

# Optimization decision support system for safe ship control

J. Lisowski

*Department of Ship Automation, Faculty of Electrical Engineering,  
Gdynia Maritime University, Poland*

## Abstract

This paper describes the application of selected methods of optimal control theory for a ship to determine a safe trajectory during the passing of other ships encountered in restricted visibility at sea. The methods of comparison of safe ship control in a collision situation – multi-stage positional non-cooperative and cooperative game, multi-step matrix non-cooperative game, dynamic optimization with neural constrains of the state control process and kinematics optimization – have been introduced here. The synthesis of computer navigator decision supporting algorithms using dynamic programming and dual linear programming methods has been presented. The considerations have been used to illustrate examples of a computer simulation of the algorithms to determine the safe and optimal own ship trajectories in a navigational situation in restricted visibility on the North Sea.

*Keywords: differential games, positional games, matrix games, dual linear programming, dynamic programming, transport engineering, safe ship operations.*

## 1 Optimal and safe ship control in the sea game environment

The number of vessels operating on the seas and oceans has been increasing year by year, as have their speeds. This has necessitated further and more precise investigation into the problem of safe navigation. In order to ensure the safety of navigation the ships are obliged to comply with the regulations of the Convention, namely the International Regulations for Preventing Collisions at Sea (COLREG). However, these Rules refer only to two ships and under the conditions of good visibility [1, 2].



In the case of a restricted visibility, the Rules only specify recommendations of a general nature and are not able to consider all the necessary conditions which determine the passing course. Consequently, the actual process of a ship passing other objects very often occurs in conditions of uncertainty and conflict accompanied by an inadequate co-operation of the ships with regard to the COLREG Rules. It is, therefore, reasonable to investigate, develop and represent the methods of a ship's safe handling using the rules of theory based on dynamic game and computational intelligence [3–5, 8].

In practice, the process of handling a ship as a multidimensional dynamic object depends both on the accuracy of the details concerning the current navigational situation obtained from the ARPA (Automatic Radar Plotting Aids) anti-collision system and on the form of the process model used for determining the rules of the handling synthesis [10, 12].

The ARPA system ensures automatic monitoring of at least 20  $j$  encountered ships, determining their movement parameters (speed  $V_j$ , course  $\psi_j$ ) and elements of approaching to own ship moving with speed  $V$  and course  $\psi$  ( $D_{min}^j = DCPA_j$  - Distance of the Closest Point of Approach,  $T_{min}^j = TCPA_j$  - Time to the Closest Point of Approach) and also assess the risk of collision  $r_j$  (Fig. 1).

While formulating the model of the process it is essential to take into consideration both the kinematics and the dynamics of the ship's movement, the disturbances, the strategy of the encountered objects and the formula assumed as the goal of the ship's handling [14, 15].

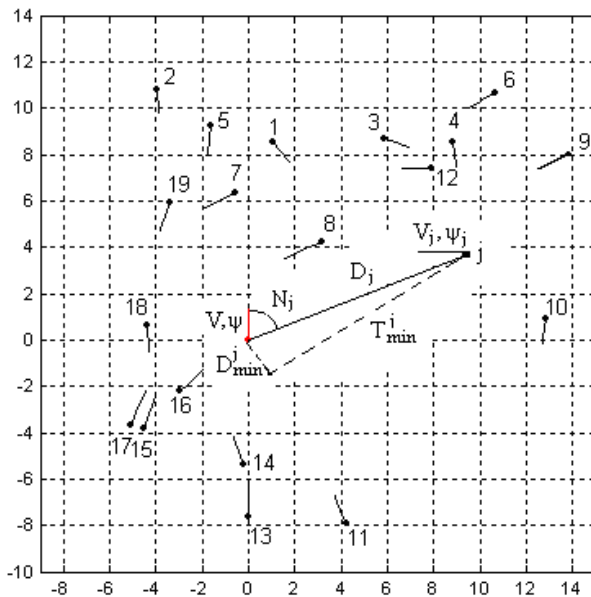


Figure 1: Navigational situation of the own ship passing  $j$  encountered ships.

