Model Predictive Control for Path Following and Collision-Avoidance of Autonomous Ships in Inland Waterways

Dhanika Mahipala[1] and Tor Arne Johansen[2]

***Abstract*— While existing algorithms for open water navigation typically address path following and COLREGS compliant collision-avoidance, the unique challenges of inland waterways require a more tailored approach. We propose a two-level control strategy that employs Model Predictive Control (MPC) and Scenario-Based Model Predictive Control (SB-MPC) for path following and collision-avoidance. The algorithm proposes integrated strategies for handling riparian land, static obstacles, and dynamic obstacles. The method is tested in simulation.**

I. Introduction

A. Background

There are two major objectives that need to be addressed when developing a control algorithm for autonomous vessels, namely, path following and collision-avoidance. While certain algorithms in the literature distinctly differentiate between path following and collision-avoidance, there are those that do not and instead aim to address both issues using interconnected components [1]. Furthermore, the majority of these algorithms are designed for autonomous surface vehicles that navigate in open waters.

Due to the increase in interest in using inland waterways such as navigable rivers and canals for cargo transportation, autonomy in inland waterway transportation has become a separate research area of interest over the recent years. Inland waterway navigation comes with its own set of unique challenges that limit the capability to apply the said algorithms designed for open waters as is. Therefore, it is vital to specifically cater towards addressing these challenges when developing algorithms for autonomous navigation and collision-avoidance in inland waterways. The authors identify the following aspects as required to be addressed by an algorithm designed for autonomous ship navigation in inland waterways.

- Static obstacle avoidance capability: It is more likely for inland waterways to encounter static obstacles such as reeks, wrecks, navigational aids, fishing zones, small islands and other static obstacles found in harbor environments e.g., moored vessels, that are unmapped.
- Dynamic obstacles: Inland waterways are often congested and densely trafficked compared to open waters.
- Riparian land aware navigation capability: In this paper, the term riparian land is used to refer to the grounding hazard on either side of the waterbody e.g., riverbank. Unlike open waters, inland waterways are narrow waterbodies which force the vessel to frequently encounter land when de-routing from the pre-planned path to avoid dynamic obstacles. Therefore, if possible, it is better to handle riparian land as a separate entity from the other static obstacles mentioned above, and implement a collision-avoidance strategy specifically catered for it to improve the overall performance of the algorithm.
- Traffic rule compliance: It is necessary for the algorithm to adhere to applicable local traffic rules when navigating and performing collision-avoidance maneuvers.
- Adaptability of the algorithm for secondary objectives: Such as use of minimum time and/or least amount of energy.

The initial step in designing such an algorithm starts with determining a set of global waypoints between the current position and the destination during the mission planning stage which is carried out before sailing. However, this is beyond the scope of this research thus, it is assumed that a sequence of waypoints is available.

B. Literature Review

There has been ample research carried out on autonomous path following and collision-avoidance over the years. The articles [1], [2] and [3] summarize many of the said algorithms. These include evolutionary algorithms, sampling-based algorithms, cell decomposition methods, directional approaches, and road map methods. Several of the algorithms also comply with Convention on the International Regulations for Preventing Collisions at Sea (COLREGS) [4].

The SB-MPC algorithm [5], proposes a reactive collision-avoidance method based on simulation and receding horizon optimization. It enforces compliance with the main rules of COLREGS and collision hazard avoidance through the evaluation of a cost function along the predicted ship and obstacle trajectories. Since the introduction, there have been many modifications and additions suggested to improve the performance of the algorithm. One such research is [6] where the authors present an additional cost component which they call a COLREGS-transitional cost that penalize control behaviors that abort a COLREGS compliant maneuver. Research was also carried out to handle kinematic uncertainty present when relying on track estimates of nearby dynamic obstacles for collision-avoidance (COLAV) using SB-MPC [7]. The authors have named this modified algorithm as Probabilistic Scenario-Based Model Predictive Control (PSB-MPC). The article [8] proposes a parallelized implementation of the PSB-MPC on a Graphical Processing Unit (GPU) that reduces the computational speed of the algorithm, allowing the MPC problem to scale linearly with increasing the number of control behaviors, static and dynamic obstacles and prediction scenarios. In [9], the authors suggest a Dynamic Bayesian Network (DBN) to model and infer the intentions of other ships. Similarly, [10], proposes two-stage trajectory prediction (2-STP) algorithm to help

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 955.768 (MSCA-ETN AUTOBarge). This publication reflects only the authors' view, exempting the European Union from any liability. Project website: http://etn-autobarge.eu/.

[1,2]Department of Engineering Cybernetics, Norwegian University of Science and Technology, Norway. (Emails: [1]dcmahipa@stud.ntnu.no and [2]tor.arne.johansen@ntnu.no)

[1]Kongsberg Maritime AS, Norway.

the own ship to be aware of intention changes of target vessels and avoid collision risks. Another study [11] aims to improve the SB-MPC algorithm by including collaboration, i.e., information exchange between ships. Overall, in our opinion, SB-MPC has proven to be a successful COLAV strategy for autonomous vessels in open waters and intend to adapt it for inland waterways as well. In particular, when utilizing SB-MPC for navigation in inland waterways, the following challenges may arise:

- Resolution insufficiency of the solutions due to being derived from a descretized finite solution pool.
- Erratic solutions due to abrupt changes in the state of the hazardous environment as a result of congestion in inland waterways.

