It could however be very convenient to also add several velocities within the decision space. The USV could then also increase or decrease its velocity in hazardous situations.
- The methods can potentially be further refined by improving the distinction for different COLREGS situations.
- The methods can also be implemented with a real ship model incorporating additional disturbances such as wind and currents.
- The location and heading direction of the dynamic obstacles is assumed to be known exactly. The performance of the methods can be analyzed when actual implementation of a suitable estimation process is adopted.
- It could be valuable to test the methods in practice by implementing the proposed guidance subsystem within a real USV and testing the true performance.

Bibliography

[1] P. Agrawal and J. M. Dolan. Colregs-compliant target following for an unmanned surface vehicle in dynamic environments. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1065–1070. doi: 10.1109/IROS.2015.7353502.

[2] A.S.K. Annamalai and A. Motwani. A comparison between lqg and mpc autopilots for inclusion in a navigation, guidance and control system. 2013. URL https://pdfs.semanticscholar.org/215b/d67544bafc7ca861347787ec6181b535da1f.pdf.

[3] M. Arzamendia, D. Gregor, D.G. Reina, and S.L. Toral. An evolutionary approach to constrained path planning of an autonomous surface vehicle for maximizing the covered area of ypacarai lake. *Soft Computing*, 2017. ISSN 1433-7479. doi: 10.1007/s00500-017-2895-x. URL https://doi.org/10.1007/s00500-017-2895-x.

[4] H. Ashrafiuon, K.R. Muske, L.C. McNinch, and R.A. Soltan. Sliding-mode tracking control of surface vessels. *IEEE Transactions on Industrial Electronics*, 55(11):4004–4012, 2008. ISSN 0278-0046. doi: 10.1109/TIE.2008.2005933.

[5] M.N. Azzeri, F.A. Adnan, and M.Z.md. Zain. Review of course keeping control system for unmanned surface vehicle. *Jurnal Teknologi (Sciences and Engineering)*, 7(5):11–20, 2015.

[6] I.R. Bertaska, B. Shah, K. von Ellenrieder, P. Švec, W. Klinger, A.J. Sinisterra, M. Dhanak, and S.K. Gupta. Experimental evaluation of automatically-generated behaviors for usv operations. *Ocean Engineering*, 106:496–514, 2015. ISSN 0029-8018. doi: https://doi.org/10.1016/j.oceaneng.2015.07.002. URL http://www.sciencedirect.com/science/article/pii/S0029801815003066.

[7] F. Beser and T. Yildirim. Colregs based path planning and bearing only obstacle avoidance for autonomous unmanned surface vehicles. *Procedia Computer Science*, 131:633–640, 2018. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2018.04.306. URL http://www.sciencedirect.com/science/article/pii/S1877050918306860.

[8] B. Braginsky and H. Guterman. Obstacle avoidance approaches for autonomous underwater vehicle: Simulation and experimental results. *IEEE Journal of Oceanic Engineering*, 41(4):882–892, 2016. ISSN 0364-9059. doi: 10.1109/JOE.2015.2506204.

[9] S. Campbell and W. Naeem. A rule-based heuristic method for colregs-compliant collision avoidance for an unmanned surface vehicle. *IFAC Proceedings Volumes*, 45(27):386–391, 2012. ISSN 1474-6670. doi: https://doi.org/10.3182/20120919-3-IT-2046.00066. URL http://www.sciencedirect.com/science/article/pii/S1474667016312605.

[10] S. Campbell, W. Naeem, and G.W. Irwin. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control*, 36(2):267–283, 2012. ISSN 1367-5788. doi: https://doi.org/10.1016/j.arcontrol.2012.09.008. URL http://www.sciencedirect.com/science/article/pii/S1367578812000430.

[11] G. Casalino, A. Turetta, and E. Simetti. A three-layered architecture for real time path planning and obstacle avoidance for surveillance usvs operating in harbour fields. In *OCEANS 2009-EUROPE*, pages 1–8. doi: 10.1109/OCEANSE.2009.5278104.

[12] C.H. Chen, G. Chen, and J.J. Chen. Design and implementation for usv based on fuzzy control. In *2013 CACS International Automatic Control Conference (CACS)*, pages 345–349, . doi: 10.1109/CACS.2013.6734158.

[13] J. Chen, W. Pan, Y. Guo, C. Huang, and H. Wu. An obstacle avoidance algorithm designed for usv based on single beam sonar and fuzzy control. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2446–2451, . doi: 10.1109/ROBIO.2013.6739838.

