Jolanta KOSZELEW[1], Piotr WOŁEJSZA[2]

Białystok University of Technology (1), Maritime University of Szczecin (2)

doi:10.15199/48.2015.02.07

Anticollision manoeuvre optimization in the NAVDEC system

***Abstract**. Authors presented the concept of improved time optimization algorithm of anticollision manoeuvre to be implemented in the NAVDEC system. The algorithm will play an important role in supporting the navigator in decision making process. Moreover implementation of the algorithm will reduce significantly fuel consumption. In conclusion, safety and economic efficiency of shipping will be improved.*

***Streszczenie.** Autorzy przedstawili ulepszony algorytm optymalizacji czasu manewru antykolizyjnego, który zostanie zastosowany w systemie NAVDEC. Algorytm odegra istotną rolę w procesie wspomagania decyzji nawigatora. Ponadto, implementacja algorytmu zredukuje znacznie zużycie paliwa. Reasumując, poziom bezpieczeństwa oraz efektywność ekonomiczna zostanie poprawiona. (**Optymalizacja manewru antykolizyjnego w systemie NAVDEC**).*

Keywords: *decision support, anticollision, optimization, NAVDEC.*
Słowa kluczowe: wspomaganie decyzji, zapobieganie kolizjom, optymizacja, NAVDEC.

Introduction

Supporting the navigator in making decisions may significantly enhance the safety and effectiveness of the transport process. The navigational decision support system is to supplement the shipborne navigational equipment, while in the future it may be a part of Integrated Bridge System (IBS). The correct operation of the system requires co-operation with other devices and systems onboard ship and the external ones in order to acquire navigational information automatically. Apart from the presentation of navigational situation and information, the basic functions of the decision support system include: 1) navigational situation analysis; 2) solving collision situations; 3) interaction with the navigator via a user/operator interface. For a system to be effective and practically used, it has to present selected information in a readable manner and to be user-friendly. In this connection, a lot of attention is paid to the issues connected with building a proper user interface. One important requirement is the compatibility with present standards of information presentation in navigational information systems[1], [2].

NAVDEC – Navigational Decision Support System

The known navigational systems in use and methods of navigational decision support perform information functions and as such are helpful in the process of safe conduct of a ship. However, none of these known systems provides a navigator with ready solutions of collision situations taking account of all the vessels in the proximity of own ship, where the Collision Regulations [3] apply. Another shortcoming of these systems is that they do not explain the assessment of a navigational situation and proposed manoeuvre parameters.

Systems presently used on ships are information systems only. NAVDEC is not only information system. It's also decision support system.

Developed at the Maritime University of Szczecin NAVDEC system is a navigation tool that performs alongside providing information typical tasks for decision support systems. NAVDEC is an important complement to navigational equipment of the ship. Is a real-time system operated by the navigator. Its proper functioning requires interaction with devices and systems on the ship. The standard configuration of the ship include: log, gyrocompass, radar, echo sounder, ARPA, GNSS (Global Navigational Satellite System), such as GPS (Global Positioning System) or DGPS (Differential Global Positioning System). In addition, AIS, ECDIS, GNSS. In the version being developed following sources of information are in use: log, gyrocompass, radar / ARPA, GPS and DGPS, AIS and ENC (Fig.1).

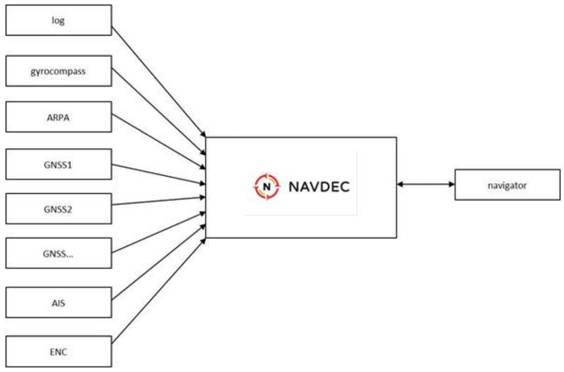

Fig. 1. Data sources for decision supporting system [4]

Actual optimization in the NAVDEC system

The purpose of optimization is the choice of minimally-time manoeuvre from all available solutions, leading to pass other targets at presumed CPA (Closest Point of Approach). To achieve this goal, the following optimization criteria were used in NAVDEC: energy optimization, time optimization.