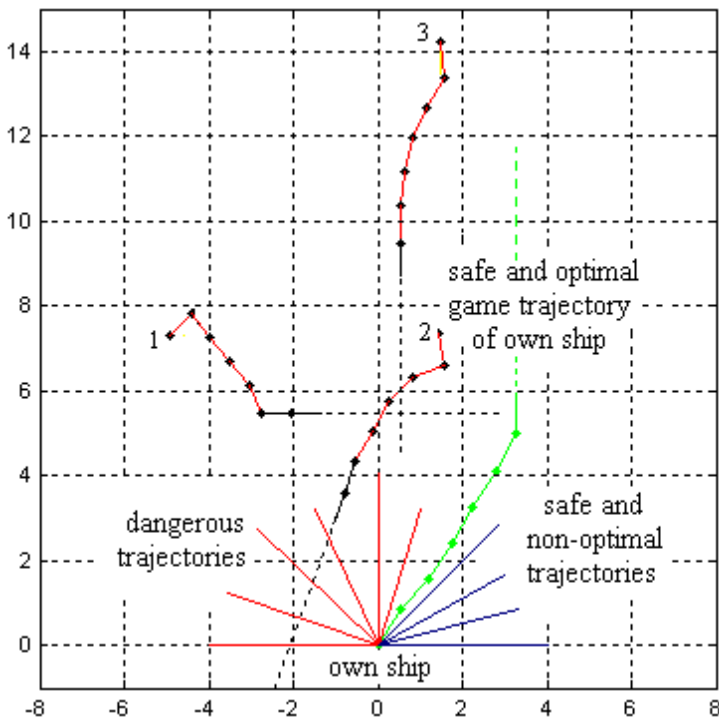


Figure 2: The possible trajectories of the own ship in the situation of passing three encountered ships.

The diversity of selection of possible models directly affects the synthesis of the ship's handling algorithms which are afterwards affected by the ship's handling device, directly linked to the ARPA system and, consequently, determines the effects of safe and optimal control. Figure 2 illustrates possible trajectories - with regard to their safety and effectiveness - using an example of a situation where the own ship passes three other encountered ships. The figure represents a set of compromises of a ship's safe handling measured in terms of a collision risk and time-optimal strategy of the ship's handling.

## 2 The optimization models of safe ship control

The way of controlling a ship - which is a multi-dimensional and non-linear dynamic object - depends on the range and accuracy of information on the prevailing navigational situation and on the adopted model of the process. The variety of the models to be adopted directly influences the synthesis of various algorithms supporting the navigator's work, and then on the effects of a safe control of the own ship's movement (Fig. 3).

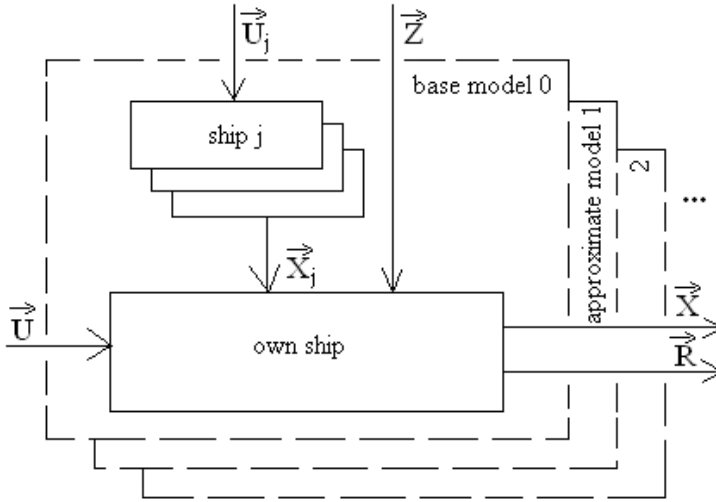


Figure 3: Block diagram of the models for control process:  $\vec{U}$  – vector of the own ship control,  $\vec{U}_j$  – control vector of the  $j$ -th ship,  $\vec{X}_j$  – state vector of the  $j$ -th ship,  $\vec{Z}$  – disturbance vector,  $\vec{X}$  – state vector of the process,  $\vec{R}$  – vector of safety the situation (risk of collision).

