Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 96 (2016) 1022 – 1031

Procedia Computer Science

20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

A trajectory base method for ship's safe path planning

Agnieszka Lazarowska[a,*]

[a]*Gdynia Maritime University, 81-87 Morska St., 81-225 Gdynia, Poland*

Abstract

The paper introduces a new algorithmic approach for ship's safe path planning. A new method utilizes a database containing a set of trajectories. An algorithm searches the trajectory base in order to find a safe trajectory that meets the optimality criterion (criteria) selected by the system operator. The method presented can be applied in a navigation Decision Support System. The method can also be adopted for applications in other environments, where similar problem occurs - in navigation of mobile robots or aircrafts. It should be emphasized that dynamic properties of an own ship are included in the calculations.

The paper describes assumptions and constraints taken into account in the solution construction process, a presentation of the Trajectory Base Algorithm (TBA) Decision Support System (DSS) and the explanation of the algorithm operation principle. A successful application of the method proposed is confirmed by simulation tests carried out with the use of MATLAB environment. The results of calculations with the use of implemented algorithm are presented and compared with solutions received by an application of an approach based on a heuristic method - the Ant Colony Optimisation. The outcome and achievements are summarized in the conclusions.

Peer-review under responsibility of KES International

Keywords:
Automation; Decision Support System; Path Planning; Safe Ship Control; Ship Navigation; Trajectory Base;

---

1. Introduction

A Decision Support System (DSS) is an advisory system that aids an operator in the decision-making process. Technological development stimulates the growth of different kinds and miscellaneous applications of DSSs. Types of DSSs employ genetic algorithms, fuzzy sets, agents and computer vision techniques. The scope of DSS applications is also very wide and continues to grow dynamically. Among exemplary applications medical diagnosis, business and management, industrial control, planning and navigation in robotics and text analytics can be mentioned. A more detailed presentation of different intelligent DSSs was introduced by Kaklauskas[1]. Here a DSS enables integration of various sources of information from different sensors and uses miscellaneous Knowledge Engineering techniques such as expert systems, case-based reasoning systems and fuzzy logic to determine optimal decisions[2]. The ship's

---

[*] Corresponding author. Tel.: +48-58-558-64-71 ; fax: +48-58-558-64-45.
E-mail address: a.lazarowska@we.am.gdynia.pl

Peer-review under responsibility of KES International
doi:10.1016/j.procs.2016.08.118

safe path planning constitutes such a complex process, with a large amount of data to analyze, derived from different navigation aids.

This paper presents a developed DSS, the purpose of which is to help in a process of ship's collision avoidance. The aim of the navigation DSS is to provide an advice to a navigator about the collision avoidance action (or a sequence of actions - a ship's safe trajectory) that should be taken in order to solve the actual navigation situation. A similar problem of safe path planning occurs in mobile robots navigation[3], navigation of aircrafts, Unmanned Aerial Vehicles (UAVs) and other collision avoidance applications.

Over the years many different approaches to solving that problem have been developed. Among these proposals a knowledge-based method has been introduced by Coenen et al.[4]. In order to create a navigation DSS, the author proposes the creation of a knowledge base that is developed using International Regulations for Preventing Collisions at Sea (COLREGs). Knowledge engineers will interpret these rules, including good seamanship practice, and based upon that will develop a proper knowledge representation in the form of if-then rules. Knowledge management is an important issue in a navigation DSS, because the system has to be developed capturing the knowledge of Subject Matter Experts (SMEs)[5,6] (navigation expert) in order to mimic the human-like decision making capability.

A similar approach of a collision avoidance expert system for Integrated Navigation Systems has been presented by Yang et al.[7]. Other approach to a ship's collision avoidance system, making use of a knowledge base, has been introduced by Lee and Kim[8]. In this method a knowledge-based system is linked up with a Candidate Sector set Selection Module, which uses a polar histogram to calculate candidate sectors. A candidate sector is a sector satisfying optimality and safety of the solution - the ship's trajectory. Other Computational Intelligence (CI) and Artificial Intelligence (AI) methods for a navigation DSS include Ant Colony Optimisation (ACO)[9,10], neural networks presented by Ahn et al.[11] and Simsir et al.[12], differential games[13], branch-and-bound method, dynamic programming and genetic algorithm[14].

