IAMU 2018 Research Project
(No. YAS201801)

Technology for Safer and Greener Shipping (Research on New Obstacle Avoidance Algorithms for Ships (NOAA))

By

Gdynia Maritime University (GMU)

August 2019

IAMU
International Association of Maritime Universities

This report is published as part of the 2018 Research Project in the 2018 Capacity Building Project of International Association of Maritime Universities, which is fully supported by The Nippon Foundation.

Published by the International Association of Maritime Universities (IAMU) Secretariat
Meiwa Building 8F, 1-15-10 Toranomon, Minato-ku,
Tokyo 105-0001, JAPAN
TEL : 81-3-6257-1812 E-mail : info@iamu-edu.org URL : http://www.iamu-edu.org

ISBN978-4-907408-30-5

IAMU
International Association of Maritime Universities

日本財団
THE NIPPON FOUNDATION

IAMU 2018 Research Project
(No. YAS201801)

Technology for Safer and Greener Shipping (Research on New Obstacle Avoidance Algorithms for Ships (NOAA))

By
Gdynia Maritime University (GMU)

Contractor : Gdynia Maritime University (GMU)
Research Coordinator : Agnieszka Lazarowska

Contents

Executive Summary

The research carried out in this project was focused on the development of new solutions and technologies for unmanned ships in order to achieve safer and greener shipping. The aim of the project was the development of a path planning module, constituting a part of an intelligent control system for ships - a Guidance, Navigation and Control (GNC) system. In order to achieve this goal, one of the main research objectives was the development of new, original, effective algorithms for the determination of a safe, optimal path for a ship in a collision situation at sea, applicable in the GNC system. Implemented algorithms were verified by simulation studies in the MATLAB environment. After that tests in real environment with the use of mobile platforms (in the laboratory conditions) and on board the research and training ship *Horyzont II* under operating conditions were carried out in order to further validate the developed methods.

For the purpose of ship's path planning and obstacle avoidance, three algorithms were implemented using a unified problem and environment representation. Two algorithms, the Discrete Artificial Potential Field (DAPF) algorithm and the Trajectory Base Algorithm (TBA), belong to the deterministic methods. The third algorithm, the Ant Colony Optimization (ACO) algorithm, is classified as a heuristic approach.

The developed algorithms were tested in the laboratory conditions. For this purpose a system composed on an Indoor Positioning System and mobile platforms was developed. Furthermore a system for the algorithms verification on board the ship was developed and implemented on board the ship *Horyzont II*. The system on board the vessel *Horyzont II* was tested during the ship's XLI voyage from 20.06.2018 to 20.07.2018 and XLII voyage from 31.08.2018 to 30.09.2018 to Spitsbergen. After that the navigational data registered on board the ship were analyzed and tests of the algorithms concerning collision situations with both static and dynamic obstacles, simple and complex encounters with navigational data registered on board the ship were conducted. The validation of algorithms included also verification of the trajectories calculated by the algorithms from the perspectives of different ships taking part in the considered encounter situations.

As a result of the project, it was found out that the Trajectory Base Algorithm proved to be more effective in terms of the optimality of solution and the run time of the algorithm for most of test cases, a heuristic Ant Colony Optimization algorithm could be useful for very complex situations, where the deterministic approach would have difficulties in finding a solution in a reasonable amount of time. Carried out simulation studies with real navigational data registered on board the ship *Horyzont II* confirmed applicability of implemented algorithms for solving the ship's trajectory planning problem. Carried out test with the use of mobile platforms proved feasibility of solutions calculated by the algorithms. Tests in real conditions carried out on board the ship *Horyzont II* showed that application of a hard drive of SSD type is required due to particularly difficult weather conditions during storms, which caused a damage to a hard drive of a HDD type used for data storage. Real data registration from the radar with ARPA proved that it would be needed to apply also data from AIS and ENC. Real data registration revealed also that an additional data filtering mechanism is needed, because sometimes some part of data can be incomplete. Analysis of achieved results of the algorithms verification (solutions feasibility, run time of the algorithms) enables to state that the developed ship's trajectory planning algorithms are suitable for application in a Decision Support System or a GNC system of an autonomous vessel and that their application will contribute to achieve safer and greener shipping.

