IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 28, 2021, accepted July 27, 2021, date of publication August 9, 2021, date of current version August 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3103323

Two Space-Time Obstacle Representations Based on Ellipsoids and Polytopes

ANDREAS B. MARTINSEN[1] AND ANASTASIOS M. LEKKAS[1,2]
[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7491 Trondheim, Norway
[2]Centre for Autonomous Marine Operations and Systems, Norwegian University of Science and Technology, 7491 Trondheim, Norway

Corresponding author: Andreas B. Martinsen (andreas.b.martinsen@ntnu.no)

ABSTRACT When operating autonomous surface vessels in uncertain environments with dynamic obstacles, planning safe trajectories and evaluating collision risk is key to navigating safely. In order to perform these tasks, it is important to have a computationally efficient and adaptable obstacle representation to allow for quick and robust predictions of the obstacle trajectory. This paper presents a novel space-time obstacle representation, which is able to predict the reachable set for a dynamic obstacle under uncertainty. This is done by projecting the area occupied by the obstacle forward in time, using a set of velocities representing the possible maneuvers that the obstacle may take. Under some mild assumptions, we show how the space-time obstacle can be implemented in a computationally efficient way, using both convex polytopes and ellipsoids. Additionally, we show how the space-time obstacle representation can be used for risk assessment, collision avoidance and trajectory planning for autonomous surface vessels.

INDEX TERMS Autonomous surface vehicles, collision avoidance, marine vehicles, motion planning, optimal control, trajectory optimization, risk assessment.

I. INTRODUCTION

With increasing interest in autonomy solutions in the maritime industry, it becomes increasingly important to develop robust and efficient methods for risk assessment and collision avoidance (COLAV). This is especially true for dynamic obstacles, for which accurate trajectory predictions is complicated by numerous uncertainties. A major component of developing robust and efficient methods, is the underlying obstacle representation. While the literature is for the most part concerned with COLAV and risk assessment methods, where the obstacle representation is chosen to fit the algorithm. The goal of this article is to create awareness around the representation, and showing some of the befits of building a COLAV and risk assessment method around an obstacle representation, instead of the obstacle representation being built around the method. In order for the obstacle representation to be practical, it needs to be able to capture the shape and movement of the obstacle in a way that is computationally efficient, allowing for real-time risk assessment, planning and decision making. Additionally the obstacle representation must be robust, allowing for both measurement uncertainties, as well as uncertainties in the obstacle behavior.

One of the first obstacle representations to be used for assessing collision risk is the closest point of approach (CPA) [1], which computes the distance and point in time when two vessels are at their closest, given that the vessels have a known constant velocity. CPA was initially developed to give human readable feedback to navigators on the risk associated with the speed and course of the vessel, but has more recently been incorporated into automated COLAV systems [2], [3]. Based on the same idea as CPA, the velocity obstacle (VO) representation [4], [5], computes the set of velocities which lead to a collision, i.e. giving a distance at the closest point of approach (dCPA) of zero, and a time of closest point of approach (tCPA) greater then zero. The VO approach has seen widespread use, as it allows for easily assessing if a given velocity vector is collision free. Additional extension to the VO representation allow for kinematic constraints, obstacle behaviour and uncertainty [6]–[8], with similar methods such as dynamic window (DW) methods, allowing for dynamic constraints [9]. While these obstacle representations are fairly computationally efficient, they come with some major drawbacks, mainly that they considering only a single maneuver, such as a constant velocity or

The associate editor coordinating the review of this manuscript and approving it for publication was Xiwang Dong.

turning rate. This makes the methods suitable for short term collision avoidance, but is only of limited use for long term planning and risk assessment of multiple complex maneuvers. In order to allow for longer term planning, a common approach is to use set based obstacle representation, where the set of points making up the obstacles is computed and projected forward in time. For surface vessels this is typically done using either circles and ellipsoids [10]–[14], or polytopes [15], [16]. While these methods allow for longer term planning, they usually assume that the obstacle velocity is exactly known over some prediction horizon, meaning that these methods are limited in terms of the robustness under both measurement and behaviour uncertainty of the obstacle. In order to account for obstacle uncertainty, the most accurate methods incorporate the obstacle uncertainty, this can be done by computing the reachable sets for the obstacle [17], [18], or using probabilistic methods [6], [19]–[22] for predicting the behaviour of the obstacles. These methods allow for accurate obstacle predictions, but are often less flexible and more computationally expensive than what is ideal for a general purpose obstacle representation.

In order to address some of the drawbacks of existing methods, we propose a novel space-time obstacle representation, which is able to predict the set of states which a dynamic obstacle may occupy, given uncertainty in both measurements, as well as the future behaviour of the obstacle. The proposed space-time representation is a set based representation, as the area occupied by the vessel is projected forward in time. Contrary to other set based approaches however, the proposed space-time representation uses a set of velocities representing the possible maneuvers that the obstacle may take. This ensures robustness to both measurement and behavioural uncertainty similarly to probabilistic methods. In addition to developing a theoretical framework for the proposed space-time representation, we also show how a space-time obstacle can be efficiently implemented both as convex polytopes and ellipsoids, in addition to showing how the space-time obstacle may be used for both risk assessment and trajectory planning. The main contributions of this paper are as follows:

- The development of a novel space-time obstacle representation for predicting the possible future trajectories of an obstacle under uncertainty.
- Implementation of the space-time obstacle using both a convex polytope representation, and an ellipsoid representation.
- We provide examples of how the space-time representation can be used both for risk assessment and trajectory planning for surface vessels.

The rest of the paper is structured as follows: Section II introduces the space-time obstacle, and shows how it can be implemented using both polytopes and ellipsoids. Section III shows how the space-time obstacle can be used for both trajectory planning and risk assessment, and Section IV concludes the paper.

II. SPACE-TIME OBSTACLE REPRESENTATION

When performing obstacle avoidance, we need a way of representing the obstacle. This can be done by representing the obstacle as the set $\mathcal{O}$ of all space-time coordinates $(\boldsymbol{x}, t)$ that the obstacle can occupy. For static obstacles, the obstacle representation remains the same at for all time, however this is no longer the case when faced with dynamic obstacles. When planning safe trajectories, it is important to be able to account for the movement of the obstacle in order to safely avoid it. Given an obstacle $\mathcal{O}_0$ at time $t = 0$, with a velocity vector $\boldsymbol{v}$ we can predict what the obstacle will look like in the future as:

$$\mathcal{O}_t = \mathcal{O}_0 + (\boldsymbol{v} \cdot t) \tag{1}$$

This will work in the case where the obstacle is deterministic, and we know its initial area $\mathcal{O}_0$ and future velocity $\boldsymbol{v}$. However in most real world applications, this is not the case, as we may only have noisy estimates of position and velocity, and the obstacle may speed up or slow down. In order to account for this, we propose using a set of feasible velocities $\mathcal{V}$, which represents the uncertainty about the measurement and behaviour of the obstacle. This gives the following obstacle prediction:

$$\mathcal{O}_t = \mathcal{O}_0 \oplus (\mathcal{V} \cdot t) \tag{2}$$

where $\oplus$ denotes the *Minkowski sum*, i.e. the point-wise sum between two sets. This can be further generalized into space-time coordinates $(\boldsymbol{x}, t)$, as a robust space-time obstacle representation:

$$\mathcal{O} = \{(\boldsymbol{x}, t) \mid \boldsymbol{x} \in \mathcal{O}_t\} \tag{3}$$

Using this robust space-time obstacle representation, where the true obstacle lies within the initial obstacle set $\mathcal{O}_0$, and the obstacle velocity lies within the velocity set $\mathcal{V}$, then space-time coordinates that do not fall within the set $\mathcal{O}$ are guaranteed to be collision free.