Despite these many proposals, the issue has not been definitely solved so far. Complexity of the problem makes it difficult to develop a solution taking into account all of the demands and constraints, such as multi-ship encounters, presence of static obstacles, manoeuvres of other ships, an own ship mission profile, weather conditions and dynamic properties of a ship. The restrictions that have to be taken into account in order to achieve a proper problem solving capability of the DSS are presented in Fig. 1. The purpose of this paper is to propose a navigation DSS utilizing a new method and aimed at elimination or at least reduction of the limitations previously introduced.

Nomenclature

tm	the time of the OS manoeuvre
x	longitude of the OS position
y	latitude of the OS position
D_j	the distance of the j-th TS from the OS
D_s	the safe distance of the TS from the OS
N	true north
N_j	the bearing of the j-th TS
OS	the own ship - the ship with the DSS
TS	the target ship - the ship that need to be avoided
V	the speed of the OS
V_j	the speed of the j-th TS
Ψ	the course of the OS
Ψ_j	the course of the j-th TS
$\Delta\Psi$	the course change of the OS

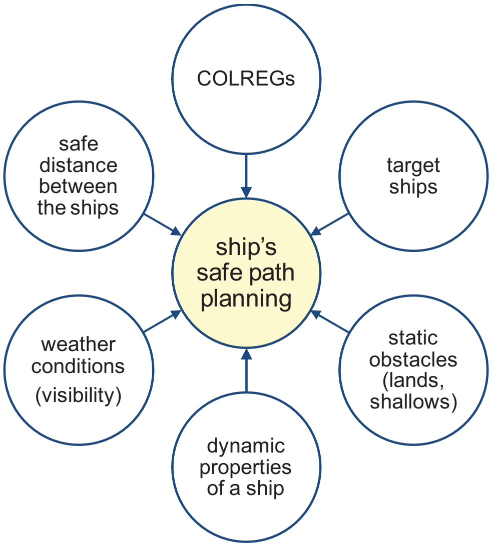

Fig. 1. Restrictions taken into account in a ship's safe path planning process.

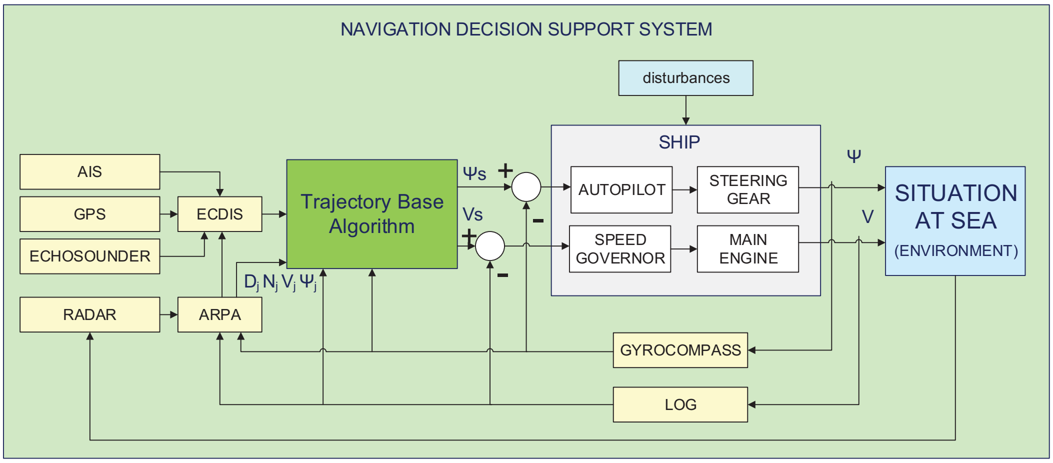

Fig. 2. The navigation DSS architecture.

