TRANSNAV
http://www.transnav.eu
the International Journal on Marine Navigation and Safety of Sea Transportation
Volume 17
Number 1
March 2023

DOI: 10.12716/1001.17.01.20

Effect of Ship Neural Domain Shape on Safe and Optimal Trajectory

J. Lisowski
Gdynia Maritime University, Gdynia, Poland

ABSTRACT: This article presents the task of safely guiding a ship, taking into account the movement of many other marine units. An optimally neural modified algorithm for determining a safe trajectory is presented. The possible shapes of the domains assigned to other ships as traffic restrictions for the particular ship were subjected to a detailed analysis. The codes for the computer program Neuro-Constraints for generating these domains are presented. The results of the simulation tests of the algorithm for a navigational situation are presented. The safe trajectories of the ship were compared at different distances, changing the sailing conditions and ship sizes.

1 INTRODUCTION

According to the International Association of Marine Aids to Navigation and Lighthouse Authorities (IALA-AISM) [1], a domain is defined as the operational zone around, above or below a vessel within which an incursion by another fixed or moving object, or another domain, may trigger reactions or processes.

Domains as a tool for avoiding the risk of collision are classified by Jurdzinski [2] as two-dimensional domains in the form of circular, elliptical, polygonal and mixed forms and three-dimensional domains in the form of an ellipsoidal and irregular solid.

Goodwin [3], Davis et al. [4] and Colley et al. [5] conducted the first domain analyses. Detailed analyses of domain properties in various applied conditions were carried out by Szlapczynski et al. [6], Starup [7], Chen et al. [8] and Wang [9]. Pietrzykowski and Wielgosz [10,11] and Marcjan et al. [12] presented the use of domains for planning anticollision maneuvers.

Domain design based on information from an AIS system was carried out by Horteborn et al. [13] and Zhang et al. [14], Smierzchalski [15] and Zhu et al. [16] made the first attempts to shape domains using artificial intelligence.

Thus far, no comparison of the possible domain shapes for the optimality of a planned safe trajectory of a ship has been performed.

The thesis of this paper is to show that through the experimental analysis of different domain shapes, it is possible to assess the optimality of cruise routes in emergency situations.

The new elements of this paper, contributing to the development of methods and tools for safe maritime transport, are:

– A detailed comparative analysis of safe trajectories for different domain shapes, determined according to the modified optimally neural algorithm, which was previously developed by Lisowski [17];

– An analysis of the optimality as a function of safe passing distances of ships and the discretization of the computer calculations.

This paper's content is presented as follows: In Section 1, we introduce a synthetic review of the literature. Section 2 presents a model of the safe and optimal ship control process. The optimally neural algorithm for safe ship trajectory is described in Section 3. Section 4 illustrates the results of the algorithm simulation studies. Section 5 concludes this paper and looks at future directions for improvement.

2 TASK OF SAFE SHIP GUIDING

The description of the task of safely guiding ships consists of kinematics and dynamics equations, in the following form:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t) \tag{1}$$

where $\boldsymbol{x}$ are the measurable variables of the object dynamics; x_1 and x_2 are the components of the ship's position; x_3 is the ship's course; x_4 is the angular velocity; x_5 is the speed; x_6 is the acceleration of the ship's motion; and t is time.

The state and control constraints are:

$$g(\mathbf{x}) \geq 0 \tag{2}$$

$$h(\mathbf{u}) \leq 0 \tag{3}$$

where $\boldsymbol{u}$ is the control quantities: u_1—rudder angle; u_2—rotational speed or pitch of the ship's propeller.

Contrary to the stabilization of the ship's course at small angles of rudder deflection, anticollision maneuvering, according to Rule 8b COLREGs [17], requires visibly larger changes of course at greater rudder deflections.

Therefore, nonlinear ship dynamics equations were used here.

On the other hand, the state constraints contain kinematic equations of the motion of passing ships. These constraints, in their simplest form, take a fixed or variable shape generated by an artificial neural network [18,19].

3 OPTIMALLY NEURAL ALGORITHM FOR SAFE SHIP TRAJECTORY

Dynamic programming with the following control quality index can be used to solve the task of calculating the optimal time to complete a safe ship trajectory:

$$\min_{u} \int_0^t x_5 dt = t^* \tag{4}$$

This leads to the determination of the ship's optimal trajectory, safely avoiding encountered ships as the neural constraints of this process states, as shown in Figure 1.