A. POLYTOPE SPACE-TIME OBSTACLE REPRESENTATION

One way of efficiently computing the robust space-time obstacle representation, is to use convex polytopes. Given a set of points $\mathcal{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots \boldsymbol{s}_n\}$ we can compute a convex polygon containing all the points as the convex hull of the points.

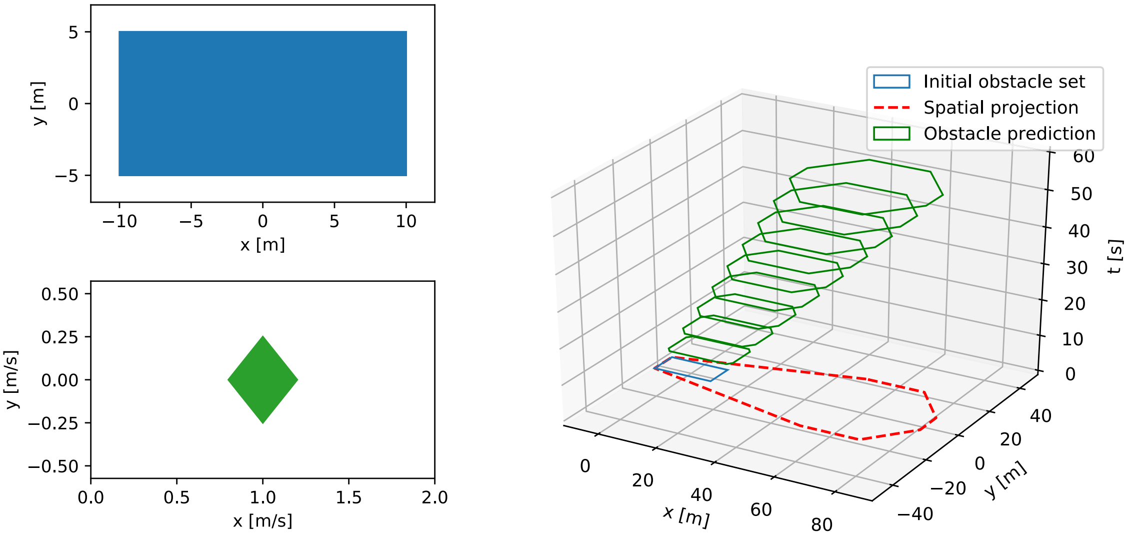
$$\mathrm{Conv}(\mathcal{S}) \tag{4}$$

Using this we can define the obstacle and velocity set in terms of the convex hull of a finite set of points.

$$\mathcal{O}_0 = \mathrm{Conv}(\{\boldsymbol{o}_1, \cdots \boldsymbol{o}_n\}), \quad \mathcal{V} = \mathrm{Conv}(\{\boldsymbol{v}_1, \cdots \boldsymbol{v}_n\}) \tag{5}$$

A useful property of the convex hull is that the convex hull and Minkowski sum are commutative operations. This means that for the two sets $\mathcal{O}_o$ and $\mathcal{V}$, the following equality holds:

$$\begin{aligned}\mathcal{O}_t &= \mathcal{O}_0 \oplus (\mathcal{V} \cdot t)\\ &= \mathrm{Conv}(\{\boldsymbol{o}_1, \cdots \boldsymbol{o}_n\}) \oplus \mathrm{Conv}(\{\boldsymbol{v}_1 \cdot t, \cdots \boldsymbol{v}_n \cdot t\})\\ &= \mathrm{Conv}(\{\boldsymbol{o}_1, \cdots \boldsymbol{o}_n\} \oplus \{\boldsymbol{v}_1 \cdot t, \cdots \boldsymbol{v}_n \cdot t\}),\end{aligned} \tag{6}$$



(a) Obstacle and velocity sets

(b) Space-time obstacle representation

FIGURE 1. Given the obstacle and velocity sets in (a), we can compute the resulting space-time obstacle representation in (b). Where the blue lines represent the initial obstacle, green lines represent the obstacle at different times, and the red dashed line represents the spatial projection of the space-time obstacle.

hence the resulting obstacle prediction remains convex [23]. In the case of a three dimensional space time, we can formulate the robust space-time obstacle representation in terms of the corresponding half-space representation:

$$\mathcal{O} = \left\{ (\boldsymbol{x}, t) \,\middle|\, \boldsymbol{A}_o \begin{bmatrix} \boldsymbol{x} \\ t \end{bmatrix} \le \boldsymbol{b}_o \right\}. \tag{7}$$

Given the extreme points (vertexes) of $\mathcal{O}_t$, in a counter clockwise order at arbitrarily chosen times t_1 and t_2 where $0 \le t_1 < t_2$, as:

$$\begin{aligned} \text{Vertex}(\mathcal{O}_{t_1}) &= \{\boldsymbol{o}_{t_1,1}, \boldsymbol{o}_{t_1,2} \ldots \boldsymbol{o}_{t_1,n}\} \\ \text{Vertex}(\mathcal{O}_{t_2}) &= \{\boldsymbol{o}_{t_2,1}, \boldsymbol{o}_{t_2,2} \ldots \boldsymbol{o}_{t_2,n}\}, \end{aligned} \tag{8}$$

the half space representation is given by the following:

$$\begin{aligned} \boldsymbol{A}_{o,i} &= \begin{bmatrix} \boldsymbol{o}_{t_2,i} +1 - \boldsymbol{o}_{t_2,i} \\ 0 \end{bmatrix} \times \begin{bmatrix} \boldsymbol{o}_{t_1,i} - \boldsymbol{o}_{t_2,i} \\ t_1 - t_2 \end{bmatrix} \\ \boldsymbol{b}_{o,i} &= \boldsymbol{A}_{o,i} \begin{bmatrix} \boldsymbol{o}_{t_2,1} \\ t_2 \end{bmatrix}, \end{aligned} \tag{9}$$

where $\boldsymbol{A}_{o,i}$ and $\boldsymbol{b}_{o,i}$ are the i-th rows of $\boldsymbol{A}_o$ and $\boldsymbol{b}_o$ respectively. We should also note that the × operator denotes the cross product of two vectors, and $\boldsymbol{o}_{t_1,n+1} = \boldsymbol{o}_{t_1,1}$. Using the above space-time obstacle representation, it is straightforward to check if a given space-time coordinate $(\boldsymbol{x}, t)$ will result in a collision with the obstacle. This is done simply by algebraically evaluating the matrix inequality given in (7), which has a computational complexity which grows linearly with the number of vertices in the initial obstacle set $\mathcal{O}_0$ and the velocity set $\mathcal{V}$.