Further works planned to be carried out include elimination of errors in the input data from the radar and continuation of tests on board the ship.

Research deliverables include:

1. Presentation of the Interim report at IAMU AGA 2018.

2. Lazarowska A., "Research On New Obstacle Avoidance Algorithms For Ships", *Proceedings of the 19th Annual General Assembly (AGA) of the International Association of Maritime Universities (IAMU)*, Barcelona, Spain, October 17-19, (2018), pp 434–441.

3. Lazarowska A., "Verification of Ship's Trajectory Planning Algorithms Using Real Navigational Data", *TransNav - The International Journal on Marine Navigation and Safety of Sea Transportation*, Vol. 13, No. 3, (2019), pp 559–564.

4. Lazarowska A., "Discrete Artificial Potential Field Approach to Mobile Robot Path Planning", *Proceedings of the 10th IFAC Symposium on Intelligent Autonomous Vehicle (IAV 2019),* Gdansk, Poland, July 3-5, *IFAC-PapersOnLine*, (2019), pp 277–282.

5. Lazarowska A., "Research on algorithms for autonomous navigation of ships", *WMU Journal of Maritime Affairs*, (2019), pp 1–18, https://doi.org/10.1007/s13437-019-00172-0.

1. Introduction

1.1 Background

According to Allianz Global Corporate & Specialty Safety and Shipping report published in 2018 [1], an annual review of ship's accidents and safety developments in shipping, 90% of cargo is transported by ships. There are over 50 000 merchant ships taking part in the international trade, while in 2017 94 ships were lost completely. According to the statistics 75% of ship accidents are caused by human error. Among the major risks busy seas and safety standards are listed. Ensuring the safety of a vessel is a critical issue in the maritime industry. The presented project was dedicated to that very important matter and its aim was to develop new solutions contributing to achievement of safer shipping.

Technological development stimulates the search process for new solutions leading to autonomous shipping. The Maritime Safety Committee (MSC), one of IMO's committees recently made a step towards the development of legal regulations in this area, by an announcement at its 99th session meeting in May 2018 that works will be done in order to address the safety, security and environmental factors of autonomous ship operations in the IMO conventions.

Recent trends within the topic of unmanned and autonomous ships include the development of a fully-electric 120 TEU autonomous container ship Yara Birkeland by Kongsberg Maritime AS [2] and the Wanshan Marine Test Field for unmanned vessels in Guangdong, China [3].

This project deals with the development of new algorithms applied for decision-making in a collision situation at sea, applicable in Decision Support Systems or in Autonomous Navigation Systems on board a ship. The aim of the presented research was the development of new solutions to achieve safer shipping and advancement in the autonomous navigation technology.

The structure of the report is as follows. Section 1 presents the problem definition, a diagram of the developed Decision Support system, description of the process constraints, requirements and assumptions. In section 2 the developed obstacle avoidance algorithms are introduced. Section 3 includes simulation studies of the developed algorithms. Section 4 is dedicated to the description of experimental studies results obtained with the use of mobile platforms, while in section 5 results of experimental studies on board the ship *Horyzont II* are presented.

1.2 Problem definition

The goal of the project is to develop and test new algorithms for ship's trajectory planning in a collision situation at sea. Safe ship control systems, containing navigational equipment such as logs, gyro-compasses, GPSs, radars with Automatic Radar Plotting Aids (ARPA), Automatic Identification Systems (AIS), Electronic Chart Display and Information Systems (ECDIS) and autopilots, enable implementation of various decision support methods. The safe control of a ship is a complex process, as it requires a detailed analysis of a large amount of information and make a manoeuvring decision under time pressure. Incorrect assessment of the current navigational situation based on the data from the navigational equipment can lead to a collision. The latest ARPA devices, such as for example the Kongsberg's K-Bridge ARPA radar, allow for manual or automatic tracking of up to 100 objects detected by radar. The device has a trial manoeuvre function, which enables for simulation of the manoeuvre planned by the navigator, but does not propose a safe course or speed change. The development of modern computational intelligence methods allows for their application in ship's safe trajectory planning algorithms.