2. The Trajectory Base Algorithm-based Decision Support System

The aim of the navigation DSS presented in this paper is the calculation of a course change manoeuvre or a speed reduction manoeuvre or a sequence of manoeuvres (a ship's safe trajectory) and a presentation of the computed solution to a navigator as an advice in his decision-making process. The navigation DSS architecture is presented in Fig. 2. The developed Trajectory Base Algorithm (TBA) DSS is composed of four modules: Data Input Module, Trajectory Base Module, Trajectory Base Algorithm Module and Solution Output Module as shown in Fig. 3. The task of the Data Input Module is to receive the data describing the actual navigation situation from different navigation aids such as a radar with an Automatic Radar Plotting Aid (ARPA), an Automatic Identification System (AIS), an Electronic Chart Display and Information System (ECDIS), a gyrocompass, a speed log, an echo sounder, a Global Positioning System (GPS) or a Differential Global Positioning System (DGPS), wind speed and direction sensors. These data include an own ship's course and speed, target ships' courses, speeds, bearings and distances from an own ship, information concerning position of static constraints (lands, islands, buoys, fairways, canals, shallows) and visibility/weather conditions. The data describing an own ship and target ship for an exemplary navigation situation are presented in Fig. 4.

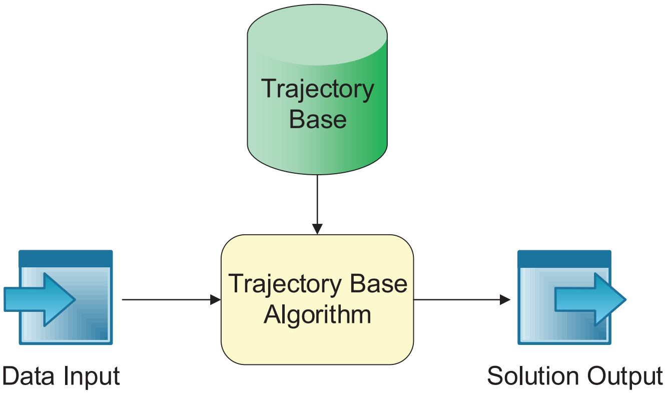

Fig. 3. The Trajectory Base Algorithm-based Decision Support System.

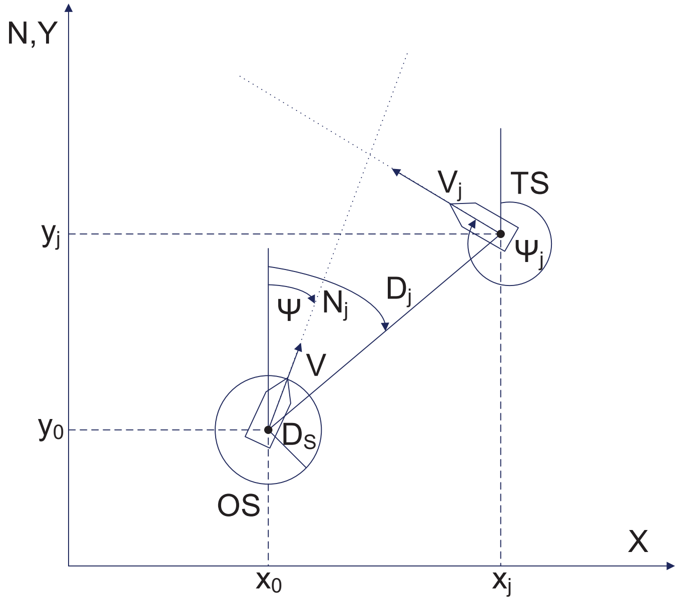

Fig. 4. An exemplary navigation situation.

