TRANSNAV
http://www.transnav.eu
the International Journal on Marine Navigation and Safety of Sea Transportation
Volume 8
Number 1
March 2014

DOI: 10.12716/1001.08.01.08

Game Strategies of Ship in the Collision Situations

J. Lisowski
Gdynia Maritime University, Gdynia, Poland

ABSTRACT: The paper introduced the basic model of process of safe ship control in a collision situation using a game model with j objects, which includes non-linear state equations and non-linear, time varying constraints of the state variables as well as the quality game control index in the forms of the game integral payment and the final payment. Approximated model of the process control as the model of multi-step matrix game in the form of dual linear programming problem has been adopted here. The Game Ship Control GSC computer program has been designed in the Matlab/Simulink software in order to determine the own ship's safe trajectory. These considerations have been illustrated with examples of a computer simulation using an GSC program for determining the safe ship's trajectory in real navigational situation. Simulation research were passed for five sets of strategies of the own ship and met ships.

1 INTRODUCTION

The process of the own ship passing other ships at sea very often occurs in conditions of uncertainty and conflict accompanied by an inadequate co-operation of the ships with regard to the International Regulations for Preventing Collisions at Sea (COLREG). 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 games and methods of computational intelligence.

In practice, the process of handling a ship as a control 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. The ARPA system ensures automatic monitoring of at least 20 j-th encountered objects, determining their movement parameters (speed V_j, course ψ_j) and elements of approaching to own ship ($D^j_{\min} = DCPA_j$ – Distance of the Closest Point of Approach, $T^j_{\min} = TCPA_j$ – Time to the Closest Point of Approach) and also assess the risk r_j of collision (Bist 2000, Cahill 2002, Gluver & Olsen 1998).

However, the range of functions of a standard ARPA system ends up with a simulation of a manoeuvre selected by navigator. The problem of selecting such a manoeuvre is very difficult as the process of control is very complex since it is dynamic, non-linear, multi-dimensional and game making in its nature (Figures 1 and 2) (Fang & Luo 2005, Fossen 2011, Lisowski 2013b, Perez 2005).

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 control. 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.

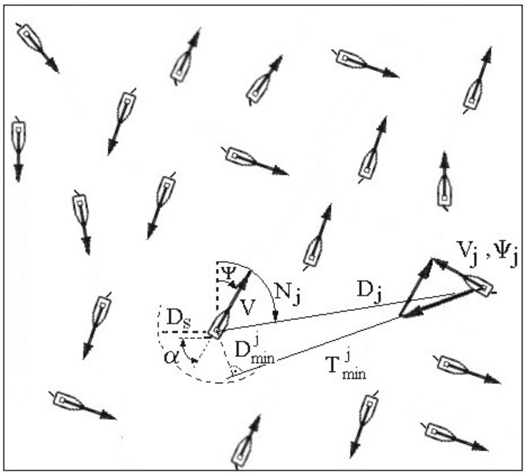


Figure 1. Parameters describing the process of the own ship passing j-th encountered ship.

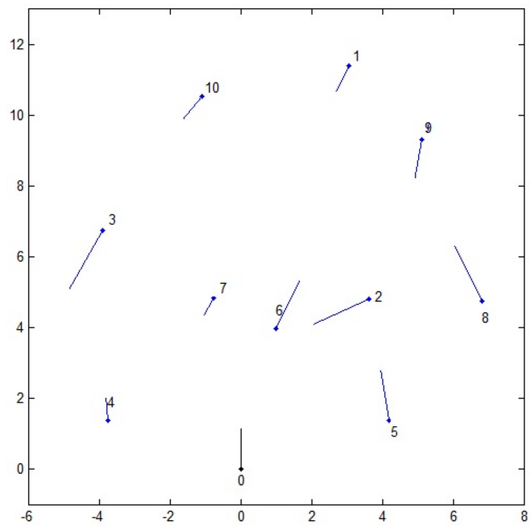


Figure 2. Vectors of own ship and encountered objects.

2 BASIC MODEL OF GAME SHIP CONTROL

The most general description of the own ship's passing the j number of other encountered ships is the model of a differential game of a j number of objects (Figure 3).

General dynamic features of the process are described by a set of state equations in the following form:

$$\dot{x}_i = f_i(x_0^{\vartheta_0},\ x_j^{\vartheta_j},\ u_0^{\nu_0},\ u_j^{\nu_j},\ t) \tag{1}$$

where $\vec{x}_0^{\vartheta_0}(t)$ is ϑ_0 dimensional vector of the process state of the own ship determined in a time span $t \in [t_0, t_k]$, $\vec{x}_j^{\vartheta_j}(t)$ is ϑ_j dimensional vector of the process state for the *j*-th object, $\vec{u}_o^{\nu_0}(t)$ is ν_0 dimensional control vector of the own ship and $\vec{u}_j^{\nu_j}(t)$ is ν_j dimensional control vector of the *j*-th object (Isaacs 1965, Keesman 2011).

The state variable $x_0^{\vartheta_0}$ is represented by the values: course, angular turning speed, speed, drift angle, rotational speed of the screw propeller and controllable pitch propeller - of the own ship and $x_j^{\vartheta_j}$ by the values: distance, bearing, course and speed - of the *j*-th object. While the control value $u_0^{\nu_0}$ is represented by: reference rudder angle, reference rotational speed screw propeller and reference controllable pitch propeller - of the own ship and $u_j^{\nu_j}$ by the values: course and speed - of the *j*-th object (Isil & Koditschek 2001).