In this research, a hybrid algorithm consisting of Model Predictive Control (MPC) and Scenario-Based Model Predictive Control (SB-MPC) [5] is therefore proposed to overcome these issues by distributing the responsibilities between the two algorithms. The proposed algorithm also fulfills the autonomous path following and collision-avoidance tasks of an inland waterway vessel identified in Section I-A as follows:

- Strategies have been proposed to handle static obstacles, dynamic obstacles and riparian land.
- The SB-MPC algorithm is COLREGs compliant.
- Even though a secondary control objective is not discussed in this paper, the proposed algorithm can easily incorporate one by adding an additional cost component.

The idea of exploiting the complimentary strengths of multiple algorithms has been previously explored by [12], [13]. In both references, two different three-layered architectures for real-time path planning and obstacle avoidance are proposed. However, these algorithms do not utilize the SB-MPC algorithm nor consider riparian land separate from the static obstacles since they are primarily developed for vessels that navigate in open waters. Moreover, [12] does not consider the COLREGS.

C. Contribution

The main contribution of this paper is enabling the adaptation of SB-MPC for COLAV to inland waterways by incorporating it in a two level hybrid control architecture. Furthermore, the riparian land is taken into consideration by proposing a novel concept of a cross-track error corridor, which serves as the limits of the maneuverable area for the vessel at a given instance. In addition, a modification to the Constant Velocity Model (CVM) is introduced for target vessel trajectory prediction that takes into account the riparian land. Finally, the proposed algorithm was verified with realistic situations by choosing simulation scenarios from Electronic Navigational Charts (ENC).

II. Overview

The hybrid control algorithm proposed in this paper is a two-level control strategy. On the top level, an MPC controller generate desired course (χ_d) and speed (U_d) commands considering the destination waypoint, static obstacles, and riparian land. Next, the SB-MPC would consider the outputs given by the MPC, χ_d and U_d, and suggest course and speed modifications χ_m and U_m, to avoid dynamic obstacles. Then, χ_m and U_m will be used to calculate the final course and speed commands χ_c and U_c (as explained in Section V) that is sent to the autopilot to drive the vessel. This system is shown in Figure 1.

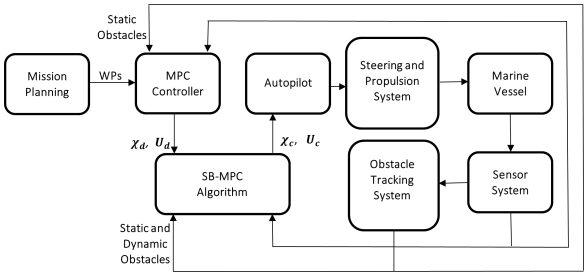


Fig. 1. Control system architecture

III. Ship Model

A. Own-ship model

A mathematical model of the own vessel is required by the MPC and SB-MPC controllers. For this research we have decided to adapt the kinematic model used in [8] as,

$$\begin{aligned} x_{k+1} &= x_k + U_k \cos(\chi_k) \\ y_{k+1} &= y_k + U_k \sin(\chi_k) \\ \chi_{k+1} &= \chi_k + \frac{1}{T_\chi}(\chi_{input,k} - \chi_k) \\ U_{k+1} &= U_k + \frac{1}{T_U}(U_{input,k} - U_k) \end{aligned} \tag{1}$$

which describes the own-ship state, $\mathbf{x}_k = [x_k, y_k, \chi_k, U_k]^T$ at time t_k, that consists of East and North position in Cartesian coordinates, course over ground (COG), and speed over ground (SOG). T_χ and T_U are course and speed time constants that can be derived from a system identification methodology. The χ_{input} and U_{input} are the course and speed commands that serve as the inputs to the system.

B. Target ship model

Target ships in the vicinity of the own ship are identified as dynamic obstacles. To generate the optimum desired course and speed modifications for the own-ship, the SB-MPC algorithm predicts the future trajectory of the dynamic obstacles. For this, we are proposing a ground-avoiding constant velocity model as follows, by assuming a narrow channel.

$$\begin{aligned} x_{k+1}^{target} &= x_k^{target} + U_k^{target} \cos\left(\chi_k^{target}\right) \\ y_{k+1}^{target} &= y_k^{target} + U_k^{target} \sin\left(\chi_k^{target}\right) \\ \chi_{k+1}^{target} &= \begin{cases} \chi_k^{target}, & \text{if } d_{LOS_g} \geq d_{critical} \\ \widetilde{\chi}_k^{target}, & \text{otherwise} \end{cases} \\ \widetilde{\chi}_k^{target} &= \begin{cases} \chi_k^{target} + \beta, & \text{if } d_{LOS_g}^{\beta+} \geq d_{LOS_g} \\ \chi_k^{target} - \beta, & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

where, $\mathbf{x}_k^{target} = [x_k^{target}, y_k^{target}, \chi_k^{target}, U_k^{target}]^T$ describes the target ship state at time t_k. The variable d_{LOS_g} is the distance from target ship to the closest point on ground along the course angle χ_k^{target}. The variable $d_{LOS_g}^{\beta+}$ is the distance from target ship to the closest point on ground along the direction defined by the course angle ($\chi_k^{target} + \beta$). The parameters $d_{critical}$ and β are turning parameters that

depend on the required ground clearance and maneuvering capabilities of each target vessel, respectively. The model assumes a straight-line trajectory for the target vessel while $d_{LOS_g} \geq d_{critical}$. When d_{LOS_g} is less than the critical range, the target vessel course angle is assumed to change direction by $\pm\beta$ depending on the course angle that gives the least line of sight distance to the ground.