The Trajectory Base Module is a database containing a set of candidate solutions (candidate trajectories). The current version of the database contains 6261 trajectories generated with the use of 34 rules. An exemplary rule is presented as Algorithm 1 and the trajectories generated with the use of this rule are shown in Fig. 5. The Trajectory Base Algorithm Module is responsible for the calculation of a solution. In the Knowledge-Based approach the solution is achieved by searching through the rule base defined in the form of if-then rules in order to find a solution appropriate for the considered navigation situation (a proper collision avoidance action). In the method proposed in this paper a TBA searches through the trajectory base instead of the rule base in order to find the best solution in terms of its safety and optimality. The Solution Output Module transmits a solution in a graphical and numerical form to the system operator. The transmitted data include a graphical presentation of a calculated ship's safe trajectory along with an own ship's course at every line segment of the trajectory, the length of the trajectory and the time of an own ship passage to the end waypoint.

Algorithm 1 Rule for generation of exemplary trajectories.

```
x=xs; y=ys;
for j:=2 to 9 do
  for k:=1 to 5 do
    trajectory(i)=[(x,y) (x+k,j) (x,y+ye)];
    i:=i+1;
  end for
end for
```

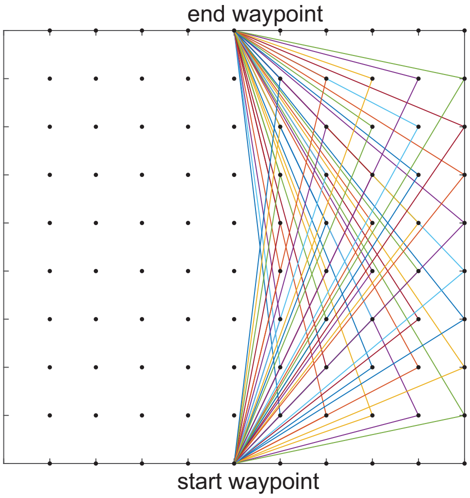

Fig. 5. Trajectories generated with the use of an exemplary rule.

3. The Trajectory Base Algorithm

Based upon the data received from the Data Input Module a relative course, bearing and speed of every target ship is calculated in relation to an own ship. Next, the first candidate trajectory is retrieved from the trajectory base. Candidate trajectories in the database are sorted according to the user's preferences. A system operator can decide whether he prefers to receive the shortest trajectory, the fastest trajectory or a compromise of both criteria. The candidate trajectory is then evaluated in terms of its compliance with static and dynamic restrictions.

The evaluation process begins with the division of the candidate trajectory into a number of steps. After that, every own ship movement from the current position to the next position (current step of an own ship movement along the trajectory) is checked whether it constitutes a course change manoeuvre. When the current step is a course alteration manoeuvre, dynamic properties of an own ship are applied in a form of a proper value of a manoeuvre time. The manoeuvre time is computed at the stage of the system application for a specified vessel and stored in a table. The manoeuvre time depends on a rudder angle, ship's speed and loading conditions. The manoeuvre time as a function of a course change for an exemplary vessel is shown in Fig. 6.

After application of a manoeuvre time, an own ship's and target ships' positions are calculated. If an own ship's and target ships' positions do not cause a collision, the next step of an own ship movement along the trajectory is checked against a collision occurrence. If a collision occurs, the candidate trajectory is rejected and the next candidate solution is retrieved from the database for evaluation. If the candidate trajectory fulfils all of the restrictions - at every step of an own ship movement along the trajectory its position does not intersect the static and dynamic obstacles - then this candidate solution becomes the final best solution and further calculations are interrupted. The algorithm terminates its operation and the solution is passed to the Solution Output Module. Every target ship is modelled with the use of a safety area called a ship's domain. The shape and size of the domain, in addition to the assurance of a safe distance between the ships, also enforces the COLREGs compliance of the solution. The flowchart of the TBA is presented in Fig. 7 and the pseudo-code is shown as Algorithm 2.

