TRANSNAV International Journal on Marine Navigation and Safety of Sea Transportation
Volume 3 Number 2 June 2009

The Comparison of Safe Control Methods in Marine Navigation in Congested Waters

J. Lisowski
Gdynia Maritime University, Gdynia, Poland

ABSTRACT: The paper introduces comparison of five methods of safe ship control in collision situation: multi-stage positional non-cooperative and cooperative game, multi-step matrix game, dynamic and kinematics optimisation with neural constrains of state control process. The synthesis of computer navigator decision supporting algorithms with using dual linear programming and dynamic programming methods has been presented. The considerations have been illustrated an examples of a computer simulation the algorithms to determine the safe own ship's trajectory in situation of passing a many of the ships encountered at sea.

1 MATHEMATICAL MODELS OF SAFE SHIP CONTROL PROCESS

1.1 *Base model*

As the process of steering the ship in collision situations, when a greater number of objects is encountered, often occurs under the conditions of indefiniteness and conflict, accompanied by an inaccurate co-operation of the objects within the context of COLREG Regulations then the most adequate model of the process which has been adopted is a model of a dynamic game, in general of *j* tracked ships as objects of steering (Cahill 2002, Lisowski 2004b, 2005d, 2007b, Sandom 2004).
The diversity of selection of possible models directly affects the synthesis of the ship's handling algorithms which are afterwards effected by the ship's handling device directly linked to the ARPA system and determines the effects of the safe and optimal control. The properties of the process are described by the state equation (Isaacs 1965):

$$\dot{x}_i = f_i[(x_0^{\vartheta_0},\ x_j^{\vartheta_j}), (u_0^{v_0},\ u_j^{v_j}), t]\ j=1,\dots,m \tag{1}$$

where $\vec{x}_0^{\vartheta_0}(t)$ - ϑ_0 dimensional vector of the process state of the own ship determined in a time span $t \in [t_0, t_k]$; $\vec{x}_j^{\vartheta_j}(t)$ - ϑ_j dimensional vector of the process state for the *j-th* met ship; $\vec{u}_o^{v_0}(t)$ - 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 (Engwerda 2005):

$$g_j(x_j^{\vartheta_j},\ u_j^{v_j}) \le 0 \tag{2}$$

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

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

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 (Lisowski 2000a, 2002, 2005a,c, 2008b, Nisan 2007, Nowak 2005, Osborna 2004).

1.2 *Approximate models*

Having regard to a high complexity of the base model in the form of a model of a dynamic game for the practical synthesis of safe steering algorithms various simplified models are formulated, such as for example:

- multi-stage positional game
  - non-cooperative game
  - cooperative game
- multi-step matrix game
- dynamic model with neural constraints

- kinematic model
  - classical model
  - fuzzy model
- static model
- speed triangle model.

The degree of simplification is dependent on a control method applied (Lavalle 2006, Lisowski 2005b, 2006b, 2007a,c, 2008d, Straffin 2001).

2 ALGORITHMS OF SAFE SHIP CONTROL

Each particular approximated model of process may be assigned respective methods of safe control of ship (Table 1).

Table 1. Algorithms of determining ship strategies.

Process models	Control methods	Computer supporting algorithms	Type of decision
Multi-stage positional game	Dual linear programming	NPG CPG	Positional game trajectory
Mult-step matrix game	Dual linear programming	MG	Risk game trajectory
Dynamic	Dynamic programming, Artificial neural network	DO	Dynamic optimal trajectory
Kinematic	Linear programming Fuzzy Control	KO	Kinematic optimal trajectory
Static	Linear programming Fuzzy control	OM	Optimal manoeuvre

In practice, methods of selecting a manoeuvre assume a form of appropriate steering 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. 1).

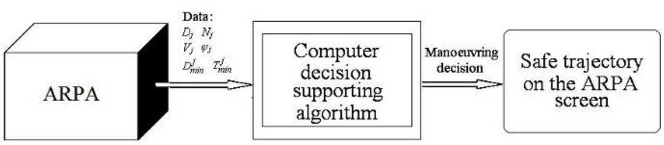

Figure 1. The system structure of computer support of navigator decision in collision situation.

2.1 *Algorithm of non-cooperative positional game NPG*

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

- for the measured position $p(t_k)$ of the steering status at the moment t_k sets of the acceptable strategies $U_j^0[p(t_k)]$ are determined for the encountered objects in relation to the own ship, and the output sets $U_0^{jw}[p(t_k)]$ of the acceptable strategies of the own ship in relation to each one of the encountered objects,
- a pair of vectors u_j^m and u_0^j, is determined in relation to each *j-th* object and then the optimal positional strategy of the own ship $u_0^*(p)$ from the condition:

$$I^* = \min_{u_0 \in U_0 = \bigcap_{j=1}^{m} U_0^j} \left\{ \max_{u_j^m \in U_j} \min_{u_0^j \in U_0^j(u_j)} S_0[x_0(t_k), L_k] \right\} = S_0^*(x_0, L_k) \tag{4}$$

$$S_0[x_0(t), L_k] = \int_{t_0}^{t_{Lk}} u_0(t)\, dt \tag{5}$$

where 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 steering 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 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 (Lebkowski 2001, Lisowski 2001a, 2008b, Segal 1998).

2.2 *Algorithm of cooperative positional game CPG*

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

$$I^* = \min_{u_0 \in U_0 = \bigcap_{j=1}^{m} U_0^j} \left\{ \min_{u_j^m \in U_j} \min_{u_0^j \in U_0^j(u_j)} S_0[x_0(t_k), L_k] \right\} = S_0^*(x_0, L_k) \tag{6}$$

2.3 *Algorithm of matrix game MG*

The dynamic game is reduced to a multi-step matrix game of a *j* number of participants (Lisowski 2001b, 2004a, 2006a, Radzik 2000). The matrix game $\boldsymbol{R} = [r_j(\nu_j, \nu_0)]$ 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 ν_0 of the own ship and those ν_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. 2).