### 2.1 Base model of the differential game

The properties of the process are described by the state equation (Isaacs 1965):

$$\dot{x}_i = f_i[x_0^{g_0}, x_j^{g_j}, (u_0^{v_0}, u_j^{v_j}), t] \quad j=1, \dots, m \quad (1)$$

where  $\vec{x}_0^{g_0}(t)$  -  $g_0$  dimensional vector of the process state of the own ship determined in a time span  $t \in [t_0, t_k]$ ;  $\vec{x}_j^{g_j}(t)$  -  $g_j$  dimensional vector of the process state for the  $j$ -th met ship;  $\vec{u}_0^{v_0}(t)$  -  $v_0$  dimensional control vector of the own ship;  $\vec{u}_j^{v_j}(t)$  -  $v_j$  dimensional control vector of the  $j$ -th met ship.

The constraints of the control and the state of the process are connected with the basic condition for the safe passing of the ships at a safe distance  $D_s$  in compliance with COLREG Rules, generally in the following form:

$$g_j(x_j^{g_j}, u_j^{v_j}) \leq 0 \quad (2)$$

Goal function has form of the payments – the integral payment and the final one:

$$I_0^j = \int_{t_0}^{t_k} [x_0^{g_0}(t)]^2 dt + r_j(t_k) + d(t_k) \rightarrow \min \quad (3)$$

The integral payment represents loss of way by the own ship while passing the encountered ships and the final payment determines the final risk of collision  $r_j(t_k)$  relative to the  $j$ -th ship and the final deflection of the own ship  $d(t_k)$  from the reference trajectory [6].

## 2.2 Approximate models

For the practical synthesis of safe control algorithms, various simplified models are formulated, such as for example:

- dual linear programming model of non-cooperative multi-stage positional game,
- dual linear programming model of cooperative multi-stage positional game,
- dual linear programming model of non-cooperative multi-step matrix game,
- dynamic programming model with neural state constraints of safe control process,
- linear programming model of kinematic safe control process [8].

The degree of model simplification is dependent on an optimal control method applied and level of cooperation between ships (Table 1).

## 3 Computer support algorithms

In practice, methods of selecting a manoeuvre assume a form of approximate control algorithms supporting navigator decision in a collision situation. Algorithms are programmed into the memory of a Programmable Logic Controller PLC.

This generates an option within the ARPA anti-collision system or a training simulator (Fig. 4).

Table 1: Algorithms for determining ship strategies.

Process Model	Optimal Control Method	Computer Support Algorithm	Type of Decision
multi-stage positional game	dual linear programming	NPG CPG	game trajectory
multi-step matrix game	dual linear programming	MG	game trajectory
dynamic	dynamic programming	DO	optimal trajectory
kinematic	linear programming	KO	optimal trajectory

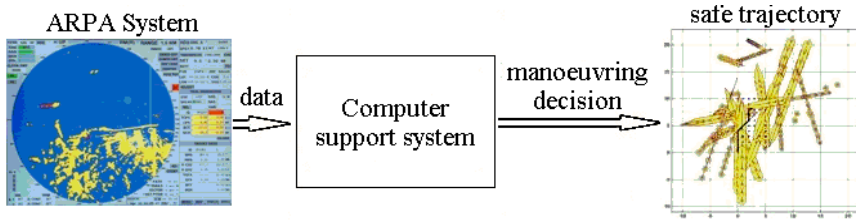


Figure 4: The structure of the computer support system of the navigator decision in a collision situation.

### 3.1 Algorithm of the non-cooperative positional game (NPG)

The optimal control of the own ship  $u_0^*(t)$ , equivalent for the current position  $p(t)$  to the optimal positional steering  $u_0^*(p)$ , is determined from the condition:

$$I^* = \min_{u_0 \in U_0 = \prod_{j=1}^m U_0^j} \left\{ \max_{u_j^* \in U_j} \min_{u_0^* \in U_0^j(u_j)} \int_{t_0}^{t_k} u_0(t) dt \right\} = S_0^*(x_0, L_k) \quad (4)$$

$S_0$  refers to the continuous function of the manoeuvring goal of the own ship, characterising the distance of the ship at the initial moment  $t_0$  to the nearest turning point  $L_k$  on the reference  $p_r(t_k)$  route of the voyage.

The optimal control of the own ship is calculated at each discrete stage of the ship's movement by applying the SIMPLEX method to solve the problem of the dual linear programming, assuming the relationship (4) as the goal function and the control constraints (2).