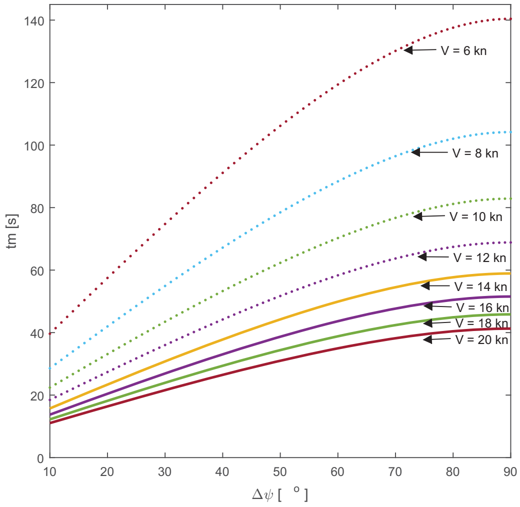

Fig. 6. The manoeuvre time as a function of a course change for an exemplary vessel - a 160 m long Mariner class vessel.

Algorithm 2 High-level pseudo-code for a Trajectory Base Algorithm.

```
for s:=1 to n do
  while (termination condition not met) do
    CheckSolutionFeasibility(s)
    if s ∈ AdmissibleSolutions then
      if $(f(s) < f(s_{best}))$ then
        $s_{best} \leftarrow s$
      end if
    end if
  end while
end for
return $s_{best}$
```

4. Simulation tests

The above described TBA was implemented in the MATLAB programming language and was tested extensively in order to evaluate its problem solving capability and the quality of results. The objective function is defined as the length of a trajectory. It is calculated according to Equation 1 as a sum of the lengths w of k line segments $(e_1, e_2, .., e_k)$ constituting a trajectory.