First step of the actual optimization method involves the determination of the courses leading to pass at presumed distance (*Safe_Courses* procedure). The courses are calculated on the basis of [5] for each pair: the own ship (number 1) and the target ship *i* (for *i* = 2 to *n*, where *n* is the number of target ship).

Input data:

- position (*x*1, *y*1), speed (*V*1) and course over ground (*KDd*1) of the own ship,
- position (*xi*, *yi*), speed (*Vi*) and course over ground (*KDdi*) of target,
- CPA – safe passing distance set up by navigator

Output data:

< $gamma_i$1, $gamma_i$2 >, <$gamma_i$3, $gamma_i$4> - sectors of safe courses for pair: the own ship and the target ship number *i*.

Safe_Courses(i):

```
{
xwz=xi-x1; ywz=yi-y1; vxwz=vxi-vx1;
vywz=vyi-vy1;
vw=sqrt(vxwz*vxwz+vywz*vywz);
D=sqrt((xwz*xwz+ywz*ywz)^2);
Adcpa1=(xwz*ywz + CPA * sqrt(D^2-(CPA)^2))
/(xwz*xwz - (CPA)^2);
Adcpa2=(xwz*ywz – CPA * sqrt(D^2-(CPA)^2))
/(xwz*xwz -(CPA)^2);
vxi=Vi*sin(KDd2); vyi=Vi*cos(KDdi);
Bdcpa1=Adcpa1*vxi-vyi;Bdcpa2=Adcpa2*vxi-vyi;
```

```
gamma_i1=2*atan((Adcpa1*V1+sqrt((Adcpa1*Adcpa
1+1)*V1*V1-Bdcpa1*Bdcpa1))/(Bdcpa1-
V1))*180/pi;
gamma_i2=2*atan((Adcpa1*V1-
sqrt((Adcpa1*Adcpa1+1)*V1*V1-
Bdcpa1*Bdcpa1))/(Bdcpa1-V1))*180/pi;
gamma_i3=2*atan((Adcpa2*V1+sqrt((Adcpa2*Adcpa
2+1)*V1*V1-Bdcpa2*Bdcpa2))/(Bdcpa2-
V1))*180/pi;
gamma_i4=2*atan((Adcpa2*V1-
sqrt((Adcpa2*Adcpa2+1)*V1*V1-
Bdcpa2*Bdcpa2))/(Bdcpa2-V1))*180/pi;
}
```

We assume, for simply, that we get as results exactly four angles in the above algorithm for each *i*=2 to *n*. We have to run the above *Safe_Courses*(*i*) procedure for each pair: the own ship and the target ship number *i* (for *i*=2 to *n*) and as the result we get all safe sectors < $gamma_i1$, $gamma_i2$ >, <$gamma_i3$, $gamma_i4$>.
Next, we execute the *Common_Safe_Sectors* procedure for all target ships as the angle intersections of all safe sectors < $gamma_i1$, $gamma_i2$ >, <$gamma_i3$, $gamma_i4$> (for *i*=2 to *n*). The details of step one can be found in [7]. Let's denoted by $gamma_j$ elements of common safe sectors.

The first phase of optimization step is energy optimization (*Energy_Optimization* procedure). It's lead to choose new courses, from obtained solutions, which requires the smallest deviation (Z) from present own ship trajectory:

Energy_Optimization:

```
{
       for each gamma_j {
          Z_j = KDd1 ± gamma_j;
          Z=min(Z_j);
        }
       return Z;
}
```

This way we received maximum two solutions which guarantee to pass with other targets on presumed CPA.

In this case, own ship course is 280^0. As a result of calculation, we received two sectors of safe courses, which lead to pass with other targets on presumed CPA (1 Nm):
- starboard sector 335.70^0 – 234.59^0,
- port sector 236.63^0 – 257.06^0.

Following solutions, which requires the smallest alteration of course, were chosen:
- alter course to port by 22.93^0 on course 257.06^0,
- alter course to starboard by 55.70^0 on course 335.70^0.

