Int. J. Appl. Math. Comput. Sci., 2013, Vol. 23, No. 2, 439–446
DOI: 10.2478/amcs-2013-0033

SENSITIVITY OF COMPUTER SUPPORT GAME ALGORITHMS OF SAFE SHIP CONTROL

JÓZEF LISOWSKI

Department of Ship Automation
Gdynia Maritime University, 83 Morska Str., 81-225 Gdynia, Poland
e-mail: jlis@am.gdynia.pl

The paper investigates the sensitivity of safe ship control to inaccurate data from the ARPA anti-collision radar system and to changes in the process control parameters. The system structure of safe ship control in collision situations and computer support programmes exploring information from the ARPA anti-collision radar are presented. Sensitivity characteristics of the multistage positional non-cooperative and cooperative game and kinematics optimization control algorithms are determined through examples of navigational situations with restricted visibility at sea.

Keywords: differential games, positional games, matrix games, dual linear programming.

1. Introduction

The process of handling a ship as a multidimensional dynamic object depends on both the accuracy of details concerning the current navigational situation obtained from a Automatic Radar Plotting Aids (ARPAs) anti-collision system and on the form of the process model used for the control synthesis. There are various methods for the avoidance of ship collision. The simplest method is determination of the manoeuvre of a change in the course or speed of an own ship in relation to the most dangerous ship encountered. A more effective method is to determine a safe trajectory (Pietrzykowski, 2011; Szlapczynski and Smierzchalski, 2009; Szynkiewicz and Błaszczyk, 2011). Most adequate to the real character of control process is determination of the game trajectory of a ship (Lisowski, 2007).

1.1. Information about the state process. The ARPA system allows tracking automatically at least 20 encountered j objects as shown in Fig. 1, as well as determining their movement parameters (speed V_j, course ψ_j) and elements of approach to the 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 assessing the collision risk r_j (Bist, 2000; Bole *et al.*, 2006). The risk value is defined by referring the current situation of approach, described by some parameters, to the assumed evaluation of the situation as safe, determined by a safe distance of approach D_s and a safe time T_s, which are necessary to execute a collision avoiding manoeuvre taking into consideration the distance D_j to the j-th ship encountered (Cahill, 2002; Zio, 2009; Fadali and Visioli, 2009). The functional scope of a standard ARPA system ends with the trial manoeuvre to alter the ship course or the ship's speed selected by the navigator (Cockcroft and Lameijer, 2006; Gluver and Olsen, 1998).

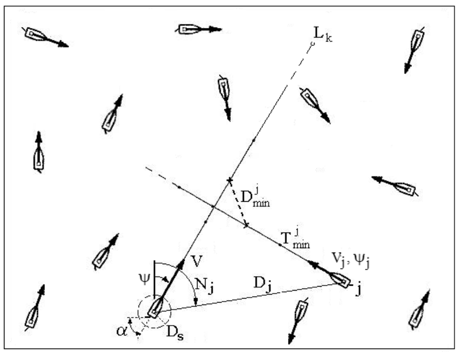

Fig. 1. Navigational situation of the passing of the own ship with j ship encountered moving at V_j speed and ψ_j course.

1.2. Processes of game ship control. Assuming that the dynamic movement of ships in time occurs under the influence of the appropriate sets of control,

$$[U_o{}^{(\eta)}, U_j{}^{(\eta)}], \tag{1}$$

where $U_o{}^{(\eta)}$ is a set of the own ship's strategies, $U_j{}^{(\eta)}$ is a set of the j-th ship's strategies, $\eta = 0$ denotes the course and trajectory stabilisation, $\eta = 1$ is denotes the execution of the anti-collision manoeuvre in order to minimize the risk of collision, which in practice is achieved by satisfying the following inequality:

$$D^j_{\min} = \min D_j(t) \geq D_s, \tag{2}$$

$D^j{}_{\min}$ being the smallest distance of approach of the own ship and the j-th encountered object, D_s is the safe approach whose distance in the prevailing conditions depends on the visibility conditions at sea, the COLREG rules and the ship's dynamics, D_j is the current distance to the j-th object taken from the ARPA anti-collision system, $\eta = -1$ refers to the manoeuvring of the ship in order to achieve the closest point of approach, for example, during the approach of a rescue ship, transfer of cargo from ship to ship, destruction of an enemy's ship, etc.

In the adopted notation we can distinguish the following types ship of steering in order to achieve a specific goal:

- basic type of steering, stabilization of the course or trajectory: $[U_o{}^{(0)}, U_j{}^{(0)}]$,
- avoidance of a collision by executing
  (a) own ship's manoeuvres: $[U_o{}^{(1)}, U_j{}^{(0)}]$,
  (b) manoeuvres of the j-th ship: $[U_o{}^{(0)}, U_j{}^{(1)}]$,
  (c) co-operative manoeuvres: $[U_o{}^{(1)}, U_j{}^{(1)}]$,
- encounter of ships: $[U_o{}^{(-1)}, U_j{}^{(-1)}]$,
- situations of a unilateral dynamic game: $[U_o{}^{(-1)}, U_j{}^{(0)}]$ and $[U_o{}^{(0)}, U_j{}^{(-1)}]$.