$$f(s) = \sum_{i=1}^{k} w(e_i) \rightarrow min \quad (1)$$

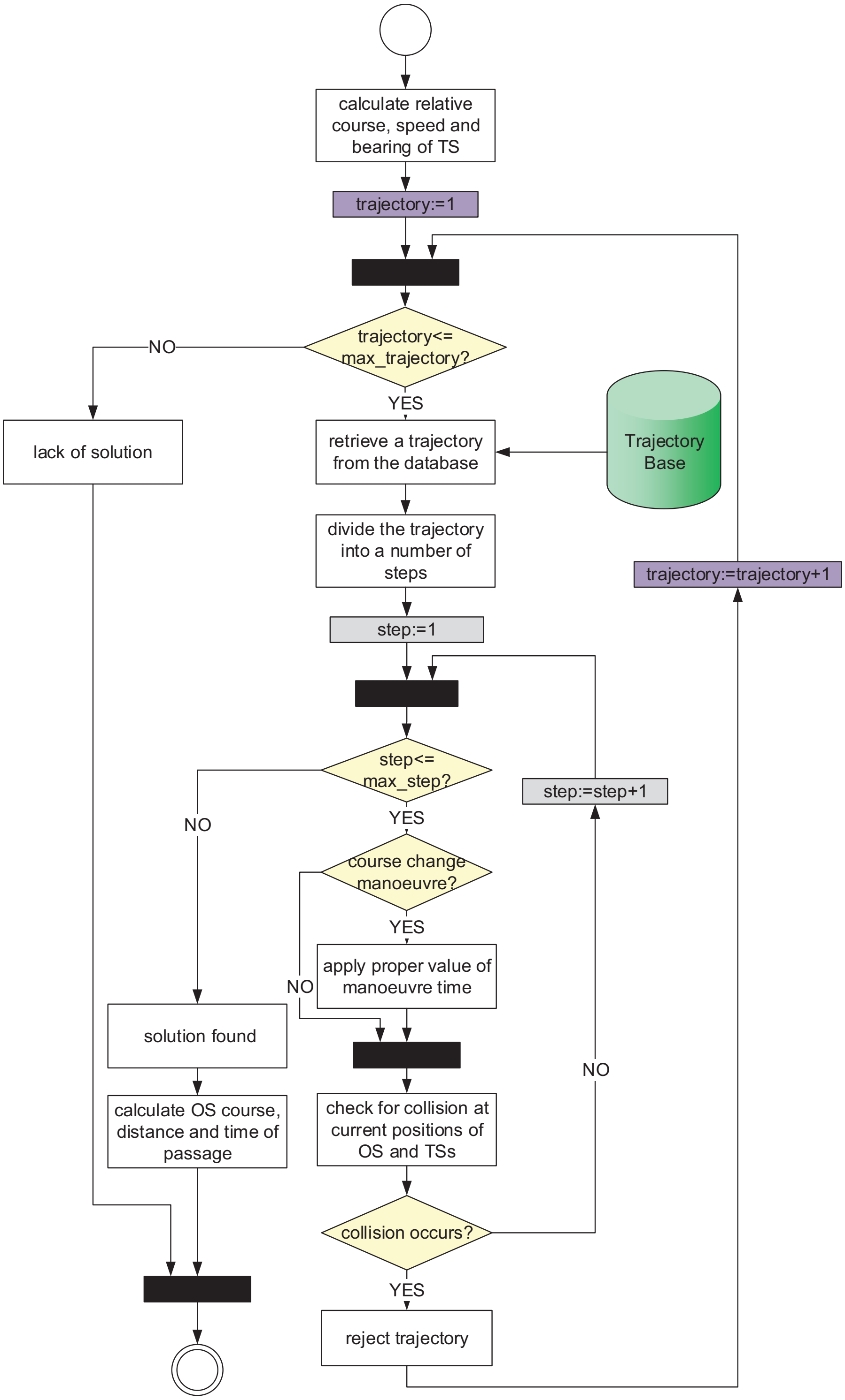

Fig. 7. A flowchart of the Trajectory Base Algorithm.

Two representative test cases were chosen for a presentation in this paper - an encounter situation with one ship and with four ships. The safe distance is assured by the application of a hexagon domain around a ship, shown in Fig. 8. The parameters marked in Fig. 8, defining the size of the hexagon domain, were set to the following values: A = 1 nm, B = 0.6 nm, C = D = 0.25 nm, E = 0.6 nm.

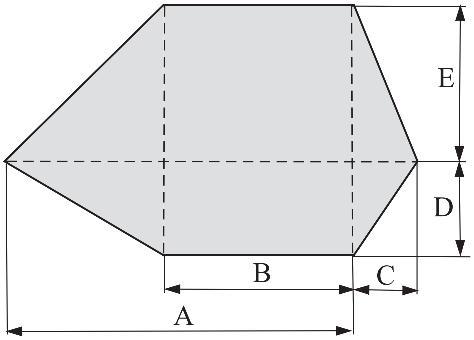

Fig. 8. A ship's hexagon domain.

The simulation tests were carried out for a Mariner class vessel with a length of 160.93 m as an own ship. The dynamic properties of this vessel were taken into account during calculations. The mathematical model of a Mariner class ship was acquired from the Marine Systems Simulator MATLAB toolbox presented by Fossen[15]. The solutions achieved with the use of the TBA were compared with the results returned by the ACO-based algorithm.

Table 1. Data of test case 1.

Ship	Ψ [°]	V [kn]	N [°]	D [nm]
OS	0	14	–	–
TS	180	12	0	10

Table 2. Results of test case 1.

Method	Length of trajectory [nm]	Δ Ψ [°]	Computational time [s]
TBA	9.22	14, 25	0.4
ACO	9.22	11, 25	19

Table 3. Data of test case 2.

Ship	Ψ [°]	V [kn]	N [°]	D [nm]
OS	0	16	–	–
TS1	90	16	315	8
TS2	315	17	45	4
TS3	180	17	355	6.5
TS4	0	15	180	3

Table 4. Results of test case 2.