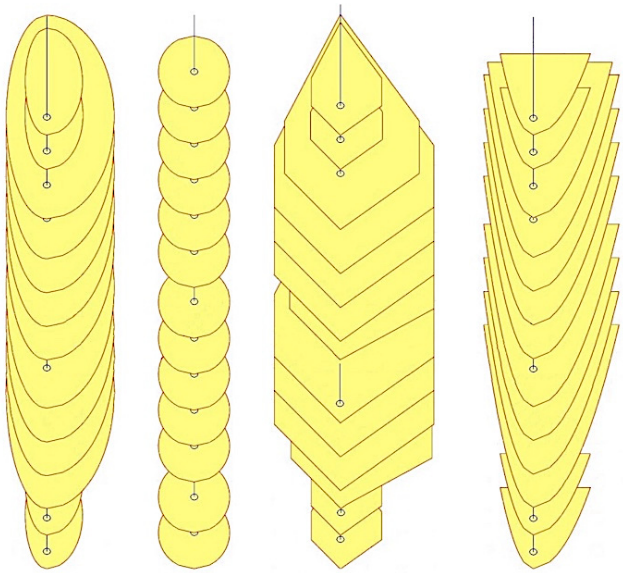

Figure 1. Movement of various shapes of ship neural domains in the shape of an ellipse, circle, hexagon and parabola, illustrating the mapping of subjectivity of the navigator assessing the risk of collision when passing another ship.

Four types of ship domain shapes, in the form of an ellipse, circle, hexagon and parabola, were examined and compared in this study. A pretrained three-layer neural network with hyperbolic tangent and logistic activation functions was used to generate these domains.

The MATLAB Neural Network Toolbox was utilized to synthesize the neural network, and an error propagation method with an adaptive learning rate and momentum was employed for its adaptation. The network was tested against various practical scenarios and its expected response from approximately 350 experienced navigators.

For each variant of the scenarios, a navigator subjectively decided, in accordance with their previous sea practice, the approximately appropriate change of course or ship speed. The neural network prepared in this way represented the average experience of a large group of navigators.

The final synthesis of the optimally neural algorithm combined dynamic programming as a discrete optimization multistep method with the neural network mapping real state constraints of the collision avoidance process with other ships.

Figure 2 illustrates the functioning of the optimally neural control algorithm, developed by the author of this paper, for determining safe route of further voyage while avoiding many ships.

An innovative solution to the task of determining the optimal and safe trajectory of a ship when passing a larger number of other ships is the design of an artificial neural network generating ship domains and the use of the Bellman dynamic programming method to synthesis the control algorithm [20-22].

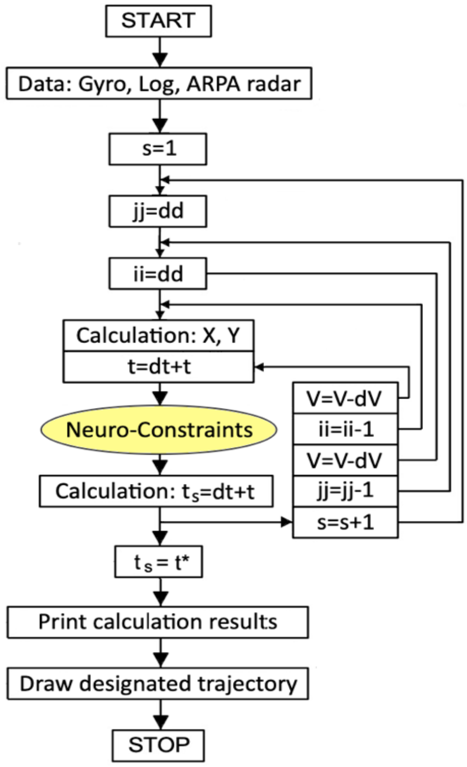

Figure 2. Optimally neural algorithm for determining a safe trajectory: V – own ship speed, s – number of stage, dd – number of nodes in one stage, ii - node number in the previous stage, jj - node number at the current stage, t^* - optimal, shortest safe trajectory time.

A computer program in the MATLAB/Simulink software representing the Neuro-Constraints procedure developed by the author of this work (Algorithm 1) is shown on the right. *The learning material this neuralnetwork consisted of navigational scenarios, 300 responses were recorded from experienced navigators in ARPA system training courses.*