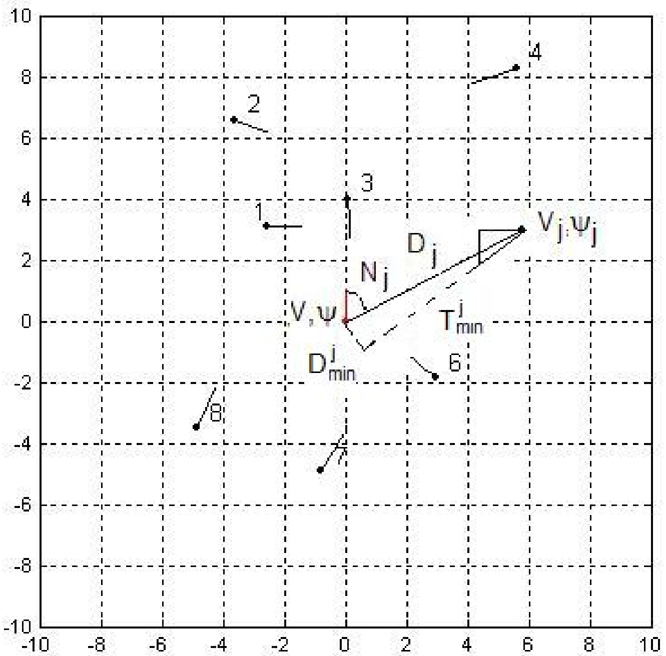

Figure 2. Navigational situation representing the passing of the own ship with the *j-th* object.

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^j_{min} and T^j_{min}, 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:

$$r_j = \left[w_1 \left(\frac{D^j_{min}}{D_s} \right)^2 + w_2 \left(\frac{T^j_{min}}{T_s} \right)^2 + \left(\frac{D_j}{D_s} \right)^2 \right]^{-\frac{1}{2}} \quad (7)$$

where the weight coefficients (w_1, w_2) are depended on the state visibility at sea, dynamic length and dynamic beam of the ship, kind of water region.

The constraints affecting the choice of strategies (ν_0, ν_j) are a result of COLREG recommendations.

Player I may use ν_0 of various pure strategies in a matrix game and player II has ν_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 $p_j(\nu_j, \nu_0)$ of using pure strategies by the players. As a result of using the following form for the steering criterion:

$$\left(I^j_0 \right)^* = \min_{\nu_0} \max_{\nu_j} r_j \quad (8)$$

the probability matrix $\boldsymbol{P} = [p_j(\nu_j, \nu_0)]$ of using particular pure strategies may be obtained.

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

$$\left(u^{\nu_0}_0 \right)^* = u^{\nu_0}_o \left\{ [p_j(\nu_j, \nu_0)]_{max} \right\} \quad (9)$$

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 (Cichuta 2000).

2.4 *Algorithm of dynamic optimisation DO*

The description of the own ship dynamic allows for the following representation of the state equations in a discrete form:

$$x_{i,k+1} = x_{i,k} + \Delta x_{i,k}(x_i, u_1, u_2) \qquad i = 1, 2, ..., 7 \quad (10)$$

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:

$$g_j(X_j, Y_j, t) \le 0 \quad (11)$$

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, Lisowski 2000b).

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:

$$I(u_1,u_2) = \int_0^{t_k} x_5\,dt \cong x_5 \int_0^{t_k} dt \rightarrow min \qquad (12)$$

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:

$$t_k^* = \min_{u_{1,k-2},u_{2,k-2}} [t_{k-1}^* + \Delta t_k(x_{1,k}, x_{2,k}, x_{1,k+1}, x_{2,k+1}, x_{5,k})] \qquad (13)$$

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 $(u_{1,k-2}, u_{2,k-2})$ at the *k-2* stage (Levine 1996).

By going from the first stage to the last one the formula (13) 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 (Nise 2008).

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 (Skogestad 2005).

2.5 *Algorithm of kinematics optimisation KO*

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

$$I^* = \min_{u_0 \in U_0 = \bigcap_{j=1}^{m} U_0^j} \{S_0[x_0(t_k), L_k]\} = S_0^*(x_0, L_k) \qquad (14)$$

3 COMPUTER SIMULATION

Computer simulation of NPG, CPG, MG, DO and KO algorithms was carried out in Matlab/Simulink software on an examples of a real navigational situations at sea of passing *j* encountered objects (Pachciarek 2007, Lisowski 2008c).

3.1 *Situation for j=4 encountered ships*

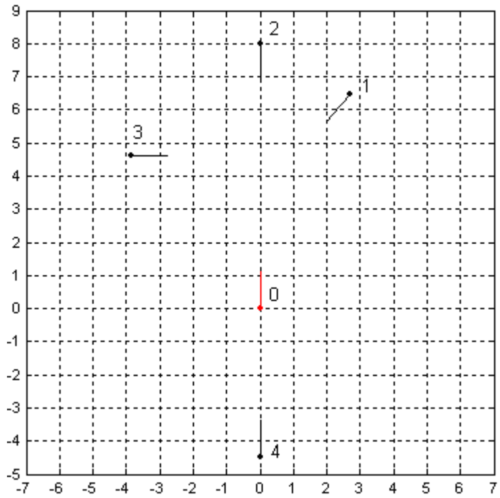

Figure 3. The 6 minute speed vectors of own and 4 encountered ships in situation in Kattegat Strait