The second step of the optimization is a time optimization (*Time_Optimization* procedure). At the beginning distances from present position (X_1, Y_1) to the closest waypoint (X_f, Y_f) through point (XS_k, YS_k) were calculated. Point (XS_k, YS_k) is the place, where ship alter course to the closest waypoint. Course alteration is exectuted when $TCPA_{fs}$ -time to closest point of approach to the farest ship equals zero. From point (X_1, Y_1) to point (XS_k, YS_k) the own ship can apply on the $gamma_j$ courses which is choosen as results of the energy optimization. Following algorithm was used to calculate the time of the course alternation.

Time Optimization

```
{
 calculate TCPA_fs;
 d_1 =  V_1*TCPA_fs;
 for each gamma_j
 {
  XS_k = d_1*sin(gamma_j) +X_1;

  YS_k = d_1*cos(gamma_j) +Y_1;

  d_2 = sqrt((X_f-XS_k)^2+(Y_f-YS_k)^2);

  t_j = TCPA_fs + d_2/V_1;
  t=min(tj);
 }
   returnt;
}
```

The shortest time solution is marked on Fig. 2. It fulfills safety criteria and enables the quickest attainment of the closest waypoint.

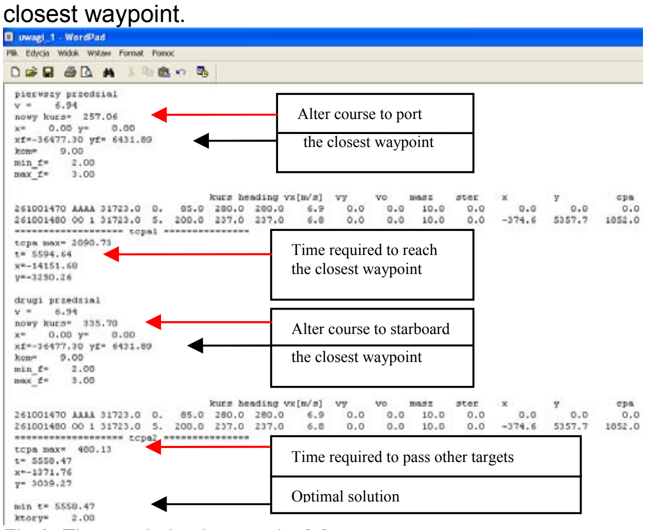

Fig.2. Time optimization results [7]

As an energy optimization, two solutions were chosen:
- alter course to port by 22.93^0 on course 257.06^0,
- alter course to starboard by 55.70^0 on course 335.70^0.

For both solutions times required to execute whole manoeuvre were calculated. Navigator received following advice: alter course to starboard, because total time is shorter (Fig. 2).

Improved optimization

The authors observe that the time optimization step, introduced in the previous section, can be improved. In actual version of this second step of the optimization, the own ship is too long on the chosen course. It is on this course as the long as $TCPA_{fs}$ – TCPA for the farthest ship, will equal zero. We propose new version of the optimization which is based on the assumption that the own ship change the course to the closest way-point, at the earliest position for which TCPA is greater than zero for at most one target ship. We introduce below the algorithm for the new improved version of the time optimization procedure.

Designations:
c_m – own ship course maneuver
c_r – own ship return course
TCPA[*i*] – TCPA for *i*-th target ship (*i* = 2..*n*)

t_p – time availability of targets data

Assumptions:

- We consider only collision situations on open waters with good visibility.
- We assume, that there is no wind and no waves.
- There is no risk of close quarters situation with each target ship.
- It's possible to generate the manoeuver for all ships in the fourth meeting phase, the latest [7]
- Only course manoeuvers are possible.
- We can receive information about positions (xi, y_i), speed (Vi) and course ($KDdi$) for each ship i=1..n, after each t_p seconds (in practice t_p is equal to few seconds)
- Each target ship can change the course during the maneuver of the own ship.

Input data:

- position ($x1$, $y1$), speed ($V1$) and course over ground ($KDd1$) of the own ship,
- position (xi, yi), speed (Vi) and course over ground ($KDdi$) of target,
- CPA – safe passing distance set up by navigator.