Algorithm 1: Neuro-Constraints

```
Function [V1] = Neuro-Constraints(V1, X1, X2, T5,zn,SpeedShip,CourseShip)
load Nowagi2Good.mat
load Area hexagon parabola ellipse circle
global KURSc Voc Gc Kc JJc tab1
if CourseShip<0
CourseShip =2*pi+ CourseShip;
end;
DeltaT = T5 ;
Course=(COURSEc*pi)/180;
BearingObject(1 :Gc, 1) = (tab1(1 :Gc, 1))*pi/180;%radian
DistanceObject(1 :Gc, 1) = tab1(1 :Gc, 2); %mile
CourseObject(1 :Gc, 1) = (tab1(1 :Gc, 4))*pi/180;%radian
if zn ==1
for Gc=1:Gc,
if Course  <= CourseObject(Gc, 1)
CourseObject (Gc, 1) = (CourseObject(Gc,1)-Course) ;
end;
if Course > CourseObject (Gc, 1)
CourseObject (Gc, 1) = ((2*pi + CourseObject (Gc,1)-Course));
end;
if CourseShip  <= BearingObject(Gc, 1)
BearingRel (Gc, 1) = (BearingObject (Gc,1)- CourseShip);
end;
if CourseShip > BearingObject (Gc, 1)
BearingRel(Gc, 1) = ((2*pi + BearingObject (Gc,1)- CourseShip ));
end;
Xpocz(Gc,1)=DistanceObject(Gc,1)*sin(BearingRel (Gc,1));
Ypocz(Gc,1)= DistanceObject (Gc,1)*cos(BearingRel(Gc,1));     AbsolX(Gc,1)=
DistanceObject(Gc,1)*sin(BearingObject (Gc,1));     AbsolY(Gc,1)=
DistanceObject(Gc,1)*cos(BearingObject(Gc,1));
AbsolXY(Gc,:)=[AbsolX(Gc,1) AbsolY(Gc,1) SpeedShip tab1(Gc,3) CourseObject(Gc,1)];
if CourseShip  <= CourseObject (Gc, 1)
CourseRel(Gc, 1) = (CourseObject(Gc,1)- CourseShip);
end;
if CourseShip > CourseObject(Gc, 1)
CourseRel(Gc, 1) = ((2*pi + CourseObject(Gc,1)- CourseShip ));
end;
Object 1( Gc,:)=[ Xbeg(Gc,1)  Ybeg(Gc,1) SpeedShip tab1(Gc,3) CourseRel(Gc,1)];
end;
save CourseObject CourseObject
save BearingRel BearingRel
Object (:,5) = CourseRel(:,1);
save AbsolXY AbsolXY
save CourseRel CourseRel
XYbeg(:,1) = Object1(:,1);
XYbeg(:,2) = Object1(:,2);
save XYbeg XYbeg
Object = Object1(1 :Gc, :);
NumOb = size(Object,1);
Vown = ones(NumbOb,1)*j* SpeedShip;
Vobject = Object(:,4).*exp(j*(pi/2 - CourseRel(:,1)));
Vrel1 = Object(:,4).*exp(j*(pi/2 - CourseObject (:,1)));
Vrel = Vobject - Vown;
Object = [Object(:,1:5) abs(Vrel)];
save Object Object
%save Vrel Vrel
if (hexagon==1)|(circle==1),
[V1] = Domains (zn, Object,Vrel,DeltaT, CourseRel, SpeedShip,XYbeg,X1,X2,AbsolXY,Vrel1);
end;
if parabola==1,
[V1] = Domainsp (zn, Object,Vrel,DeltaT, CourseRel, SpeedShip,XYbeg,X1,X2,AbsolXY,Vrel1);
end;
if ellipse==1,
[V1] = Domainse (zn, Object,Vrel,DeltaT, CourseRel, SpeedShip,XYbeg,X1,X2,AbsolXY,Vrel1);
end
else
load AbsolXY AbsolXY
load CourseObject CourseObject
load Object Object
load XYbeg XYbeg
for Gc=1:Gc,
if CourseShip  <= CourseObject(Gc, 1)
CourseRel(Gc, 1) = (CourseObject(Gc,1)- CourseShip);
end;
if SpeedShip > CourseObject(Gc, 1)
CourseRel(Gc, 1) = ((2*pi + CourseObject(Gc,1)- SpeedShip));
end;
end;
NumbOb = size(Object,1);
Object(:,5) = CourseRel(:,1);
Object(:,3) = SpeedShip;
Vown = ones(NumbOb,1)*j* SpeedShip;
Vobject = Object(:,4).*exp(j*(pi/2 - CourseRel(:,1)));
Vrel1 = Object(:,4).*exp(j*(pi/2 - CourseObject(:,1)));
Vrel = Vobject - Vown;
Object = [Object(:,1:5) abs(Vrel)];
if (hexagon==1)|(circle==1),
[V1] = Domains (zn, Object,Vrel,DeltaT, CourseRel, SpeedShip,XYbeg,X1,X2,AbsolXY,Vrel1);
end;
if parabola==1,
[V1] = Domainsp (zn, Object,Vrel,DeltaT, CourseRel, SpeedShip,XYbeg,X1,X2,AbsolXY,Vrel1);
end;
if ellipse==1,
 [V1] = Domainse (zn, Object,Vrel,DeltaT, CourseRel, SpeedShip,XYbeg,X1,X2,AbsolXY,Vrel1);
end;
end;
```

4 ALGORITHM SIMULATION STUDIES

The algorithm was subjected to simulation tests in MATLAB/Simulink software for the scenario of passing ten ships (Table 1 and Figure 3).