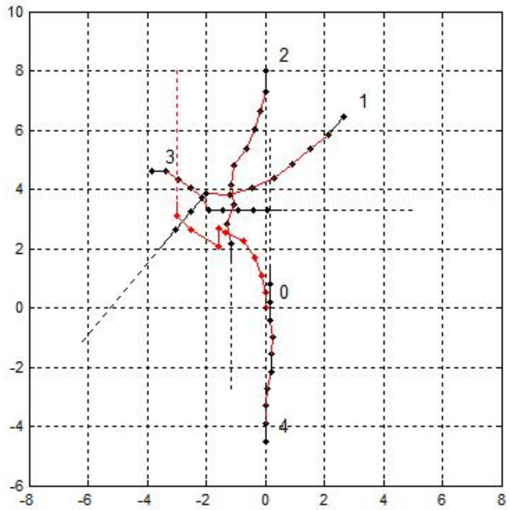

Figure 4. The safe trajectory of own ship for NPG algorithm in good visibility D_s=1 nm in situation of passing *j=4* encountered ships, $r(t_k)=0$, $d(t_k)=2.99$ *nm*.

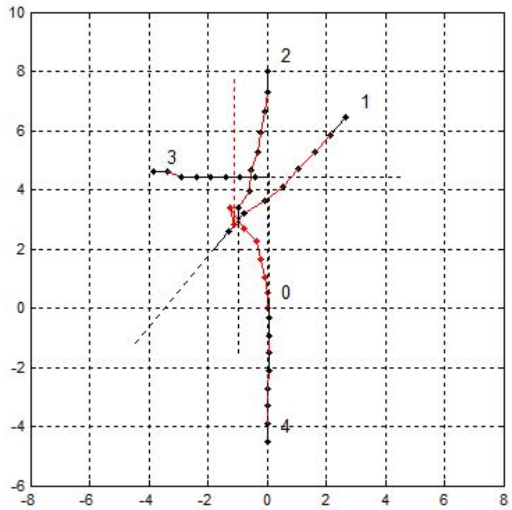

Figure 5. The safe trajectory of own ship for CPG algorithm in good visibility D_s=1 nm in situation of passing *j=4* encountered ships, $r(t_k)=0$, $d(t_k)=1.10$ *nm*.

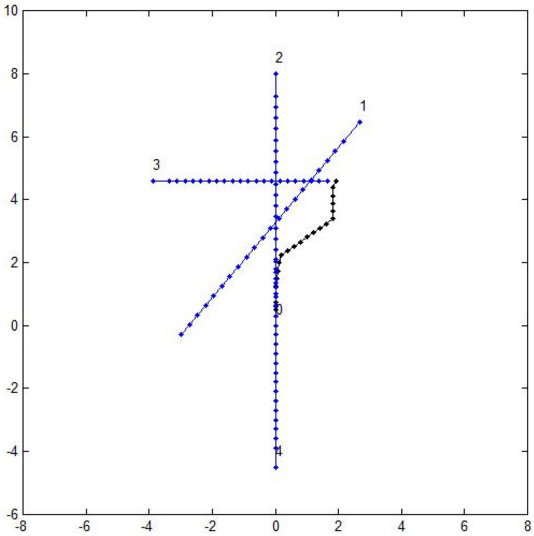

Figure 6. The safe trajectory of own ship for MG algorithm in good visibility $D_s=1$ *nm* in situation of passing $j=4$ encountered ships, $r(t_k)=0$, $d(t_k)=0.83$ *nm*.

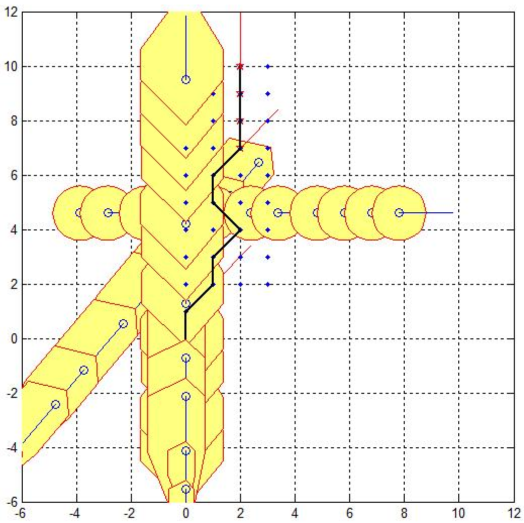

Figure 7. The safe trajectory of own ship for DO algorithm in good visibility $D_s=1$ *nm* in situation of passing $j=4$ encountered ships, $t_K^* = 1.16\ h$.

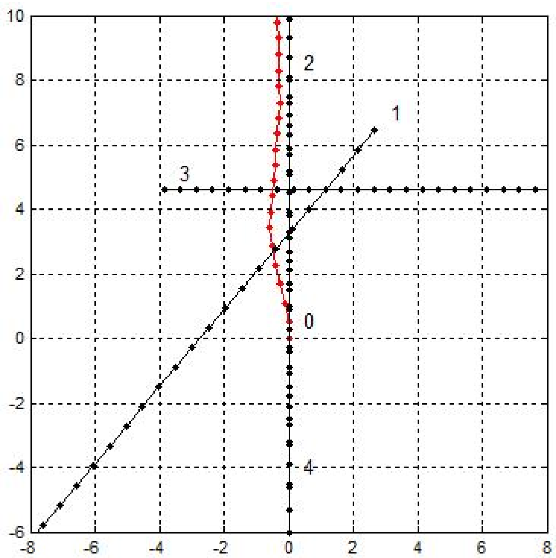

Figure 8. The safe trajectory of own ship for KO algorithm in good visibility $D_s=1$ *nm* in situation of passing $j=4$ encountered ships, $r(t_k)=0$, $d(t_k)=0.38$ *nm*.