Dangerous situations resulting from a faulty assessment of the approaching process by one party with the other party's failure to conduct observations include the following:

- one ship is equipped with a radar or an anti-collision system, the other with a damaged radar or without this device,
- chasing situations which refer to a typical conflicting dynamic game: $[U_o{}^{(-1)}, U_j{}^{(1)}]$ and $[U_o{}^{(1)}, U_j{}^{(-1)}]$.

The first case usually represents regular optimal control, the second and third are unilateral games, while the fourth and fifth cases represent conflicting games (Basar and Olsder, 1982; Galuszka and Swierniak, 2005).

1.3. Computer support for the 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, non-stationary and game making in its nature. In practice, methods of selecting a manoeuvre take the form of appropriate steering algorithms supporting the navigators decision in a collision situation for realization by an autopilot (Fang and Luo, 2005; Landau *et al.*, 2011; Tomera, 2010; Witkowska *et al.*, 2007).

2. Basic model of game control

The model of the process consists of both the kinematics and dynamics of the ship movement, the strategies of the encountered ships and the performance control index of the own ship (Mesterton-Gibbons, 2001; Perez, 2005; Clarke, 2003; Tomera and Smierzchalski, 2006).

2.1. State equations. The most general description of the own ship passing j other encountered ships is the model of a differential game of j moving control ships (Fig. 2). The properties of the control process are

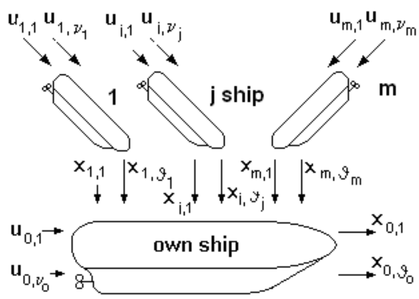

Fig. 2. Block diagram of the basic differential game model of the safe ship control process.

described by the state equation

$$\dot{x}_i = f_i\left[(x_{o,\theta_o}, \ldots, x_{m,\theta_m}), (u_{o,\nu_o}, \ldots, u_{m,\nu_m}), t\right], \tag{3}$$

where $\vec{x}_{o,\theta_o}(t)$ is a θ_o dimensional vector of the process state of the own ship, $\vec{x}_{j,\theta_j}(t)$ is a θ_j dimensional vector of the process state for the j-th ship, $\vec{u}_{j,\nu_j}(t)$ is a ν_j dimensional control vector of the own ship, $\vec{x}_{j,\nu_j}(t)$ is a ν_j dimensional control vector of the j-th ship (Isaacs, 1965; Lisowski, 2010; Engwerda, 2005).

Taking into consideration the equations reflecting the own ship's hydromechanics and equations of the own ship's movement relative to the j-th encountered ship, the equations of the general state of the process (3) take the form (4) (Fossen, 2011; Keesman, 2011):

$$\begin{aligned}
\dot{x}_{o,1} &= x_{o,2},\\
\dot{x}_{o,2} &= a_1 x_{o,2} x_{o,3} + a_2 x_{o,3} \left|x_{o,3}\right| + b_1 x_{o,3} \left|x_{o,3}\right| u_{o,1},\\
\dot{x}_{o,3} &= a_4 x_{o,3} \left|x_{o,3}\right| \left|x_{o,4}\right| x_{o,4} \left(1 + x_{o,4}\right)\\
&\quad + a_5 x_{o,2} x_{o,3} x_{o,4} \left|x_{o,4}\right| + a_6 x_{o,2} x_{o,3} x_{o,4}\\
&\quad + a_7 x_{o,3} \left|x_{o,3}\right| + a_8 x_{o,5} \left|x_{o,5}\right| x_{o,6}\\
&\quad + b_2 x_{o,3} x_{o,4} \left|x_{o,3} u_{o,1}\right|,\\
\dot{x}_{o,4} &= a_3 x_{o,3} x_{o,4} + a_4 x_{o,3} x_{o,4} \left|x_{o,4}\right| + a_5 x_{o,2} x_{o,2}\\
&\quad + a_9 x_{o,2} + b_2 x_{o,3} u_{o,1},\\
\dot{x}_{o,5} &= a_{10} x_{o,5} + b_3 u_{o,2},\\
\dot{x}_{o,6} &= a_{11} x_{o,6} + b_4 u_{o,3},\\
\dot{x}_{j,1} &= -x_{o,3} + x_{j,2} x_{o,2} + x_{j,3} \cos x_{j,3},\\
\dot{x}_{j,2} &= -x_{o,2} x_{j,1} + x_{j,3} \sin x_{j,3},\\
\dot{x}_{j,3} &= -x_{o,2} + b_{4+j} x_{j,3} u_{j,1}\\
\dot{x}_{j,4} &= a_{11+j} x_{j,4} \left|x_{j,4}\right| + b_{5+j} u_{j,2}.
\end{aligned} \tag{4}$$