Using the function of *lp – linear programming* from the Optimisation Toolbox MATLAB, the positional multi-stage game non-cooperative manoeuvring NPG program has been designed for the determination of the own ship safe trajectory in a collision situation.

### 3.2 Algorithm of the cooperative positional game (CPG)

Goal function (4) for cooperative game has the form:

$$I^* = \min_{u_0 \in U_0 = \prod_{j=1}^m U_0^j} \left\{ \min_{u_j^* \in U_j} \min_{u_0^* \in U_0^j(u_j)} \int_{t_0}^{t_k} u_0(t) dt \right\} = S_0^*(x_0, L_k) \quad (5)$$

### 3.3 Algorithm of the matrix game (MG)

The dynamic game is reduced to a multi-step matrix game of a  $j$  number of participants. The matrix game  $R = [r_j(v_j, v_0)]$  includes the values determined previously on the basis of data taken from an anti-collision system ARPA the value a collision risk  $r_j$  with regard to the determined strategies  $v_0$  of the own ship and those  $v_j$  of the  $j$ -th encountered objects.

The matrix risk contains the same number of columns as the number of participant I (own ship) strategies and the number of lines which correspond to a joint number of participant II ( $j$  objects) strategies (Fig. 1).

The value of the risk of the collision  $r_j$  is defined as the reference of the current situation of the approach described by the parameters  $D_{min}^j$  and  $T_{min}^j$ , to the assumed assessment of the situation defined as safe and determined by the safe distance of approach  $D_s$  and the safe time  $T_s$  – which are necessary to execute a manoeuvre avoiding a collision with consideration actual distance  $D_j$  between own ship and encountered  $j$ -th ship:

$$r_j = \left[ w_1 \left( \frac{D_{min}^j}{D_s} \right)^2 + w_2 \left( \frac{T_{min}^j}{T_s} \right)^2 + \left( \frac{D_j}{D_s} \right)^2 \right]^{\frac{1}{2}} \quad (6)$$

where the weight coefficients ( $w_1, w_2$ ) depend on the state visibility at sea, dynamic length and dynamic beam of the ship and the type of water region.

The constraints affecting the choice of strategies ( $v_0, v_j$ ) are a result of COLREG recommendations.

Player I may use  $v_0$  of various pure strategies in a matrix game and player II has  $v_j$  of various pure strategies. As the game, most frequently, does not have saddle point, the state of balance is not guaranteed – there is a lack of pure strategies for both players in the game.

The problem of determining an optimal strategy may be reduced to the task of solving dual linear programming problem. Mixed strategy components express the distribution of probability  $p_j(v_j, v_0)$  of using pure strategies by the players.

As a result of using the following form for the control goal:

$$(I_0^j)^* = \min_{v_0} \max_{v_j} r_j \quad (7)$$

the probability matrix  $P=[p_j(v_j, v_0)]$  of using particular pure strategies may be obtained.

The solution for the control goal is the strategy of the highest probability:

$$(u_0^{v_0})^* = u_0^{v_0} \{ [p_j(v_j, v_0)]_{max} \} \quad (8)$$

Using the function of  $lp$  – *linear programming* from the Optimisation Toolbox MATLAB, the matrix multi-step game manoeuvring MG program has been designed for the determination of the own ship safe trajectory in a collision situation [6].

### 3.4 Algorithm of dynamic optimisation (DO)

The own ship dynamic is represented by the state equations in a discrete form:

$$x_{i,k+1} = x_{i,k} + \Delta x_{i,k}(x_i, u_1, u_2) \quad i = 1, 2, \dots, 7 \quad (9)$$

where:  $x_1=X_0, x_2=Y_0, x_3 = \psi, x_4 = \dot{\psi}_{max}, x_5=V, x_6=\dot{V}, x_7=t, u_1=\alpha_r/\alpha_{max}, u_2=n_r/n_{max}$



The basic criterion for the ship's control is to ensure safe passing of the objects, which is considered in the state constraints:

$$g_j(X_j, Y_j, t) \leq 0 \tag{10}$$

This dependence is determined by the area *ship's domain* of the collision hazard and which assumes the form of a circle, parable, ellipse or hexagon (Baba 2001).

The ships domains may have a permanent or variable shapes generated, for example, by Neural Network Toolbox MATLAB.

