Preprint - 22nd European Control Conference (ECC), Stockholm, Sweden, IEEE Xplore, 2024, pp. 1131-1138.
doi.org/10.23919/ECC64448.2024.10591229

Towards Decision Support in Vessel Guidance Using Multi-Agent Modelling

Luka Grgičević[1], Erlend M. Coates[1], Robin T. Bye[1], Thor I. Fossen[2] and Ottar L. Osen[1]

***Abstract*— This paper presents a decision support system for marine vehicle collision avoidance that utilizes agent-based modelling. It generates waypoints, consecutive strategies of heading changes, and if necessary, speed changes to avoid risky collision situations in multi-vessel encounters. The global collision risk metric is defined as a weighted sum of cost functions, which are computed for each target vessel based on factors such as distance and relative velocity. An evolutionary game theory algorithm based on the replicator dynamics concept is applied to determine the best strategy using competitive agents. The proposed method aims to optimize vessel trajectories considering the risk of collision and deviated path length. The feasibility of the approach is demonstrated using simulation in the NetLogo modelling tool and provides insights into how to define an appropriate model for a scalable agent-based application for vessel guidance algorithm verification.**

I. INTRODUCTION

The decision support system (DSS) can be part of a vessel guidance system that generates rerouting in the shape of heading or speed setpoints, as a reference trajectory, or in the shape of waypoints for the vessel to follow. International Maritime Organisation (IMO) defines the DSS within the first degree of autonomy [1]. Since 2018, IMO has been defining an *S-Mode* concept, developed under the e-Navigation initiative, which attempts to standardize vessel bridge design across different vendors [2]. This implies a complete redesign of the human-machine interfaces onboard or at the remote operating centres (e.g. see [3] and [4]). An integrated navigation system is meant to unify the information from an electronic chart display and information collection system, automatic radar plotting aid, and a situational awareness system. Justifying and conveying the autonomous system information to humans is an open research question.

A system that can be considered as decision support could provide an explainable and computable trajectory or a path for the crew on board or in a remote operating centre. The given advice is generally a function of states of situational awareness, The International Regulations for Preventing Collisions at Sea 1972 (COLREGs) [5], various risk factors, and more. Here, we focus on one of the subsystems, namely collision avoidance. The ideas described in this paper can also be applied to the design of higher levels of autonomy algorithms for collision avoidance (e.g. see [6]). Authors of recent work in [7] and [8], gave a comprehensive review and analysis of such algorithms.

Algorithms for collision avoidance based on model predictive control (MPC) are widely considered the most promising solution. One example is research in [9], with the anticipation of the inclusion of the route exchange in an automatic identification system. A variation of scenario-based MPC was studied in [10], where the collision probability was obtained by integrating the probability distribution function of the states of the target vessel (TV) that intersect the circular safety zone of the own vessel (OV).

Collision avoidance from a differential game perspective has been solved for two vessel encounters, considering minimal turning ability for both the TV and the OV, such that vessels do not violate their safety zones [11].

The velocity obstacle algorithm (e.g., see [12]) finds a collision-free velocity, such that the relative velocity vector is tangential to a desired circular safety zone. It computes admissible velocity sets and generates a tree of solutions, where the nodes with branches of velocities are recomputed each time horizon. Apart from a tunable time horizon, this method allows a provisional cost function for optimizing the consecutive paths. Some improvements of the algorithm that are designed for proactive dynamic obstacles are reciprocal velocity obstacle and hybrid velocity obstacle, introduced by [13] and [14], respectively. Since any speed increase causes quadratic growth of power needed to achieve it, it may be beneficial to have heading and speed strategies (phenotypes) separated. An approach using clean strategies and making a game matrix from heading and speed changes of OV and TVs has been shown in [15].

The DSS presented in this paper finds a set of strategies, decomposed in velocity magnitude and heading, and avoids risky collision situations. The base for the strategy evaluation is the choice of the global collision risk metric. Here, we design it to be a weighted sum of individual cost functions, each corresponding to one instance of the OV, whose function value depends on the number of TVs it has to involve in situational awareness. These cost functions are used to construct a payoff matrix, a concept from game theory and an analogy to the episodic return in a reinforcement learning context. Using a variation of an algorithm based on replicator dynamics (e.g. see [16] and [17]), the OV's best strategy for a given situation is found from the predetermined set.

Replicator dynamics has been used in the comparison of velocity obstacle methods, where each method represents a competitive strategy in a finite population [18]. This

[1] Luka Grgičević, Erlend M. Coates, Robin T. Bye and Ottar L. Osen are with the Department of ICT and Natural Sciences, Norwegian University of Science and Technology (NTNU), Ålesund 6025, Norway, {luka.grgicevic, erlend.coates, robin.t.bye, ottar.osen}@ntnu.no

[2] Thor I. Fossen is with the Department of Engineering Cybernetics, NTNU, Trondheim 7491, Norway, thor.fossen@ntnu.no

evolutionary game theory algorithm has also been shown to match the dynamics of several reinforcement learning algorithms [17]. Paths resulting from the reinforcement learning algorithms may suffer from high curvature, so here we take the opposite approach, by minimizing the number of waypoints that match the number of targets.

The platform for developing and validating the DSS is the NetLogo modelling tool [19]. It has been used in related research on the safety assessment of marine traffic in channels [20], where the authors used the standard protocol for describing the agent-based models [21]. The software has also been used in cluster computing for distributed path search [22].