C. Path coordinate system

The proposed algorithm is executed in a local coordinate frame referred here as the Path Coordinate System denoted by $\{P_j\}\,(j = 1, 2, \ldots)$ based on the connected waypoint paths. This formation is adapted from [14] as follows.

As shown in Figure 2, in the path coordinate system $\{P_j\}\,(j = 1, 2, \ldots)$, X_{p_j} is along the line connecting adjacent waypoints, and Y_{p_j} is orthogonal to that line pointing $\pi/2$ counterclockwise. O_{p_j} is the origin of the jth path coordinate system located at the jth waypoint connecting reference path j and $j+1$. The angle with respect to the East-North-Up (ENU) inertial frame $\{n\}$, of reference path j is denoted as ψ_j.

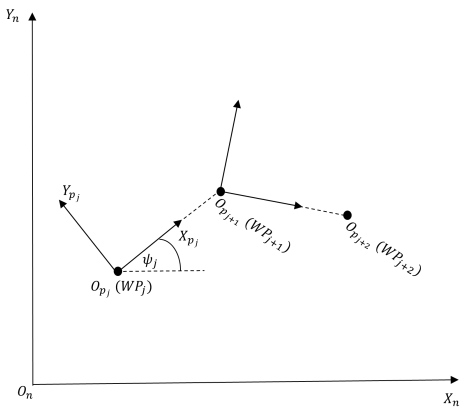


Fig. 2. Inertial coordinate system and path coordinate system in the horizontal plane

The ENC data and outputs of the mission planning stage i.e., waypoints will be made available in the inertial coordinate frame. Therefore, these data points need to be transformed from $\{n\}$ to $\{P_j\}$. Equation 3 can be used to transform a single point:

$$\begin{bmatrix} x_{p_j} \\ y_{p_j} \end{bmatrix} = \begin{bmatrix} \cos\psi_j & \sin\psi_j \\ -\sin\psi_j & \cos\psi_j \end{bmatrix} \begin{bmatrix} x - x_{WP_j} \\ y - y_{WP_j} \end{bmatrix} \tag{3}$$

where, (x_{WP_j}, y_{WP_j}) is the coordinate of the waypoint j, being the origin of $\{P_j\}$ in $\{n\}$, (x, y) is a point in $\{n\}$ that needs to be transformed to $\{P_j\}$ coordinate system.

When switching from $\{P_j\}$ to $\{P_{j+1}\}$, the angle difference between the two adjacent x axes is $\psi_{j+1} - \psi_j$. Since $\{P_j\}$ and $\{P_{j+1}\}$ are connected, the new origin $O_{P_{j+1}}$ has coordinates $(l_j, 0)$ relative to the old coordinate system $\{P_j\}$. Therefore, a coordinate can be transformed from $\{P_j\}$ to $\{P_{j+1}\}$ (when the waypoints switch) as follows,

$$\begin{bmatrix} x_{p_{j+1}} \\ y_{p_{j+1}} \end{bmatrix} = \begin{bmatrix} \cos(\psi_{j+1} - \psi_j) & \sin(\psi_{j+1} - \psi_j) \\ -\sin(\psi_{j+1} - \psi_j) & \cos(\psi_{j+1} - \psi_j) \end{bmatrix} \begin{bmatrix} x_{p_j} - l_j \\ y_{p_j} - 0 \end{bmatrix} \tag{4}$$

IV. MPC Controller

In our control algorithm, the MPC controller serves as the top-level guidance controller that generates the desired course and speed angles depending on the destination coordinates, static obstacles, and riparian land limits. The MPC is used to solve the optimal control problem (OCP) repeatedly [15]. The continuous-time OCP is formulated as,

$$\begin{aligned} \min_{\mathbf{x}(.),\mathbf{u}(.)} \quad & \int_0^T \phi(\mathbf{x}(\tau),\ \mathbf{u}(\tau))d\tau \\ s.t. \quad & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t),\ \mathbf{u}(t)) \quad \forall\ t \in [0,\ t_{max}] \\ & \mathbf{h}(\mathbf{x}(t),\ \mathbf{u}(t)) \leq 0 \\ & \mathbf{x}(0) = \bar{\mathbf{x}}(t_0) \end{aligned} \tag{5}$$

where, ϕ is the objective function, $\mathbf{x}(.)$ is the own-ship states; $\mathbf{x} = [x, y, \chi, U]^T$, $\mathbf{u}(.)$ contains control actions; $\mathbf{u} = [\chi_d, U_d]^T$, $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ is the ship model described in Equation 1, $\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t))$ represents inequality constraints, and $\bar{\mathbf{x}}(t_0)$ denotes the current state of the ship.

Next, the OCP is discretized and converted to a numerical Nonlinear Programming Problem (NLP) for N_p number of steps (horizon) using the multiple shooting method [16].

$$\begin{aligned} \min_{\mathbf{w}} \quad & \phi(\mathbf{w}) \\ s.t. \quad & \mathbf{g}(\mathbf{w}) = 0 \\ & \mathbf{h}(\mathbf{w}) \leq 0 \end{aligned} \tag{6}$$

where, $\mathbf{w} = \left[\mathbf{x}_0^T, \mathbf{u}_0^T, \ldots, \mathbf{x}_{N_p-1}^T, \mathbf{u}_{N_p-1}^T, \mathbf{x}_{N_p}^T\right]^T \in R^{6N_p+4}$ denotes the decision variables in the discretized-time formulation. The cost function ϕ is formulated as,

$$\phi(\mathbf{w}) = \sum_{k=1}^{N_p} \|\mathbf{x}_k - \mathbf{x}_k^{ref}\|_Q^2 + \|\mathbf{u}_k - \mathbf{u}_{k-1}\|_M^2 \tag{7}$$

where, $\mathbf{x}_k$ and $\mathbf{x}_k^{ref}$ denote the current state and reference state respectively. $\mathbf{x}^{ref}$ is the vessel state consisting the coordinates of the next waypoint, and expected course angle and speed of the ownship at that waypoint. Here, $\|Z\|_Q^2$ denotes the Euclidean norm of a vector Z, i.e., Z^TQZ. The first part of the Equation 7 penalizes the error in ship state while the second part penalizes the control effort. Matrices Q and M consist of tuning parameters. We have used the Runge-Kutta Order 4 (RK4) method to calculate the next ship state.