The state variables are represented by the following values:

$x_{o,1} = \psi$: course of the own ship,

$x_{o,2} = \dot{\psi}$: angular turning speed of the own ship,

$x_{o,3} = V$: speed of the own ship,

$x_{o,4} = \beta$: drift angle of the own ship,

$x_{o,5} = n$: rotational speed of the screw propeller of the own ship,

$x_{o,6} = H$: pitch of the adjustable propeller of the own ship,

$x_{j,1} = D_j$: distance to the j-th object, or x_j—its coordinate,

$x_{j,2} = N_j$: bearing of the j-th object, or y_j—its coordinate,

$x_{j,3} = \psi_j$: course of the j-th object, or β_j—relative meeting angle,

$x_{j,4} = V_j$: speed of the j-th object,

where $\theta_o = 6, \theta_j = 4$.

The control values are represented as follows:

$u_{o,1} = \theta_r$: reference rudder angle of the own ship, or angular turning speed of the own ship, or course of the own ship, depending on the kind of approximated model of the process,

$u_{o,2} = n_r$: reference rotational speed of the own ship's screw propeller, or force of the propeller thrust of the own ship, or speed of the own ship,

$u_{o,3} = H_r$: reference pitch of the adjustable propeller of the own ship,

$u_{j,1} = \psi_j$: course of the j-th object, or $\dot{\psi}_j$ angular turning speed of the j-th object,

$u_{j,2} = V_j$: speed of the j-th object, or force of the propeller thrust of the j-th object,

where $\nu_o = 3, \nu_j = 2$. For $j = 20$ objects, the basic game model is represented by $i = 86$ state variables of the process control.

2.2. State and control constraints. The constraints of the control and the state of the process are connected with the basic condition for the safe passing of ships at a safe distance D_s in compliance with the International Regulations for Preventing Collisions at Sea (COLREGS Rules), generally in the following form:

$$g_j\left(x_{j,\theta_j}, u_{j,\nu_j}\right) = D_s - D^j_{\min} \leq 0. \tag{5}$$

2.3. Performance index for control. 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 the j-th ship is the minimization of the objective function in the form of an integral cost and the final one:

$$I_{0,j} = \int_{t_0}^{t_k} [x_{0,\theta_0}(t)]^2 \mathrm{d}t + r_j(t_k) + d(t_k) \to \min. \tag{6}$$

The integral cost represents the ship's losing her way while passing the encountered ships and the final cost determines the final risk of collision $r_j(t_k)$ relative to the j-th ship and the final deflection of the ship $d(t_k)$ from the reference trajectory (Fig. 3) (Modarres, 2006; Nisan *et al.*, 2007).

3. Game control algorithms

For practical synthesis of safe control algorithms, various simplified models are formulated:

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

The degree of model simplification depends on an optimal control method applied and the level of cooperation between ships (Table 1) (Findeisen *et al.*, 1980; Fletcher, 1987).

3.1. Multi-stage non-cooperative positional game algorithm PG_{nc}. The optimal steering of the own ship $u_0^*(t)$, for the current position $p(t)$ is equivalent to the optimal positional steering $u_0^*(p)$. The sets of acceptable strategies $U_j^0\left[p\left(t_k\right)\right]$ are determined for the encountered ships relative to the own ship and initial sets $U_0^{jw}\left[p\left(t_k\right)\right]$

Table 1. Algorithms for determining ship strategies.

Approximate model	Support algorithm	Method of optimization	Form of decision
multi-stage non-cooperative positional game	PG_{nc}	triple linear programming	game trajectory
multi-stage cooperative positional game	PG_c	triple linear programming	game trajectory
kinematic	KO	linear programming	optimal trajectory

of acceptable strategies of the own ship relative to each of the encountered ships. The pair of vectors u_j^m and u_0^j relative to the j-th ship is determined and then the optimal positional strategy for the own ship $u_0^*(p)$ from the condition

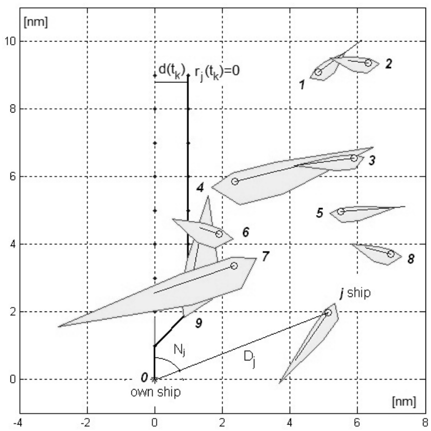

Fig. 3. Final risk of collision $r_j(\mathrm{t}_k)$ relative and the final deflection $d(t_k)$ from the reference trajectory in a situation of passing the j-th ship encountered: nm—nautical mile.