Improved_Optimization procedure works as follows:
To the line (1) we determine the new course for the own ship taking into account the energy optimization. This first part of the optimization process is not different from the current version which is realized in NAVDEC. From the line (1) works the new version of the time optimization procedure. After each t_p seconds we get the actual information about each ship such as the actual position, the speed and the course. Next, we calculate the actual safe courses sectors and TCPA for each target ship and run *Common_Safe_Sectors* procedure. The most important fragment of the improved version of the time optimization algorithm is selected in the frame (begins in the (2) line) We determine in this step the earliest position (XS_k, YS_k) on the own ship trajectory defined by its course manoeuvre, for which TCPA[i] > 0 for at most one target ship i (i = 2..n). On the base of this position the return course c_r to the closest way-point (X_f, Y_f) is determined. The loop which is continued as long as the own ship return trajectory defined by the return c_r intersects at most one domain of target ships. The time which stops the loop is the result of the improved optimization algorithm.
Thanks to this that we use dynamic approach to determination of (XS_k, YS_k) point, the return distance to the nearest way-point may be significantly shorter. This effect can result in significant fuel savings. We consider this savings in the next section. Moreover, the optimization procedure currently implemented in NAVDEC assumes that all target ships don't change the course and speed during the anti-collision manoeuvre realization. In the *Improved_Optimization* procedure these parameters can be varied.
In the Fig.3a Fig.3b main differences between the current and the improved version of the time optimization algorithm are illustrated. We see in Fig. 3b that the improved version of the optimization choose the (XS_k, YS_k) point 2 minutes earlier than in the actual version.

Improved_Optimization

```
{
  for i=2 to N
 {
    Safe_Courses(i);
    determine TCPA[i];
 }
determine Common_Safe_Sectors;
Z = Energy_Optimization;
c_m= change KDd1 with Z deviation;
(1)t=0;
 {
      for (i=1 to n) get information about:
      actual(xi,y_i), Vi oraz KDdi;
      for i=2 to n
       {
          Safe_Courses(i);
          determine TCPA[i];
       }
   determine Common_Safe_Sectors;
(2)
   (XS_k, YS_k) = the earliest position on the
own ship
                 trajectory defined by c_m,
                 for which TCPA[i] > 0 at most
one i=2..n;
 c_r = course from (XS_k, YS_k) to  (X_f, Y_f);
 t=t+t_p;
} while ( the own ship trajectory defined by
c_r
                 intersects at most one
               domain of ships
                i=2 ..n);
return t;
}
```

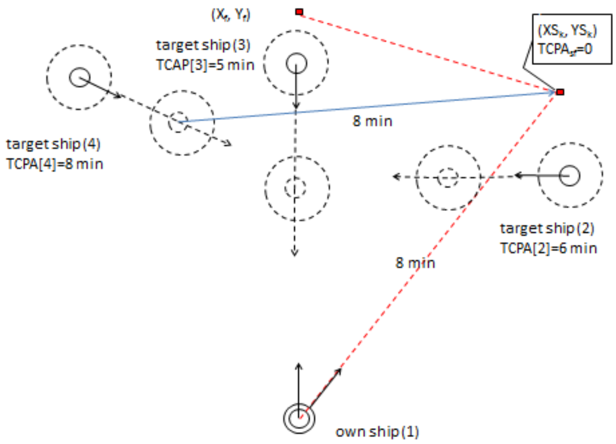

Fig. 3a. Actual time optimization example (head on orientation). Return course c_r starts after 8 minutes

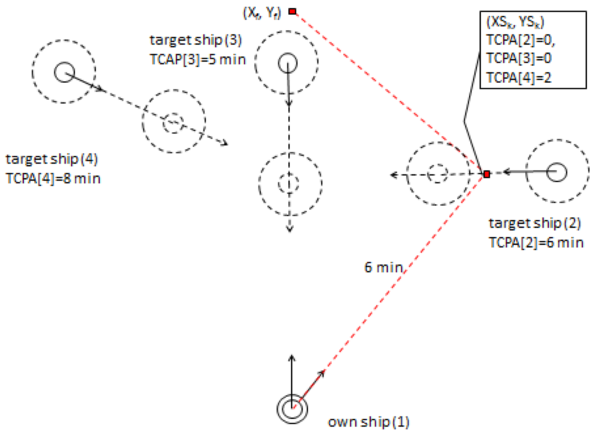

Fig.3b. Improved time optimization example (head on orientation). Return course c_r starts after 6 minutes