A. Contributions of this paper

The authors' main contributions are twofold. First, we introduce the concept of a global collision risk metric that is defined as a payoff matrix for an evolutionary algorithm, which in turn is used for separately and subsequently finding suboptimal heading change and speed change strategies. The goal is to find the minimal heading change strategy before reducing speed, which is more energy-consuming. A related example of a global risk factor for multi-vessel encounters can be found in [23] but here we are using a rather different design to construct a multidimensional spatiotemporal game environment. The DSS designed in this paper could be adapted by human-machine interface experts and communicated to human operators visually. However, this is outside of the intended scope.

Second, we reformulate a known kinematic model and make it suitable for guidance algorithm verification using multi-agent modelling. The proposed ideal particle model simply captures the transient response of a change of velocity vector in a *North-East-Down (NED)* frame of reference, using time constants related to the vessel dimensions. A similar model is found in [11] where the modelled change in speed depends on actuator thrust and drag coefficient. Thus, it includes an assumption on dynamics in the kinematic model. It is worth emphasising that the rudder effect in *Nomoto's* model (e.g. see [24]) serves as a means to obtain sway motion. Here, we bypass such modelling in a *BODY* frame of reference by considering closed-loop autopilots.

B. Paper structure and methodology overview

In section II we describe the kinematic model of an ideal particle following a path. Any change of a velocity magnitude or heading has a transient response. The methodology employed in this paper is visually outlined in Fig. 1 and involves a sequence of key steps to enhance decision-making in dynamic maritime scenarios. To commence, the OV first acquires essential information, its own current states and the states of the TVs, upon which various knowledge arrays are built. With this information at hand, the OV proceeds to generate a range of possible scenarios, each corresponding to different path angles. Each of these path angles is associated with an agent, representing a potential course of action that the OV could take. These agents inherit and maintain knowledge arrays that are used for the computation of the expected return value of the particular strategy. In sections III and IV, we delve into the assessment of the time-varying global collision risk metric and the construction of a payoff matrix. This risk metric provides insights into the likelihood of collisions associated with each scenario. For every strategy change, a specific point in time, denoted as T_k, marks the initiation of reward collection. During this period, the agents accumulate rewards based on their chosen course of action. These rewards may encompass various factors, such as safety considerations or efficiency gains. After the reward collection phase concludes at T_{k+1}, the payoff matrix is constructed, which is then used to select the bounded optimal strategy from the chosen set [41]. In section V we show some results from the simulation of a scenario with two TVs, where the DSS produce two imputed waypoints in between the initial path. Remarks on performance requirements, future work and an intuitive way forward are summarized in section VI.

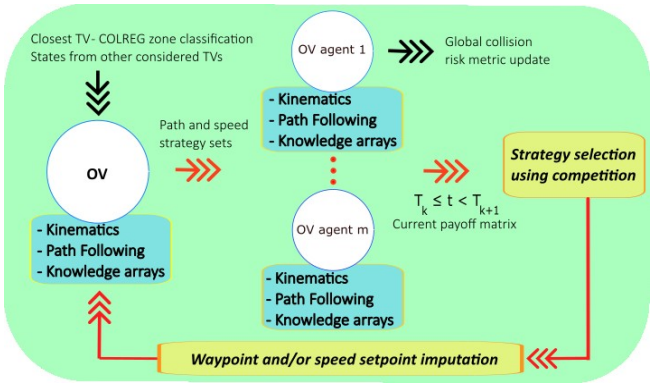

Fig. 1. Methodology block diagram.

II. VESSEL KINEMATICS

The vessel kinematics are designed to be as simple as possible but complex enough to capture the maximum turning ability and speed transient response. Similar models are control system reference models inspired by *mass-spring-damper* systems and they are used frequently due to their simplicity [24]. These models refer to the desired dynamics in the frame of reference attached to the rigid body, the *BODY* frame of reference. Recent research in [25] and references therein have the description of a range of models often used for collision avoidance. The most relevant reason for taking this modelling approach is a reduction in computation time. This becomes particularly important when we have multiple instances of the OV, and each has to compute future position, path following error and the collision probability metrics.

A. Speed and heading kinematics

Each OV model has two degrees of freedom, speed and heading kinematics, and a path-following algorithm. TV models are simply constant velocity obstacles. We model the speed (1) and heading kinematics (2) as decoupled,

inhomogeneous linear differential equations with constant coefficients. The speed transient response is captured by a first order system and the heading with a critically damped second order system. The rate of turn transient response generally has the same degree as the step response in speed [24].

$$\tau_u \dot{U} + U = U_s \tag{1}$$

$$\tau_\psi^2 \ddot{\psi} + 2\zeta\tau_\psi \dot{\psi} + \psi = \psi_s \tag{2}$$

Control inputs are step signals in the *NED* frame of reference, namely the velocity magnitude U and heading angle ψ, respectively. They represent two separate closed-loop systems. Speed U and heading ψ reach desired setpoints, U_s and ψ_s. An assumption can be made that the vessel autopilots generally have negligible overshoots, so the damping factor ζ is set to one. Therefore, the only tuning factors are the time constants τ_ψ and τ_u, which can be estimated using the vessel's turning radius and required stopping distance.

The turning cycle test is performed at a maximum rudder angle for the given test speed. IMO defines tactical radius, a result of the turning cycle test, to be at most 5 vessel lengths and advance no more than 4.5 vessel lengths, whilst the stopping criterion is set to be no longer than 15 vessel lengths [26]. An example of a relation between *Nomoto's* model time constant and gain, and the turning ability of a vessel, via the *Norrbin's* number (*P*), is elaborated in [27].

The weather influence is considered as incorporated within the autopilot models by assuming it does not cause a steady state error nor significantly prolongs the time to reach the steady state but merely increases the control effort. We consider this to be out of scope for verification of guidance algorithms based on multi-agent modelling.