The equality constraints $\mathbf{g}(\mathbf{w})$ are used to close the shooting gaps from the integration steps.

$$\mathbf{g}(\mathbf{w}) = \begin{bmatrix} \bar{\mathbf{x}} - \mathbf{x}_0 \\ F(\mathbf{x}_0, \mathbf{u}_0) - \mathbf{x}_1 \\ F(\mathbf{x}_1, \mathbf{u}_1) - \mathbf{x}_2 \\ \vdots \\ F\left(\mathbf{x}_{N_p-1}, \mathbf{u}_{N_p-2}\right) - \mathbf{x}_{N_p} \end{bmatrix} \tag{8}$$

The inequality constraints $\mathbf{h}(\mathbf{w})$ are used to specify control input constraints ($\mathbf{h}_u$), static obstacle avoidance constraints ($\mathbf{h}_{sobs}$) and riparian land constraints ($\mathbf{h}_{rland}$). First,

$$\mathbf{h}_u(\mathbf{w}) = \begin{bmatrix} \chi_{lb} - \chi_d \\ -(\chi_{ub} - \chi_d) \\ U_{lb} - U_d \\ -(U_{ub} - U_d) \end{bmatrix} \tag{9}$$

where χ_{lb}, χ_{ub}, U_{lb} and U_{ub} denote the minimum and maximum allowable course angles and speed of the vessel respectively. Next,

$$\mathbf{h}_{sobs}\left(\mathbf{w}, L^i\right) = \begin{bmatrix} L^i - d_1^i \\ L^i - d_2^i \\ \vdots \\ L^i - d_{N_p}^i \end{bmatrix} \tag{10}$$

where d_k^i is the Euclidean distance between own-ship and the static obstacle i at a given time, while L^i is the minimum allowable distance between the two as further explained in Section IV-A. Finally,

$$\mathbf{h}_{rland}\left(\mathbf{w}, g_{1:N_p}^{lh}, g_{1:N_p}^{rh}\right) = \begin{bmatrix} y_1 - \max(g_{y_1}^{lh}, g_{y_1}^{rh}) \\ -(y_1 - min(g_{y_1}^{lh}, g_{y_1}^{rh})) \\ y_2 - \max(g_{y_2}^{lh}, g_{y_2}^{rh}) \\ -(y_2 - \min\left(g_{y_2}^{lh}, g_{y_2}^{rh}\right)) \\ \vdots \\ y_{N_p} - \max(g_{y_{N_p}}^{lh}, g_{y_{N_p}}^{rh}) \\ -(y_{N_p} - \min\left(g_{y_{N_p}}^{lh}, g_{y_{N_p}}^{rh}\right)) \end{bmatrix} \tag{11}$$

where y_k is the y-axis coordinate of the own-ship in the path coordinate system $\{P_j\}$ and $g_{y_k}^{lh}, g_{y_k}^{rh}$ are the y-axis coordinate of the corresponding ground positions on the riparian land on either side, in the path coordinate system $\{P_j\}$. These parameters are further explained in Section IV-B.

A. Static obstacle avoidance

In this research, reeks, wrecks, navigational aids, fishing zones and small islands are identified as static obstacles. We propose to handle these as constraints on the MPC problem by considering these obstacles as circular regions.

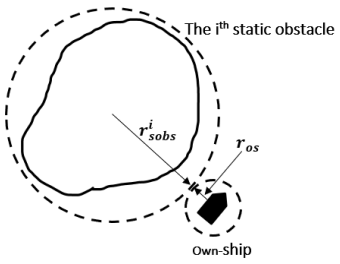


Fig. 3. Creating static obstacle constraints.

As shown in Figure 3, both the static obstacle and the own ship are considered as circular constraints. r_{obs}^i and r_{os} are the radius of the circles enclosing the ith static obstacle and own ship, respectively. One can select appropriate margins when determining said radius. Therefore, L^i and d^i in Equation 10 can be calculated as follows,

$$\begin{aligned} L^i &= r_{sobs}^i + r_{os} \\ d^i &= \|\mathbf{p}_{sobs}^i - \mathbf{p}_{os}\| \end{aligned} \tag{12}$$

where, $\mathbf{p}_{sobs}^i$ and $\mathbf{p}_{os}$ are center position of the enclosing circles of ith static obstacle and own ship respectively.

Even though this method is suitable for most cases, deducing the shape of a static obstacle to a single circle can be a waste of navigable area depending on how the circular area is formulated. The SeaCharts package introduced in [17], proposes an algorithm called 'EnclosingCircles' which calculates local polygon approximation circles, based on the currently visible shoreline or depth contour corresponding to the ship draught, from any given ship position, which can be used to address this issue.