Method	Length of trajectory [nm]	Δ Ψ [°]	Computational time [s]
TBA	9.4	18, 45, 27	1.6
ACO	10.1	16, 61	22

Test case 1 is an encounter between an own ship and one target ship. The initial data describing this situation are listed in Table 1. The solution obtained by the TBA is composed of two course alteration manoeuvres, the first one of 14 degrees and the second one of 25 degrees, while the trajectory computed by ACO-based algorithm also consists of two course change manoeuvres, but with values of 11 degrees and 25 degrees. The results received by both algorithms are similar, but the difference in computational time is significant in favour of the TBA. The results of this test case are presented in Table 2 and graphically in Fig. 9 (a).

Test case 2 is an encounter between an own ship and four target ships defined by data presented in Table 3. The solution calculated by the TBA consists of three course change manoeuvres, the first one of 18 degrees, the second one of 45 degrees and the third one of 27 degrees. The trajectory computed by ACO-based algorithm is composed of two course alteration manoeuvres with values of 16 degrees and 61 degrees. The ship's safe path returned by the TBA is shorter by 0.7 nm and its computational time is much shorter for this case as compared to that of ACO-based algorithm. The solutions for this test case are compared in Table 4 and graphically in Fig. 9 (b).

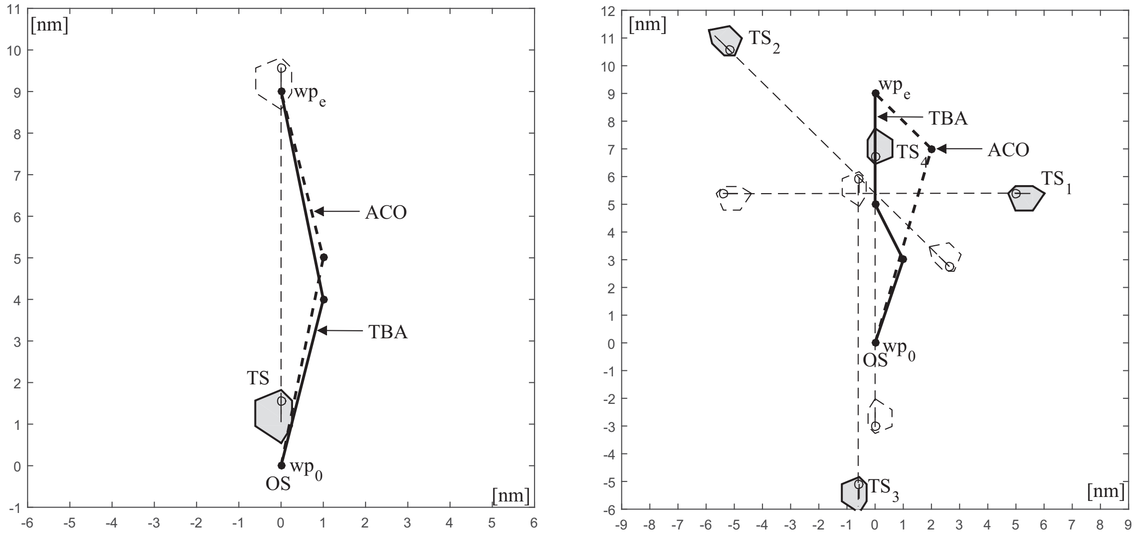

Fig. 9. (a) OS trajectory computed for test case 1; (b) OS trajectory computed for test case 2.

5. Conclusions

The paper presents a DSS developed to aid the user in a decision-making process in a dynamic environment. The aim of the developed navigation DSS is to achieve advancement of automation in the ship control process. The system needs to utilize the knowledge of SMEs (navigation experts) concerning application of COLREGs and collision risk assessment to achieve human-like decision making in a complex environment with many static and dynamic obstacles and big data to analyze from many sensors. The DSS needs to have an ability of returning repeatable decisions and results in a reasonable amount of time.