Moreover, a criterion of optimisation is taken into consideration in the form of smallest possible way loss for safe passing of the objects, which, at a constant speed of the own ship, leads to the time-optimal control:

$$I(u_1, u_2) = \int_0^{t_k} x_5 dt \cong x_5 \int_0^{t_k} dt \rightarrow \min \tag{11}$$

Determination of the optimal control of the ship in terms of an adopted control quality index may be performed by applying Bellman's principle of optimisation.

The optimal time for the ship to go through k stages is as follows:

$$t_k^* = \min_{u_{1,k-2}, u_{2,k-2}} [t_{k-1}^* + \Delta t_k(x_{1,k}, x_{2,k}, x_{1,k+1}, x_{2,k+1}, x_{5,k})] \tag{12}$$

The optimal time for the ship to go through the k stages is a function of the system state at the end of the k-1 stage and control (u<sub>1,k-2</sub>, u<sub>2,k-2</sub>) at the k-2 stage.

By going from the first stage to the last one the formula (12) determines the Bellman's functional equation for the process of the ship's control by the alteration of the rudder angle and the rotational speed of the screw propeller.

The constraints for the state variables and the control values generate the *NEUROCONSTR* procedure in the dynamic optimal control DO program for the determination of the own ship safe trajectory in a collision situation [7, 11].

### 3.5 Algorithm of kinematics optimisation (KO)

Goal function (4) for kinematics optimization has the form:

$$I^* = \min_{u_0 \in U_0 = \prod_{j=1}^m U_j^0} \int_0^{t_{Lk}} u_0(t) dt = S_0^*(x_0, L_k) \tag{13}$$

## 4 Computer simulation

Computer simulation of NPG, CPG, MG, DO and KO algorithms was carried out in MATLAB/SIMULINK software on an example of the real navigational situation of passing *j=19* encountered ships on the North Sea in restricted visibility when *D<sub>s</sub>=3 nm* (Figs. 5-10).



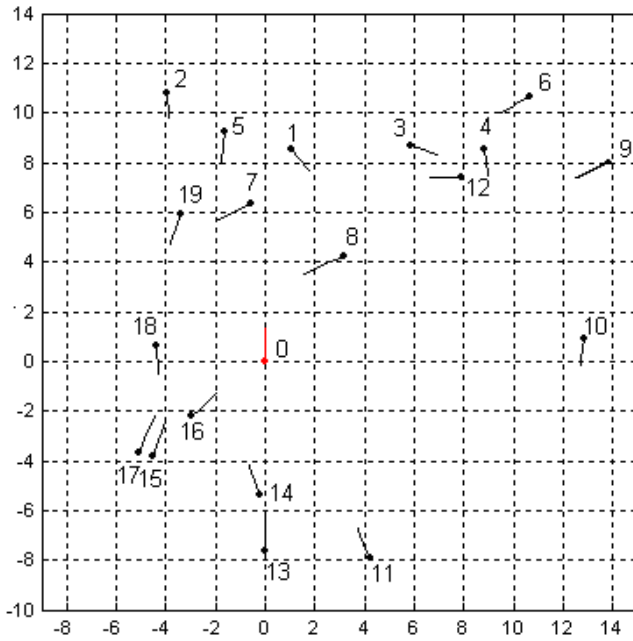


Figure 5: The 6 minute speed vectors of own and 19 encountered ships in situation on the North Sea.

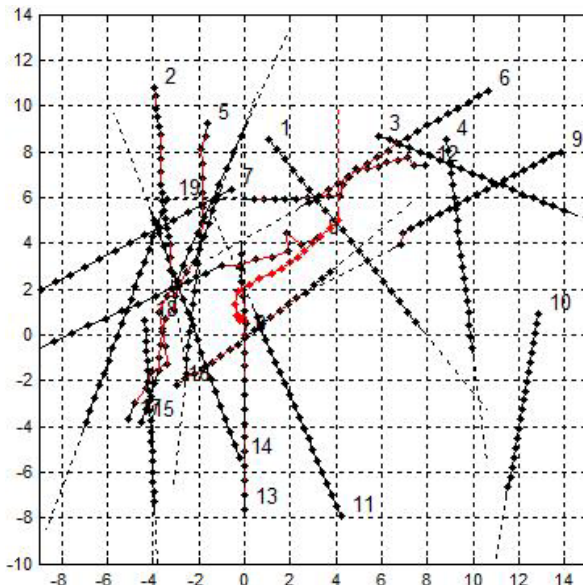


Figure 6: The safe trajectory of own ship for the NPG algorithm in restricted visibility  $D_s=3\text{ nm}$  in the situation of passing  $j=19$  encountered ships,  $r(t_k)=0$ ,  $d(t_k)=4.11\text{ nm}$ .