The *North-East* velocities magnitudes v_N and v_E are components of the velocity vector with magnitude U (3). In practice, we obtain them by differentiating the latitude and longitude positions.

$$\begin{bmatrix} v_N \\ v_E \end{bmatrix} = \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix} U \tag{3}$$

B. Path following

Generated waypoints from the DSS constitute a path to follow for our model. The saturated proportional control law, the line of sight (LOS) algorithm (e.g. see [28] and [29]), requires the current *NED* coordinates of the OV (x, y), the heading (ψ) and the coordinates of the starting and ending waypoints (x_{wp1}, y_{wp1}) and (x_{wp2}, y_{wp2}), subsequently. The output of the algorithm is the heading setpoint ψ_s (4) for the heading autopilot model (2), which is the sum of the path angle χ (5) and the desired heading angle $\psi_d \in (-\pi, \pi]$.

$$\psi_s = \chi + \psi_d \tag{4}$$

Considering that the vessel has a planar motion, the path angle can be computed using (5).

$$\chi = \text{atan2}((y_{wp2} - y_{wp1}), (x_{wp2} - x_{wp1})) \tag{5}$$

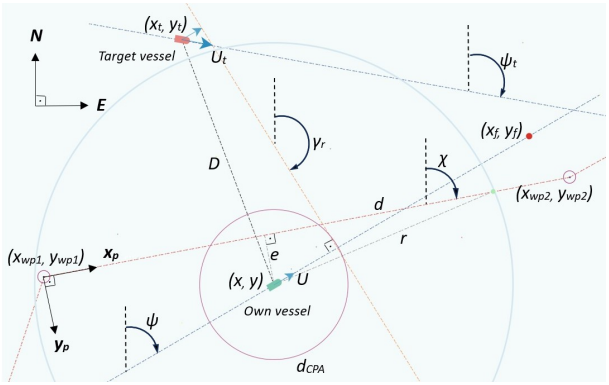

Fig. 2. Visualization of the important symbols used throughout the paper.

The function atan2 computes $\chi \in (-\pi, \pi]$, just like four quadrants arctan. The desired heading angle ψ_d (8) follows the arctan function, which depends on the *cross-track* error e and a *look-ahead* distance d. Here, we introduce a new frame of reference called *PATH*, which has an origin at the starting waypoint (x_{wp1}, y_{wp1}) whose x-axis x_p has been rotated by the path angle χ [24]. The cross-track error e (6) is the distance between a vessel's position (x, y) and its projection on the x-axis in *PATH*. The look-ahead distance d (7) is the distance between the projection and an intersection of a circle centred at (x, y) with a radius r that has to be bigger than e (Fig. 2). The point that we need lies on the x-axis of the *PATH* closer to the waypoint (x_{wp2}, y_{wp2}). The choice for radius r depends entirely on the designer who, naturally, balances between performance and efficiency. In other words, by taking the distance smaller, the control law (8) becomes more aggressive. This radius is usually taken to be an integer multiple of the vessel's length and it can be adaptively tuned to reduce the energy consumption.

$$e = (y - y_{wp1})\cos(\chi) - (x - x_{wp1})\sin(\chi) \tag{6}$$

$$d = \sqrt{r^2 - e^2} \tag{7}$$

$$\psi_d = \arctan\left(\frac{-e}{d}\right) \tag{8}$$

The path angle χ has to be updated in a switching manner after the vessel approaches the next waypoint at a distance closer than the threshold [30]. Function *ssa* (9), used in (4) and (2), wraps the angle α to $(-180, 180]$, as in [24].

$$ssa(\alpha) = (\alpha + 180) \bmod (360) - 180 \tag{9}$$

III. COLLISION RISK METRIC

In this section, we present a method of environment design for competitive agents.

The metric presented here has some resemblance with object future states probability fields that are used for collision risk assessment in [31]. Modelling rewards (positive or negative) for future states or constraints with a convex function can also be found in literature describing control barrier functions [32], artificial potential field functions [33] and risk assessment functions (e.g. see [34] and [35]).

A. Preliminaries

We assume that the OV wants to consider a strategy change if any of the TVs is in the D_w range. The distance D_w is taken provisionally as the radius in which estimates of TVs' positions and velocities are made and represents the sufficient distance to start rerouting if the speeds are not exceeding some maximum value. It is a tunable parameter that influences the global collision risk metric and it can be kept at $5\mathrm{nmi}$ (nautical miles). The bigger the radius, the higher the uncertainties in estimates, so the trade-off has to be made not to increase the computational cost. In section III-C, we elaborate on the appropriate scaling of the global collision risk metric.

A risk of collision with one TV is represented as one convex cost function with a maximum located at the coordinates of the OV when the time to closest point of approach (CPA) is equal to zero. The coordinates of the OV at that time are denoted as (x_f, y_f), and the ones from TV as (x_{cpa}, y_{cpa}). To obtain those coordinates we assume that the TV, located at (x_t, y_t), and OV have constant velocities. The TV's velocity components are v_{Nt} and v_{Et}. It is located at the distance D from the OV. We can compute the relative velocity angle using (10). The location of the TV when the time to CPA (t_{cpa}) is zero, can be computed with (11) and (12).

$$\gamma_r = \operatorname{atan2}((v_{Nt} - v_N), (v_{Et} - v_E)) \tag{10}$$

$$x_{cpa} = x_t + (y_{cpa} - y_t)\tan(\gamma_r) \tag{11}$$

$$y_{cpa} = \frac{x - x_t + y_t\tan(\gamma_r) - y\tan(\gamma_r - \pi/2)}{\tan(\gamma_r) - \tan(\gamma_r - \pi/2)} \tag{12}$$