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 (Mesterton-Gibbons 2001):

$$g_j[x_j^{\vartheta_j}(t), u_j^{\nu_j}(t)] \le 0 \qquad j = 1, 2, ..., m \tag{2}$$

The constraints (2) as „ship's domains" take a form of a circle, ellipse, hexagon or parable and may be generated, for example, by the neural network (Figure 4) (Cockcroft & Lameijer 2006, Landau et al. 2011, Lisowski 2014a, Millington & Funge 2009, Zio 2009).

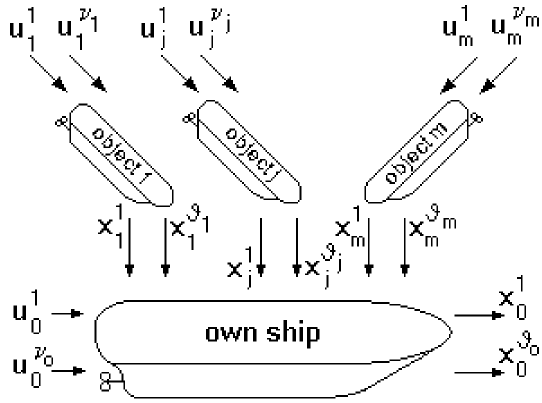


Figure 3. Block diagram of a model ship's differential game including j participants.

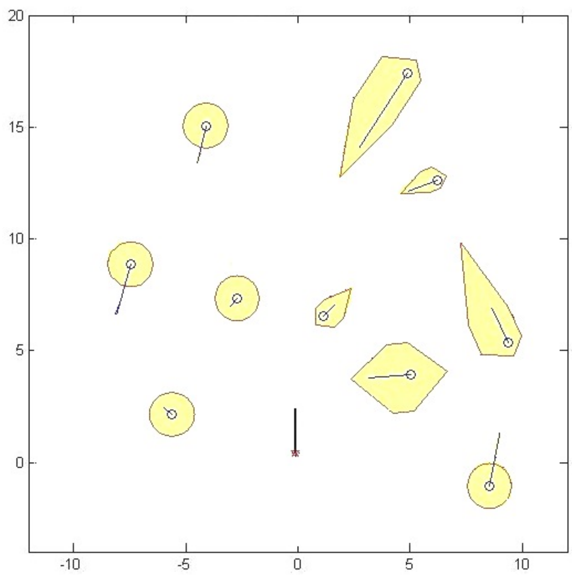


Figure 4. The shapes of the neural ship's domains in the situation of 10 encountered objects.

The synthesis of the decision making pattern of the ship's handling leads to the determination of the optimal strategies of the players who determine the most favourable, under given conditions, conduct of the process. For the class of non-coalition games, often used in the control techniques, the most beneficial conduct of the own ship as a player with *j*-th object is the minimization of her goal function in the form of the payments – the integral payment and the final one:

$$I_0^j = \int_{t_0}^{t_k} [x_0^{\vartheta_0}(t)]^2 dt + r_j(t_k) + d(t_k) \quad \rightarrow \min \tag{3}$$

The integral payment determines the loss of way of the own ship to reach a safe passing of the encountered objects and the final one determines the risk of collision and final game trajectory deflection from reference trajectory (Straffin 2001).

Generally two types of the steering goals are taken into consideration - programmed steering $u_0(t)$ and positional steering $u_0[x_0(t),t]$. The basis for the decision making steering are the decision making patterns of the positional steering processes, the patterns with the feedback arrangement representing the differential games.

The application of reductions in the description of the own ship's dynamics and the dynamic of the *j*-th encountered object and their movement kinematics lead to the approximated matrix game model (Cymbal et al. 2007, Engwerda 2005, Lisowski 2013a).

3 APPROXIMATED MODEL OF GAME SHIP CONTROL

3.1 *State and control variables*

The differential game is reduced to a matrix game of a j number of participants who do not co-operate among them (Figure 5) (Lisowski 2014b).

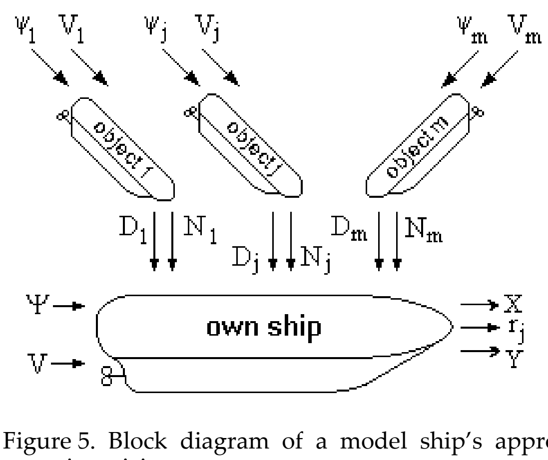


Figure 5. Block diagram of a model ship's approximated game j participants.

The state and control variables are represented by the following values:

$$\begin{aligned} & x_0^1 = X,\ x_0^2 = Y,\ x_j^1 = D_j,\ x_j^2 = N_j, \\ & u_0^1 = \psi,\ u_0^2 = V, \\ & u_j^1 = \psi_j,\ u_j^2 = V_j \\ & j = 1, 2, ..., m \end{aligned} \tag{4}$$

3.2 *Risk of collision*

The matrix game includes the values determined previously on the basis of data taken from an anti-collision system ARPA the value a collision risk rj with regard to the determined strategies of the own ship and those of the j-th encountered objects.