Table 1. Navigation scenario measurement data.

Ship, k	Speed, V_k (kn)	Course, ψ_k (deg)	Distance, D_k (nm)	Bearing, N_k (deg)
0	16.2	0	0	0
1	15.8	261	7.9	45
2	10.5	120	4.2	320
3	14.2	180	7.9	349
4	12.8	351	4.5	62
5	9.3	358	4.2	24
6	4.1	0	8.1	30
7	6.7	180	6.7	336
8	3.9	180	3.8	301
9	11.9	330	7.6	18
10	8.1	0	3.6	10

First, Figures 4 and 5 show a comparison of the safe trajectories in situations of good and restricted visibility at sea for different shapes of ship domains and without domains.

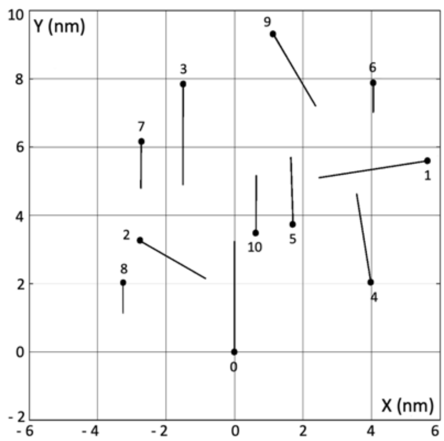

Figure 3. Vessel traffic navigation scenario.

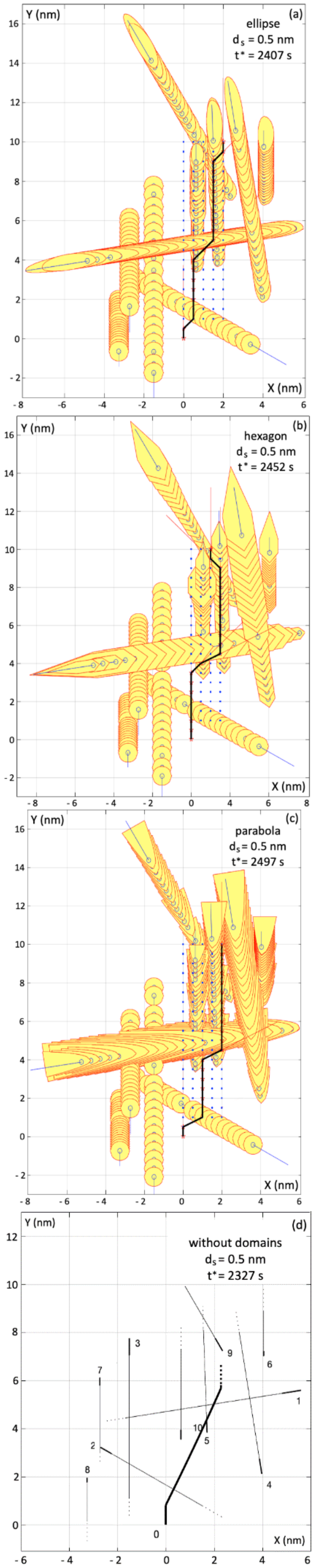

Figure 4. Comparison of safe trajectories in the situation of good visibility at sea for d_s = 0.5 nm, for different shapes of ship domains: (a) – ellipse, (b) – hexagon, (c) - parabola and without domains – (d).