Next, we compute the distance to the CPA (d_{cpa}) using (13). Assuming at least one speed to be nonzero, the t_{cpa} could be computed using (14).

$$d_{cpa} = \sqrt{(y_{cpa} - y)^2 + (x_{cpa} - x)^2} \tag{13}$$

$$t_{cpa} = \sqrt{\left|\frac{d_{cpa}^2 - D^2}{U^2 + U_t^2 - 2(v_{Nt}v_N + v_{Et}v_E)}\right|} \tag{14}$$

Finally, the estimates of coordinates of the OV are then computed using (15) and (16). Those are the coordinates of individual cost function maxima.

$$y_f = y + v_E t_{cpa} \tag{15}$$

$$x_f = x + v_N t_{cpa} \tag{16}$$

B. Individual cost function

In the rest of the paper, we add a subscript $j \in \mathbb{N}^+, j \leq m$ to OV agents and $i \in \mathbb{N}^+, i \leq n$ to TVs.

Along the OV agents' path, n cost functions are generated, each belonging to one TV in the range D_w and approaching in distance. An individual cost function f_i depends on the coordinates and velocities of OV and the TV, as stated in (17).

$$f_i = s_i \cdot \exp\left(-\frac{(p_x - x_f)^2 + (p_y - y_f)^2}{s_i^4}\right) \tag{17}$$

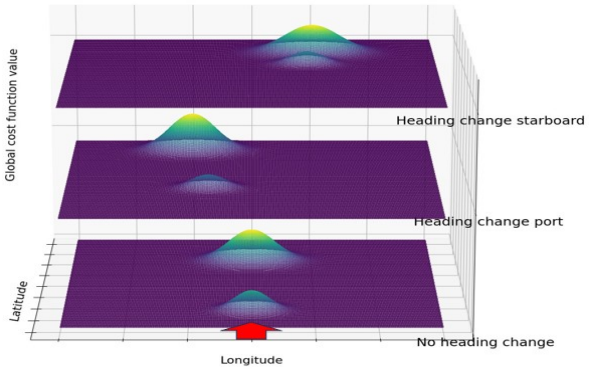

Fig. 3. Global cost functions for three equiangular OV's strategies encountering two targets each.

The NetLogo environment is made of square *patches* with coordinates of the centre (p_x, p_y). It should be noted that the coordinates of the patches' centres and the coordinates of a vessel are taken as unitless values. We define scalar s_i, with (18).

$$s_i = c\left(1 - \frac{d_{cpa}}{D_w}\right) \tag{18}$$

The scaler s_i will linearly increase up to a maximum value of c, as the d_{cpa} shrinks. Constant c defines an individual function's maximum value. It will influence the total reward individual agents collect but changing it will not modify the outcome of the strategy competition.

As an illustrative example, a snapshot of time-varying cost functions for three strategies (turn port, turn starboard and stay on the course) encountering two TVs can be depicted in Fig. 3.

The cost for approaching static obstacles would take the shape that follows the diffusion or continuity equation. Authors in [6] and [36] used piecewise linear land contours as bounds in optimization. The diffusive cost could drop in a direction perpendicular to an individual piecewise-linear approximation of the obstacle.

C. Mixture of cost functions

The total cost function of an individual OV agent j is a weighted sum of individual cost functions (18).

$$F_j = \sum_{i=0}^{n} w_i f_i \tag{19}$$

For the sake of clarity, these weights might be denoted by both subscripts since they depend on the states of both OV and the TVs. The weighting factors w_i for constructing the global cost function F_i represents how important a particular TV is to a given OV, in cost function reduction. They are defined by factors such as distance D_i between OV and TVs, time t_{cpa}, the ratio between OVs and TVs' length and mass, environmental influence on the controllability of a vessel and situational awareness. An alternative could be simply to estimate w_i as a multiplication of normalized relative bearing and normalized distance to TVs. Recent

research in [37] uses fuzzy sets for similar tasks and related to COLREGs interpretation. To conclude, with this we create a spatiotemporal environment for each agent.

IV. WAYPOINTS GENERATION

Suppose we generate m OV agents that inherit all the states and properties of the original OV model. A predetermined set of m strategies is made of the desired headings $\{h_1, h_2, ..., h_m\}$. Those strategies are equiangular lines that intersect at the coordinates of the OV. Angle in between strategies is a tunable parameter. This set should not be symmetric with respect to the OV heading at the time the agents are generated. It is custom to classify the TVs into four groups based on zone entry criteria for COLREGs 13-17: *crossing give-away*, *crossing stand-on*, *overtaking* and *head-on* [38]. Having this in mind, the set should be generated such that the desired strategies fall into one of the zones.

Each generated agent is assigned a path to follow, which has path angle χ (5). Agents follow the strategies using LOS (20).

$$\chi = h_j \tag{20}$$

A. Payoff matrix for competitive agents

An individual agent collects rewards by summing values stored in patches. Each patch stores those values for every agent separately, determined by their global cost function, and changes them dynamically. So, each patch stores and maintains a list of global cost functions, whose length depends on the number of agents.

We propose that the total accumulated reward for an individual agent takes into account the global cost functions of other agents as well. In other words, each agent integrates all of the global cost functions in parallel as it moves over the patches and maintains an array of their values, whose length depends on the number of agents.

We can denote enumerated patch as p, with a known square edge length and center coordinates, to be a part of the set P that is made of patches which are intersected by the agent's path.