Conclusion and future work

Implementation of new optimization method enables to reduce fuel consumption. Instead of 5558 seconds in previous option, the manoeuvres based on new method took only 5503 seconds. Instead of 38603 meters, own ship had to pass 38191 meters. Daily fuel consumption of medium size container ship (8.000TEU-Twenty Feet Equivalent Unit) is around 260 tons. The manoeuvre shorter by 55 seconds gave total savings of 170 liters of fuel. Taking into account that such manoeuvre is executed many times per day, fuel savings are significant.

Authors presented the concept of improved time optimization algorithm of anticollision manoeuvre. This concept involves a number of simplifications for the meeting stages, which have a place in the decision making process. In addition, implementation of the algorithm in the NAVDEC system will require taking into account the limitations of restricted area during the manoeuver and moving away from the assumption that ships can only manoeuver by course and not by course and speed. Besides, the key point of the proposed algorithm, selected by the frame will require the use of the heuristic method with the low time complexity. In the first step authors will attempt to apply solutions based on game control [6], genetic algorithms described in [8] and [9] or evolutionary algorithms described in [10]. Exit condition of the loop will also require the use of fast solutions in the field of computational geometry.

Acknowledgement

The authors gratefully acknowledge support from Polish Ministry of Science and Higher Education at the Maritime Academy in Szczecin (grant 1/S/ITM/2013) and Bialystok University of Technology (grant S/WI/1/2014).

REFERENCES

[1] Pietrzykowski Z., Magaj J., The problem of route determination in ship movement in a restricted area. *Annual of Navigation* 19, part 2, 2012, 53–69.

[2] Pietrzykowski Z., Fuzzy Control in Solving collision Situations at Sea. *Computational Intelligence: Methods and Applications*, Eds. L. Rutkowski, R. Tadeusiewicz, L.A. Zadeh, J. Żurada, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2008, 103–111.

[3] COLREGs 1972, Convention on the international regulations for preventing collisions at sea, International Maritime Organization.

[4] Koszelew J., Wołejsza P., Last minute manoeuvre as a part of maritime transport logistic system, *Logistyka*, No 4, 2014.

[5] Lenart A., Manoeuvring to required approach parameters – CPA distance and time. *Annual of Navigation* No 1, Gdynia, 1999, 99-108.

[6] Lisowski J., The sensitivity of computer support game algorithms of a safe ship control. *International Journal Applied Mathematics and Computer Science*, Vol. 23, No 2, 2013, pp. 10.

[7] Wołejsza P., Multiagent decision support system in collision situation, dissertation, Maritime University of Szczecin, 2008.

[8] Koszelew J., Ostrowski K., A genetic algorithm with multiple mutation which solves orienteering problem in large networks, *Computational Collective Intelligence* - Berlin : Springer-Verlag, 2013, 356-365.

[9] Szłapczyński R., Szłapczyńska J., On evolutionary computing in multi-ship trajectory planning, *ApplIntell* (2012) 37:155–174, DOI 10.1007/s10489-011-0319-7

[10] Śmierzchalski R., Michalewicz Z.: Modelling of a ship trajectory in collision situations at sea by evolutionary algorithm. *IEEE Transaction on Evolutionary Computation*, Vol. 4, No. 3, 2000, 227–244.

***Authors**: dr Jolanta Koszelew, Politechnika Białostocka, Wydział Informatyki, ul. Wiejska 45a, 15-351 Białystok, E-mail: j.koszelew@pb.edu.pl; dr inż. Piotr Wołejsza, Akademia Morska w Szczecinie, Wydział Nawigacyjny, ul. Wały Chrobrego 1-2, 70-500 Szczecin, E-mail: piotr@am.szczecin.pl.*