Using the proposed polytope space-time obstacle representation, with the obstacle and velocity set given as follows:

$$\mathcal{O}_0 = \text{Conv}\left(\begin{bmatrix} -10 \\ -5 \end{bmatrix}, \begin{bmatrix} -10 \\ 5 \end{bmatrix}, \begin{bmatrix} 10 \\ 5 \end{bmatrix}, \begin{bmatrix} 10 \\ -5 \end{bmatrix} \right) \tag{10}$$

$$\mathcal{V} = \text{Conv}\left(\begin{bmatrix} 0.8 \\ 0.00 \end{bmatrix}, \begin{bmatrix} 1.0 \\ -0.25 \end{bmatrix}, \begin{bmatrix} 1.2 \\ 0.00 \end{bmatrix}, \begin{bmatrix} 1.0 \\ 0.25 \end{bmatrix} \right), \tag{11}$$

we get the space-time obstacle representation given in Figure 1. From the figure we see that the obstacle representation continues to grow with time. This is caused by the uncertainty in the velocity being compounded over time.

B. ELLIPSOID SPACE-TIME OBSTACLE REPRESENTATION

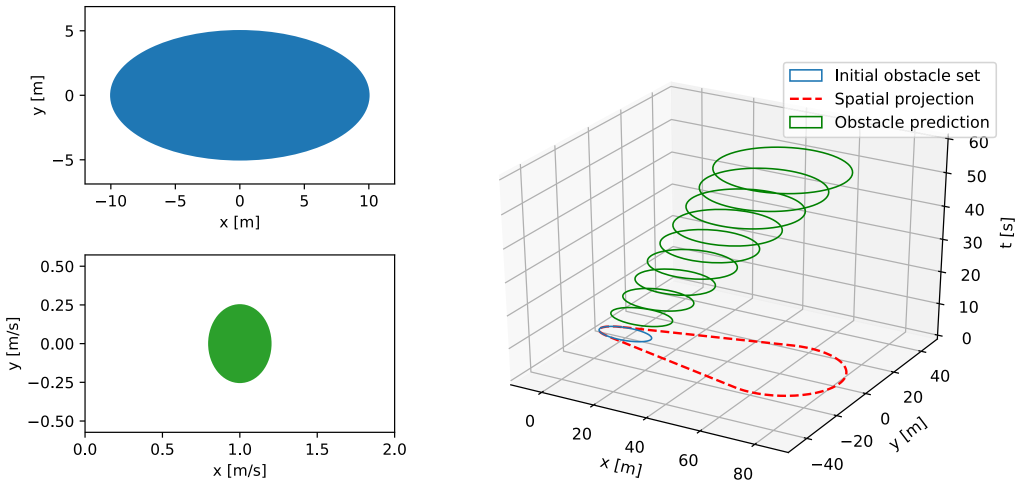
The robust space-time obstacle representation may also be efficiently computed using an ellipsoidal set representation. We use the basic definition of an ellipsoid:

$$E(\boldsymbol{p}, \boldsymbol{Q}) = \{\boldsymbol{x} \in \mathbb{R}^n | (\boldsymbol{x} - \boldsymbol{p})^\top \boldsymbol{Q}^{-1} (\boldsymbol{x} - \boldsymbol{p}) \le 1\}, \tag{12}$$

where $\boldsymbol{p} \in \mathbb{R}^n$ is the ellipse center, and $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ is a positive definite shape matrix. In order to compute the robust space-time obstacle representation we must be able to compute the Minkowski sum $E(p_1, Q_1) \oplus E(p_2, Q_2)$ between two arbitrary ellipsoids. Unfortunately, the Minkowski sum of two ellipsoid is in general not an ellipsoid. However it is possible to formulate an ellipsoidal outer approximation:

$$\begin{aligned} & E(\boldsymbol{p}_1, \boldsymbol{Q}_1) \oplus E(\boldsymbol{p}_2, \boldsymbol{Q}_2) \\ & \quad \subset E(\boldsymbol{p}_1 + \boldsymbol{p}_2, (1 + c^{-1})\boldsymbol{Q}_1 + (1 + c)\boldsymbol{Q}_2) \quad \forall c > 0 \end{aligned} \tag{13}$$

Moreover, the minimizer of the trace and hence the sum of the eigenvalues of the resulting symmetric shape matrix is



(a) Obstacle and velocity sets

(b) Space-time obstacle representation

FIGURE 2. Given the obstacle and velocity sets in (a), we can compute the resulting space-time obstacle representation in (b). Where the blue lines represent the initial obstacle, green lines represent the obstacle at different times, and the red dashed line represents the spatial projection of the space-time obstacle.

analytically given as:

$$c = \sqrt{\frac{\mathrm{Tr}(\boldsymbol{Q}_1)}{\mathrm{Tr}(\boldsymbol{Q}_2)}} \tag{14}$$

Given an initial obstacle and velocity estimate as the ellipsoidal sets:

$$\mathcal{O}_0 = E(\boldsymbol{p}_o, \boldsymbol{Q}_o), \quad \mathcal{V} = E(\boldsymbol{p}_v, \boldsymbol{Q}_v), \tag{15}$$

where the scaled velocity ellipsoid $\mathcal{V} \cdot t$ is given as:

$$\mathcal{V} \cdot t = E\left(\boldsymbol{p}_v \cdot t, \boldsymbol{Q}_v \cdot t^2\right), \tag{16}$$

we can use (13) to formulate an outer approximation of the robust obstacle representation (2) as follows:

$$\mathcal{O}_t \subset E(\boldsymbol{p}_t, \boldsymbol{Q}_t) \tag{17}$$

where

$$\begin{aligned} \boldsymbol{p}_t &= \boldsymbol{p}_o + \boldsymbol{p}_v \cdot t \\ \boldsymbol{Q}_t &= \left(1 + \frac{t}{d}\right)\boldsymbol{Q}_o + \left(t^2 + d \cdot t\right)\boldsymbol{Q}_v \\ d &= \sqrt{\frac{\mathrm{Tr}(\boldsymbol{Q}_o)}{\mathrm{Tr}(\boldsymbol{Q}_v)}} \end{aligned} \tag{18}$$

Using this ellipsoidal outer approximation, we define the ellipsoidal robust space-time obstacle representation as follows:

$$\mathcal{O} = \{(\boldsymbol{x}, t) \mid (\boldsymbol{x} - \boldsymbol{p}_t)^\top \boldsymbol{Q}_t^{-1}(\boldsymbol{x} - \boldsymbol{p}_t) \leq 1\}. \tag{19}$$

We can note that using the above space-time obstacle representation, it is straightforward to check if a given space-time coordinate $(\boldsymbol{x}, t)$ will result in a collision with the obstacle, as it simply involves algebraically evaluating the inequality in (19), which has a constant computation time.

Using the proposed ellipsoid space-time obstacle representation, with the obstacle and velocity set given as follows:

$$\mathcal{O}_0 = E\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}10^2 & 0\\0 & 5^2\end{bmatrix}\right) \tag{20}$$

$$\mathcal{V} = E\left(\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0.2^2 & 0\\0 & 0.25^2\end{bmatrix}\right), \tag{21}$$

we get the space-time obstacle representation given in Figure 2. Similarly to the polytope representation, we see that the obstacle representation continues to grow with time in order to account for the increasing uncertainty.