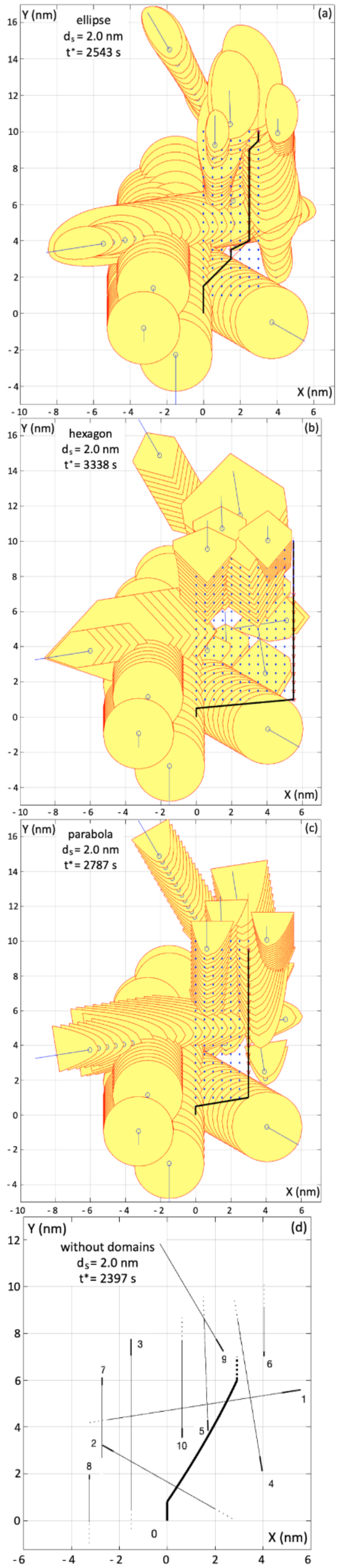

Figure 5. Comparison of safe trajectories in the situation of restricted visibility at sea for d_s = 2.0 nm, for different shapes of ship domains: (a) – ellipse, (b) – hexagon, (c) - parabola and (d) - without domains.

Then, in Figures 6 and 7, a comparison of the safe trajectories with course maneuvering only and when maneuvering the course and speed for different shapes of ship domains and without domains was carried out.

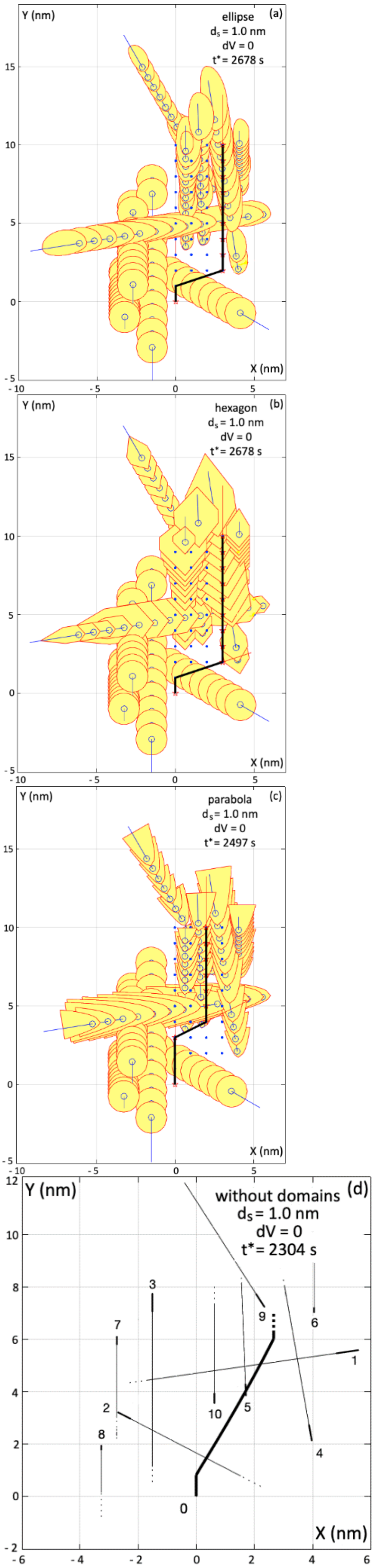

Figure 6. Comparison of safe trajectories in a situation of restricted visibility at sea for d_s = 1.0 nm with course maneuvering only with dV = 0, for different shapes of ship domains: (a) – ellipse, (b) – hexagon, (c) - parabola and (d) - without domains.