B. Cross-track error corridor

The main difference between navigating inland waters compared to open sea is the presence of riparian land. The riparian land is not considered as a static obstacle above, and is therefore handled using a different strategy. We call it the Cross-track Error Corridor. If this is to be handled by SBMPC alone, it would require careful consideration when developing the cost components and tuning of the weights of respective cost components. Furthermore, the difference between consecutive course and speed commands generated at an area with high traffic density could be too drastic for inland waters due to the volatility of the environment. However, since information about the riparian land on the planned route is available well ahead as ENC data, it is possible to handle this as constraints of the MPC problem for the considered horizon.

To calculate the cross-track error corridor, we need to extract ENC chart data. More details on how ENC chart data were extracted for this research is explained in Section VI-A. Once extracted, it would provide riparian land as polygons which can be used to calculate the cross-track error corridor using Algorithm 1.

The input S to Algorithm 1 is an array with points of the predicted ship trajectory. This is usually derived by finding a predetermined number of intermediate waypoints on a straight-line connecting two subsequent waypoints. The parameter RG is a single polygon (sequence of points) created by merging the individual polygons representing riparian land within a predetermined radius from own ship. Figure 4 shows the error corridor generated for a part of the Beitstadsundet Fjord, Norway. Notice the red dotted line (the generated corridor) slightly inwards from the polygon boundaries. This margin is a buffer zone against navigation inaccuracies.

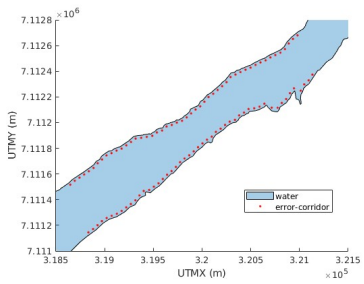


Fig. 4. Cross-track error corridor generated for a part of Beitstadsundet Fjord, Norway.

The input Δd is the maximum width of the corridor on either side of the ship if riparian land is not present. The function would return two arrays consisting of points of the corridor boundaries on either side of the ship. These points are generated w.r.t the inertial coordinate system. Next, they

Algorithm 1: Calculating cross-track error corridor

Data: Riparian land polygon data RG, an array of points S, that consists of intermediate points between two adjacent waypoints sampled at a predetermined length interval, and maximum corridor width Δd

Result: Two arrays of points, g_k^{lh} and g_k^{rh} that mark the error corridor on either side of the vessel

$g_k^{lh} \leftarrow 0$;
$g_k^{rh} \leftarrow 0$;
while $i < length(S)$ **do**
  $perp\_line \leftarrow$ perpendicular line to the line connecting two adjacent points in S;
  $point\_maxWidth\_rh \leftarrow$ point on $perp\_line$ at $a + \Delta d$ distance along the y axis (cross-track distance) from the corresponding point on S;
  $point\_maxWidth\_lh \leftarrow$ point on $perp\_line$ at $a - \Delta d$ distance along the y axis (cross-track distance) from the corresponding point on S;
  $line\_rh \leftarrow$ line connecting $point\_maxWidth\_rh$ and the corresponding point on S;
  $line\_lh \leftarrow$ line connecting $point\_maxWidth\_lh$ and the corresponding point on S;
  if $line\_rh$ *intersect with* RG **then**
    $g_i^{rh} \leftarrow$ coordinate of the intersection point
  else
    $g_i^{rh} \leftarrow point\_maxWidth\_rh$
  end
  if $line\_lh$ *intersect with* RG **then**
    $g_i^{lh} \leftarrow$ coordinate of the intersection point
  else
    $g_i^{lh} \leftarrow point\_maxWidth\_lh$
  end
end

are transformed to the path coordinate system introduced in Section III-C. Finally, the transformed coordinate points are then used to obtain $g_{y_k}^{lh}$, $g_{y_k}^{rh}$ parameters of Equation 11. The coordinate system transformation is performed, as it allows directly setting constraints for the position along the y-axis of the MPC solution. Figure 5 illustrates the generated error corridor for a situation where only one side of the vessel has riparian land present.

V. SB-MPC Controller

The SB-MPC algorithm that acts as the reactive controller in our approach is based on [5]. The main objective of the SB-MPC controller is to compute modifications to the nominal course (χ_d) and speed (U_d) generated from the top-level MPC controller to avoid dynamic obstacles. In SB-MPC, the following discrete control actions are commonly used:

- Course offset in degrees (χ_m): -90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75, 90.
- Speed factor (U_m): 0, 0.5, 1.

The modifications are in turn applied to the desired decisions from the MPC (χ_d, U_d) to obtain final course and speed commands (χ_c, U_c) using $\chi_c = \chi_d + \chi_m$ and $U_c = U_d \cdot U_m$ equations. Therefore, choosing $\chi_m = 0$ and $U_m = 1$ simply recovers the nominal course and speed (χ_d, U_d). This

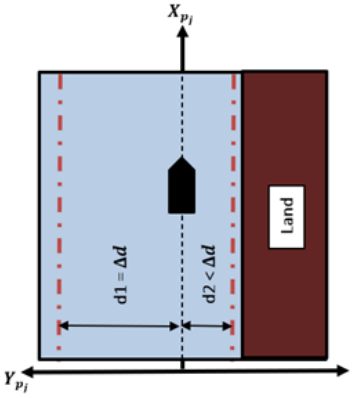


Fig. 5. Example for cross-track error corridor generation.

parametrization leads to a total of $N_s = 13 \times 3 = 39$ possible control behaviors to be simulated and evaluated over a time horizon to obtain the optimum course and speed commands (χ_c^*, U_c^*) for each time step. It is also worth noting that when evaluating certain course and speed combination, the course offset and speed factor (χ_m, U_m) are assumed to remain constant over the time horizon.