The form of such a game is represented by the risk matrix $\boldsymbol{R}=[r_j(\nu_0, \nu_j)]$ containing the same number of columns as the number of participant I (own ship) strategies. She has; e.g. a constant course and speed, alteration of the course 20° to starboard, to 20° port etc., and contains a number of lines which correspond to a joint number of participant II (j-th object) strategies:

$$R = [r_j(\nu_0, \nu_j)] = \begin{vmatrix} r_{11} & r_{12} & .... & r_{1,\nu_0-1} & r_{1\nu_0} \\ r_{21} & r_{22} & .... & r_{2,\nu_0-1} & r_{2\nu_0} \\ .... & .... & .... & .... & .... \\ r_{\nu_1 1} & r_{\nu_1 2} & .... & r_{\nu_1,\nu_0-1} & r_{\nu_1\nu_0} \\ .... & .... & .... & .... & .... \\ r_{\nu_j 1} & r_{\nu_j 2} & .... & r_{\nu_j,\nu_0-1} & r_{\nu_j\nu_0} \\ .... & .... & .... & .... & .... \\ r_{\nu_m 1} & r_{\nu_m 2} & .... & r_{\nu_m,\nu_0-1} & r_{\nu_m\nu_0} \end{vmatrix} \tag{5}$$

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:

$$r_j = \left[\varepsilon_1 \left(\frac{D_{\min}^j}{D_s} \right)^2 + \varepsilon_2 \left(\frac{T_{\min}^j}{T_s} \right)^2 + \varepsilon_3 \left(\frac{D_j}{D_s} \right)^2 \right]^{-\frac{1}{2}} \tag{6}$$

where the weight coefficients ε_1, ε_2 and ε_3 are depended on the state visibility at sea (good or restricted), kind of water region (open or restricted), speed V of the ship, static L and dynamic L_d length of ship, static B and dynamic B_d beam of ship, and in practice are equal (Figures 6 and 7):

$$1 \le (\varepsilon_1, \varepsilon_2, \varepsilon_3) \le 20 \tag{7}$$

$$L_d = 1.1\,(L + 0.345\,V^{1.6}) \tag{8}$$

$$B_d = 1.1\,(B + 0.767\,LV^{0.4}) \tag{9}$$

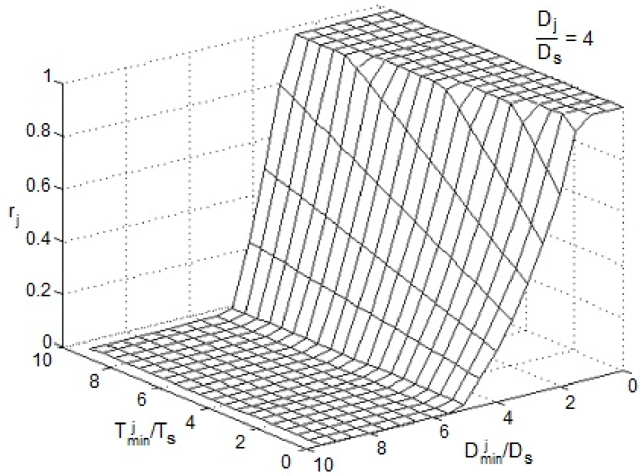


Figure 6. The surface of the collision risk value rj in dependence on relative values distance and time of j-th object approach.

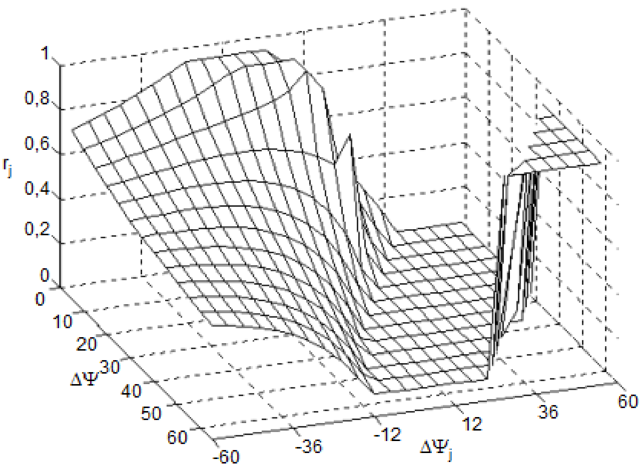


Figure 7a. Dependence of the collision risk on the strategy the own ship and that of the j-th encountered object to approaching from the LB.

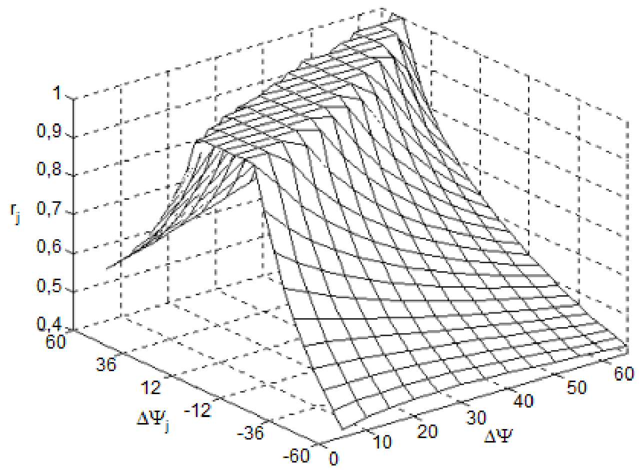


Figure 7b. Dependence of the collision risk on the strategy the own ship and that of the j-th encountered object to approaching from the SB.