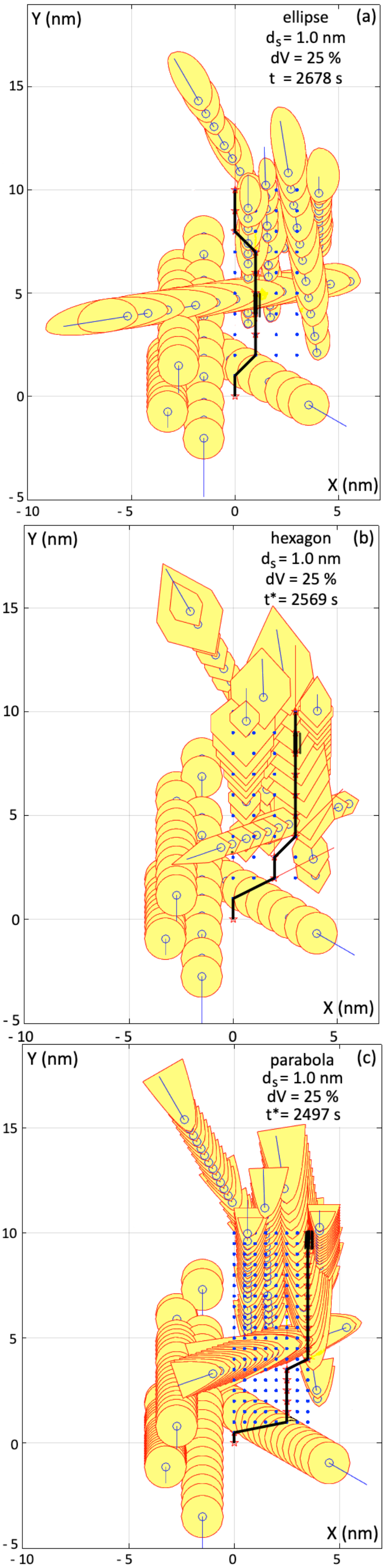

Figure 7. Comparison of safe trajectories in a situation of restricted visibility at sea d_s = 1.0 nm when maneuvering course and speed with dV = 25 %, for different shapes of ship domains: (a) – ellipse, (b) – hexagon, (c) - parabola.

Figure 8 illustrates the two-dimensional optimality of the safe trajectory depending on the discretization of the calculations and the required safe passing distance for domain forms as an ellipse, hexagon and parabola.

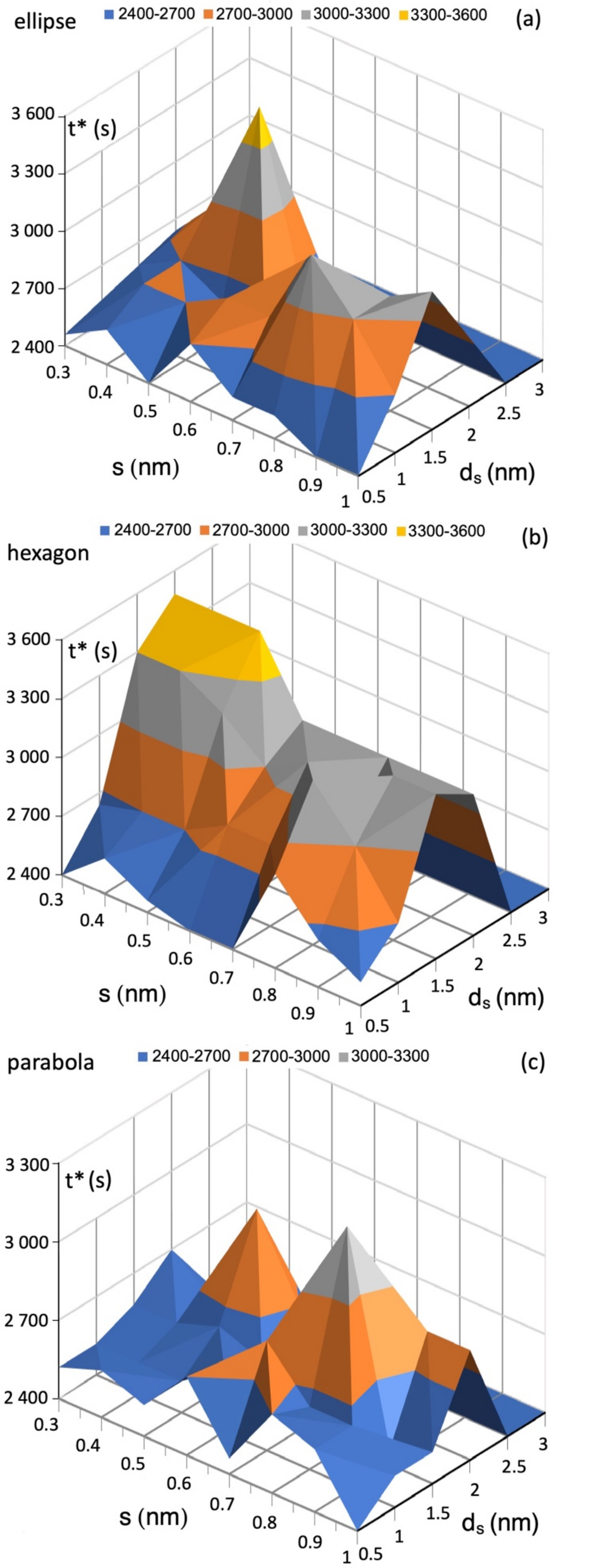

Figure 8. The optimal time t^* for the execution of a safe trajectory depending on discretization s of calculations at different safe distances d_s for domains: (a) – ellipse, (b) – hexagon, (c) - parabola.