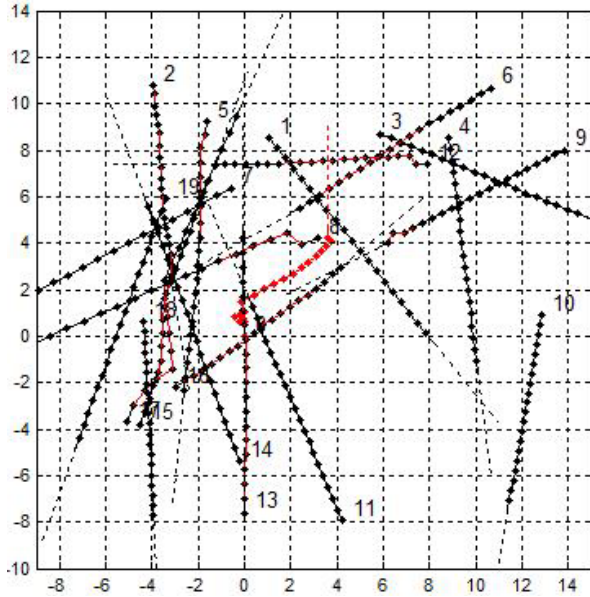


Figure 7: The safe trajectory of own ship for the CPG algorithm in restricted visibility  $D_s=3$  nm in the situation of passing  $j=19$  encountered ships,  $r(t_k)=0$ ,  $d(t_k)=3.63$  nm.

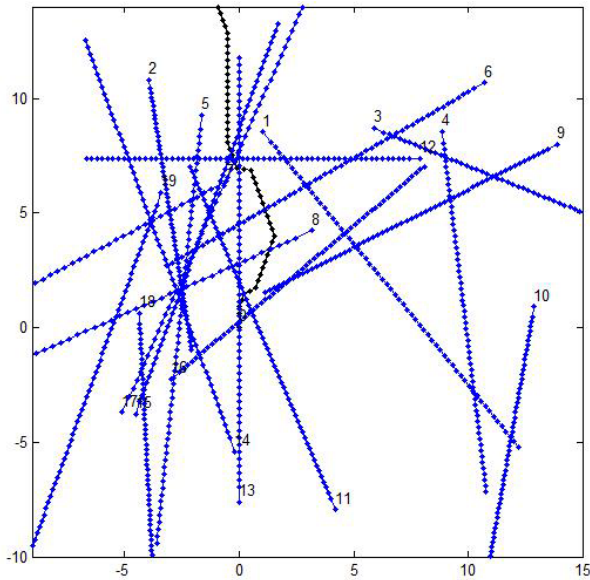


Figure 8: The safe trajectory of own ship for the MG algorithm in restricted visibility  $D_s=3$  nm in the situation of passing  $j=19$  encountered ships,  $r(t_k)=0$ ,  $d(t_k)=4.36$  nm.

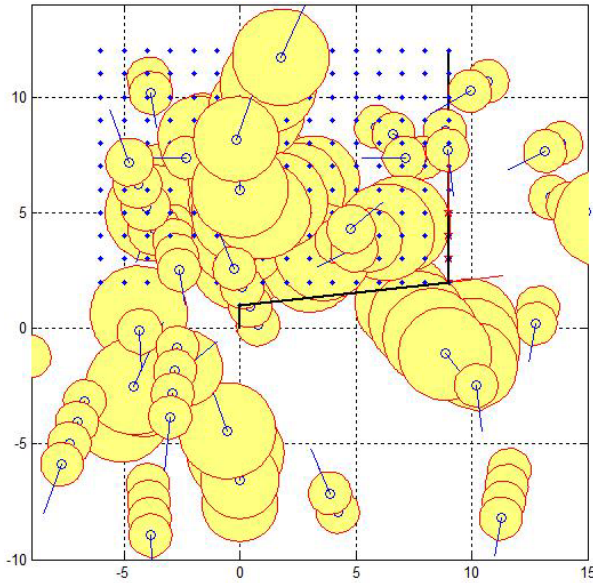


Figure 9: The safe trajectory of own ship for the DO algorithm in restricted visibility  $D_s=3\text{ nm}$  in the situation of passing  $j=19$  encountered ships,  $t_K^* = 1.67\text{ h}$ .