[14] H. L. Chiang and L. Tapia. Colreg-rrt: An rrt-based colregs-compliant motion planner for surface vehicle navigation. *IEEE Robotics and Automation Letters*, 3(3):2024–2031, 2018. ISSN 2377-3766. doi: 10.1109/LRA.2018.2801881.

[15] R. Craig Couter. Implementation of the pure pursuit path tracking algorithm. 1992.

[16] M.K. Eidal. *COLREGS Compatible Motion Planning for Autonomous Surface Vessels*. PhD thesis, Norwegian University of Science and Technology, 2018.

[17] Z. Feng, J. Zhu, and R. Allen. Design of continuous and discrete lqi control systems with stable inner loops. *Shanghai Jiaotong University Journal*, 12(6):787–792, 2007.

[18] T.I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. 2011. doi: 10.1002/9781119994138.

[19] R. Halterman and M. Bruch. Velodyne hdl-64e lidar for unmanned surface vehicle obstacle detection. In *SPIE Defense, Security, and Sensing*, volume 7692, page 8. SPIE.

[20] Liang Hu, Wasif Naeem, Eshan Rajabally, Graham Watson, Terry Mills, Zakirul Bhuiyan, and Ivor Salter. Colregs-compliant path planning for autonomous surface vehicles: A multiobjective optimization approach**the authors should like to thank innovate uk, grant reference, tsb 102308, for the funding of this project. *IFAC-PapersOnLine*, 50(1):13662–13667, 2017. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2017.08.2525. URL http://www.sciencedirect.com/science/article/pii/S2405896317334468.

[21] T. Huntsberger, H. Agharian, A. Howard, and D.C. Trotz. Stereo vision–based navigation for autonomous surface vessels. *Journal of Field Robotics*, 28(1):3–18, 2010. ISSN 1556-4959. doi: 10.1002/rob.20380. URL https://doi.org/10.1002/rob.20380.

[22] H. Kim, S. Kim, M. Jeon, J. Kim, S. Song, and K. Paik. A study on path optimization method of an unmanned surface vehicle under environmental loads using genetic algorithm. *Ocean Engineering*, 142:616–624, 2017. ISSN 0029-8018. doi: https://doi.org/10.1016/j.oceaneng.2017.07.040. URL http://www.sciencedirect.com/science/article/pii/S0029801817304122.

[23] W.B. Klinger, I.R. Bertaska, K.D. von Ellenrieder, and M.R. Dhanak. Control of an unmanned surface vehicle with uncertain displacement and drag. *IEEE Journal of Oceanic Engineering*, 42(2):458–476, 2017. ISSN 0364-9059. doi: 10.1109/JOE.2016.2571158.

[24] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger. Safe maritime autonomous navigation with colregs, using velocity obstacles. *IEEE Journal of Oceanic Engineering*, 39(1):110–119, 2014. ISSN 0364-9059. doi: 10.1109/JOE.2013.2254214.

[25] J. Larson, M. Bruch, R. Halterman, J. Rogers, and R. Webster. *Advances in Autonomous Obstacle Avoidance for Unmanned Surface Vehicles*. 2007.

[26] A. Lazarowska. *Multi-criteria ACO-based Algorithm for Ship's Trajectory Planning*, volume 11. 2017. doi: 10.12716/1001.11.01.02.

[27] S. Lee, K. Kwon, and J. Joh. *A fuzzy logic for autonomous navigation of marine vehicle satisfying COLREG guidelines*, volume 2. 2004.

[28] E. Lefeber, K.Y. Pettersen, and H. Nijmeijer. Tracking control of an underactuated ship. *IEEE Transactions on Control Systems Technology*, 11(1):52–61, 2003. ISSN 1063-6536. doi: 10.1109/TCST.2002.806465.

[29] C. Li, Y. Zhao, G. Wang, Y. Fan, and Y. Bai. Adaptive rbf neural network control for unmanned surface vessel course tracking. In *2016 Sixth International Conference on Information Science and Technology (ICIST)*, pages 285–290. doi: 10.1109/ICIST.2016.7483425.

[30] Y. Liu and R. Bucknall. *The angle guidance path planning algorithms for unmanned surface vehicle formations by using the fast marching method*, volume 59. 2016. doi: 10.1016/j.apor.2016.06.013.

[31] Y. Liu, W. Liu, R. Song, and R. Bucknall. Predictive navigation of unmanned surface vehicles in a dynamic maritime environment when using the fast marching method. *International Journal of Adaptive Control and Signal Processing*, 31(4):464–488, 2015. ISSN 0890-6327. doi: 10.1002/acs.2561. URL https://doi.org/10.1002/acs.2561.