Finally, the dependence of the optimal time t^* for the execution of a safe trajectory at the advance time t_a of the overtaking maneuver, which was assigned to the ship's deadweight, is shown in Figure 9. The knowledge from the theory of automatic control was used here, and the ship dynamics are presented in the form of the maneuver advance time.

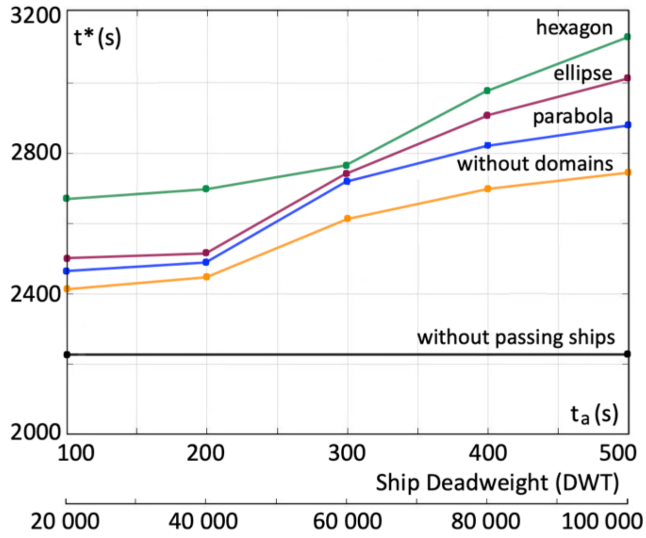

Figure 9. The dependence of the safe trajectory execution time on the ship's dynamics for various shapes of its domains.

The advance time t_a consists of the correction time of the set course change, approximately equal to three time constants of the ship's dynamics as the subject of automatic control.

The sensitivity of the optimal time t^* for the safe trajectory of the ship to changes in the safe passing distance d_s and to the density s of the dynamic programming grid ranges from 38% to 50% and is the lowest for domains in the form of a parabola.

On the other hand, the sensitivity of the optimal time t^* to complete the safe trajectory of the ship, depending on its size DWT, ranges from 10% to 35% and is the highest for hexagonal domains.

The algorithm, in order to take into account Rule 19 COLREGs to keep out of the way of another ship approaching from starboard, assigns port ships circular domains of constant radius equal to the safety distance d_s, and starboard ships elliptical or hexagonal or parabola domains of variable size, generated by an artificial neural network.

5 CONCLUSIONS

This study explored the scientific and research considerations of the optimally neural algorithm for a ship's safe path synthesis, taking into account the various shapes of the passing ship domains, which allowed for the following conclusions to be drawn:

- In terms of visibility at sea, ellipse-shaped domains are the least sensitive in terms of extending the safe trajectory when visibility conditions at sea deteriorate;
- In terms of the ability to maneuver a ship, when maneuvering only the ship's course, the most

effective domains are in the shape of a parabola, and when maneuvering the course and speed, they are domains in the shape of an ellipse;
– In terms of taking into account the size of a ship, the greatest optimality is provided by a parabola-shaped domain;
– In terms of the discretization of calculations, hexagonal domains show the greatest sensitivity.

Future research should consider:
– Testing the sensitivity of determining a safe path to inaccuracy and uncertainty of information regarding the state of the process from various measurement sources;
– Modification of the algorithm taking into account unforeseen maneuvering of other vessels, contrary to COLREGs, during the execution of a safe trajectory;
– Developing a version of the algorithm that takes into account the specificity of ship navigation in confined waters and in open waters.

ACKNOWLEDGMENT

This research was funded by a research project of the Gdynia Maritime University in Poland, no. WE/2023/PZ/02, “Control algorithms synthesis of autonomous objects”.