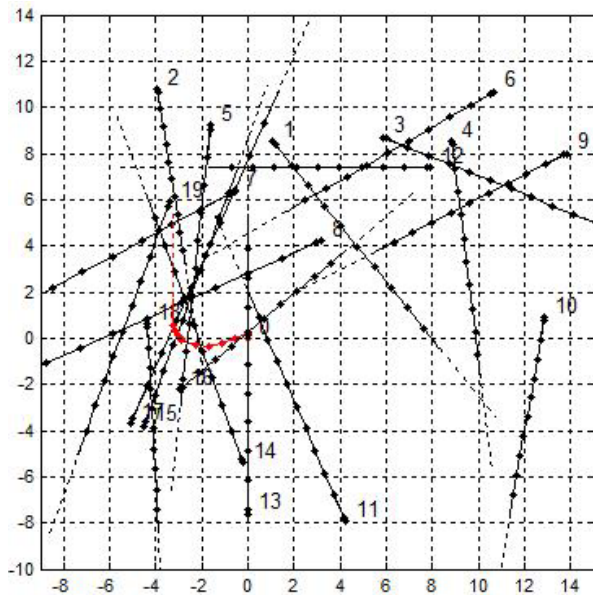


Figure 10: The safe trajectory of own ship for the KO algorithm in restricted visibility  $D_s=3\text{ nm}$  in the situation of passing  $j=19$  encountered ships,  $r(t_k)=0$ ,  $d(t_k)=3.25\text{ nm}$ .

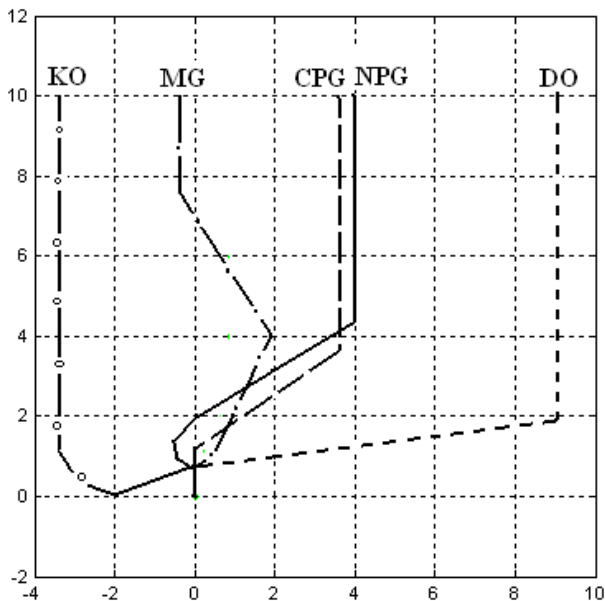


Figure 11: The comparison of own ship safe trajectories in restricted visibility  $D_s=3$  nm in the situation of passing  $j=19$  encountered ships on the North Sea.

Figure 11 introduces the comparison of five computer support algorithms of safe ship control in the sea game environment using different methods of optimization [9, 13, 16].

## 5 Conclusion

In order to ensure safe navigation, ships are obliged to observe legal requirements contained in the COLREG Rules. However, these Rules refer exclusively to two ships under good visibility conditions, in the case of restricted visibility the Rules provide only recommendations of general nature and they are unable to consider all necessary conditions of the real process.

Therefore, the real process of the ships passing exercises occurs under the conditions of indefiniteness and conflict accompanied by an imprecise co-operation among the ships in the light of the legal regulations.

A necessity to consider simultaneously the strategies of the encountered ships and the dynamic properties of the ships as control objects is a good reason for the application of the differential game model - often called the dynamic game.

The control methods considered in this paper are, in a certain sense, formal models for the thinking processes of a navigating officer steering of own ships. Therefore, they may be applied in the construction of both appropriate training simulators at the maritime training centre and also for various options of the basic module of the ARPA anti-collision system.