$$I_0^* = \min_{u_0 \in \bigcap_{j=1}^{m} U_0^j} \max_{u_j^m \in U_j} \min_{u_0^j \in U_0^j} \int_{t_0}^{t_{L_k}} u_0(t)\,\mathrm{d}t = S_0^*(x_0, L_k) \tag{7}$$

is established.

The function 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 $p_r(t_k)$ voyage route (Isil-Bozma and Koditschek, 2001; Millington and Funge, 2009; Osborne, 2004). 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 triple linear programming, assuming the relationship (7) to be the goal function along with the control constraints (5) (Błaszczyk *et al.*, 2007; Luus, 2000; Mehrotra, 1992; Pantoja, 1998).

3.2. Multi-stage cooperative positional game algorithm PG_c.

The performance index of control for a cooperative game has the form

$$I_0^* = \min_{u_0 \in \bigcap_{j=1}^{m} U_0^j} \min_{u_j^m \in U_j} \min_{u_0^j \in U_0^j} \int_{t_0}^{t_{L_k}} u_0(t)\,\mathrm{d}t = S_0^*(x_0, L_k). \tag{8}$$

3.3. Multi-stage non-game kinematic optimization algorithm KO.

The goal function for kinematics optimization has the form

$$I_0^* = \min_{u_0 \in \bigcap_{j=1}^{m} U_0^j} \int_{t_0}^{t_{L_k}} u_0(t)\,\mathrm{d}t = S_0^*(x_0, L_k). \tag{9}$$

Using the function lp from the Matlab Optimization Toolbox, the positional multi-stage game non-cooperative manoeuvring PG_{nc}, the positional multi-stage game cooperative manoeuvring PG_c and the multi-stage non-game kinematic optimization KO programs have been designed for the determination of the own ship's safe trajectory in a collision situation (Lisowski, 2009).

4. Sensitivity of safe ship control

4.1. Definition of safe control sensitivity.

The sensitivity analysis of game control makes, for sensitivity analysis of the game, the final cost (6) measured as the final deviation of $d(t_k) = d_k$ of the safe game trajectory from the reference trajectory. Taking into consideration the practical application of the game control algorithm for the own ship in a collision situation, it is recommended to perform the sensitivity analysis of safe control with regard to the accuracy degree of the information received from the anti-collision ARPA radar system in the current approach situation, on the one hand, and also with regard to the changes in kinematic and dynamic parameters of the control process on the other (Lisowski, 2011; Straffin, 2001; Wierzbicki, 1984).

Admissible average errors that can be contributed by sensors of the anti-collision system can have the following values:

– for radar,

- bearing: $\pm 0.22°$,
- form of cluster: $\pm 0.05°$,

- form of impulse: ±20 m,
- margin of antenna drive: ±0.5°,
- sampling of bearing: ±0.01°,
- sampling of distance: ±0.01 nm,

– gyrocompass: ±0.5°,
– log: ±0.5 knots,
– GPS: ±15 m.

The algebraic sum of all errors, influencing the picturing of a navigational situation, cannot exceed ±6% or ±3°.

4.2. Sensitivity of safe ship control to the inaccuracy of information from the ARPA system. Let $X_{0,j}$ represent state process control information on the navigational situation such that

$$X_{0,j} = \{V, \psi, V_j, \psi_j, D_j, N_j\}. \tag{10}$$

Let then $X'_{0,j}$ represent information from the ARPA system containing errors of measurement and processing parameters:

$$X'_{0,j} = \{V \pm \delta V, \psi \pm \delta\psi, V_j \pm \delta V_j, \psi_j \pm \delta\psi_j, D_j \pm \delta D_j, N_j \pm \delta N_j\}. \tag{11}$$

The relative measure of the sensitivity of the final cost in the game s_1 as a final deviation of the ship's safe trajectory d_k from the reference trajectory will be

$$s_1 = (X'_{0,j},\ X_{0,j}) = \frac{d'_k(X'_{o,j})}{d_k(X_{0,j})}, \tag{12}$$

$$s_1 = \{s^V, s^\psi, s^{V_j}, s^{\psi_j}, s^{D_j}, s^{N_j}\}. \tag{13}$$

4.3. Sensitivity of safe ship control to process parameter alterations. Let X_p represent a set of parameters of state process control:

$$X_p = \{t_m, D_s, \Delta t_k, \Delta V\}. \tag{14}$$

Let then X'_p represent information containing errors of measurement and processing parameters:

$$X'_p = \{t_m \pm \delta t_m, D_s \pm \delta D_s,\ t_k \pm \delta t_k, \Delta V \pm \delta\Delta V\}. \tag{15}$$