The navigation DSS utilizes a new algorithmic approach for solving the ship's safe path planning problem - the Trajectory Base Algorithm. In this method a database containing a set of trajectories is searched through in order to find the best solution - a safe trajectory that best fulfils the optimality criterion. The dynamic properties of a vessel are taken into account during calculations in the form of a manoeuvre time. The algorithm was implemented in the MATLAB programming language and simulation tests were performed to confirm the problem solving capability of the proposed method. The TBA performance was compared with the efficiency of one of the heuristic methods - the ACO-based algorithm. The received results proof that the TBA achieves better performance than the ACO-base algorithm with regard to the optimality of solution. The trajectories calculated by the TBA are shorter. The computational time of the TBA is significantly shorter than that attained by the ACO-base algorithm. The TBA DSS presented in this paper is applicable in commercial systems, because it takes into account the most important restrictions of the ship's safe path planning process, its operation principle is simple, it is able to return the solution in near real-time and the solution is repeatable for every run of calculations.

References

1. Kaklauskas A. *Biometric and Intelligent Decision Making Support*. 1st ed. Switzerland: Springer International Publishing; 2015.
2. Tweedale JW. A review of cognitive decision-making within future mission systems. *Procedia Computer Science.* 18th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems - KES2014. Elsevier; 2014;**Vol. 35**:1043-1052.
3. Cook G. *Mobile Robots: Navigation, Control and Remote Sensing*. 1st ed. New Jersey: Wiley-IEEE Press; 2011.
4. Coenen FP, Smeaton GP, Bole AG. Knowledge-based Collision Avoidance. *J Navigation* 1989;**42**:107-116.
5. Tweedale JW, Jain LC *Embedded Automation in Human-Agent Environment*. 1st ed. Springer-Verlag Berlin Heidelberg; 2012.
6. Tweedale JW, Jain LC. Advances in Knowledge-Based Information Systems. In: Tweedale JW, Jain LC, Watada J, Howlett RJ, editors. *Knowledge-Based Information Systems in Practice*. Springer International Publishing; 2015. p. 1-18.
7. Yang Ch, Kose K, Phan S, Kuo P. A Simulation-Based Procedure for Expert System Evaluation. In: Logananthara R, Palm G, Ali M, editors. *Intelligent Problem Solving. Methodologies and Approaches*. Berlin Heidelberg: Springer; 2003. p. 149-160.
8. Lee YI, Kim YG. A Collision Avoidance System for Autonomous Ship Using Fuzzy Relational Products and COLREGs. In: Yang ZR, Yin H, Everson RM, editors. *Intelligent Data Engineering and Automated Learning - IDEAL 2004*. Berlin Heidelberg: Springer; 2004. p. 247-252.
9. Lazarowska A. Ant Colony Optimization based navigational decision support system. *Procedia Computer Science.* 18th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems - KES2014. Elsevier; 2014;**Vol. 35**:1013-1022.
10. Tomera M. Ant Colony Optimization Algorithm Applied to Ship Steering Control. *Procedia Computer Science.* 18th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems - KES2014. Elsevier; 2014;**Vol. 35**:83-92.
11. Ahn JH, Rhee KP, You YJ. A study on the collision avoidance of a ship using neural networks and fuzzy logic. *Appl Ocean Res* 2012;**37**:162-173.
12. Simsir U, Amasyali MF, Bal M, Celebi UB, Ertugrul S. Decision support system for collision avoidance of vessels. *Appl Soft Comput* 2014;**25**:369-378.
13. Lisowski J. Computational Intelligence Methods of a Safe Ship Control. *Procedia Computer Science.* 18th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems - KES2014. Elsevier; 2014;**Vol. 35**:634-643.
14. Mohamed-Seghir M. The Branch-and-bound Method, Genetic Algorithm, and Dynamic Programming to Determine a Safe Ship Trajectory in Fuzzy Environment. *Procedia Computer Science.* 18th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems - KES2014. Elsevier; 2014;**Vol. 35**:348-357.
15. Fossen TI. *Handbook of Marine Craft Hydrodynamics and Motion Control*. 1st ed. John Wiley and Sons, Ltd.; 2011.