With (21) and (22), we define a payoff that agent j, playing strategy h_j, gets by competing against agent m, playing strategy h_m, and vice versa. Rewards accumulation is periodical, and on its end, the strategy competition is played. The kth reward accumulation lasts over one time period Δ of simulation starting at T_k and ending at T_{k+1}.

$$R_{jm} = \sum_{p \in P} b_j F_m(p) \tag{21}$$

$$R_{mj} = \sum_{p \in P} b_m F_j(p) \tag{22}$$

Weights b are designed to keep the vessel on the original path so that imputed waypoints are not far from the original track. They are further elaborated in subsection IV-C.

We propose that Δ depends on the largest t_{cpa} that is calculated for each TV at times T_1, T_2, T_3 etc, the time agents' strategies are determined. In this way, we increase the number of TVs that define the individual agent's payoff, therefore the winning strategy for that time period Δ. This has the downside of having large values of Δ since it depends on how large is the D_w. For example, the solution might simply be to let the agents collect the rewards from patches for half an hour (not simulation time), because this is usually the time upper bound for relatively accurate trajectory estimates of large and slow maritime vehicles.

An example of the total payoff matrix for strategies competing against each other can be depicted in Table 1 In Fig. 4, an example situation is presented. After the decision to evaluate heading change has been initiated, the TVs' position is estimated by considering constant velocity, movement of the OV is frozen at T_0, and 10 agents have taken different strategies. Each agent embeds the global cost functions into patches sequentially at every environment refresh rate δ. Small circles with a dot in the middle represent waypoints along the agents' paths for better visualization. Only one function, F_4, is plotted along the position of the three CPAs, marked as red dots, and three d_{cpa} circles.

TABLE I

TIME-VARYING PAYOFF MATRIX

	OV_1	OV_2	OV_j	OV_m
OV_1	R_1	R_{12}, R_{21}	R_{1j}, R_{j1}	R_{1m}, R_{m1}
OV_2	R_{21}, R_{12}	R_2	R_{2j}, R_{j2}	R_{2m}, R_{m2}
OV_j	R_{j1}, R_{1j}	R_{j2}, R_{2j}	R_j	R_{jm}, R_{mj}
OV_m	R_{m1}, R_{1m}	R_{m2}, R_{2m}	R_{mj}, R_{jm}	R_m

B. Strategy selection using replicator dynamics

Each strategy j is a member of a finite population of size N, which every playing cycle t stays constant. One playing cycle lasts until all the agents have played against the rest, in a round-robin fashion. We start with a provisional but equal number of agents that play one of the strategies from the predefined set. The number of agents playing a particular strategy j in the next cycle $a_j(t+1)$ (25) is determined by the growth function g (24). This function value depends simply on the payoff matrix and a fraction of agents playing other strategies. Since this type of replication does not have the counterforce to stabilize the population number, we normalize the population size N (23) at the end of each cycle. The strategy that plays the one-shot game with all the rest is chosen sequentially.

$$N = \sum_{j=1}^{m} a_j \tag{23}$$

$$g_j(t) = \frac{a_1(t)}{N} R_{j1} + \frac{a_2(t)}{N} R_{j2} + ... + \frac{a_m(t)}{N} R_{jm} \tag{24}$$

$$a_j(t+1) = a_j(t)(1 + g_j(t)) \tag{25}$$

It is important to emphasise this algorithm does not depend on the *time* used in the simulation but converges rather

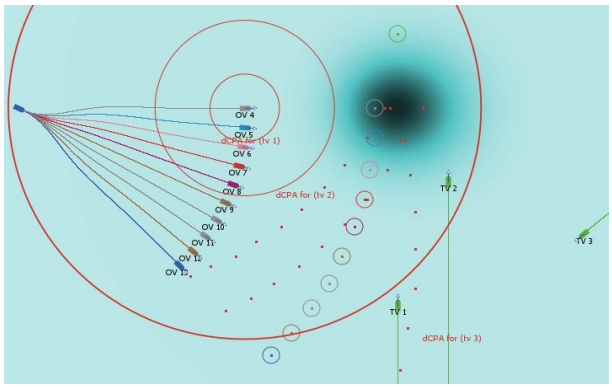

Fig. 4. 10 strategies in a vicinity of 3 TVs.

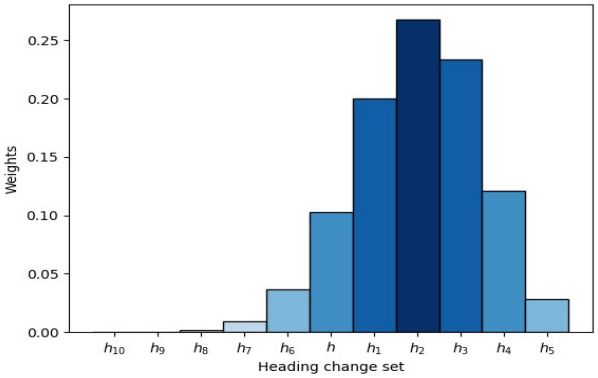

Fig. 5. Weights b for 11 strategies, where the h_2 has the minimal relative bearing towards the original waypoint.

quickly to a Nash equilibrium in a finite number of cycles marked as t in (24) and (25).

C. Consecutive strategies

The chosen strategy will be the one with the biggest population after the competition is terminated. The relatively short convergence time depends on the number of strategies and the payoff matrix entries. After the j^{th} strategy has been chosen, the environment is reset and the agents are eliminated. We resume the simulation such that OV has a new waypoint to reach, which is generated at the coordinates of the mean of the closest individual cost function f_i, computed after time period Δ. Those would be the coordinates of the winning agent when the first t_{cpa}, in the list it maintained, reached zero. When the OV approaches the waypoint closer than the threshold, new time T_{k+1} is recorded and a new Δ is computed or set. The procedure is repeated until there are no COLREGs-applicable situations to resolve or static obstacles along the path.