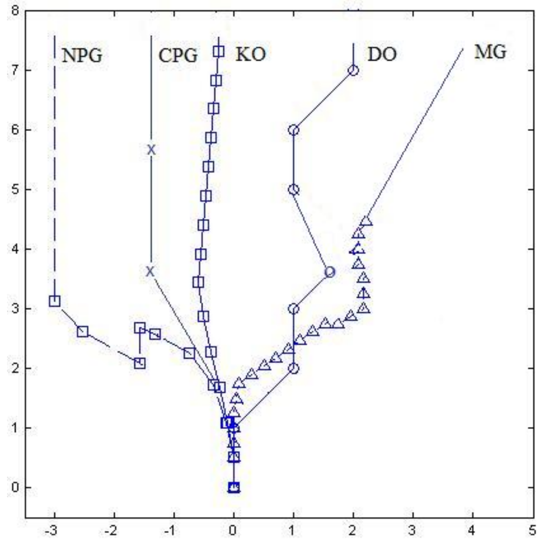

Figure 9. The comparison of own ship safe trajectories in good visibility $D_s=1$ *nm* in situation of passing $j=4$ encountered ships.

3.2 *Situation for j=8 encountered ships*

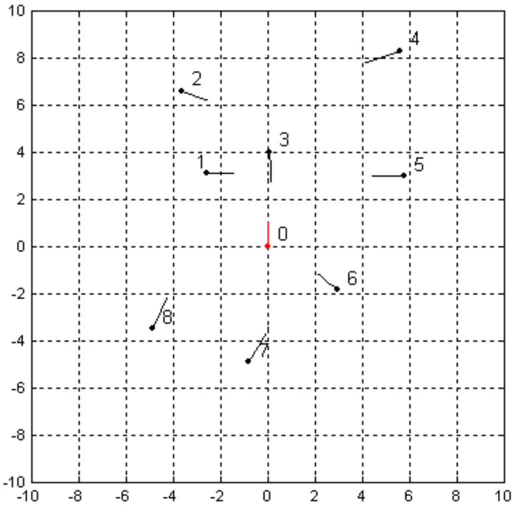

Figure 10. The 6 minute speed vectors of own and 8 encountered ships in situation in Kattegat Strait.

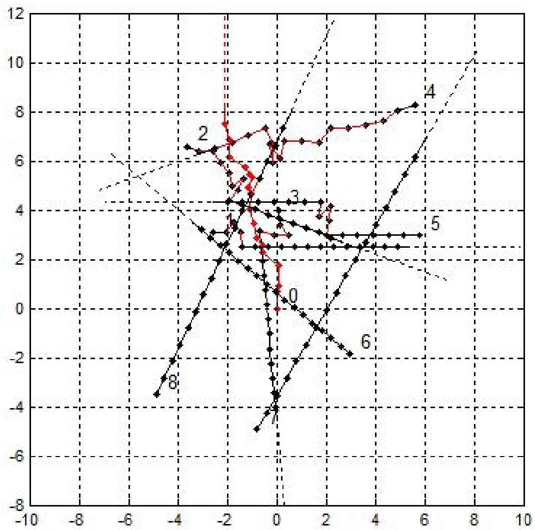

Figure 11. The safe trajectory of own ship for NPG algorithm in good visibility $D_s=1$ nm in situation of passing $j=8$ encountered ships, $r(t_k)=0$, $d(t_k)=2.10$ *nm*.

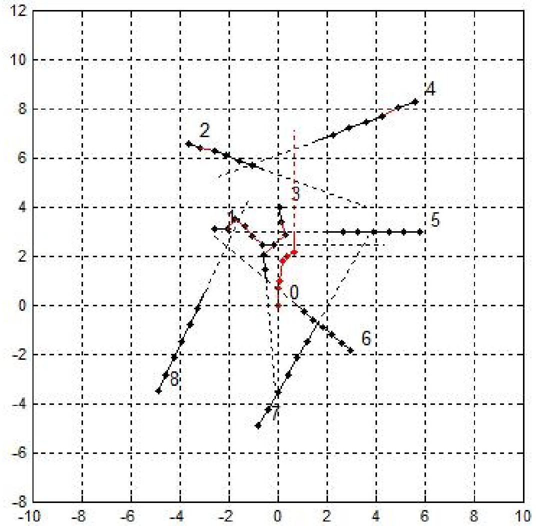

Figure 12. The safe trajectory of own ship for CPG algorithm in good visibility D_s=1 nm in situation of passing *j=8* encountered ships, *r(t$_k$)=0, d(t$_k$)=0.68 nm*.

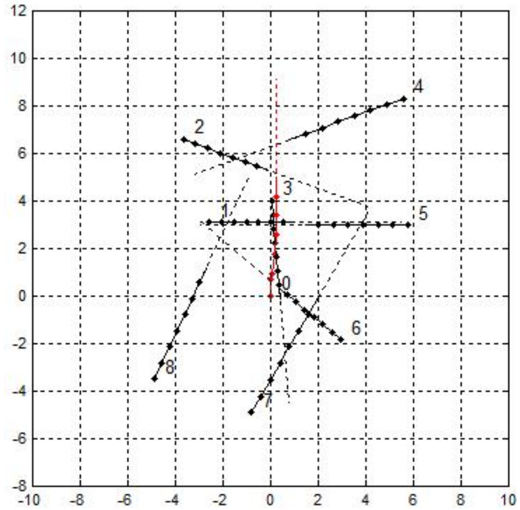

Figure 15. The safe trajectory of own ship for KO algorithm in good visibility $D_s=1$ *nm* in situation of passing *j=8* encountered ships, *r(t$_k$)=0, d(t$_k$)=0.26 nm*.