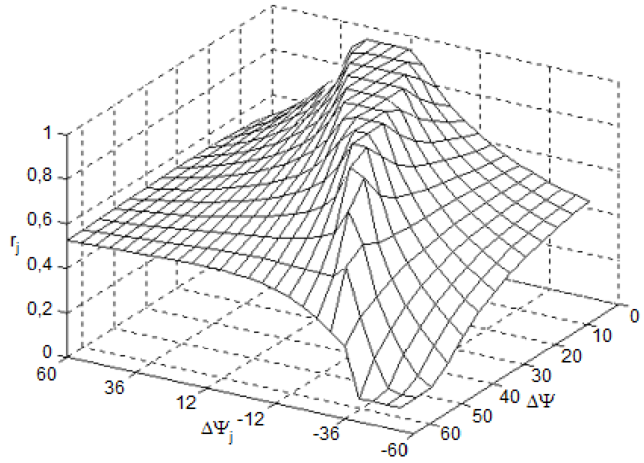


Figure 7c. Dependence of the collision risk on the strategy the own ship and that of the j-th encountered object to approaching from the stern.

The constraints affecting the choice of strategies are a result of the recommendations of the way priority at sea. Player I (own ship) may use ν_0 of various pure strategies in a matrix game and player II (encountered object) has ν_j of various pure strategies (Osborne 2004).

3.3 *Control algorithm*

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. In order to solve this problem dual linear programming may be used (Pantoja 1988, Seghir 2012, Speyer & Jacobson 2010).

In a dual problem player I having ν_0 various strategies to be chosen tries to minimize the risk of collision (Modares 2006):

$$I_0 = \min_{\nu_0} r_j \tag{10}$$

while player II having ν_j strategies to be chosen try to maximize the risk of collision (Mehrotra 1992):

$$I^j = \max_{\nu_j} r_j \tag{11}$$

The problem of determining an optimal strategy may be reduced to the task of solving dual linear programming problem:

$$\left(I_0^j\right)^* = \min_{\nu_0} \max_{\nu_j} r_j \tag{12}$$

Mixed strategy components express the probability distribution $\boldsymbol{P}=[p_j(\nu_0, \nu_j)]$ of using pure strategies by the players:

$$P=[p_j(\nu_0,\nu_j)]=\begin{vmatrix} p_{11} & p_{12} & \dots & p_{1,\nu_0-1} & p_{1\nu_0} \\ p_{21} & p_{22} & \dots & p_{2,\nu_0-1} & p_{2\nu_0} \\ \dots & \dots & \dots & \dots & \dots \\ p_{\nu_1 1} & p_{\nu_1 2} & \dots & p_{\nu_1,\nu_0-1} & p_{\nu_1\nu_0} \\ \dots & \dots & \dots & \dots & \dots \\ p_{\nu_j 1} & p_{\nu_j 2} & \dots & p_{\nu_j,\nu_0-1} & p_{\nu_j\nu_0} \\ \dots & \dots & \dots & \dots & \dots \\ p_{\nu_m 1} & p_{\nu_m 2} & \dots & p_{\nu_m,\nu_0-1} & p_{\nu_m\nu_0} \end{vmatrix} \quad (13)$$

The solution for the steering goal is the strategy of the highest probability and will also be the optimal value approximated to the pure strategy:

$$\left(u_0^{\nu_0}\right)^* = u_o^{\nu_0}\left\{[p_j(\nu_0,\nu_j)]_{\max}\right\} \quad (14)$$

The safe trajectory of the own ship has been treated here as a sequence of changes course and speed.

The values established are as follows: safe passing distances among the ships under given visibility conditions at sea D_s, time delay of manoeuvring and the duration of one stage of the trajectory as one calculation step. At each step the most dangerous object is determined with regard to the value of the collision risk r_j. Consequently, on the basis of the semantic interpretation of the COLREG Regulations the direction of a turn of the own ship is selected to the most dangerous encountered object (Flechter 1987).

The collision matrix risk $\boldsymbol{R}$ is determined for the admissible strategies of the own ship ν_0 and those ν_j for j-th object encountered. By applying dual linear programming in order to solve the matrix game you obtain the optimal values of the own course and that of the j-th object at the smallest deviation from their initial values.

If, at a given step, no solution can be found at a speed of the own ship V, the calculations are repeated at the speed reduced by 25% until the game has been solved. The calculations are repeated step by step until the moment when all elements of the matrix $\boldsymbol{R}$ become equal to zero and the own ship, after having passed the encountered objects, returns to her initial course and speed.

In this manner optimal safe trajectory of the ship is obtained in a collision situation (Fadali & Visioli 2009).

Using the function of *lp – linear programming* from the Optimization Toolbox contained in the Matlab software, the Game Ship Control GSC program has been designed for the determination of the safe ship's trajectory in a collision situation.

4 COMPUTER SIMULATION

4.1 *GSC1 program*

Simulation tests in Matlab/Simulink of the GSC program have been carried out with reference to real situation of passing j=10 encountered ships.