REFERENCES

[1] „Ship Domain – International Dictionary of Marine”, IALA – AISM International Association of Marine Aids to Navigation and Lighthouse Authorities, 2023, https://www.iala-aism.org/wiki/dictionary/index.php/Ship_Domain.
[2] M. Jurdzinski, “The ship's domain in marine navigation”, Works of the Navigation Faculty of Gdynia Maritime University, no. 32, pp. 9-20, 2017 (in polish), doi: 10.12716/1002.32.01.
[3] E. M. A. Goodvin, “Statistical study of ship domains”, Journal of Navigation, , vol. 28, pp. 328-334, 1975, ISSN: 0373-4633.
[4] P. V. Davis, M. J. Dove and C. T. Stockel, “A computer simulation of marine traffic using domains and areas”, Journal of Navigation, , no. 33, pp. 215-222, 1980, doi: 10.1017/S0373463300035220.
[5] B. A. Colley, R. G. Curtis and C. T. Stockel, “Manoeuvring Times, Domains and Arenas”, Journal of Navigation, vol. 36, pp. 324–328, 1983, doi:10.1017/S0373463300025030.
[6] R. Szlapczynski, P. Krata and J. Szlapczynska, “A Ship Domain-Based Method of Determining Action Distances for Evasive Manoeuvres in Stand-On Situations”, Journal of Advanced Transportation, vol. 2018, doi: 10.11.55/2018/3984962.
[7] M. Starup, “Ship Domain in Open waters- the size and shape of the navigator’s declarative ship domain”, University of South-Eastern Norway, Faculty of Technology, Natural Sciences and Maritime Sciences, Master Thesis, May 2018.
[8] P. F. Chen, P. van Gelder and J. M. Mou, “Integration of Elliptical Ship Domains and Velocity Obstacles for Ship Collision Candidate Detection”, TransNav - the International Journal on Marine Navigation and Safety of Sea Transportation, vol. 13, no. 4, pp. 751-758, 2019, doi:10.12716/1001.13.04.07.
[9] Y. Y. Wang, „An Empirical Model of Ship Domain for Navigation in Restricted Waters”, National University of Singapore, Ph. D. Thesis, Singapore, 22 August 2012, http://scholarbank.nus.edu.sg/handle/10635/36604.
[10] Z. Pietrzykowski and M. Wielgosz, “Effective ship domain – Impact of ship size and speed”, Ocean Engineering, vol. 219, 108423, 2021, doi.10.1016/j.oceaneng.2020.108423.
[11] M. Wielgosz and Z. Pietrzykowski, “The ship domain in navigational safety assessment”, PLoS ONE, vol. 17, no. 4, 2022, e0265681, doi:10.1371/journal.pone.0265681.
[12] K. Marcjan, L. Gucma and D. Kotkowska, „The Collision Risk Management Method for Ships Navigating on Coastal Waters Based on Ship Domain and Near-Miss Concept”, European Research Studies Journal , vol. XXIV, no. 4, pp. 127-146, 2021, doi: 10.35808/ersj/2567.
[13] A. Hörteborn, J. W. Ringsberg, M. Svanberg, and H. Holm, “A Revisit of the Definition of the Ship Domain based on AIS Analysis”, The Journal of Navigation, vol. 72, no. 3, pp. 777-794, 2019, doi:10.1017/S0373463318000978.
[14] E. Tu, G. Zhang, L. Rachmawati, E. Rajabally and G. B. Huang, "Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey From Data to Methodology," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 5, pp. 1559-1582, May 2018, doi: 10.1109/TITS.2017.2724551.
[15] R. Smierzchalski, “Ships domains as a collision risk at sea in the evolutionary trajectory planning”, in Risk Analysis II, WIT Press, pp. 43-52, 2000, doi: 10.2495/RISK000041.
[16] X. Zhu, H. Xu and J. Lin, “Domain and Its Model Based on Neural Networks” Journal of Navigation, vol. 54, no.1, pp. 97-103, 2001, doi:10.1017/S0373463300001247.
[17] “COLREG : Convention on the International Regulations for Preventing Collisions at Sea, 1972”, International Maritime Organization: London, 2002, ISBN: 928-0-151-38X.
[18] J. Lisowski, “Multistage Dynamic Optimization with Different Forms of Neural-State Constraints to Avoid Many Object Collisions Based on Radar Remote Sensing”, Remote Sensing, vol. 12, no. 6, 2020, pp. 1-13. doi: 10.3390/rs12061020.
[19] J. Lisowski, “Optimization Methods in Maritime Transport and Logistics”, Polish Maritime Research, vol. 25, no. 4, pp. 30-38, 2018, doi: 10.2478/pomr-2018-0129.
[20] J. Lisowski, “Synthesis of a Path-Planning Algorithm for Autonomous Robots Moving in a Game Environment during Collision Avoidance”, Electronics, vol. 10, no. 6, 2021, pp. 1-14. doi: 10.3390/electronics10060675.
[21] J. Lisowski, “Game Control Methods Comparison when Avoiding Collisions with Multiple Objects Using Radar Remote Sensing”, Remote Sensing, vol. 12, no. 10, pp. 1-19, 2020, doi: 10.3390/rs12101573.
[22] A. Lew and H. Mauch, “Dynamic Programming – A Computational Tool”, Springer: Berlin, Germany, 2007, ISBN: 978-3-540-37014-7.