1.3 Safe ship control system

The task of the safe ship control is the determination of safe controls, a safe course change and/or speed of the ship or a sequence of safe course changes and/or speed changes of the ship - a safe trajectory of a ship. This goal is accomplished by means of an appropriate control algorithm.

The input data to the algorithm, describing the current navigational situation include:
- the course and speed of an own ship,
- courses and speeds of target ships,
- bearings of target ships,
- distances of target ships from an own ship,
- data concerning placement of static navigational obstacles (lands, shallows).

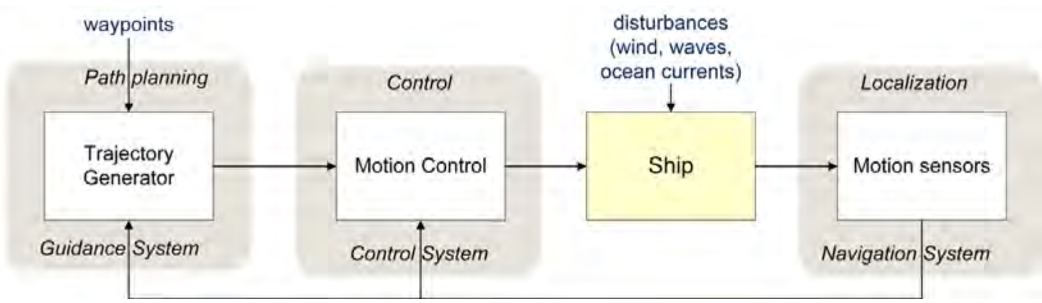


Fig. 1.1. Safe ship control system

The diagram of a safe ship control system, developed on the basis of [1], is shown in Fig. 1.1.

The task of safe ship control is accomplished by determining such control values, so that an own ship position does not exceed the static (lands) and dynamic constraints (target ships) during movement along the determined trajectory. The safe ship trajectory has to fulfill the following equation:

$$D_{jmin} = min(D_j(t)) > D_S \quad (1)$$

where the minimum distance between an own ship and a target ship has to be larger than the specified safe distance D_S. Dynamic obstacles are taken into account by using the ship's domain term. The ship's domain has been defined by E. M. Goodwin [5] as an area around the ship that the navigator wants to keep free from other ships and static obstacles.

Depending on the final conditions, two types of safe ship control tasks are distinguished, in which the purpose is to:

- avoid a collision and return to the set course for open waters,
- avoid a collision and return to the set end point of the trajectory for restricted waters.

1.4 Ship's trajectory planning method

Assumptions

Determining a ship's safe trajectory in a collision situation at sea is a complex process requiring consideration of many factors affecting correct assessment of the current navigational situation.

Factors that must be taken into account in this process are:

- compliance of the determined trajectory with The International Regulations for Preventing Collisions at Sea (COLREGs) [6] - clear, readable manoeuvres to the appropriate side;
- static navigation restrictions such as lands, fairways, shallows, buoys;
- dynamic navigation restrictions - target ships,
- dynamic properties of an own ship;
- returning to the appropriate end point (for restricted waters);
- the method of obtaining information about the current navigational situation (ARPA, AIS).

In the ship's safe trajectory algorithms the following assumptions are taken into account:

- static and dynamic obstacles;
- determining the trajectory to the specified end point;
- kinematic model of the process;
- taking into account the dynamic properties by the use of the manoeuvre time;
- data concerning target ships from the ARPA system;
- manoeuvres fulfilling the COLREGs;
- hexagon domains around target ships;
- run time of the algorithm not exceeding sixty seconds,
- repeatable results for every run of calculations with the same input data;
- the strategies of the target ships not taken into account.