The application of approximate models of the dynamic game to synthesis of optimal control allows the determination of safe trajectory in situations of passing a greater number of met objects as sequence of course and speed manoeuvres.

The algorithms NPG and CPG determine game and safe trajectory of the ship with relation to of all objects and permits to take into account the degree of their cooperation.

The algorithm MG determines game and safe trajectory of the ship with relation to of the object of most dangerous.

The algorithms DO and KO determine the optimal and safe trajectory of the ship most nearing to the received trajectory from the training simulator ARPA.

The developed algorithms take also into consideration the Rules of the COLREG Rules and the advance time of the manoeuvre, approximating the ship's dynamic properties, and evaluates the final deviation of the real trajectory from the reference value.

These algorithms can be used for computer supporting of navigator safe manoeuvring decision in a collision situations using information from ARPA anti-collision radar system. The use from these algorithms by the own ship is not relative to this whether other ships use the same software. The NPG, CPG and MG game algorithms take into account changes of course and the speeds of other cooperating or not cooperating in accordance with the COLREG Rules ships.

In DO and KO not game algorithms the changes of parameters of the other ships movement are tracked by the anti-collision system ARPA and taken into account in the algorithm of determining the safe trajectory of the own ship.

The detailed analysis of this problem will be introduced in the paper on the next conference.

## References

- [1] Bist, D.S., *Safety and security at sea*, Butterworth Heinemann: Oxford – New Delhi, 2000.
- [2] Cahill, R.A., *Collisions and their causes*, The Nautical Institute: London, 2002.
- [3] Cockcroft, A.N. & Lameijer, N.F., *Collision avoidance rules*, Elsevier: Amsterdam – Tokyo, 2006.
- [4] Engwerda, J.C., *LQ Dynamic Optimization and Differential Games*, John Wiley and Sons: West Sussex, 2005.
- [5] Gluver, H. & Olsen, D., *Ship collision analysis*, Balkema: Rotterdam, 1998.
- [6] Lisowski J., The dynamic game theory methods applied to ship control with minimum risk of collision (Chapter). *Risk Analysis V - Simulation and Hazard Mitigation*, ed. C.A. Brebbia, WIT Press: Southampton - Boston, pp. 293-302, 2006.
- [7] Lisowski, J., Application of dynamic game and neural network in safe ship control. *Polish Journal of Environmental Studies*. **16**, pp. 114-120, 2007.
- [8] Lisowski, J., Optimal and game ship control algorithms avoiding collisions at sea (Chapter). *Risk Analysis VI - Computer Simulation Risk Analysis and*



- Hazard Mitigation*, ed. C.A. Brebbia, WIT Press: Southampton – Boston, pp. 1-10, 2008.
- [9] Lisowski, J. & Mohamed-Seghir, M., Safe ship control methods based on fuzzy set theory. *Polish Journal of Environmental Studies*, **17(3C)**, pp. 55-58, 2008.
- [10] Lisowski J., The comparison of safe control methods in marine navigation in congested waters (Chapter). *Marine navigation and safety of sea transportation*, ed. A. Weintrit, CRC Press: Boca Raton – Leiden, pp. 507-515, 2009.
- [11] Lisowski J., Dynamic programming of safe ship trajectory with neural state constraints. *Polish Journal of Environmental Studies*, **18(4B)**, pp. 126-129, 2009.
- [12] Lisowski J., Computer simulation of optimal and safe ship trajectory in collision situation at sea. *Polish Journal of Environmental Studies*, **18(4B)**, pp. 130-134, 2009.
- [13] Lisowski J., Sensitivity of safe game ship control on base information from ARPA radar (Chapter). *Radar Technology*, ed. V. Kordic, In-Tech: Vienna, pp. 1-26, 2009. (in printing)
- [14] Nisan, N., Roughgarden, T., Tardos, E. & Vazirani, V.V., *Algorithmic Game Theory*, Cambridge University Press: New York, pp. 717-733, 2007.
- [15] Osborne, M.J, *An Introduction to Game Theory*. Oxford University Press: New York, 2004.
- [16] Pachciarek, A., *Comparative analysis of safe ship trajectory determining on computer simulation*. B.Sc. thesis: Gdynia Maritime University, 2007. (in polish)