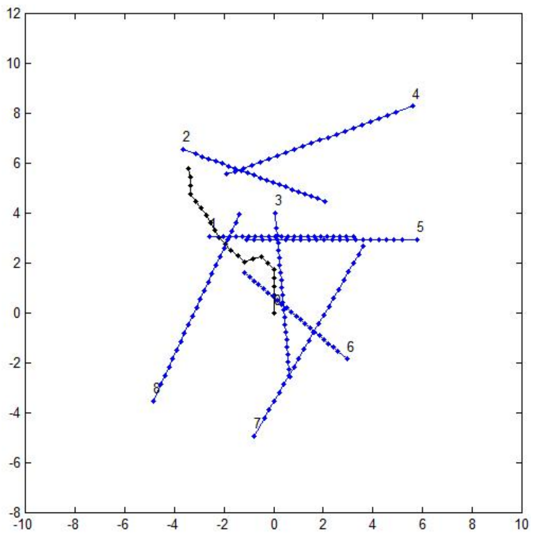

Figure 13. The safe trajectory of own ship for MG algorithm in good visibility $D_s=1$ *nm* in situation of passing *j=8* encountered ships, *r(t$_k$)=0, d(t$_k$)=2.74 nm*.

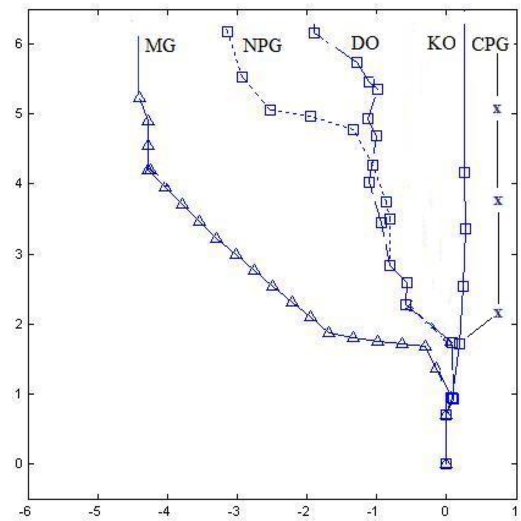

Figure 16. The comparison of own ship safe trajectories in good visibility $D_s=1$ *nm* in situation of passing *j=8* encountered ships.

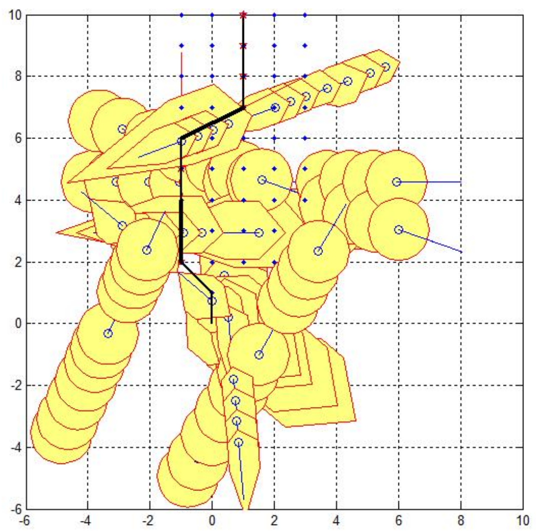

Figure 14. The safe trajectory of own ship for DO algorithm in good visibility $D_s=1$ *nm* in situation of passing *j=8* encountered ships, $t_K^* = 0.93\,h$.

3.3 *Situation for j=19 encountered ships*

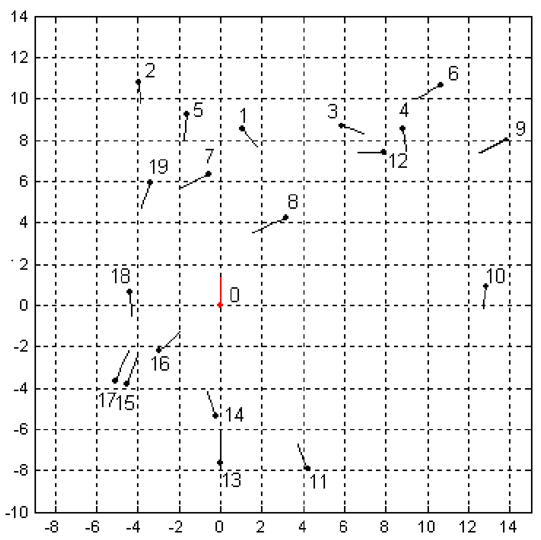

Figure 17. The 6 minute speed vectors of own and 19 encountered ships in situation on the North Sea.

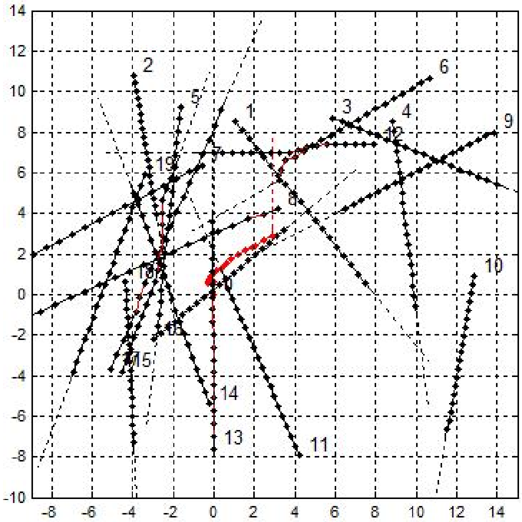

Figure 18. The safe trajectory of own ship for NPG algorithm in good visibility D_s=1 nm in situation of passing *j=19* encountered ships, $r(t_k)=0$, $d(t_k)=2.92$ *nm*.