C. SPACE-TIME OBSTACLE RISK ASSESSMENT

The robust space-time obstacle representations presented in the previous sections, show how we can compute a space-time volume which the obstacle may occupy, given an initial obstacle area and a set of obstacle velocities. For some applications, this may be overly restrictive, as the true obstacle will take only one velocity within the set of possible velocities. In this case it may be more useful to reason about the risk associated with the space-time obstacle. One way of reasoning about the risk is to consider the size of the velocity set $\mathcal{V}$. Writing the velocity uncertainty as:

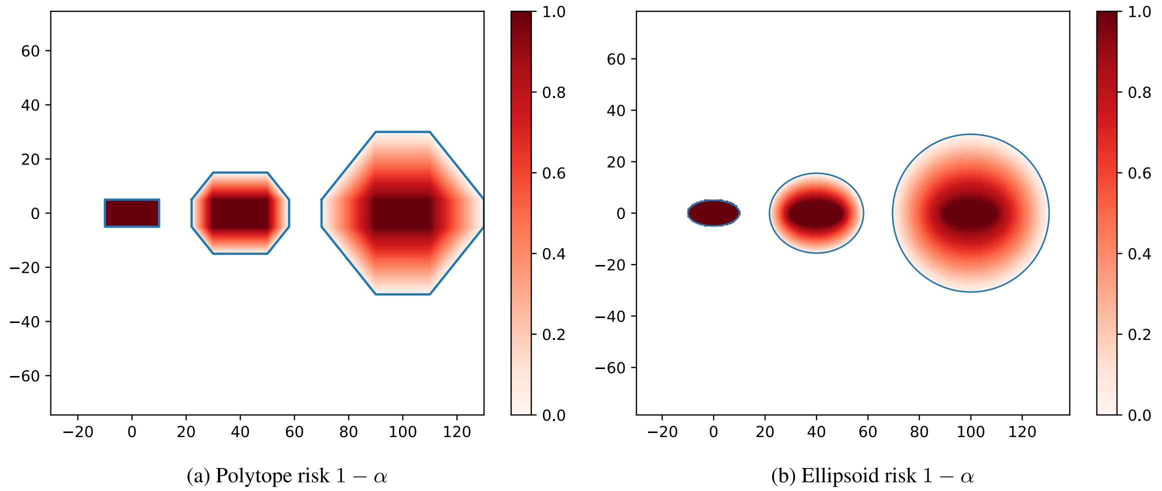
$$\mathcal{V} = \boldsymbol{v} + \alpha \cdot \mathcal{V}_0 \tag{22}$$

where $\boldsymbol{v}$ is the geometric center of the velocity set $\mathcal{V}$, $\mathcal{V}_0$ is the velocity uncertainty with geometric center at the origin, and $\alpha \in [0, 1]$ is the velocity uncertainty scaling. Using this we can define the scaled obstacle prediction as:

$$\mathcal{O}_{t,\alpha} = \mathcal{O}_0 \oplus (\boldsymbol{v} + \alpha \cdot \mathcal{V}_0) \cdot t, \tag{23}$$

and the scaled space-time obstacle as:

$$\mathcal{O}_\alpha = \{(\boldsymbol{x}, t) \mid \boldsymbol{x} \in \mathcal{O}_{t,\alpha}\}. \tag{24}$$



(a) Polytope risk $1 - \alpha$

(b) Ellipsoid risk $1 - \alpha$

FIGURE 3. Risk for the polytope and ellipsoid space-time obstacle at 0, 40 and 100 seconds.

Using the above formulation, we can represent the risk as $1 - \alpha$ where α is chosen as the minimum scaling for which a space-time coordinate $(\boldsymbol{x}, t)$ is within the space-time obstacle $\mathcal{O}_\alpha$. This can be formulated as the following optimization problem:

$$\begin{aligned} \min_{\alpha} \; & \alpha \\ \text{s.t. } & (\boldsymbol{x}, t) \in \mathcal{O}_\alpha \\ & \alpha \in [0, 1] \end{aligned} \tag{25}$$

Solving the above optimization problem is in general quite computationally expensive, however utilizing the properties of the polytope and ellipsoid representation, the constrained optimization problem above, has a closed form solution in the case of the polytope, and can be transformed to a unconstrained optimization problem for the ellipsoid. For the examples given in Figure 1 and 2, we get the risk seen in Figure 3. It should be noted that this measure of risk is not a measure of probability, but rather represents the degree of uncertainty in the set of velocities that the obstacle can take.

III. APPLICATION

In this section we will show how the proposed space-time obstacle representation can be used for risk assessment, collision avoidance (COLAV) and trajectory planning for autonomous surface vessels (ASVs).

A. VO AND CPA CONVERSION

Some of the most common COLAV methods used today rely on evaluating multiple candidate velocities using either closest point of approach (CPA), or velocity obstacles (VO). Given a candidate velocity $\boldsymbol{v}$ and an initial position $\boldsymbol{x}_0$, making up the straight line trajectory $\boldsymbol{x}_v(t) = \boldsymbol{x}_0 + \boldsymbol{v} \cdot t$, we can evaluate the CPA and VO by finding the time that minimizes the distance between the $\boldsymbol{x}_v(t)$ and a point $(\boldsymbol{x}, t)$ in the space-time obstacle. This can be formulated as the following optimization problem:

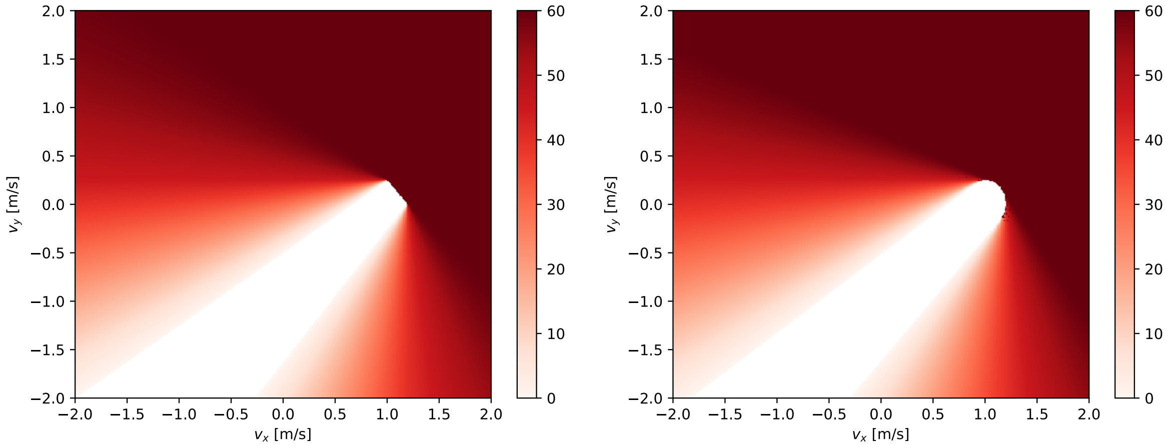
$$\begin{aligned} \min_{\boldsymbol{x},t} \; & ||\boldsymbol{x}_v(t) - \boldsymbol{x}||^2 \\ \text{s.t. } & (\boldsymbol{x}, t) \in \mathcal{O}, \\ & t \geq 0. \end{aligned} \tag{26}$$

Given the solution $(\boldsymbol{x}, t)$ of the optimization problem, t is the tCPA, $\boldsymbol{x}_v(t)$ is the closest point of approach, and $||\boldsymbol{x}_v(t) - \boldsymbol{x}||$ is the dCPA. If the distance $||\boldsymbol{x}_v(t) - \boldsymbol{x}||$ at the CPA is zero, then the candidate velocity lies within the VO, and is considered unsafe. We can note that the optimization problem in (26) is a quadratic programming problem for the polytope representation, and a nonlinear programming problem for the elliptical obstacle representation.