[32] Y. Liu, R. Bucknall, and X. Zhang. The fast marching method based intelligent navigation of an unmanned surface vehicle. *Ocean Engineering*, 142:363–376, 2017. ISSN 0029-8018. doi: https://doi.org/10.1016/j.oceaneng.2017.07.021. URL http://www.sciencedirect.com/science/article/pii/S0029801817303979.

[33] Z. Liu, Y. Zhang, X. Yu, and C. Yuan. Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*, 41:71–93, 2016. ISSN 1367-5788. doi: https://doi.org/10.1016/j.arcontrol.2016.04.018. URL http://www.sciencedirect.com/science/article/pii/S1367578816300219.

[34] H. Lyu and Y. Yin. Ship's trajectory planning for collision avoidance at sea based on modified artificial potential field. In *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*, pages 351–357. doi: 10.1109/ICRAE.2017.8291409.

[35] H. Lyu and Y. Yin. *COLREGS-Constrained Real-time Path Planning for Autonomous Ships Using Modified Artificial Potential Fields*. 2018. doi: 10.1017/S0373463318000796.

[36] J. H. Mei, M. R. Rizal, and J. R. Tang. *Colregs-compliant path planning for riverine autonomous surface vessle*, volume 10. 2015.

[37] B. Metcalfe, B. Thomas, A. Treloar, Z. Rymansaib, A. Hunter, and P. Wilson. A compact, low-cost unmanned surface vehicle for shallow inshore applications. In *2017 Intelligent Systems Conference (IntelliSys)*, pages 961–968. doi: 10.1109/IntelliSys.2017.8324246.

[38] S.Y. Moussavi Alashloo, D Ghosh, Y Bashir, and W.I. Yusoff. Influence of error in estimating anisotropy parameters on vti depth imaging. *International Journal of Geophysics*, 2016:1–6, 01 2016. doi: 10.1155/2016/2848750.

[39] W. Naeem, T. Xu, R. Sutton, and A. Tiano. The design of a navigation, guidance, and control system for an unmanned surface vehicle for environmental monitoring. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 222(2):67–79, 2008. ISSN 1475-0902. doi: 10.1243/14750902JEME80. URL https://doi.org/10.1243/14750902JEME80.

[40] W. Naeem, S.C. Henrique, and L. Hu. A reactive colregs-compliant navigation strategy for autonomous maritime navigation. *IFAC-PapersOnLine*, 49(23):207–213, 2016. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2016.10.344. URL http://www.sciencedirect.com/science/article/pii/S2405896316319310.

[41] Wasif Naeem, George W. Irwin, and Aolei Yang. Colregs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics*, 22(6):669–678, 2012. ISSN 0957-4158. doi: https://doi.org/10.1016/j.mechatronics.2011.09.012. URL http://www.sciencedirect.com/science/article/pii/S0957415811001553.

[42] H. Niu, Y. Lu, A. Savvaris, and A. Tsourdos. Efficient path planning algorithms for unmanned surface vehicle. *IFAC-PapersOnLine*, 49(23):121–126, 2016. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2016.10.331. URL http://www.sciencedirect.com/science/article/pii/S2405896316319188.

[43] R. Polvara, S. Sharma, J. Wan, A. Manning, and R. Sutton. Obstacle avoidance approaches for autonomous navigation of unmanned surface vehicles. *Journal of Navigation*, 71(1):241–256, 2017. doi: 10.1017/S0373463317000753.

[44] Y. Qin and X. Zhang. Robust obstacle detection for unmanned surface vehicles. In *Tenth International Symposium on Multispectral Image Processing and Pattern Recognition (MIPPR2017)*, volume 10611, page 6. SPIE.

[45] J.B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, 20(3):38–52, 2000.

[46] G. Rigatos, P. Siano, and N. Zervos. A nonlinear h-infinity control approach for autonomous navigation of underactuated vessels. In *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, pages 1143–1148. doi: 10.1109/ICCAS.2016.7832456.

[47] J.A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996. doi: 10.1073/pnas.93.4.1591. URL http://www.pnas.org/content/pnas/93/4/1591.full.pdf.

[48] B. Shah. *Planning for Autonomous Operation of Unmanned Surface Vehicles*. PhD thesis, University of Maryland, 2016.

[49] S.K. Sharma, R. Sutton, A. Motwani, and A. Annamalai. Non-linear control algorithms for an unmanned surface vehicle. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 228(2):146–155, 2014. doi: 10.1177/1475090213503630. URL http://journals.sagepub.com/doi/abs/10.1177/1475090213503630.