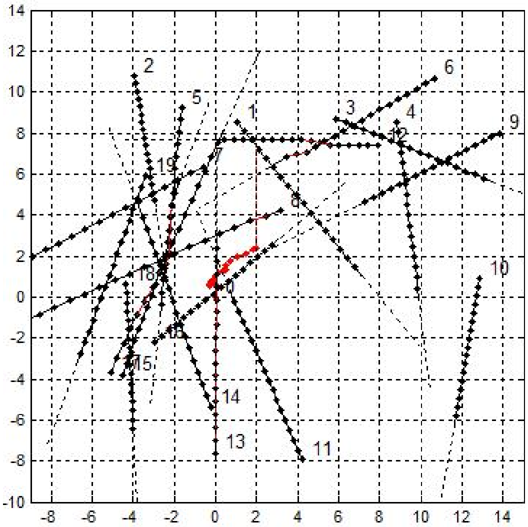

Figure 19. The safe trajectory of own ship for CPG algorithm in good visibility D_s=1 nm in situation of passing *j=19* encountered ships, $r(t_k)=0$, $d(t_k)=1.95$ *nm*.

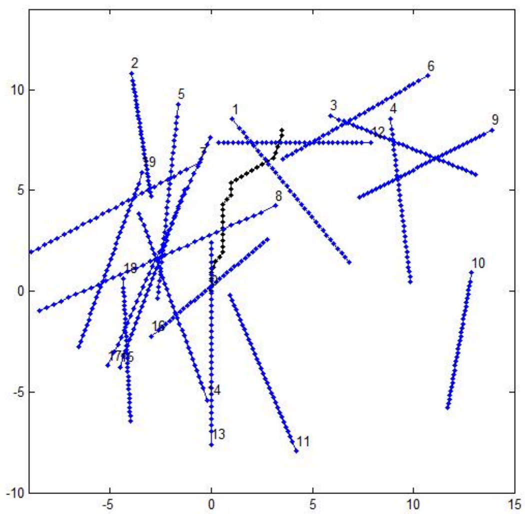

Figure 20. The safe trajectory of own ship for MG algorithm in good visibility $D_s=1$ *nm* in situation of passing *j=19* encountered ships, $r(t_k)=0$, $d(t_k)=3.81$ *nm*.

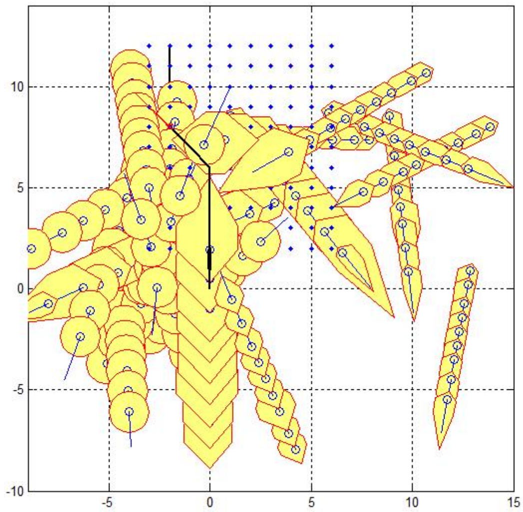

Figure 21. The safe trajectory of own ship for DO algorithm in good visibility $D_s=1$ *nm* in situation of passing *j=19* encountered ships, $t_K^* = 1.10\,h$.

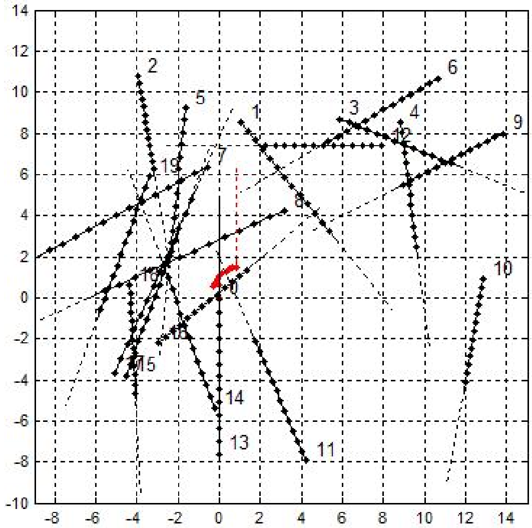

Figure 22. The safe trajectory of own ship for KO algorithm in good visibility $D_s=1$ *nm* in situation of passing *j=19* encountered ships, $r(t_k)=0$, $d(t_k)=0.84$ *nm*.

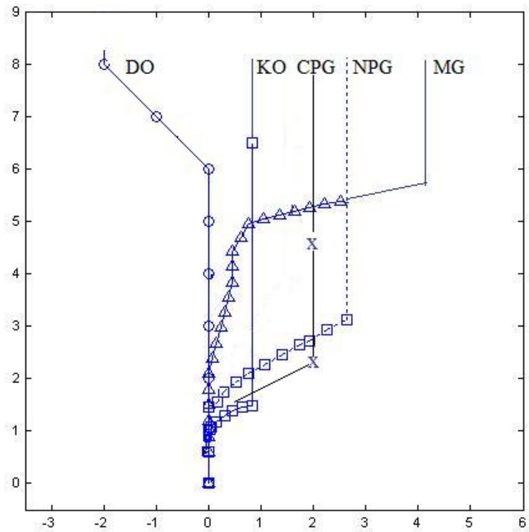

Figure 23. The comparison of own ship safe trajectories in good visibility $D_s=1$ *nm* in situation of passing *j=19* encountered ships.

3.4 *Situation for j=47 encountered ships*

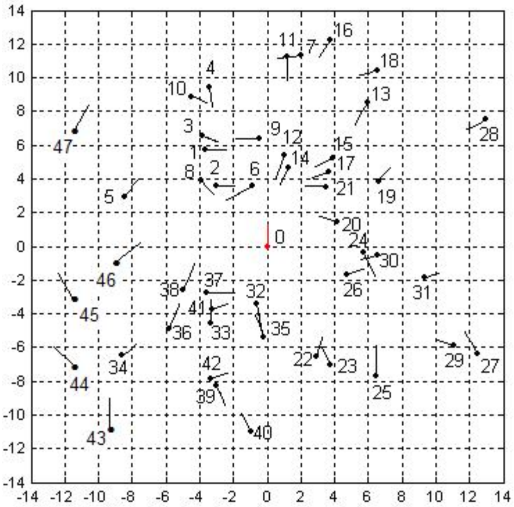

Figure 24. The 6 minute speed vectors of own and 47 encountered ships in situation in the English Channel.