For the space-time obstacles in Figure 1 and 2, and an initial position $\boldsymbol{x}_0 = [50, 50]^\top$, we get the CPA and VO seen in Figure 4. Using this, automatic COLAV can be performed by choosing a velocity outside of the VO. Additionally, collisions can be avoided with a specified margin, by choosing a velocity with a large enough dCPA.

B. SIMPLE PATH-TIME PLANNER

In some circumstances, a preplanned path may be given, and the goal during transit, is to follow the path as closely as possible. When dynamic obstacles are introduced, the trajectory planning problem is reduced to safely regulating the velocity along the preplanned path in order to perform COLAV. This problem is particularly interesting in confined waters, where we often find predetermined shipping lanes for larger vessels, and set routes for regularly scheduled traffic such as ferries. One such path-time decomposition approach to COLAV, called path-velocity decomposition, was first introduced in [24], where a path-time obstacle representation was used together with graph search methods in order to plan collision free trajectories following predetermined paths. Since then the method has been used in a number of applications, including COLAV for small urban passenger



(a) Polytope

(b) Ellipsoid

FIGURE 4. Distance to obstacle at closest point of approach (dCPA) shaded, and velocity obstacle (VO) in white.

ferries in confined waters [25]. In this section we will show how our proposed space-time obstacle representation, is a generalisation of path-time coordinates used in [24], and how the space-time obstacle representation can be used to easily introduce uncertainty and risk into the path-time decomposition approach.

1) SPACE-TIME TO PATH-TIME PROJECTION

Given a space-time obstacle, the corresponding path-time obstacle is given as the projection of the space-time obstacle along the desired path. In the case of a polytopic space time obstacle on the form:

$$\mathcal{O} = \left\{ (\boldsymbol{x}, t) \,\middle|\, \boldsymbol{A}_o \begin{bmatrix} \boldsymbol{x} \\ t \end{bmatrix} \leq \boldsymbol{b}_o \right\}, \tag{27}$$

and a straight line path on the form:

$$\boldsymbol{x}(s) = \boldsymbol{x}_0 + \boldsymbol{n} \cdot s, \tag{28}$$

where $\boldsymbol{x}_0$ is the initial position, $\boldsymbol{n}$, where $||\boldsymbol{n}|| = 1$, is the path direction, and s is the distance along the path. We can compute the path-time obstacle $\mathcal{O}_p$ as the projection of the space-time obstacle $\mathcal{O}$ as follows:

$$\mathcal{O}_p = \left\{ (s, t) \,\middle|\, \boldsymbol{A}_o \begin{bmatrix} \boldsymbol{n} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s \\ t \end{bmatrix} \leq \boldsymbol{b}_o - \boldsymbol{A}_o \begin{bmatrix} \boldsymbol{x}_0 \\ t_0 \end{bmatrix} \right\}. \tag{29}$$

Given the space-time obstacle and path in Figure 5a, the path projection gives th path-time diagram in Figure 5b.

2) EXAMPLE

Given a desired path, which is collision free with respect to static obstacles such as a land, we can combine the proposed space-time obstacle representation, and the path-time planning approach from [24], in order to plan an optimal velocity profile along the predetermined desired path, which ensures COLAV with respect to dynamic space-time obstacles. The resulting planning algorithm can be described as follows:

1) From obstacle tracks, compute the space-time representation of the vessel, predicting the obstacle movement and uncertainty into the future (Figure 6a).
2) Project the space-time obstacles onto the predetermined desired path, giving a path-time obstacle diagram as seen in Figure 6b.
3) Using a graph search algorithm such as Dijkstra *et al.* [26] or A* [27] with a given cost function, plan a sequence of constant velocities (straight lines on the path-time diagram in Figure 6b), between vertices of the path-time obstacles, which do not intersect with the path-time obstacles. The planned sequence of velocities then give an optimal collision free trajectory.

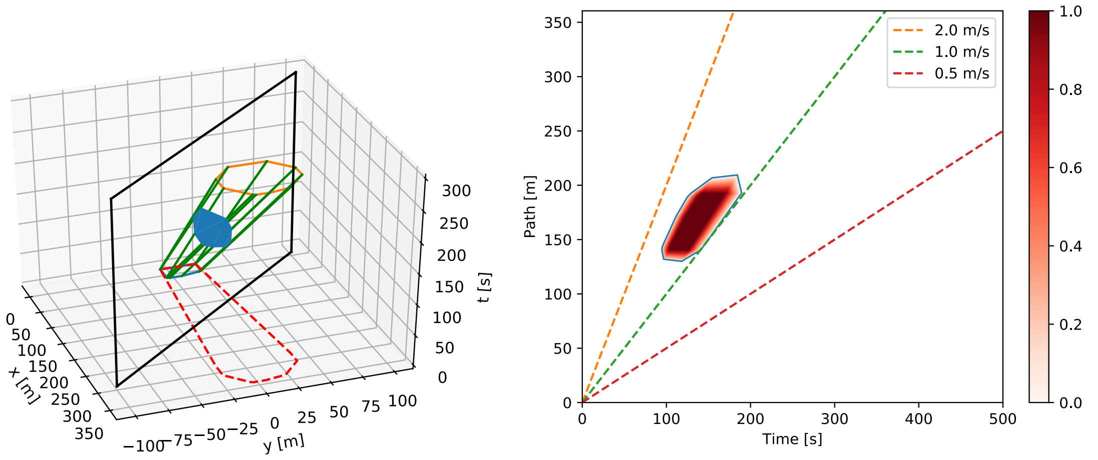
In Figure 6, we show a simple scenario from the Trondheim harbour, where three dynamic obstacles moving with different velocity uncertainties. Using time as the optimization objective with three different maximum velocities, we get the time optimal trajectories seen in Figure 6b. This planning method can be further generalized to allow for switching between multiple paths, similar to [25], and can be modified to allow for trajectories with none-zero risk, in order to allow for planning more optimal trajectories at the cost of higher risk.

C. SIMPLE SPACE-TIME PLANNER

For the path-time planning approach in the previous section, the trajectory is restricted to lie on a predetermined path. In many situations, this may be overly restrictive, and allowing for free movement is the better option. This includes following the International Regulations for Preventing Collisions at Sea (COLREGs), were clear maneuvers conveying the vessel intentions is required, and situations where evasive action is needed to avoid collision.