The relative measure of sensitivity of the final cost in the game s_2 as a final deviation of the ship's safe trajectory d_k from the assumed trajectory will be

$$s_2 = (X'_p,\ X_p) = \frac{d'_k(X'_p)}{d_k(X_p)}. \tag{16}$$

The relative measure of sensitivity of the final cost in the game s_2 as a final deviation of the ship's safe trajectory d_k from the assumed trajectory will be

$$s_2 = \{s^{t_m}, s^{D_s},\ s^{\Delta t_k}, s^{\Delta V}\}, \tag{17}$$

where t_m is the predicted time of the manoeuvre with respect to the dynamic properties of the own ship, t_k the duration of one stage of the ship's trajectory, D_s the safe distance, ΔV the reduction of the own ship's speed for a deviation from the course greater than 30° (Baba and Jain, 2001).

5. Sensitivity characteristics of safe ship control in restricted visibility at sea

Computer simulations of PG_{nc}, PG_c and KO programs, as computer software supporting the navigator manoeuvring decision, were carried out on an example of real navigational situations of passing $j = 12$ and $j = 20$ encountered ships. The situations were registered in the Skagerrak Strait on board r/v HORYZONT II, a research and training vessel of Gdynia Maritime University, on the radar screen of the ARPA anti-collision system Raytheon (Fig. 4).

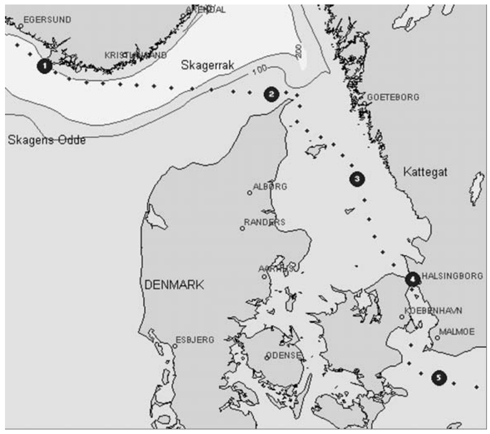

Fig. 4. Place for identification of navigational situations in the Skagerrak Strait.

5.1. Sensitivity characteristics for 12 ships encountered. Computer simulation of PG_{nc}, PG_c and KO programs was carried out in the Matlab/Simulink software on an example of the real navigational situation of passing $j = 12$ encountered ships in the Skagerrak Strait in restricted visibility when $D_s = 2$ nm. In addition, sensitivity characteristics (Figs. 5–11) are

$$\Delta X_{0,j} = \frac{\delta X_{0,j}}{X_{0,j}}, \tag{18}$$

$$\Delta X_p = \frac{\delta X_p}{X_p}. \tag{19}$$

6. Conclusions

Application of simplified models of the dynamic game of the process to the synthesis of optimal control allows

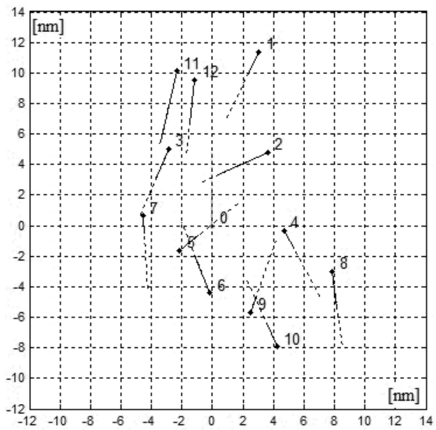

Fig. 5. Twelve minute speed vectors of the own ship and twelve encountered ships in a situation occurring in the Skagerrak Strait.

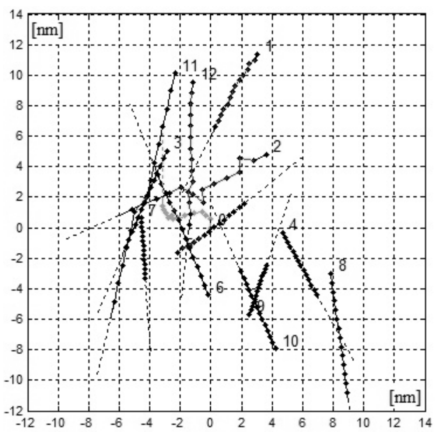

Fig. 6. Safe trajectory of the own ship for the PG_{nc} algorithm in restricted visibility $D_s = 2$ nm in a situation of passing $j = 12$ encountered ships, $r(t_k) = 0$, $d(t_k) = 3.20$ nm.

determination of the own ship safe trajectory in situations of passing a greater number of encountered ships as a certain sequence of the course and speed manoeuvres. The developed algorithms take also into consideration the COLREGS rules and the predicted time of the manoeuvre approximating the ship's dynamic properties and evaluates the final deviation of the real trajectory from the reference value.