Since there is no guarantee that OV will pass on a safe distance from the TVs, it is mandatory to check if the winning agent's d_{cpa} of the closest TV is smaller than the safety distance. More precisely, if an agent, following a particular strategy, encounters the closest TV with a miss distance (d_{cpa}) that is smaller than the safety zone, that strategy should be removed from the game as a candidate. Afterwards, we repeat the game with a truncated payoff matrix.

Moreover, in order to keep the generated waypoints not too far from the original path we propose to use a simple spring mechanism to penalize strategies that give greater relative bearing towards the original waypoint before imputing the intermediate ones. Adding additional weights b in each cost function from (21) and (22), such that the sum is always one, should serve the purpose. In Fig. 5, five strategies are generated from the starboard and five from the port, while the initial desired waypoint lies towards the starboard. The weight distribution is a skewed binomial. In this case, we increase the cost function of the strategies that are not deviating from the original plan.

If there is no heading change strategy that does not violate TVs' safety zones, the same procedure could be run for a velocity magnitude U increase or reduction strategy.

V. PRELIMINARY RESULTS

In this section, we show a simple scenario with two TVs. After the targets approached closer than D_w, we generated 9 agents that inherited all the states and properties from the OV. The offset between equiangular paths was 5°. The reward collection time period Δ was chosen to be constant. In Fig. 6, only the global cost function associated with agent 3 is plotted, along with two d_{cpa} circles associated with the two TVs. The first chosen strategy was the one from agent 6. At this point, we eliminated all the agents and set the patch values to zero. The simulation continued by imputing the new intermediate waypoint for the OV, having coordinates of the mean (x_f, y_f) of the first f_i on the agent's 6 path. TVs continued with constant velocities. After the OV reached this waypoint, as it can be seen in Fig. 7, nine new agents were generated and the game was played again in which agent 34 prevailed. Since there were no more targets after the second imputed waypoint was reached, OV continued to its original goal. The resulting OV trajectory and the two additional imputed waypoints, can be depicted in Fig. 8. We noticed that the winning strategy tends to traverse the TV from the aft in the crossing situations. This could be explained by depicting the location of f_i functions' maxima for the closer TV, as they form an arch made of red dots falling from left to right in Fig. 6. OV agents that are traversing the TV from the fore are collecting more rewards from the other agents that are traversing the TV from the aft. The kinematic equations were solved numerically using backward Euler approximation.

VI. CONCLUSIONS

We have presented a method that simplifies potentially complex multi-vessel encounters to a multidimensional spatiotemporal environment that could serve as a base for intelligent agents' learning, competition or cooperation. In this case, we used a game theory algorithm for strategy competition. COLREGs compliance level could be improved by generating strategies that are inside one of the four

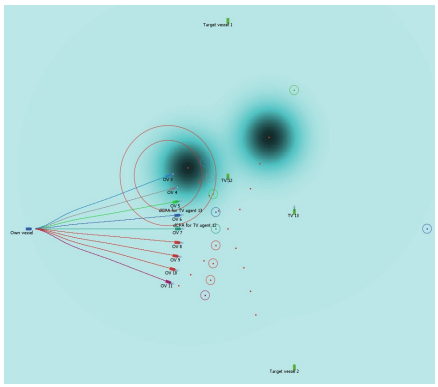

Fig. 6. Situation before the first strategy was chosen.

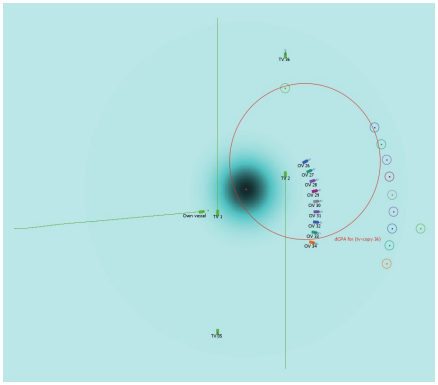

Fig. 7. Agents were generated again to deal with the second target.

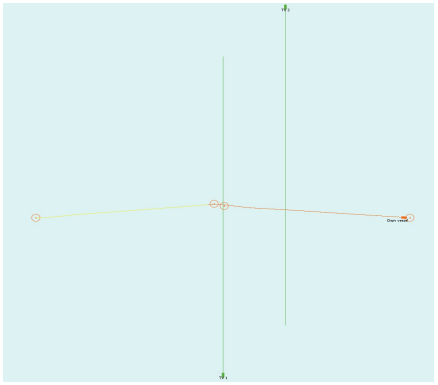

Fig. 8. OV has two imputed waypoints from agent 6 and 34, subsequently.

permissible groups, as mentioned in section III.

A. Implementation and preformance

To get an idea about the requirements for practical implementation we have to consider recomputation time (δ) of the global collision risk metric (environment refresh rate), well-filtered estimates of TVs' and OV's velocity vectors and environment patch size. The environment refresh rate should depend on the biggest speed of TV that is considered, i.e. closer than D_{wp}. Since the time frame between computations of the multidimensional spatiotemporal environment (δ) determines the payoff matrix, it is important to balance between computation cost and the desired accuracy. Computations in section III require good estimates of the vessel's position, speed and heading. For example, research in [39] describes how to obtain speed and course over ground using only global navigation satellite system receiver. The major issue remains in reducing uncertainty for the trajectory estimates of the vessels using intentions probability network models (e.g., see [40]) or using destination (or future waypoints) knowledge. For Python implementation, environment patches could be hexagons of provisionally fine resolution.