The trajectory of an own ship is determined to a specified end point, what makes it possible to apply this solution to navigational situations in restricted waters, where, due to the limited manoeuvring area, the majority of collision situations takes place. It was assumed that data concerning target ships such as courses, speeds, bearings and distances from an own ship will be obtained from the ARPA system. The algorithm used in the safe ship control system must be characterized by the determination of a repetitive solution in near-real time. In this work it was assumed that the run time of the algorithm up to sixty seconds is fulfilling the above mentioned criterion. It was decided not to take the changes in the strategy of the target ships into account. It was assumed that when a change in the strategy of the target ship will be detected, the safe trajectory of an own ship is calculated again based on new input data defining the current navigational situation. This is not a limitation, because the trajectory of an own ship can be repeatedly calculated in a very short time.

The ship's safe trajectory planning algorithm has to be characterized by:

- safety
- efficiency
- reliability
- near-real run time
- simplicity.

The purpose of the safe ship control process is to determine such a course and/or speed change of an own ship or a sequence of course and/or speed changes that will assure avoiding a collision with target ships and static obstacles. This task is achieved by the implementation of a safe ship control algorithm.

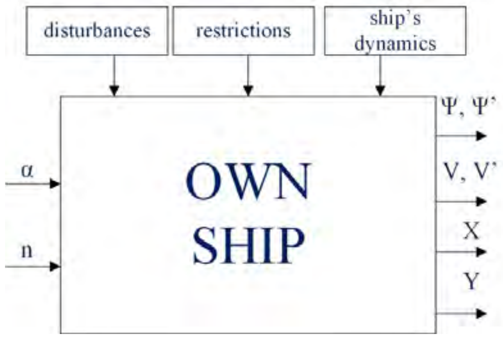


Fig. 1.2. Safe ship control process

A block diagram of the safe ship control process in a collision situation at sea is shown in Fig. 1.2, where α is the rudder angle, n is the rotational speed of the propeller, Ψ is the course of an own ship, Ψ' is the angular speed of an own ship, V is the speed of an own ship, V' is the acceleration of an own ship, X is the latitude of the own ship's position and Y is the longitude of the own ship's position.

Navigational environment has boundaries defined with the use of equation (2) and consists of a free area Ω_{free} and area occupied by navigational obstacles Ω_{obs} - equation (3). The area of navigational obstacles includes space occupied by static obstacles Ω_{stat} and dynamic obstacles, changing their position in time $\Omega_{dyn}(t)$ - equation (4).

$$\Omega = \{(X, Y) \in R^2 : k \leq x \leq l, m \leq y \leq n\} \quad (2)$$

$$\Omega = \Omega_{free} \cup \Omega_{obs} \quad (3)$$

$$\Omega_{obs} = \Omega_{stat} \cup \Omega_{dyn}(t) \quad (4)$$

Static constraints, which include lands, shallows, channels, waterways, are modelled as concave and convex polygons. An example of the land area representation in the safe ship control algorithm is shown in Fig. 1.3.

Dynamic constraints are modelled using the ship's domain. The developed methods use a hexagon-shaped domain. The size and shape of the domain enforces compliance of the calculated own ship trajectory with COLREGs.

An extension of the ship's domain in the bow direction causes in a crossing situation (rule 15 of COLREGs) passage of an own ship behind the stern of the target ship. Enlargement of the ship's domain to the starboard side will force in the head-on situation (rule 14 of COLREGs) changing the course of an own ship to the starboard side.