The internal objective of the SB-MPC is to evaluate the control behaviors $k \in \{1, 2, \ldots, N_s\}$ for each obstacle vessel $i \in \{1, 2, \ldots, N_o\}$ (N_o represents the number of obstacles) at time t_0 and select the control behavior with index k that minimizes the cost $H^k(t_0)$. Specifically,

$$k^*(t_0) = \arg\min_k \; H^k(t_0) \tag{13}$$

where,

$$H^k(t_0) = \max_i \max_{t \in D(t_0)} \left(C_i^k(t) R_i^k(t) + \kappa_i \mu_i^k(t) + \zeta^k(t)\right) + \Gamma(U_m^k, \chi_m^k) \tag{14}$$

The terms of the cost function in Equation 14 are defined as follows,

- The cost associated with collision with obstacle i at time t in scenario k, is $C_i^k(t)$, and the corresponding collision risk factor, $R_i^k(t)$.
- The COLREGS rules violation cost is $\kappa_i \mu_i^k(t)$, where $\mu_i^k(t)$ is a binary indicator of COLREGs rule violation and κ_i is a tuning parameter.
- The cost of maneuvering effort associated with scenario k is considered from $\Gamma(U_m^k, \chi_m^k)$.

Each scenario is evaluated at discrete sample times over the prediction horizon T using the discretization interval T_s, as $D(t_0) = \{t_0, t_0 + T_s, \ldots, t_0 + T\}$. More information about each cost component in detail can be found in [5].

Even though static obstacles and riparian land is handled by the constraints of the MPC controller, when modifying the desired course and speed commands (χ_d, U_d), the SB-MPC algorithm is not aware of them. Therefore, it needs its own way of handling this in case a collision-avoidance maneuver for dynamic obstacle suggested by the SB-MPC drive the vessel towards the ground. Therefore, authors of this paper propose adding an additional cost component inspired by [8] to the cost function mentioned above.

The grounding cost associated with control behavior k at time t is,

$$\zeta^k(t) = \begin{cases} e^{-(\eta_1 \left| d_i^k(t) - d_{safe} \right| + \eta_2 (t - t_0))}, & \text{if } d_i^k(t) \leq d_{close} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where, $d_i^k(t)$ is distance to the ground and d_{safe} is the circular own-ship safety margin around the ship. Only obstacles within the cicular region with radius d_{close} is considered a hazard, and η_1 and η_2 are tuning parameters.

VI. Results

A. Test setup

For the purpose of this research, we have parametrized ENC map data as two-dimensional polygons, read from shape files using the MATLAB built-in API. Each polygon encompasses a land segment in the map. The said shapefiles are generated using ENC charts corresponding to the relevant map region considered. Each shapefile in the regions of interest is read individually and merged to create a single polygon. However, a more general approach would be to use the SeaCharts library [17]. For this research, we have used the shapefiles in the Norwegian Fjord Catalog from the Norwegian Mapping Authority (NCA) [18].

B. Simulations

In Figures 6-13, blue region represents the water while the white region represents the riparian land. White circles in the middle of the water body represent artificially added static obstacles. The red cross and square represent starting and terminal waypoints, respectively. The solid red line bordering the riparian land is the generated cross-track error corridor. Own ship is denoted by a blue circle and target vessels are denoted by black, magenta and orange. The solid and dashed lines represent past and future trajectories of the vessels respectively.

In the first test simulation we would like to demonstrate the performance of the MPC controller on its own. For this, as the test bed, we have chosen part of the Beitstadsundet Fjord, Norway. In this test there are no dynamic obstacles present and therefore, the SB-MPC algorithm has been turned off. Figure 6 depict the simulation result of the test case. As shown, the MPC is perfectly capable of navigating between two waypoints avoiding static obstacles and riparian land.

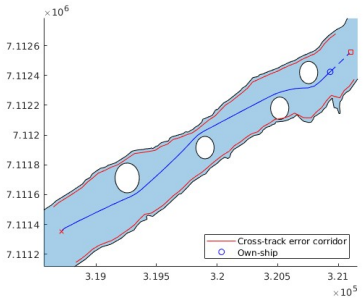


Fig. 6. Simulation 1 result

Next, we created seven test scenarios where, artificially added static obstacles, riparian land and dynamic obstacles are present. These test cases put the hybrid algorithm in challenging situations where different collision scenarios are mimicked. As the test bed for these different scenarios, part of the Trondheim Fjord, Norway was utilized. Figure 7-13 depict the results of the said test scenarios. In each of these figures, sub-figure (a) depicts the overview of the vessel trajectories while the sub-figure (b) show the distance between the ownship and each target vessel from the beginning of the simulation up to the instance showed in sub-figure (a).

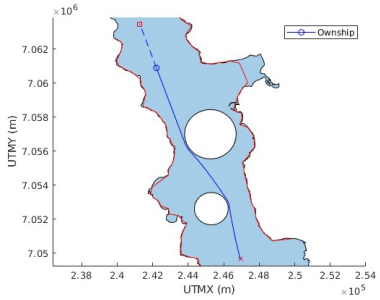
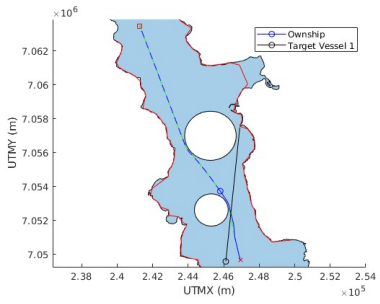
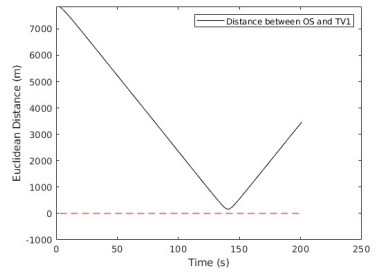