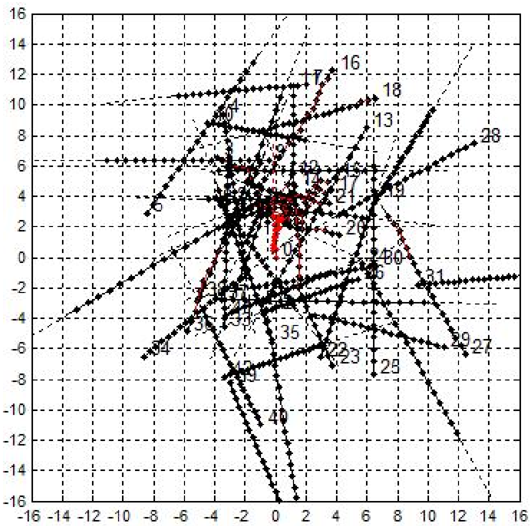

Figure 25. The safe trajectory of own ship for NPG algorithm in good visibility D_s=1 nm in situation of passing *j=47* encountered ships, $r(t_k)=0$, $d(t_k)=0.11$ *nm*.

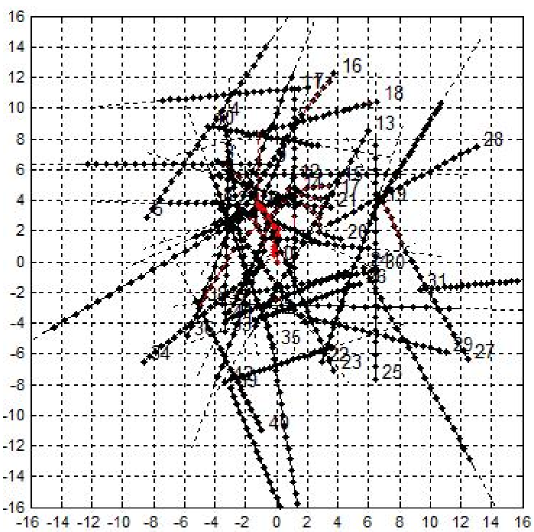

Figure 26. The safe trajectory of own ship for CPG algorithm in good visibility D_s=1 nm in situation of passing *j=47* encountered ships, $r(t_k)=0$, $d(t_k)=1.17$ *nm*.

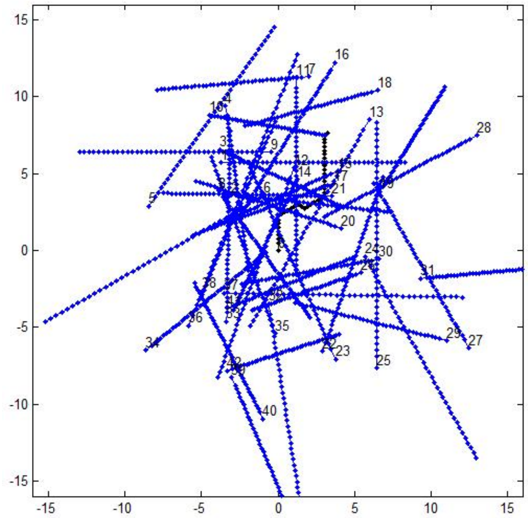

Figure 27. The safe trajectory of own ship for MG algorithm in good visibility $D_s=1$ *nm* in situation of passing *j=47* encountered ships, $r(t_k)=0$, $d(t_k)=3.83$ *nm*.

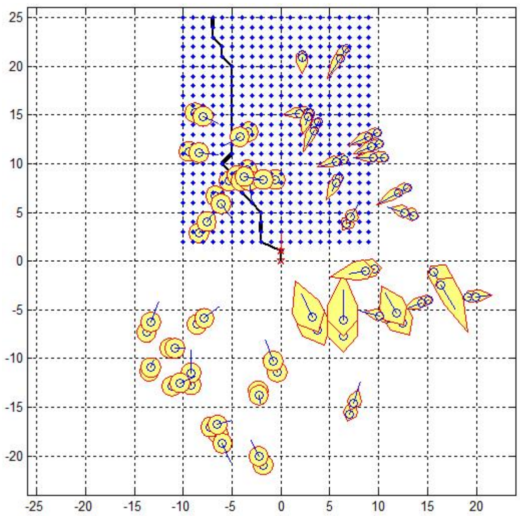

Figure 28. The safe trajectory of own ship for DO algorithm in good visibility $D_s=1$ *nm* in situation of passing *j=47* encountered ships, $t_K^* = 3.03\,h$.

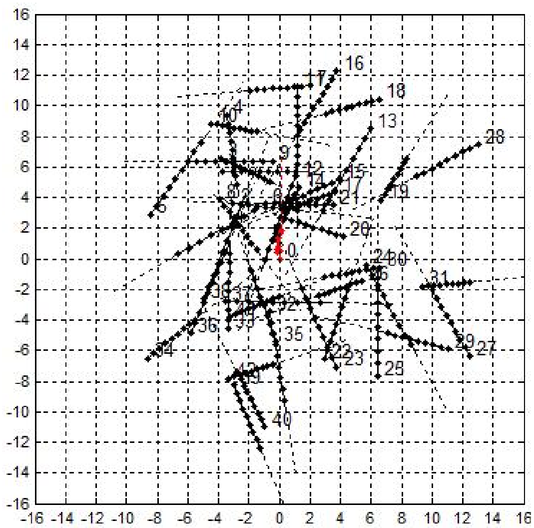

Figure 29. The safe trajectory of own ship for KO algorithm in good visibility $D_s=1$ *nm* in situation of passing *j=47* encountered ships, $r(t_k)=0$, $d(t_k)=0.11$ *nm*.