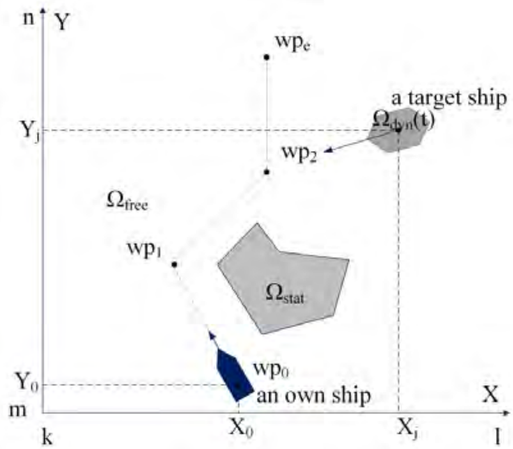


Fig. 1.3. Model of the navigational environment in a collision situation at sea

The ship's trajectory is composed of a number of waypoints – Fig. 1.3. Each waypoint is defined by geographical coordinates: latitude - *X* and longitude - *Y*.

Solving the problem of collision avoidance at sea is based upon calculation of a ship's safe trajectory from the current position of an own ship to the next waypoint with the smallest possible deviation from the current trajectory. A safe, optimal trajectory of an own ship T_{opt} is described by equation (5) and consists of $(e + 1)$ states from the initial state wp_0 to the final state wp_e.

$$T_{opt} = \{wp_0 = [X_0, Y_0, \psi_0, V_0], wp_1 = [X_1, Y_1, \psi_1, V_1], \ldots, wp_e = [X_e, Y_e, \psi_e, V_e]\} \quad (5)$$

Fig. 1.4 shows an exemplary encounter situation with one target ship with marked parameters characterizing a collision situation at sea, such as:

- the course and speed of an own ship,
- the course and speed of the jth target ship,
- the bearing on the jth target ship,
- the distance of the jth target ship from an own ship.

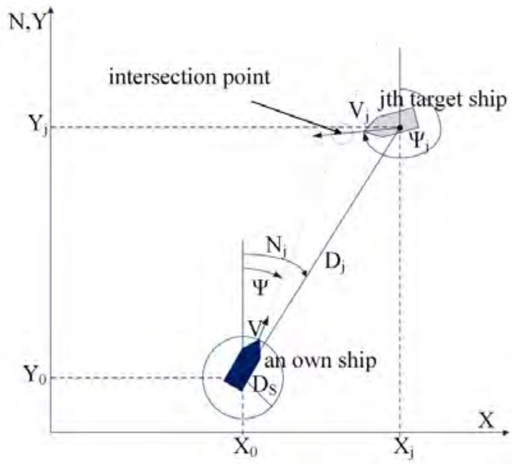


Fig. 1.4. An encounter situation of an own ship with jth target ship

It was assumed that target ships do not change their courses and speeds. The collision avoidance process at sea was described with the use of the kinematic model of an own ship and target ships movement, where state and control variables are defined as follows: $x_1 = X$, $x_2 = Y$, $x_{2j+1} = X_j$, $x_{2j+2} = Y_j$, $u = \Psi$, where $j = 1,2, ..., m$ - number of target ships. The kinematic model of an own ship and target ships motion in the form of the process state equations is described by equations (6).

$$
\begin{gathered}
\dot{x}_1 = V \cdot \sin u(t) = V \cdot \sin \Psi(t) \\
\dot{x}_2 = V \cdot \cos u(t) = V \cdot \cos \Psi(t) \\
\dot{x}_{2\cdot j+1} = V_j \cdot \sin \Psi_j(t) \\
\dot{x}_{2\cdot j+2} = V_j \cdot cos\, \Psi_j(t)
\end{gathered} \tag{6}
$$

The dynamic properties of an own ship are taken into account at the stage of the determined trajectory assessment by the use of the manoeuvre time parameter.

The project has been divided into the stages presented in Fig. 1.5.