1) DUBINS TRAJECTORY PLANNER

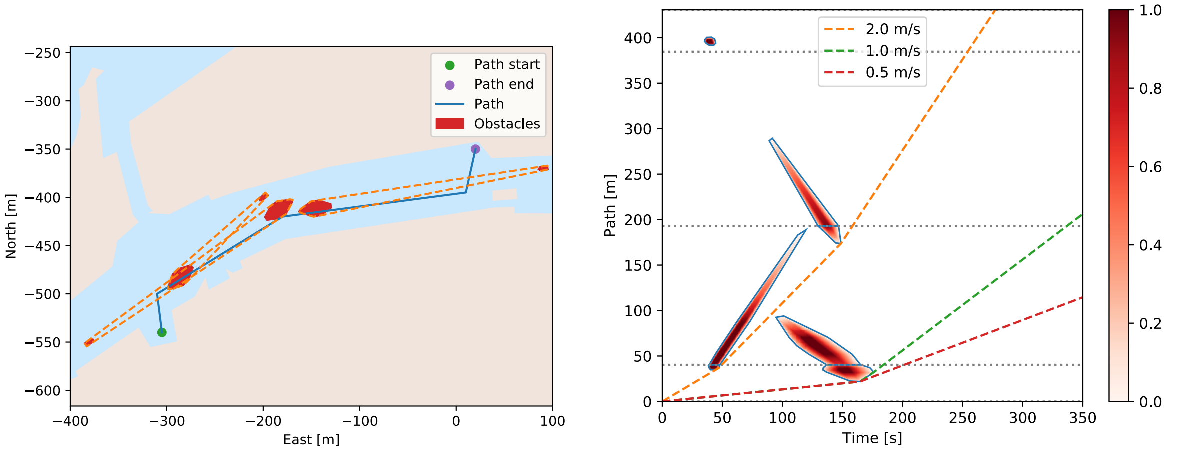
Based on the path-time planner in the previous section, it is possible to generalize the approach into a space-time planner, where the goal is to plan a collision free trajectory that avoid intersecting the space-time obstacle. This can be easily done by using sampling based methods such as probabilistic



(a) Space-time obstacle projected onto the path in black results in the path-time obstacle in blue.

(b) Path-time obstacle, where constant velocity lines that do not intersect the obstacle are considered to be safe.

FIGURE 5. Example of how a polytopic space-time obstacle can be projected along a desired path (a), and the resulting path-time diagram (b). Note that straight lines in the path-time diagram correspond to constant velocities along the path.



(a) Confined waters with dynamic obstacles at 0 seconds and predicted obstacles at 120 seconds. The goal is to find a safe velocity profile along the path.

(b) Path-time diagram, with collision free trajectories for different maximum velocities. The dotted lines mark transitions between different straight line path segments.

FIGURE 6. Example of how the path-time obstacles can be used for planning safe trajectories under obstacle uncertainty. Given the desired path and obstacles predicted motion in (a), we get the path-time diagram and planned velocity profiles in (b).

roadmaps (PRM) [28] and rapidly-exploring random tree (RRT) [29]. For our implementation however, we utilize the geometry of the space-time obstacle itself, and plan a collision free trajectory by connecting trajectory segments along the edges of the space-time obstacle representation. In order to ensure that the path is feasible with respect to the maximum turning rate of the vessel, we use Dubins paths [30], which consist of straight line segments and circular arcs of a maximum curvature. Using a graph search algorithm such as Dijkstra or A* with a given cost function, an optimal path can be found by connecting the edges of the space-time obstacles with Dubins paths.

2) EXAMPLE

Given a desired trajectory in an environment with dynamic obstacles, we can use the space-time obstacle representation in order to plan an optimal trajectory, which is collision free, and dynamically feasible. The resulting planning algorithm can be described as follows:

1) From obstacle tracks, compute the space-time representation of the vessel, predicting the obstacle movement and uncertainty into the future.
2) Using a graph search algorithm such as Dijkstra or A* with a given cost function, plan a sequence of connected Dubins paths (constant curvature circle segments and straight lines, see Figure 7), between edges of the space-time obstacles, which do not intersect with the space-time obstacles. The planned connected Dubins path then gives an optimal collision free trajectory.

In order to implement the planner, we chose to use the objective of finding the path that minimizes the squared space-time

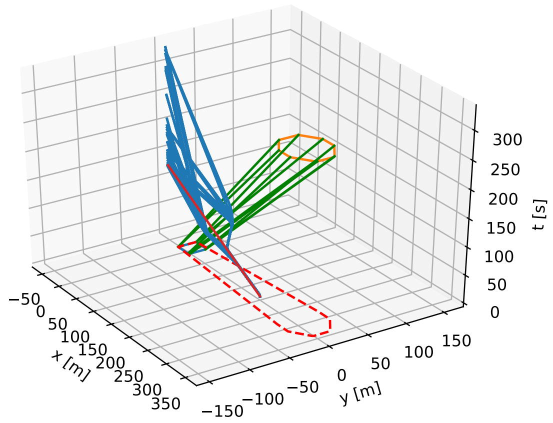


FIGURE 7. Possible collision free trajectories found during planning step.

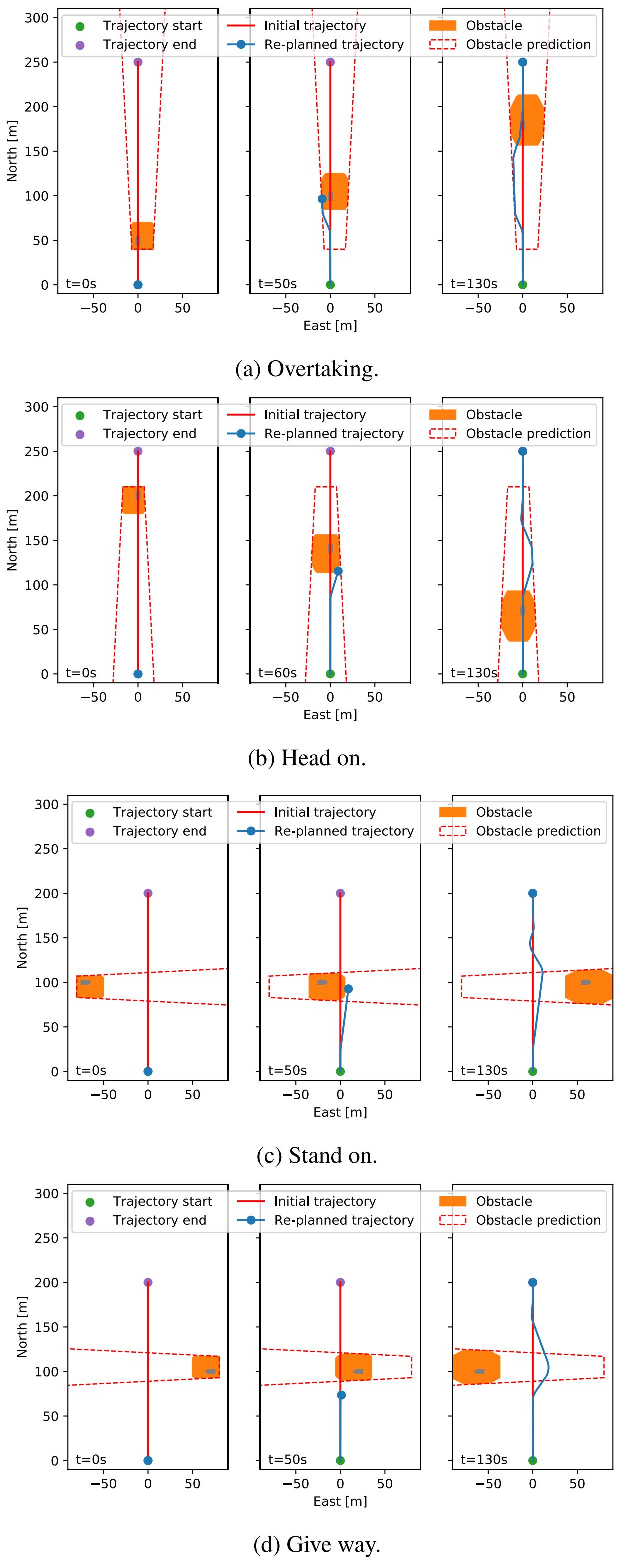
error between a desired trajectory and the planned trajectory, defined as:

$$\int_0^1 (x - x_d)^2 + (y - y_d)^2 + q \cdot (t - t_d)^2 ds, \qquad (30)$$

where the trajectory and desired trajectory are functions of the path variable $s \in [0, 1]$, and q is a weight factor for weighting the time versus position error. Using this cost function is useful, as it encourages the planned trajectory to follow the desired trajectory, making evasive maneuvers that keep the vessel as close to the desired trajectory as possible. In order to discretize the search space and make the planner computationally feasible, a finite number of vessel velocities were considered, and the space-time obstacle intersections, were computed based on the finite velocities. For the implementation, we designed the initial obstacle with a large forbidden region in front and to the right of the obstacle vessel, making COLREGs compliant trajectories optimal in terms of the planning objective given by the space-time error (30). Running the planner for overtaking, head on, stand on and give way scenarios, we got the results seen in Figure 8. From the results, we see that the planned trajectory initially follows the desired trajectory, before taking an evasive maneuver, ensuring COLAV, while adhering to the COLREGs. We can note that in the stand on scenario, the planned trajectory keeps the desired course and speed initially, as is expected in a stand on situation, however due to the uncertainty in the future obstacle position, maintaining the course and speed can in the worst case scenario lead to a collision, and the vessel must deviate from the desired path in order to ensure collision avoidance. For a physical implementation, the planner can be run iteratively, in order to replan the trajectory, as new information about the space-time obstacle becomes available, decreasing the obstacle uncertainty and hence improving the planned trajectories.

D. OPTIMIZATION BASED PLANNER

A common approach for planning optimal trajectories, is to formulate the problem as an model based optimization problem, and solving it using numerical optimization. For these methods, ellipsoids are commonly used for representing obstacles [11]–[13], as they the ellipsoid representation is computationally cheap, and simple to implement into a



(a) Overtaking.

(b) Head on.

(c) Stand on.

(d) Give way.

FIGURE 8. Planned trajectories for different scenarios when using the Dublins trajectory planner. Note the obstacle asymmetry which ensures COLREGs compliance.

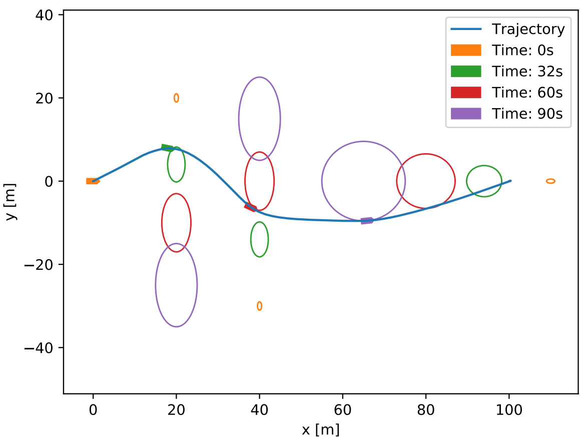


FIGURE 9. **Optimal trajectory, given dynamic obstacles with uncertainty. The four timestamps show the vessel at the initial position, as well as the closest point of approach to each of the three obstacles.**

numerical optimization problem. In this section we will show how the ellipsoid space-time obstacle can be used together with numerical optimization in order to plan optimal collision free trajectories.

1) OPTIMAL CONTROL PROBLEM

Given a cost function $J(\boldsymbol{x}, \boldsymbol{u})$ and a continuous time model of the vessel on the form:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}), \tag{31}$$

where $\boldsymbol{x}$ are the vessel states, including position heading and velocity, and $\boldsymbol{u}$ are the control surface. We can formulate an optimal control problem, which is to find the trajectory $\boldsymbol{x}(t)$ and controls $\boldsymbol{u}(t)$ which minimize the cost $J(\cdot)$ and are dynamically feasible with respect to the continuous time model over a time interval $[0, T]$. In order to account for the ellipsoidal obstacle, we can directly use the obstacle constraint from (19), giving the following optimal control problem:

$$\begin{aligned} \min_{\boldsymbol{x}(t),\boldsymbol{u}(t)} \quad & \int_0^T J(\boldsymbol{x}(t), \boldsymbol{u}(t))dt \\ \text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \\ & (\boldsymbol{x}(t) - \boldsymbol{p}_t)^\top \boldsymbol{Q}_t^{-1}(\boldsymbol{x}(t) - \boldsymbol{p}_t) \geq 1 \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0. \end{aligned} \tag{32}$$

2) EXAMPLE

In order to test the optimization based planner, we used the Cybership II model (see section IV) together with a quadratic cost function, and three dynamic obstacles. Using a collocation based transcription method to convert the optimization problem into an nonlinear programming problem, we got the optimal trajectory seen in Figure 9. From the results, we see that using the space-time obstacles, the planned trajectory is able to keep clear of the different obstacles in a way that accounts for the uncertainty obstacle uncertainty over time. This is a conservative strategy, but guarantees COLAV under obstacle uncertainty.

For more complex scenarios, it is possible to use other optimization objectives, such as time, energy, and distance [31]. It is also possible to include risk in the objective or constraints, allowing for a certain amount of risk to be taken when planning the optimal trajectory. This type of planner is also possible to implement as nonlinear model predictive control scheme by re-planning the trajectory at each time step, allowing for the planner to incorporate less conservative obstacle estimates as they become available over time.

IV. CONCLUSION

We have presented a novel space-time obstacle representation, which can be used to predict the reachable set of dynamic obstacles under uncertainty. Additionally, we have shown how the proposed space-time representation can be efficiently computed using a convex polygon half-space representation, as well as an ellipsoid representation. Finally, we demonstrated how the space-time obstacle representation can be used for risk assessment, collision avoidance and planning for surface vessels in various environments with uncertain dynamic obstacles.

Based on the example applications we have demonstrated how the proposed space-time obstacle representation offers a flexible framework for representing and predicting obstacle trajectories in way that is computationally efficient. For future work it would be interesting to look at the possibility of extending the method to allow for time varying velocity uncertainties. Using the space-time obstacle representation together with other trajectory planning and collision avoidance methods would also be interesting, as well as further studying how to best represent the initial obstacle shapes in order to promote COLREGs compliance.