The sensitivity of the final game cost: is the least for changes of the duration of one stage trajectory and for changes in the predicted time manoeuvre, the greatest for changes in the own and met ships' speeds and courses, grows with the degree of the ships cooperation for the purpose of avoiding collision, grows with the number of meeting ships and with the quantity of admissible strategies for the own ship and passing ships. The control algorithms considered are, in some sense, formal models of the thinking process of a navigator steering the

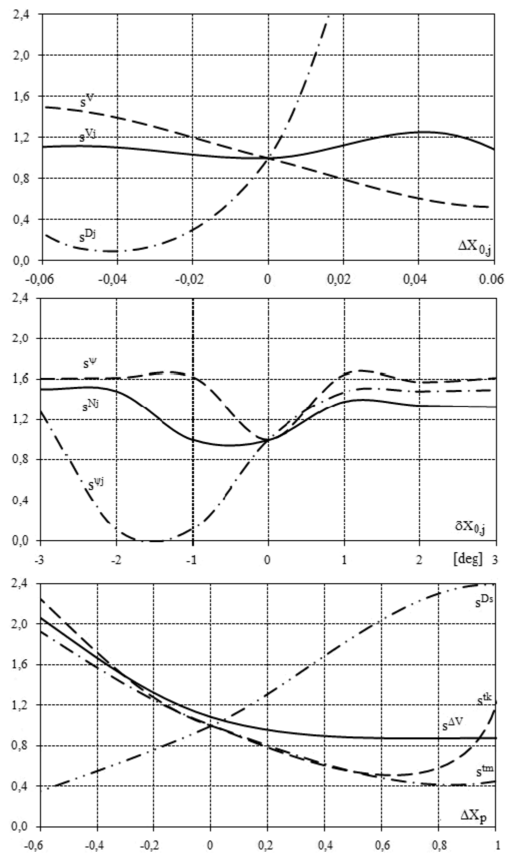

Fig. 7. Sensitivity characteristics of safe ship control according to the PG_{nc} programme on an example of the navigational situation $j = 12$ occurring in the Skagerrak Strait.

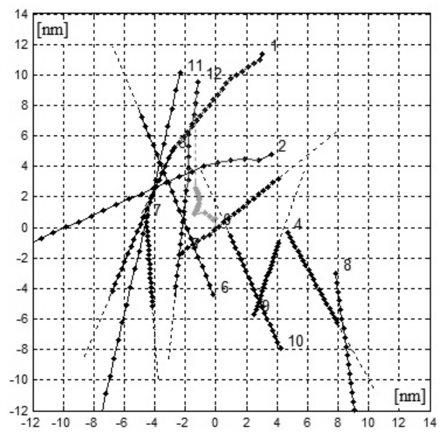

Fig. 8. Safe trajectory of the own ship for the PG_c algorithm in restricted visibility $D_s = 2$ nm in a situation of passing $j = 12$ encountered ships, $r(t_k) = 0$, $d(t_k) = 1.40$ nm.

ship's movement and making up manoeuvring decisions. Therefore, they may be applied in the construction of a new model of the ARPA system containing a computer supporting the navigator's decision making.

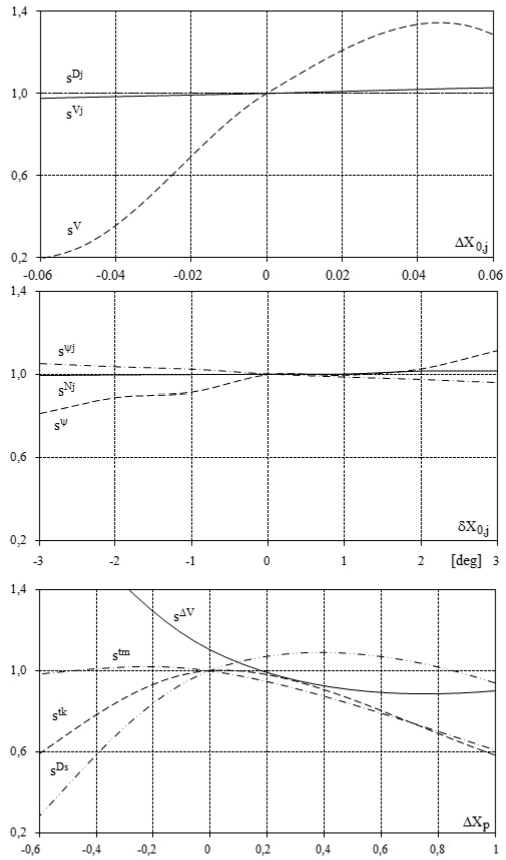

Fig. 9. Sensitivity characteristics of safe ship control according to the PG_c programme on an example of a navigational situation $j = 12$ occurring in the Skagerrak Strait.