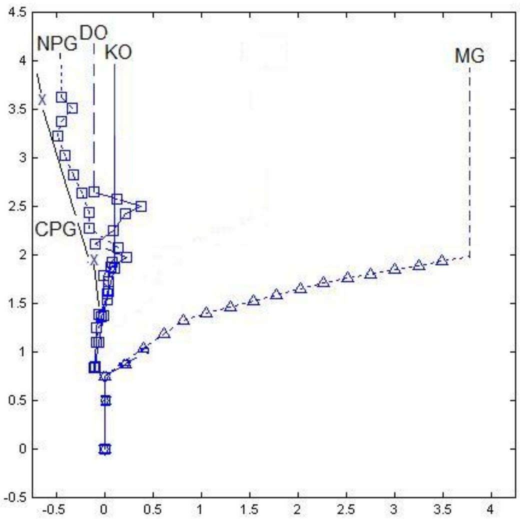

Figure 30. The comparison of own ship safe trajectories in good visibility $D_s=1$ *nm* in situation of passing $j=47$ encountered ships.

4 CONCLUSION

In order to ensure safe navigation the 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 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 takes 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 sensitivity of the final game payment:

- is least relative to the sampling period of the trajectory and advance time manoeuvre,
- most is relative to changes of the own and met ships speed and course,
- it grows with the degree of playing character of the control process and with the quantity of admissible strategies.

REFERENCES

Baba, N. & Jain, L.C. 2001. *Computational Intelligence in Games*. New York: Physica-Verlag.

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

Cichuta, M. & Dalecki, M. 2000. *Study of computer program determining safe ship's game trajectory with consideration risk of collision*. M.Sc. thesis, Gdynia Maritime Academy, (in Polish).

Engwerda, J.C. 2005. *LQ Dynamic Optimization and Differential Games*. West Sussex: John Wiley and Sons.

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

Lavalle, S.M. 2006. *Planning algorithm*. New York: Camridge.

Lebkowski, A. 2001. *Study and computer simulation of game positional control algorithm with consideration ship's speed changes in Matlab*. M.Sc. thesis, Gdynia Maritime Academy, (in Polish).

Levine, W.S. 1996. *The Control Handbook*. Florida: CRC Press.

Lisowski, J. 2006a. *The dynamic game theory methods applied to ship control with minimum risk of collision*. Risk Analysis V: Simulation and Hazard Mitigation. Southampton, Boston: WIT Press: 293-302.

Lisowski, J. 2006b. Game theory methods in control synthesis of marine moving objects – mathematical models and computer support algorithms. *Proc. of the 10th IEEE International Conference on Methods and Models in Automation and Robotics*, Miedzyzdroje: 21-39.

Lisowski, J. 2007a. Intelligent safe ship control methods in collision avoidance. *Proc. of European Control Conference*, Kos: 1-6.
Lisowski, J. 2007b. *The dynamic game models of safe navigation*. Chapter in Monograph TransNav'2007: Advances in marine navigation and safety of sea transportation. Gdynia: Gdynia Maritime University-The Nautical Institute in London: 23-30.
Lisowski, J. 2007c. Application of dynamic game and neural network in safe ship control. *Polish Journal of Environmental Studies*. Vol. 16: 114-120.
Lisowski, J. 2008a. *Optimal and game ship control algorithms avoiding collisions at sea*. Risk Analysis VI: Computer Simulation Risk Analysis and Hazard Mitigation. Southampton, Boston: WIT Press: 1-10.
Lisowski, J. 2008b. *Game and optimal safe ship control*. Chapter in monograph Recent Advances in Control and Automation. Warszawa: Exit: 302-312.
Lisowski, J. 2008c. Sensitivity of safe game ship control. *Proc. of 16th IEEE Mediterranean Conference on Control and Automation*, Corsica: 220-225.
Lisowski, J. & Mohamed-Seghir, M. 2008d. Safe ship control methods based on fuzzy set theory. *Polish Journal of Environmental Studies*. Vol. 17, No 3C: 55-58.
Nisan, N., Roughgarden, T., Tardos, E. & Vazirani, V.V. 2007. *Algorithmic Game Theory*. New York: Cambridge University Press: 717-733.
Nise, N.S. 2008. *Control systems engineering*. New York: John Wiley & Sons.
Nowak, A.S. & Szajowski, K. 2005 *Advances in Dynamic Games – Applications to Economics, Finance, Optimization and Stochastic Control*. Boston, Basel, Berlin: Birkhauser.
Osborne, M.J, 2004. *An Introduction to Game Theory*. New York: Oxford University Press.
Pachciarek, A. 2007. *Comparative analysis of safe ship trajectory determining on computer simulation*. B.Sc. thesis, Gdynia Maritime University, (in Polish).
Radzik, T. 2000. Characterization of optimal strategies in matrix games with convexity properties. *International Journal of Game Theory.* Vol. 29: 211-228.
Sandom, C. & Harvey, R.S. 2004. *Human factors for engineers*. London: The Institution of Electrical Engineers.
Segal, A. & Miloh, T. 1998. A new three dimensional differential game of pursuit and evasion. *Proc. of the 8th International Symposium on Dynamic Games and Applications.* Maastricht.
Skogestad, S. & Postlethwaite, I. 2005. *Multivariable feedback control*. New York: John Wiley & Sons.
Straffin, P.D. 2001. *Game Theory and Strategy*. Warszawa: Scholar (in Polish).