Fig. 7. Results of the test Scenario 1 in Simulation 2



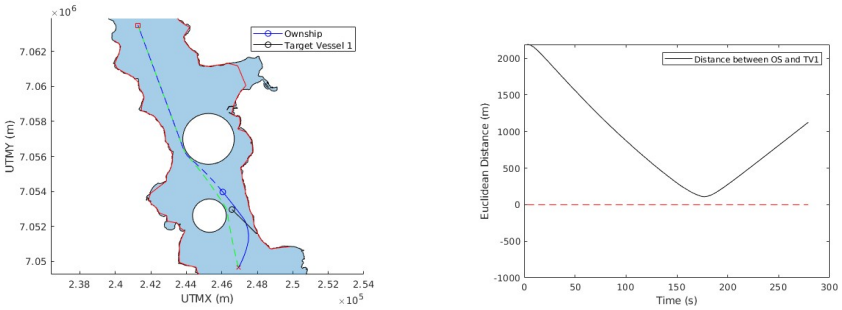
(a) Trajectory overview



(b) Distance between vessels

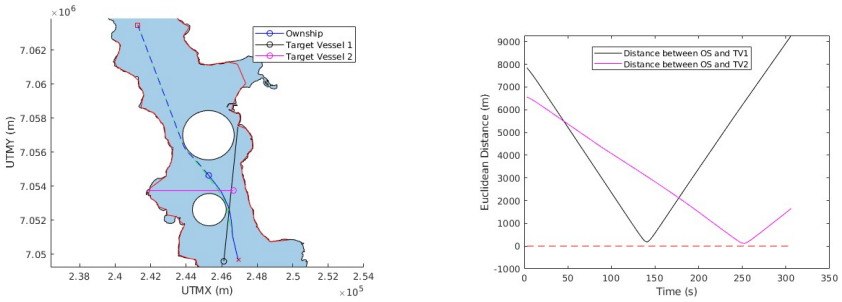
Fig. 8. Results of the test Scenario 2 in Simulation 2

In Figure 7, no dynamic obstacles were added, to test the performance of the hybrid algorithm without any target vessels and to obtain a reference trajectory to compare with the rest of the test cases. In Figures 8a-13a, the green dashed line mark the nominal own ship trajectory from Figure 7. In Figure 8 and 9, head-on and crossing scenarios are simulated and it can be seen the own ship takes a starboard turn correctly to avoid the collision situations. In Figure 10, the own ship is faced with both head-on and crossing target vessels in close proximity. In Figure 11, an overtaking situation is simulated immediately when the own ship passes the second static obstacle, while in Figure 12 the complexity of the situation was slightly elevated by adding a crossing vessel to the same overtaking scenario. In the scenario represented in Figure 13, the target vessel movements in Figure 8-10 were combined to design a complex collision



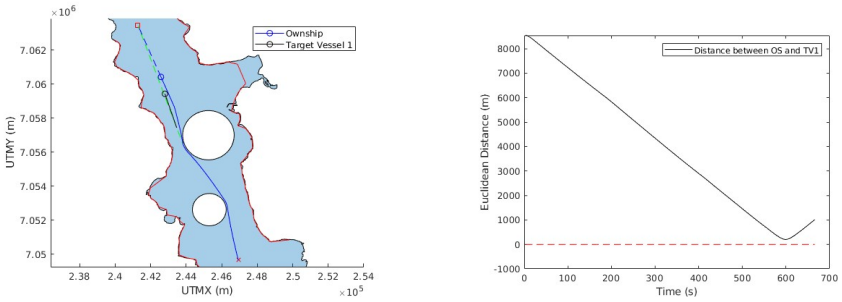
(a) Trajectory overview (b) Distance between vessels

Fig. 9. Results of the test Scenario 3 in Simulation 2



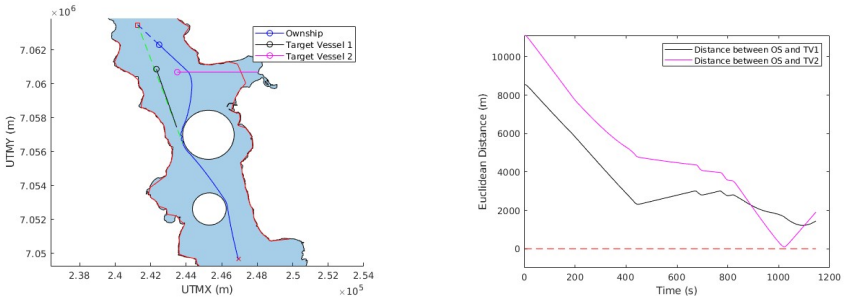
(a) Trajectory overview (b) Distance between vessels

Fig. 10. Results of the test Scenario 4 in Simulation 2



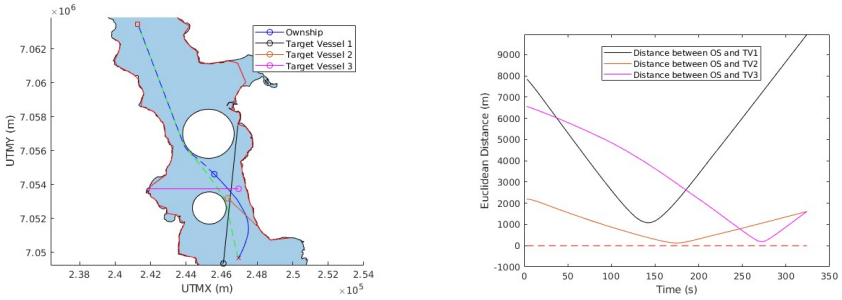
(a) Trajectory overview (b) Distance between vessels