For the first base version GSC1 of the program, the following values for the strategies have been adopted (Figure 8) (Lisowski & Lazarowska 2013, Nisan et al. 2007):

$$\nu_0 = 13 \rightarrow \left|0^o \div 60^o\right| \text{ for each of the } 5^o,$$

$$\nu_j = 25 \rightarrow \left(-60^o \div +60^o\right) \text{ for each of the } 5^o.$$

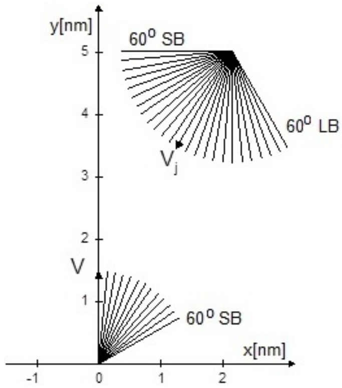


Figure 8. Possible mutual strategies of the own ship and those of the j-th encountered object in program GSC1.

The computer simulation, performed on version of the GSC1 program is presented on Figure 9.

4.2 *GSC2 program*

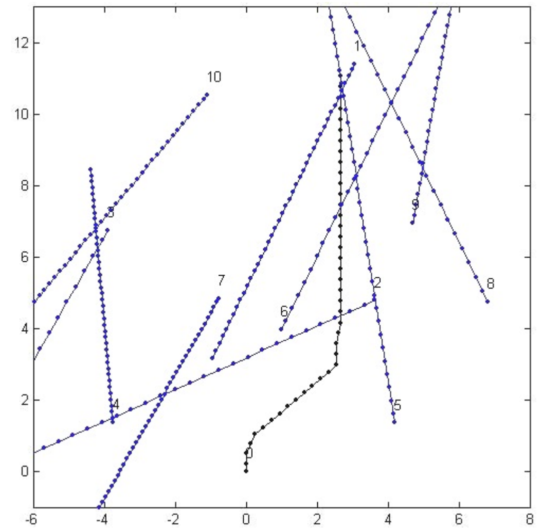
For the second version GSC2 of the program, the number of own ship strategies has been reduced to (Figure 10):

$$\nu_0 = 13 \rightarrow \left|0^o \div 60^o\right| \text{ for each of the } 5^o,$$

$$\nu_j = 3 \rightarrow \left(-30^o, 0^o, +30^o\right).$$

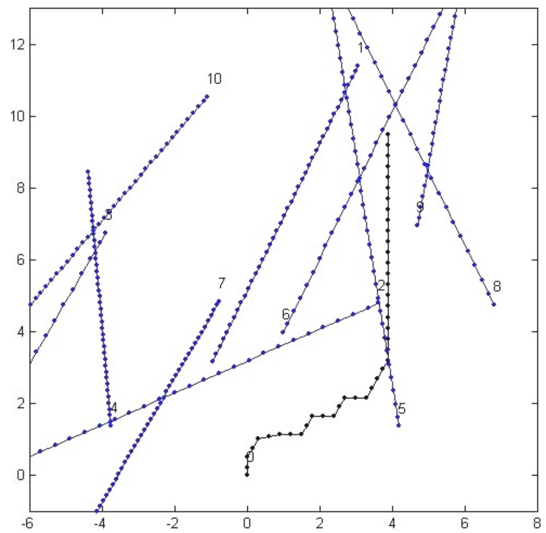
The computer simulation, performed on version of the GSC2 program is presented on Figure 11.

Good visibility: D_s=0.5 nm



r(t_k)=0, d(t_k)=2.89 nm

Restricted visibility: Ds=2.5 nm



r(t_k)=0, d(t_k)=4.73 nm

Figure 9. The ship's game trajectories for the GSC1 algorithm.

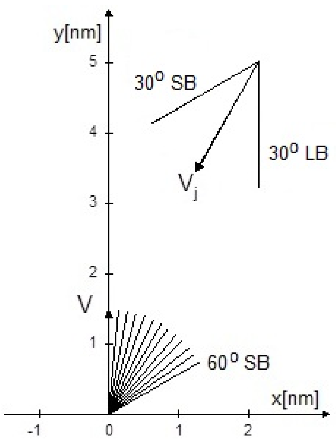
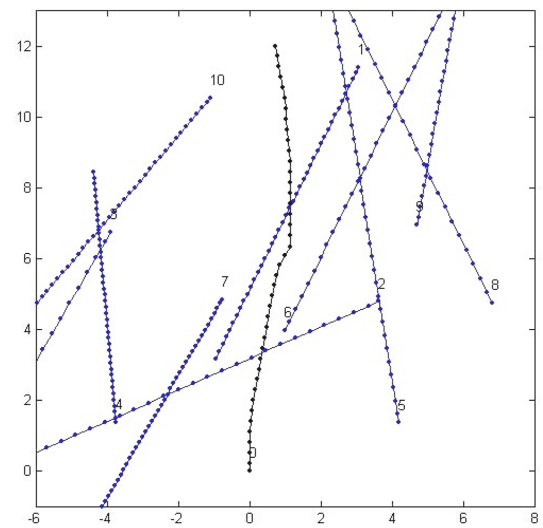


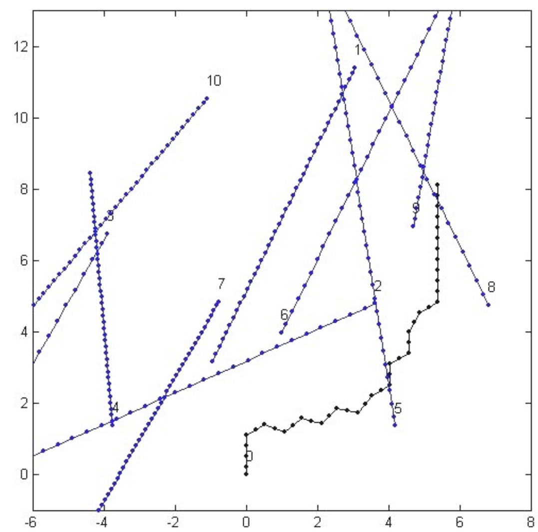
Figure 10. Possible mutual strategies of the own ship and those of the *j*-th encountered object in program GSC2.