B. Future work

We plan to use Netlogo's *behavior space* feature to generate scenarios with a provisional number of TVs with different weights w_i, headings, and speeds. The goal is to analyze the results after generating a provisional number of equiangular OV strategies and changing the heading differences in between. The natural way of analysis would be using the simplex method for n strategies to study the dynamics of the game with a payoff matrix that depends on a traffic situation. To increase the computational reducibility, the collected cost after following the initial path can be approximated by the explicit integral of the global cost function along the path, if we assume NetLogo patches in limit shrink to points. Bounds of integration would depend on reward collection time frame Δ. Extending the game such that TVs are also players would lead to a similar methodology of constructing the payoff matrix. In that case, strategies are turned to available actions and the obtained Nash equilibrium for the *local* traffic situation would be computed centrally in the Vessel Traffic Service station and advised to individual vessels. Moreover, we plan to collaborate with the human-machine interface experts and conceptualize the prototype for collision avoidance assistance that utilizes knowledge arrays and easily visualisable global collision risk metrics and knowledge arrays.

VII. ACKNOWLEDGMENT

We would like to thank Agus Hasan and Eirik Fagerhaug, affiliated with the Cyber-Physical Systems (CPS) Laboratory research group at the Department of Information and Communication Technology (ICT) and Natural Sciences at NTNU in Ålesund, for fruitful discussions and constructive feedback over the past year. This research is funded by Doctoral Programme for Integrated Research Activities to Unlock a Potential of Top-level Researchers in Digital Transformation for Sustainability (PERSEUS), the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement number 101034240. This research is also funded by SFI AutoShip, an 8-year research-based innovation Centre focusing on safe and sustainable autonomous ship operations. We would like to thank our partners, including the Research Council of Norway under Project number 309230.

REFERENCES

[1] International Maritime Organisation, Maritime Safety Committee, 100th session, 3-7 December 2018

[2] International Maritime Organisation, Sub-Committee on Navigation, Communications and Search and Rescue (NCSR), 5th session, 19-23 February 2018

[3] K. Nordby, E. Gernez and S. C. Mallam, OpenBridge: designing for consistency across user interfaces in multi-vendor vessel bridges, *Ergovessel 2019*, pp. 60-68, 2019, ISBN: 978-82-93677-04-8

[4] O. A. Alsos, E. Veitch, L. Pantelatos, K. Vasstein, E. Eide, F. Petermann and M. Breivik, NTNU Shore Control Lab: Designing shore control centres in the age of autonomous ships, *Journal of Physics: Conference Series*, vol. 2311, no. 1, pp. 012030, 2022, DOI: 10.1088/1742-6596/2311/1/012030

[5] I.M.O.(IMO), COLREGS - International Regulations for Preventing Collisions at Sea, *Convention on the International Regulations for Preventing Collisions at Sea, 1972*, 1972

[6] E.H. Thyri and M. Breivik, A domain-based and reactive COLAV method with a partially COLREG-compliant domain for ASVs operating in confined waters, *Field Robotics*, vol. 2, pp. 637–677, 2022, DOI: 10.55417/fr.2022022

[7] A. Vagale, R. Oucheikh, R. T. Bye, O. L. Ossen 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, 2021, DOI: 10.1007/s00773-020-00787-6

[8] A. Vagale, R. T. Bye, R. Oucheikh, O. L. Ossen 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, 2021, DOI: 10.1007/s00773-020-00790-x

[9] 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, 2022, DOI: 10.1016/j.ifacol.2022.10.439

[10] T. Tengesdal, E. Brekke and T. A. Johansen, On Collision Risk Assessment for Autonomous Ships Using Scenario-Based MPC, *IFAC-PapersOnLine*, vol. 53, pp. 14509-14516, 2020, DOI: 10.1016/j.ifacol.2020.12.1454

[11] T. Miloh and S. D. Sharma, Maritime Collision Avoidance as a Differential Game, *Schifftechnik*, vol. 24, no. 1, pp. 69-88, 1977, DOI: 10.15480/882.476

[12] P. Fiorini and Z. Shiller, Motion Planning in Dynamic Environments Using Velocity Obstacles, *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760-772, 1998, DOI: 10.1177/027836499801700706

[13] J. van den Berg, M. Lin, and D. Manocha, Reciprocal velocity obstacles for real-time multi-agent navigation, *IEEE International Conference on Robotics and Automation*, pp. 1928-1935, 2008, DOI: 10.1109/ROBOT.2008.4543489

[14] J. Snape, J. van den Berg, S. Guy and D. Manocha, The Hybrid Reciprocal Velocity Obstacle, *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696-706, 2011, DOI: 10.1109/TRO.2011.2120810

[15] J. Lisowski, Synthesis of a Path-Planning Algorithm for Autonomous Robots Moving in a Game Environment during Collision Avoidance, *Electronics*, vol. 10, no. 6, pp. 675, 2021, DOI: 10.3390/electronics10060675

[16] I. M. Bomze, Dynamical aspects of evolutionary stability, *Monatshefte für Mathematik*, vol. 110, no. 3-4, pp. 189–206, 1990, DOI: 10.1007/BF01301675

[17] D. Bloembergen, K. Tuyls, D. Hennes and M. Kaisers, Evolutionary dynamics of multi-agent learning: a survey, *Journal of Artificial Intelligence Research*, vol. 53, no. 1, pp. 659–697, 2015, DOI: 10.5555/2831071.2831085

[18] D. Hennes, D. Claes and K. Tuyls, Evolutionary Advantage of Reciprocity in Collision Avoidance, *Proceedings of the AAMAS 2013 Workshop on Autonomous Robots and Multirobot Systems*, 2013

[19] U. Wilensky, NetLogo, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

[20] F. Xiao, H. Ligteringen, C. van Gulijk and B. Ale, Nautical traffic simulation with multi-agent system for safety, *16th International IEEE Conference on Intelligent Transportation Systems*, pp. 1245-1252, 2013, DOI: 10.1109/ITSC.2013.6728402