APPENDIX

CYBERSHIP II MODEL

Cybership II is a 1 : 70 scale model supply ship. The length of the ship is 1.3 m and the weight about 24 kg. The maximum actuated surge force is 2N, the maximum sway force is 1.5N and the maximum yaw moment is 1.5Nm. Given the pose $\boldsymbol{\eta} = [x, y, \psi]^\top$ in terms of the position (x, y) and heading ψ, velocity $\boldsymbol{\nu} = [u, v, r]^\top$ in surge, sway and yaw, the Cybership II can be modeled as follows:

$$\underbrace{\begin{bmatrix} \dot{\boldsymbol{\eta}} \\ \dot{\boldsymbol{\nu}} \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu} \\ -\boldsymbol{M}^{-1}\left(\boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{\tau}\right) \end{bmatrix}}_{f(\boldsymbol{x},\boldsymbol{u})},$$

where the inertia matrix, Coriolis matrix, damping matrix, and transformation matrix are given as:

$$\boldsymbol{M} = \begin{bmatrix} 25.8 & 0 & 0 \\ 0 & 33.8 & 1.0115 \\ 0 & 1.0115 & 2.76 \end{bmatrix}$$

$$\boldsymbol{C}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -33.8v - 1.0115r \\ 0 & 0 & 25.8u \\ 33.8v + 1.0115r & -25.8u & 0 \end{bmatrix}$$

$$\boldsymbol{D}(\boldsymbol{\nu}) = \begin{bmatrix} 0.72 + 1.33|u| & 0 & 0 \\ 0 & 0.86 + 36.28|v| & -0.11 \\ 0 & -0.11 - 5.04|v| & 0.5 \end{bmatrix}$$

$$\boldsymbol{J}(\boldsymbol{\eta}) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

REFERENCES

[1] J. D. Luse, "Collision avoidance systems and the rules of the nautical road," *Navigation*, vol. 19, no. 1, pp. 80–88, Mar. 1972, doi: 10.1002/j.2161-4296.1972.tb00129.x.

[2] M. R. Benjamin, J. A. Curcio, J. J. Leonard, and P. M. Newman, "Navigation of unmanned marine vehicles in accordance with the rules of the road," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2006, pp. 3581–3587, doi: 10.1109/ROBOT.2006.1642249.

[3] D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, "Proactive collision avoidance for ASVs using a dynamic reciprocal velocity obstacles method," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2402–2409, doi: 10.1109/IROS.2018.8594382.

[4] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using the relative velocity paradigm," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1993, pp. 560–565, doi: 10.1109/ROBOT.1993.292038.

[5] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: A collision cone approach," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 28, no. 5, pp. 562–574, Sep. 1998, doi: 10.1109/3468.709600.

[6] B. Kluge and E. Prassler, "Reflective navigation: Individual behaviors and group behaviors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 4, Apr./May 2004, pp. 4172–4177, doi: 10.1109/ROBOT.2004.1308926.

[7] D. Wilkie, J. van den Berg, and D. Manocha, "Generalized velocity obstacles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 5573–5578, doi: 10.1109/IROS.2009.5354175.

[8] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 696–706, Aug. 2011, doi: 10.1109/TRO.2011.2120810.

[9] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997, doi: 10.1109/100.580977.

[10] Y. Yavin, C. Frangos, and T. Miloh, "Computation of feasible control trajectories for the navigation of a ship around an obstacle in the presence of a sea current," *Math. Comput. Model.*, vol. 21, no. 3, pp. 99–117, 1995, doi: 10.1016/0895-7177(94)00218-D.

[11] B.-O.-H. Eriksen and M. Breivik, "MPC-based mid-level collision avoidance for ASVs using nonlinear programming," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2017, pp. 766–772, doi: 10.1109/CCTA.2017.8062554.

[12] G. Bitar, V. N. Vestad, A. M. Lekkas, and M. Breivik, "Warm-started optimized trajectory planning for ASVs," *IFAC-PapersOnLine*, vol. 52, no. 21, pp. 308–314, 2019, doi: 10.1016/j.ifacol.2019.12.325.

[13] G. Bitar, B.-O.-H. Eriksen, A. M. Lekkas, and M. Breivik, "Energy-optimized hybrid collision avoidance for ASVs," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 2522–2529, doi: 10.23919/ECC.2019.8795645.

[14] S. Moe, K. Y. Pettersen, and J. T. Gravdahl, "Set-based collision avoidance applications to robotic systems," *Mechatronics*, vol. 69, Aug. 2020, Art. no. 102399, doi: 10.1016/j.mechatronics.2020.102399.

[15] M. Erdmann and T. Lozano-Pérez, "On multiple moving objects," *Algorithmica*, vol. 2, no. 1, pp. 477–521, 1987, doi: 10.1007/BF01840371.

[16] J. Larson, M. Bruch, and J. Ebken, "Autonomous navigation and obstacle avoidance for unmanned surface vehicles," *Proc. SPIE*, vol. 6230, pp. 53–64, May 2006, doi: 10.1117/12.663798.

[17] Y. Cho and J. Kim, "Collision probability assessment between surface ships considering maneuver intentions," in *Proc. OCEANS Aberdeen*, Jun. 2017, pp. 1–5, doi: 10.1109/OCEANSE.2017.8084791.

[18] Y. Lin and S. Saripalli, "Collision avoidance for UAVs using reachable sets," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2015, pp. 226–235, doi: 10.1109/ICUAS.2015.7152295.

[19] M. R. Akella and K. T. Alfriend, "Probability of collision between space objects," *J. Guid., Control, Dyn.*, vol. 23, no. 5, pp. 769–772, Sep. 2000, doi: 10.2514/2.4611.

[20] J. Park and J. Kim, "Predictive evaluation of ship collision risk using the concept of probability flow," *IEEE J. Ocean. Eng.*, vol. 42, no. 4, pp. 836–845, Oct. 2017, doi: 10.1109/JOE.2016.2614870.

[21] C. T. Shelton and J. L. Junkins, "Probability of collision between space objects including model uncertainty," *Acta Astronaut.*, vol. 155, pp. 462–471, Feb. 2019, doi: 10.1016/j.actaastro.2018.11.051.

[22] T. Tengesdal, E. F. Brekke, and T. A. Johansen, "On collision risk assessment for autonomous ships using scenario-based-MPC," in *Proc. IFAC World Congr.*, 2020, pp. 1–8.

[23] M. van Kreveld, O. Schwarzkopf, M. de Berg, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Cham, Switzerland: Springer, 2000, doi: 10.1007/978-3-540-77974-2.

[24] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 72–89, 1986, doi: 10.1177/027836498600500304.

[25] E. H. Thyri, M. Breivik, and A. M. Lekkas, "A path-velocity decomposition approach to collision avoidance for autonomous passenger ferries in confined waters," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14628–14635, 2020.

[26] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.

[27] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968, doi: 10.1109/TSSC.1968.300136.

[28] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996, doi: 10.1109/70.508439.

[29] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[30] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, no. 3, pp. 497–516, 1957, doi: 10.2307/2372560.

[31] A. B. Martinsen, A. M. Lekkas, and S. Gros, "Optimal model-based trajectory planning with static polygonal constraints," 2020, *arXiv:2010.14428*. [Online]. Available: http://arxiv.org/abs/2010.14428

ANDREAS B. MARTINSEN received the M.Sc. degree in engineering cybernetics from Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 2018, where he is currently pursuing the Ph.D. degree in engineering cybernetics.

His research interests include reinforcement learning, optimal control, and machine learning, with a focus on marine and maritime applications.

ANASTASIOS M. LEKKAS received the M.Sc. degree in mechanical and aeronautical engineering from the University of Patras, Patras, Greece, in 2008, and the Ph.D. degree in engineering cybernetics from Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 2014.

He is currently an Associate Professor of autonomous systems with the Department of Engineering Cybernetics, NTNU. His research interest includes merging artificial intelligence with control engineering in order to develop cyber-physical systems with increased autonomy. He is also the Project Manager of the EXAIGON Project, where the main goal is to develop explainable AI methods for safety-critical applications and business-critical applications.