Fig. 11. Results of the test Scenario 5 in Simulation 2



(a) Trajectory overview (b) Distance between vessels

Fig. 12. Results of the test Scenario 6 in Simulation 2



(a) Trajectory overview (b) Distance between vessels

Fig. 13. Results of the test Scenario 7 in Simulation 2

risk situation for the own ship. It is worth mentioning that in Figures 9,12 and 13, the ownship significantly deviates from the green line to comply with COLREGs Rule 15 [4]. The above simulations prove that the proposed control algorithm is capable of handling multiple target vessels and static obstacles in close proximity while limited by a narrow maneuverable area.

VII. Conclusion

In this paper, a novel collision-avoidance and guidance control algorithm is proposed, comprising of two layers that work in tandem to enable effective inland waterway navigation. The top layer incorporates an MPC controller that guides the vessel, performs path following and static obstacle avoidance, while the bottom layer contains an SB-MPC controller that modifies the output of the MPC to handle dynamic obstacles. The proposed algorithm is tested on several simulation scenarios, and simulation results show the algorithm's effectiveness in preventing collision scenarios.

References

[1] A. Vagale, R. Oucheikh, R. T. Bye, O. L. Osen, and T. I. Fossen, "Path planning and collision avoidance for autonomous surface vehicles I: a review," *Journal of Marine Science and Technology*, vol. 26, no. 4, pp. 1292–1306, Dec. 2021. [Online]. Available: https://doi.org/10.1007/s00773-020-00787-6

[2] A. Vagale, R. T. Bye, R. Oucheikh, O. L. Osen, and T. I. Fossen, "Path planning and collision avoidance for autonomous surface vehicles II: a comparative study of algorithms," *Journal of Marine Science and Technology*, vol. 26, no. 4, pp. 1307–1323, Dec. 2021. [Online]. Available: https://doi.org/10.1007/s00773-020-00790-x

[3] Y. Huang, L. Chen, P. Chen, R. R. Negenborn, and P. H. A. J. M. van Gelder, "Ship collision avoidance methods: State-of-the-art," *Safety Science*, vol. 121, pp. 451–473, Jan. 2020.

[4] "Convention on the International Regulations for Preventing Collisions at Sea, 1972 (COLREGs)." [Online]. Available: https://www.imo.org/en/About/Conventions/Pages/COLREG.aspx

[5] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship Collision Avoidance and COLREGS Compliance Using Simulation-Based Control Behavior Selection With Predictive Hazard Assessment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3407–3422, Dec. 2016.

[6] I. B. Hagen, D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, "MPC-based Collision Avoidance Strategy for Existing Marine Vessel Guidance Systems," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 7618–7623, iSSN: 2577-087X.

[7] T. Tengesdal, E. F. Brekke, and T. A. Johansen, "On Collision Risk Assessment for Autonomous Ships Using Scenario-Based MPC," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14 509–14 516, Jan. 2020.

[8] T. Tengesdal, T. A. Johansen, T. D. Grande, and S. Blindheim, "Ship Collision Avoidance and Anti Grounding Using Parallelized Cost Evaluation in Probabilistic Scenario-Based Model Predictive Control," *IEEE Access*, vol. 10, pp. 111 650–111 664, 2022.

[9] S. V. Rothmund, T. Tengesdal, E. F. Brekke, and T. A. Johansen, "Intention modeling and inference for autonomous collision avoidance at sea," *Ocean Engineering*, vol. 266, p. 113080, Dec. 2022.

[10] H. Tran, T. A. Johansen, and R. Negenborn, "A collision avoidance algorithm with intention prediction for inland waterways ships," in *IFAC World Congress*, 2023, (preprint).

[11] M. Akdağ, T. I. Fossen, and T. A. Johansen, "Collaborative Collision Avoidance for Autonomous Ships Using Informed Scenario-Based Model Predictive Control," *IFAC-PapersOnLine*, vol. 55, no. 31, pp. 249–256, Jan. 2022.

[12] 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*, May 2009, pp. 1–8.

[13] B.-O. H. Eriksen, G. Bitar, M. Breivik, and A. M. Lekkas, "Hybrid Collision Avoidance for ASVs Compliant With COLREGs Rules 8 and 13–17," *Frontiers in Robotics and AI*, vol. 7, 2020.

[14] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Predictive path following with arrival time awareness for waterborne AGVs," *Transportation Research Part C: Emerging Technologies*, vol. 70, pp. 214–237, Sep. 2016.

[15] ——, "Trajectory tracking of autonomous vessels using model predictive control," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 8812–8818, Jan. 2014.

[16] B.-O. H. Eriksen and M. Breivik, *MPC-Based mid-level collision avoidance for ASVs using nonlinear programming*. Institute of Electrical and Electronics Engineers (IEEE), 2017, accepted: 2018-01-24T14:14:43Z Publication Title: 766-772.

[17] S. Blindheim and T. A. Johansen, "Electronic Navigational Charts for Visualization, Simulation, and Autonomous Ship Control," *IEEE Access*, vol. 10, pp. 3716–3737, 2021, conference Name: IEEE Access.

[18] "Miljødirektoratet - Kartkatalog." [Online]. Available: https://kartkatalog.miljodirektoratet.no/dataset/Details/501?lang=en-us