[50] R.G. Simmons. Structured control for autonomous robots. *IEEE Transactions on robotics and automation*, 1994.

[51] A. Sinha, P. Bhardwaj, B. Vaibhav, and N. Mohommad. Research and development of ro-boat: an autonomous river cleaning robot. In *IST/SPIE Electronic Imaging*, volume 9025, page 8. SPIE.

[52] A.J. Sinisterra, M.R. Dhanak, and K. Von Ellenrieder. Stereovision-based target tracking system for usv operations. *Ocean Engineering*, 133:197–214, 2017. ISSN 0029-8018. doi: https://doi.org/10.1016/j.oceaneng.2017.01.024. URL http://www.sciencedirect.com/science/article/pii/S002980181730032X.

[53] X. Sun, G. Wang, Y. Fan, D. Mu, and B. Qiu. An automatic navigation system for unmanned surface vehicles in realistic sea environments. 2018.

[54] T. Temel and H. Ashrafiuon. Sliding-mode speed controller for tracking of underactuated surface vessels with extended kalman filter. *Electronics Letters*, 51(6):467–469, 2015. ISSN 0013-5194. doi: 10.1049/el.2014.4516.

[55] M. Tsou, S. Kao, and C. Su. Decision support from genetic algorithms for ship collision avoidance route planning and alerts. *Journal of Navigation*, 63(1):167–182, 2010. ISSN 0373-4633. doi: 10.1017/S037346330999021X. URL https://www.cambridge.org/core/article/decision-support-from-genetic-algorithms-for-ship-collision-avoidance-route-planning-and-alerts/FF72B65C58902B3C4DB3BF10CADC8560.

[56] N. Wang, S. Guo, J. Yin, Z. Zheng, and H. Zhao. Hybrid feedforward-feedback robust adaptive extreme learning control for euler-lagrange systems. In *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, pages 712–716, . doi: 10.1109/ICACI.2018.8377548.

[57] S. Wang, L. Xie, F. Ma, C. Wu, and K. Xu. Research of obstacle recognition method for usv based on laser radar. In *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, pages 343–348, . doi: 10.1109/ICTIS.2017.8047787.

[58] T. Wang, X. Yan, Y. Wang, and Q. Wu. A collision risk-based ship domain method approach to model the virtual force field. 2018. doi: 10.1155/2018/3984962. URL http://psam14.org/proceedings/paper/paper_279_1.pdf.

[59] TengFei Wang, X. Yan, Yang Wang, and Qing Wu. *Ship Domain Model for Multi-ship Collision Avoidance Decision-making with COLREGs Based on Artificial Potential Field*, volume 11. 2017. doi: 10.12716/1001.11.01.09.

[60] T. Weerakoon, K. Ishii, and A. Ali Forough Nassiraei. An artificial potential field based mobile robot navigation method to prevent from deadlock. 5(3):189–203, 2015. doi: 10.1515/jaiscr-2015-0028.

[61] S. Xie, P. Wu, Y. Peng, J. Luo, D. Qu, Q. Li, and J. Gu. The obstacle avoidance planning of usv based on improved artificial potential field. In *2014 IEEE International Conference on Information and Automation (ICIA)*, pages 746–751. doi: 10.1109/ICInfA.2014.6932751.

[62] J. Yang, C. Tseng, and P.S. Tseng. Path planning on satellite images for unmanned surface vehicles. *International Journal of Naval Architecture and Ocean Engineering*, 7(1):87–99, 2015. ISSN 2092-6782. doi: https://doi.org/10.1515/ijnaoe-2015-0007. URL http://www.sciencedirect.com/science/article/pii/S2092678216301030.

[63] F. Yue-wen, L. Yu-lei, W. Bo, L. Ye, S. Hai-long, and L. Xiao. An improved unscented kalman filter and the application to heading control of micro unmanned surface vehicles. In *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, pages 1466–1471. doi: 10.1109/CGNCC.2016.7829005.

[64] H. Zheng, R.R. Negenborn, and G. Lodewijks. Trajectory tracking of autonomous vessels using model predictive control. *IFAC Proceedings Volumes*, 47(3):8812–8818, 2014. ISSN 1474-6670. doi: https://doi.org/10.3182/20140824-6-ZA-1003.00767. URL http://www.sciencedirect.com/science/article/pii/S1474667016430041.

[65] E. Zitzler and L. Thiele. Evolutionary algorithms for multiobjective optimization: The strength pareto approach. *TIK-Report*, May 1998. doi: https://doi.org/10.3929/ethz-a-004288833.