Good visibility: D_s=0.5 nm



r(t_k)=0, d(t_k)=2.83 nm

Restricted visibility: Ds=2.5 nm



r(t_k)=0, d(t_k)=6.75 nm

Figure 11. The ship's game trajectories for the GSC2 algorithm.

4.3 *GSC3 program*

For the version GSC3 of the program, the number of own ship strategies has been reduced to (Figure 12):

$$\nu_0 = 4 \rightarrow \left| 0^o, 20^o, 40^o, 60^o \right|,$$

$$\nu_j = 3 \rightarrow \left(-30^o, 0^o, +30^o \right),$$

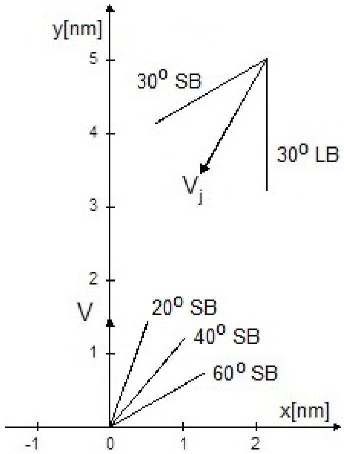


Figure 12. Possible mutual strategies of the own ship and those of the j-th encountered object in program GSC3.

The computer simulation, performed on version of the GSC3 program is presented on Figure 13.

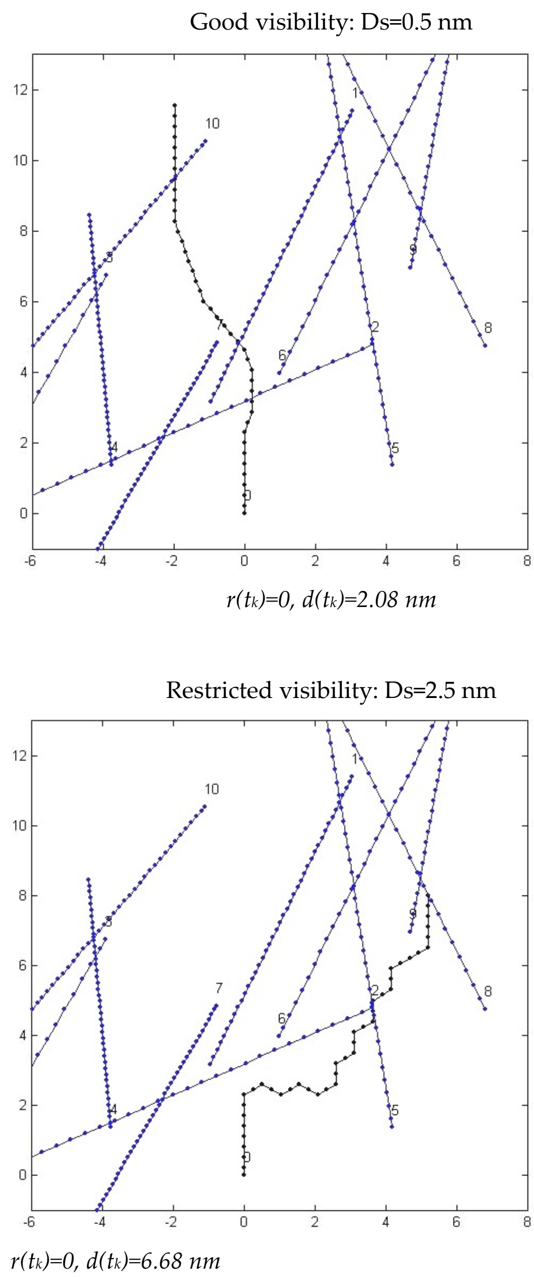


Figure 13. The ship's game trajectories for the GSC3 algorithm.

4.4 *GSC4 program*

For the version GSC4 of the program, the number of own ship strategies has been reduced to (Figure 14):

$$\nu_0 = 3 \rightarrow \left|0^o, 30^o, 60^o\right|,$$

$$\nu_j = 3 \rightarrow \left(-30^o, 0^o, +30^o\right),$$

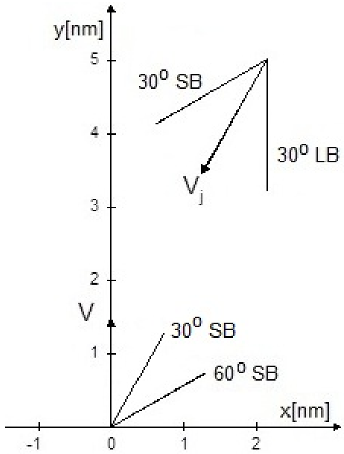


Figure 14. Possible mutual strategies of the own ship and those of the j-th encountered object in program GSC4.

The computer simulation, performed on version of the GSC4 program is presented on Figure 15.