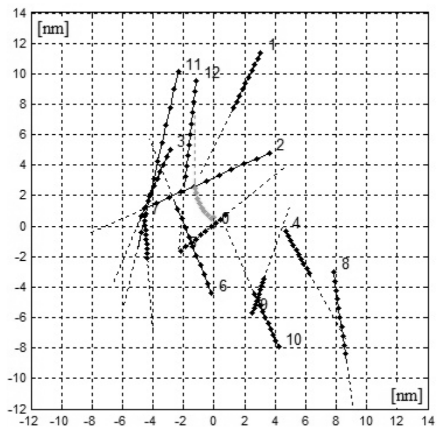

Fig. 10. Safe trajectory of the own ship for the KO algorithm in restricted visibility $D_s = 2$ nm in a situation of passing $(j = 12)$ encountered ships, $r(t_k) = 0$, $d(t_k) = 1.23$ nm.

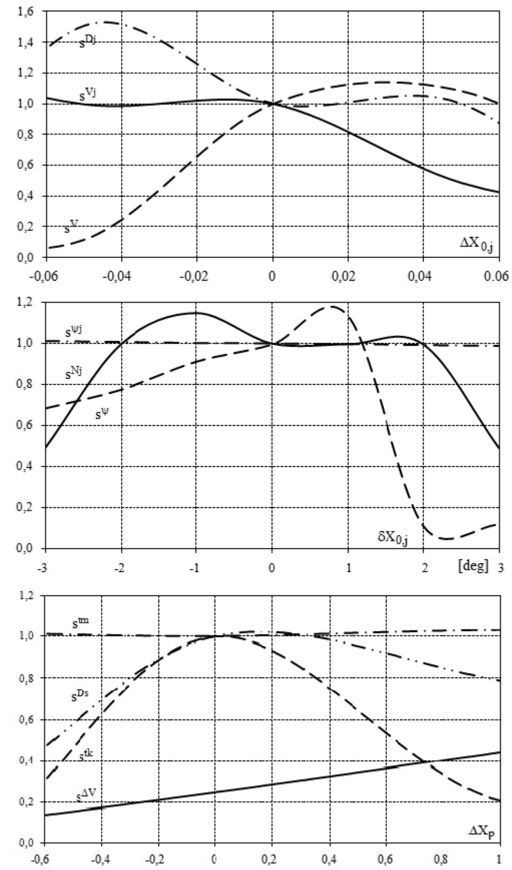

Fig. 11. Sensitivity characteristics of safe ship control according to the KO programme on an example of a navigational situation $(j = 12)$ occurring in the Skagerrak Strait.

References

Baba, N. and Jain, L. (2001). *Computational Intelligence in Games*, Physica-Verlag, New York, NY.

Basar, T. and Olsder, G. (1982). *Dynamic Non-Cooperative Game Theory*, Academic Press, New York, NY.

Bist, D. (2000). *Safety and Security at Sea*, Butter Heinemann, Oxford/New Delhi.

Błaszczyk, J., Karbowski, A. and Malinowski, K. (2007). Object library of algorithms for dynamic optimization problems: Benchmarking SQP and nonlinear interior point methods, *International Journal of Applied Mathematics and Computer Science* **17**(4): 515–537, DOI: 10.2478/v10006-007-0043-y.

Bole, A., Dineley, B. and Wall, A. (2006). *Radar and ARPA Manual*, Elsevier, Amsterdam/Tokyo.

Cahill, R. (2002). *Collisions and Their Causes*, Nautical Institute, London.

Clarke, D. (2003). The foundations of steering and manoeuvering, *Proceedings of the IFAC Conference on Manoeuvering and Control Marine Crafts, Girona, Spain*, pp. 124–132.

Cockcroft, A. and Lameijer, J. (2006). *The Collision Avoidance Rules*, Elsevier, Amsterdam/Tokyo.

Engwerda, J. (2005). *LQ Dynamic Optimization and Differential Games*, John Wiley & Sons, West Sussex.

Fadali, M. and Visioli, A. (2009). *Digital Control Engineering*, Elsevier, Amsterdam/Tokyo.

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

Findeisen, W., Szymanowski, J. and Wierzbicki, A. (1980). *The Nonlinear Hydrodynamic Model for Simulating a Ship Steering in Waves with Autopilot System*, Polish Scientific Publishers, Warsaw.

Fletcher, R. (1987). *Practical Methods of Optimization*, John Wiley & Sons, New York, NY.

Fossen, T. (2011). *Marine Craft Hydrodynamics and Motion Control*, Wiley, Trondheim.