[21] V. Grimm, U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S. K. Heinz, G. Huse, A. Huth, J. U. Jepsen, C. Jørgensen, W. M. Mooij, B. Müller, G. Pe'er, C. Piou, S. F. Railsback, A. M. Robbins, M. M. Robbins, E. Rossmanith, N. Rüger, E. Strand, S. Souissi, R. A. Stillman, R. Vabø, U. Visser, and D. L. DeAngelis, A standard protocol for describing individual-based and agent-based models, *Ecological Modelling*, vol. 198, pp. 115-126, 2006, DOI: 10.1016/j.ecolmodel.2006.04.023

[22] I. Muscalagiu, H. E. Popa and J. M. Vidal, Clustered Computing with NetLogo for the evaluation of asynchronous search techniques, *12th IEEE International Conference on Intelligent Software Methodologies, Tools and Techniques*, pp. 115-120, 2013, DOI: 10.1109/SoMeT.2013.6645651

[23] Z. Liu, Z. Wu, Z. Zheng, A cooperative game approach for assessing the collision risk in multi-vessel encountering, *Ocean Engineering*, vol. 187, 2019, DOI: 10.1016/j.oceaneng.2019.106175

[24] T. I. Fossen, Handbook of Marine Craft Hydrodynamics and Motion Control, 2nd Edition, John Wiley and Sons, 2021, ISBN: 978-1-119-57505-4

[25] 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, 2020, DOI: 10.1016/j.ssci.2019.09.018

[26] International Maritime Organisation, Maritime Safety Committee, Resolution MSC.137(76), Standards for vessel Manoeuvrability, 4 December 2002

[27] D. Clarke, P. Gendling and G. Hine, The Application of Manoeuvring Criteria in Hull Design Using Linear Theory, *The Royal Institution of Naval Architects*, 1982

[28] T. I. Fossen and K. Y. Pettersen, On uniform semiglobal exponential stability (USGES) of proportional line-of-sight guidance laws, *Automatica*, vol. 50, no. 11, pp. 2912-2917, 2014, DOI: 10.1016/j.automatica.2014.10.018

[29] T. I. Fossen, M. Breivik and R. Skjetne, Line of sight path following of underactuated marine craft, *IFAC Conference on Manoeuvring and Control of Marine Craft*, vol. 36(21), pp. 211–216, 2003, DOI: 10.1016/S1474-6670(17)37809-6

[30] A. M. Lekkas and T. I. Fossen, Line of sight guidance for path following of marine vehicles, *Advanced in Marine Robotics - section 5*, pp. 63-92, 2013, ISBN: 978-3-659-41689-7

[31] E. A. Williams, Y. Jin and P. C. Schulze, Development of Probability Fields for Collision Avoidance, *AIAA AVIATION 2020 FORUM*, 2020, DOI: 10.2514/6.2020-2863

[32] M. Marley, R. Skjetne, E. Basso and A. R. Teel, Maneuvering with safety guarantees using control barrier functions, *13th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles*, vol. 54, no. 16, pp. 370-377, 2021, DOI: 10.1016/j.ifacol.2021.10.118

[33] M. D. Pedersen and T. I. Fossen, Marine Vessel Path Planning and Guidance Using Potential Flow, *9th IFAC Conference on Manoeuvring and Control of Marine Craft*, vol. 45, no. 27, pp. 188-193, 2012, DOI: 10.3182/20120919-3-IT-2046.00032

[34] A. Vagale, O. L. Osen and R. T. Bye, Evaluation of Path Planning Algorithms of Autonomous Surface Vehicles Based on Safety and Collision Risk Assessment, *Oceans 2020 Singapore - U.S. Gulf Coast*, pp. 1-8, 2020, DOI: 10.1109/IEEECONF38699.2020.9389481

[35] 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*, vol. 31, pp. 464–488, 2017, DOI: 10.1002/acs.2561

[36] S. Blindheim and T. A. Johansen, Electronic Navigational Charts for Visualization, Simulation, and Autonomous Ship Control, *IEEE Access*, vol. 10, pp. 3716-3737, 2022, DOI: 10.1109/ACCESS.2021.3139767

[37] A. Bakdi and E. Vanem, Fullest COLREGs Evaluation Using Fuzzy Logic for Collaborative Decision-Making Analysis of Autonomous Ships in Complex Situations, *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18433-18445, 2022, DOI: 10.1109/TITS.2022.3151826

[38] K. Woerner, M. Benjamin, M. Novitzky and J. Leonard, Quantifying Protocol Evaluation for Autonomous Collision Avoidance, *Autonomous Robots*, vol. 43, no. 4, pp. 967-991, Springer-Verlag, 2018, DOI: 10.1007/s10514-018-9765-y

[39] S. Fossen and T. I. Fossen, Five-State Extended Kalman Filter for Estimation of Speed over Ground (SOG), Course over Ground (COG) and Course Rate of Unmanned Surface Vehicles (USVs): Experimental Results, *Sensors*, vol. 21, no. 23, pp. 7910, 2021, DOI: 10.3390/s21237910

[40] S. V. Rothmund, T. Tengesdal, E. F. Brekke, T. A. Johansen, Intention modeling and inference for autonomous collision avoidance at sea, *Ocean Engineering*, vol. 266, pp. 13080, 2022, DOI: 10.1016/j.oceaneng.2022.113080

[41] S. J. Russell and D. Subramanian, Provably bounded-optimal agents. *Journal of Artificial Intelligence Research*, vol. 2, no. 1, pp. 575–609, 1994, ISSN: 1076-9757