4.5 *GSC5 program*

For the version GSC5, the number of the own ship strategies has been reduced to (Figure 16):

$$\nu_0 = 2 \rightarrow \left|0^o, 60^o\right|,$$

$$\nu_j = 3 \rightarrow \left(-30^o, 0^o, +30^o\right).$$

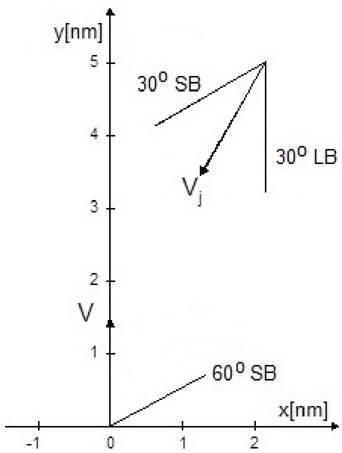


Figure 16. Possible mutual strategies of the own ship and those of the j-th encountered object in program GSC5.

The computer simulation, performed on version of the GSC5 program is presented on Figure 17.

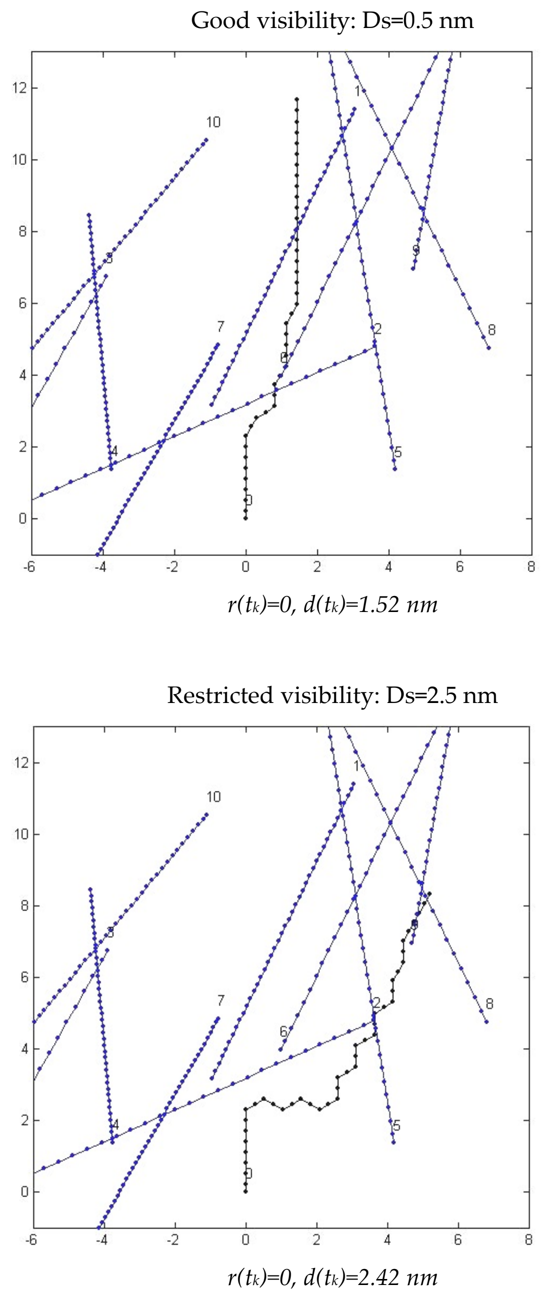


Figure 15. The ship's game trajectories for the GSC4 algorithm.

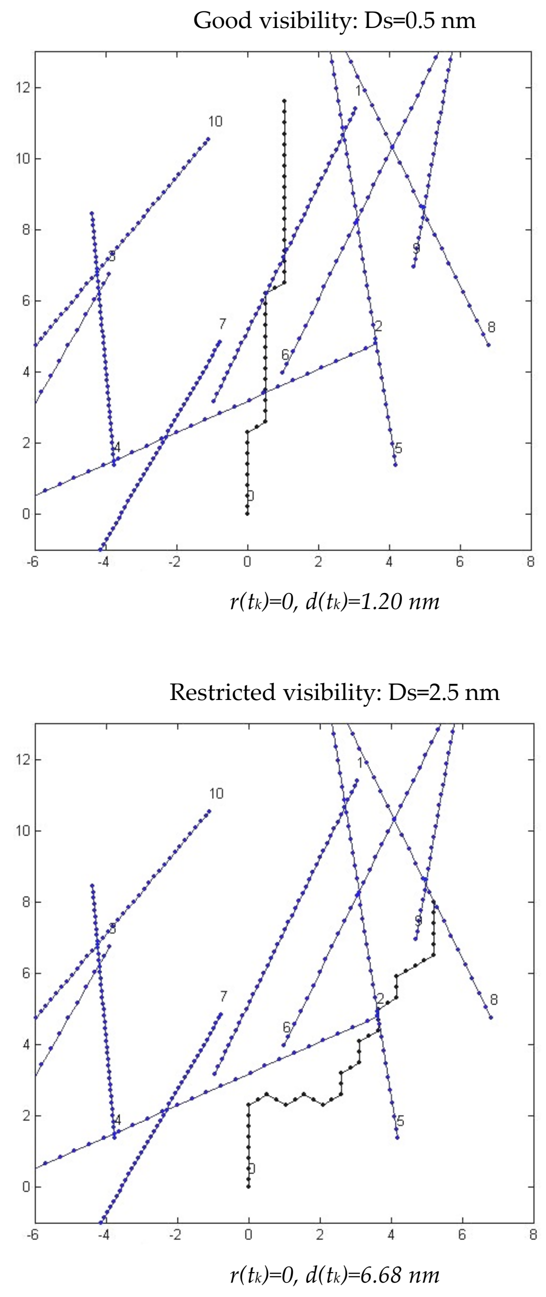


Figure 17. The ship's game trajectories for the GSC5 algorithm.