Gałuszka, A. and Świerniak, A. (2005). Non-cooperative game approach to multi-robot planning, *International Journal of Applied Mathematics and Comuter Science* **15**(3): 359–367.

Gluver, H. and Olsen, D. (1998). *Ship Collision Analysis*, A.A. Balkema, Rotterdam/Brookfield.

Isaacs, R. (1965). *Differential Games*, John Wiley & Sons, New York, NY.

Isil-Bozma, H. and Koditschek, D. (2001). Assembly as a non-cooperative game of its pieces: Analysis of ID sphere assemblies, *Robotica* **19**(3): 93–108.

Keesman, K. (2011). *System Identification*, Springer, New York, NY.

Landau, I., Lozano, R., Saadand, M. and Karimi, A. (2011). *Adaptive Control*, Springer, London/New York, NY.

Lisowski, J. (2007). The dynamic game models of safe navigation, *in* A. Weintrit (Ed.), *Marine Navigation and Safety of Sea Transportation*, Gdynia Maritime University, Gdynia, pp. 23–30.

Lisowski, J. (2009). Sensitivity of safe game ship control on base information from ARPA radar, *in* G. Kouemou (Ed.), *Radar Technology*, In-Teh, Vukovar, pp. 61–86.

Lisowski, J. (2010). Optimization decision support system for safe ship control, *in* C.A. Brebbia (Ed.), *Risk Analysis*, WIT Press, Southampton/Boston, MA, pp. 259–272.

Lisowski, J. (2011). The sensitivity of safe ship control in restricted visibility at sea, *in* A. Weintrit (Ed.), *Marine Navigation and Safety of Sea Transportation*, Gdynia Maritime University, Gdynia, pp. 75–84.

Luus, R. (2000). *Iterative Dynamic Programming*, CRC Press, Boca Raton, FL.

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

Mesterton-Gibbons, M. (2001). *An Introduction to Game Theoretic Modeling*, American Mathematical Society, Providence, RI.

Millington, I. and Funge, J. (2009). *Artificial Intelligence for Games*, Elsevier, Amsterdam/Tokyo.

Modarres, M. (2006). *Risk Analysis in Engineering*, Taylor & Francis Group, Boca Raton, FL.

Nisan, N., Roughgarden, T., Tardos, E. and Vazirani, V. (2007). *Algorithmic Game Theory*, Cambridge University Press, New York, NY.

Osborne, M. (2004). *Algorithmic Game Theory*, Oxford University Press, New York, NY.

Pantoja, J. (1998). Differential dynamic programming and Newton's method, *International Journal of Control* **47**(5): 1539–1553.

Perez, T. (2005). *Ship Motion Control*, Springer, London.

Pietrzykowski, Z. (2011). *The Navigational Decision Support System on a Sea-Going Vessel*, Maritime University, Szczecin.

Straffin, P. (2001). *Game Theory and Strategy*, Scholar, Warsaw.

Szlapczynski, R. and Smierzchalski, R. (2009). Supporting navigator's decisions by visualizing ship collision risk, *Polish Maritime Research* **1**(59): 83–88.

Szynkiewicz, W. and Błaszczyk, J. (2011). Optimization-based approach to path planning for closed chain robot systems, *International Journal of Applied Mathematics and Computer Science* **21**(4): 659–670, DOI: 10.2478/v10006-011-0052-8.

Tomera, M. (2010). Nonlinear controller design of a ship autopilot, *International Journal of Applied Mathematics and Computer Science* **20**(2): 271–280, DOI: 10.2478/v10006-010-0020-8.

Tomera, M. and Smierzchalski, R. (2006). Sliding controller for ship course steering, *Proceedings of the IFAC Conference on Manoeuvering and Control of Marine Crafts, Lisbon, Portugal*, pp. 211–219.

Wierzbicki, A. (1984). *Models and Sensitivity of Control Systems*, Elsevier, Amsterdam.

Witkowska, A., Tomera, M. and Smierzchalski, R. (2007). A backstepping approach to ship course control, *International Journal of Applied Mathematics and Computer Science* **17**(1): 73–85, DOI: 10.2478/v10006-007-0007-2.

Zio, E. (2009). *Computational Methods for Reliability and Risk Analysis*, Series on Quality, Reliability and Engineering Statistics, Word Scientific, Chennai.

Józef Lisowski, Technical University of Gdańsk: M.Sc. (1967), Ph.D. (1973), D.Sc. (1979); Gdynia Maritime Academy: Prof. (1990). Specializing in ship electro-automation, control theory application in marine technology, synthesis of computer algorithms of safe ship control in collision situations using artificial intelligence and dynamic game theory methods. He has published 6 books and 226 refereed journal and conference papers. Currently Józef Lisowski is a professor and the head of the Department of Ship Automation at Gdynia Maritime University.

Received: 10 March 2012
Revised: 20 May 2012
Re-revised: 10 November 2012