5 CONCLUSIONS

Analysis of the computer simulation studies of GSC program for different amounts of possible strategies of own ship and met objects allows to draw the following conclusions:

- The synthesis of an optimal on-line control on the base of model of a multi-step matrix game makes it possible to determine the safe game trajectory of the own ship in situations when she passes a greater *j* number of the encountered objects;
- The trajectory has been described as a certain sequence of manoeuvres with the course and speed;
- The computer program designed in the Matlab also takes into consideration the following: regulations of the Convention on the International Regulations for Preventing Collisions at Sea, advance time for a manoeuvre calculated with regard to the ship’s dynamic features and the assessment of the final deflection between the real trajectory and its assumed values;
- The essential influence to form of safe and optimal trajectory and value of deflection between game and reference trajectories has the number of admissible strategies of own ship and encountered objects;
- It results from the performed simulation testing this algorithm is able to determine the correct game trajectory when the ship is not in a situation when she approaches too large number of the observed objects or the said objects are found at long distances among them;

– In the case of the high traffic congestion the program is not able to determine the safe game manoeuvre. This sometimes results in the backing of the own object which is continued until the time when a hazardous situation improves.

REFERENCES

Bist, D.S. 2000. *Safety and security at sea*. Oxford-New Delhi: Butter Heinemann.

Cahill, R.A. 2002. *Collisions and thair causes*. London: The Nautical Institute.

Cockcroft, A.N. & Lameijer, J.N.F. 2006. *The collision avoidance rules*. Amsterdam-Tokyo: Elsevier.

Cymbal, N. N., Burmaka, I.A. & Tupikov, I.I. 2007. *Elastic strategies of the passing ships*. Odessa: KP OGT, (in Russian).

Engwerda, J.C. 2005. *LQ dynamic optimization and differential games*. West Sussex: John Wiley & Sons.

Fadali, M.S. & Visioli, A. 2009. *Digital control engineering*. Amsterdam-Tokyo: Elsevier.

Fang, M.C. & Luo, J.H. 2005. The nonlinear hydrodynamic model for simulating a ship steering in waves with autopilot system. *Ocean Engineering* 11-12(32):1486-1502.

Fletcher, R. 1987. *Practical methods of optimization*. New York: John Wiley and Sons.

Fossen, T.I. 2011. *Marine craft hydrodynamics and motion control*. Trondheim: Wiley.

Gluver, H. & Olsen, D. 1998. *Ship collision analysis*. Rotterdam-Brookfield: A.A. Balkema.

Isaacs, R. 1965. *Differential games*. New York: John Wiley & Sons.

Isil Bozma, H. & Koditschek, D.E. 2001. Assembly as a non-cooperative game of its pieces: Analysis of ID sphere assemblies. *Robotica* 19:93-108.

Keesman, K.J. 2011. *System identification*. London-NewYork: Springer.

Landau, I.D., Lozano, R., M'Saad, M. & Karimi, A. 2011. *Adapive control*. London-New York: Springer.

Lisowski, J. 2013. The sensitivity of computer support game algorithms of a safe ship control. *International Journal of Applied Mathematics and Computer Science* 23(2): 439-446.

Lisowski, J. 2013. Optimal and game control algorithms of ship in collision situations at sea. *Control and Cybernetics* (42)4: 1-20.

Lisowski, J. & Lazarowska, A. 2013. The radar data transmission to computer support system of ship safety. *Solid State Phenomena* 196: 95-101.

Lisowski, J. 2014. Optimization-supported decision-making in the marine game environment. *Solid State Phenomena* 210: 215-222.

Lisowski, J. 2014. The characteristics of the matrix game sensitivity of the safe passing of ships at sea. *Solid State Phenomena* 210: 258-264.

Mehrotra, S. 1992. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization* 4(2):575-601.

Mesterton-Gibbons, M. 2001. *An introduction to game theoretic modeling*. Providence: American Mathematical Society.

Millington, I. & Funge, J. 2009. *Artificial intelligence for games*. Amsterdam-Tokyo: Elsevier.

Modarres, M. 2006. *Risk analysis in engineering*. Boca Raton: Taylor & Francis Group.

Nisan, N., Roughgarden, T., Tardos, E. & Vazirani, V.V. 2007. *Algorithmic game theory*. New York: Cambridge University Press.

Osborne, M.J. 2004. *An introduction to game theory*. New York: Oxford University Press.

Pantoja, J.F.A. 1988. Differential dynamic programming and Newton's method. *International Journal of Control* 5(47):1539-1553.

Perez, T. 2005. *Ship motion control*. London: Springer.

Straffin, P.D. 2001. *Game theory and strategy*. Warszawa: Scholar (in polish).

Seghir, M.M. 2012. Safe ship's control in a fuzzy environment using genetic algorithm. *Solid State Phenomena* 180: 70-75.

Speyer, J.L. & Jacobson, D.H. 2010. *Primer on optimal control theory*. Philadelphia: SIAM.