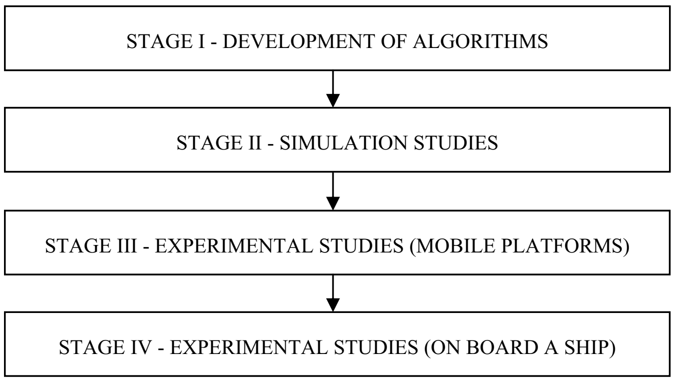


Fig. 1.5. Stages of the project

2. Stage I – development of algorithms

2.1 Aim

The aim of this stage of the project was the development of advanced optimization algorithms for application in a safe ship control systems, known as a Guidance, Navigation and Control (GNC) system. The structure of such system is shown in Fig. 1.1. The GNC system is composed of three main subsystems: the Guidance System, the Control System and the Navigation System. The Guidance System is responsible for path planning. The Control System calculates appropriate control forces in order to steer the ship along the path determined by the Guidance System. The Navigation System measures motion parameters of the ship (position, course and speed). The main component of the Guidance System is the Trajectory Generator (TG), which calculates a safe, optimal path for a ship with the use of an appropriate algorithm.

2.2 Literature review

The analysis of the recent research dedicated to the development of ship's trajectory planning algorithms allowed for the distinction of the following methods. The most recent approaches to ship's path planning and collision avoidance include the Maze Routing algorithm [13], the Branch and Bound method [14], the Cooperative Path Planning algorithm [15], the Dynamic Programming [16], the Fast Marching Method [17] and the Dynamic Games theory [18, 19, 20].

Among the artificial intelligence methods the following approaches can be distinguished: the Genetic Algorithm [21], the Fuzzy Logic based method [22, 23], the Evolutionary Algorithms [24, 25, 26], the Swarm Intelligence method [27, 28], the Artificial Potential Field approach [29, 30], the A* algorithm [31, 32] and the Artificial Neural Networks [33].

A detailed analysis of the methods can be found in the works of Statheros, Howells and Maier [34], Tam, Bucknall and Greig [35] and Campbell, Naeem and Irwin [36].

2.3 Methodology

Algorithms for ship's safe trajectory planning are classified into two groups of methods. These are deterministic and stochastic methods. Deterministic methods guarantee obtaining an optimal solution, but are characterized for the worst case by exponential computational complexity. This means that if the size of the problem increases, the computational time also increases significantly, which often makes the solution unreachable. Stochastic methods do not guarantee an optimal solution, but guarantee finding a sub-optimal solution at a relatively low computational cost.

For the purpose of the conducted research the following ship's trajectory planning algorithms were developed:

- Ant Colony Optimization (ACO) (stochastic),
- Discrete Artificial Potential Field (DAPF) (deterministic),
- Trajectory Base Algorithm (TBA) (deterministic).

It was decided to develop and test the above mentioned algorithms due to the promising results of similar approaches received in the field of mobile robotics. It was decided to develop both deterministic and heuristic approaches in order to have the possibility to evaluate both types of methods.

Ant Colony Optimization algorithm

The Ant Colony Optimization algorithm belongs to the Swarm Intelligence (SI) methods. The term Swarm Intelligence was introduced by Beni and Wang [7] in relation to cellular robotic systems. Bonabeau, Dorigo and Theraulaz [8] extended the definition of swarm intelligence, defining it as any attempt to build an algorithm inspired by the collective behaviour of a colony of insects or other animal communities.

The insect colony is a decentralized system for solving problems, composed of many relatively simple mutually interacting individuals, characterized by:

- self-organization,
- flexibility,
- robustness.

Flexibility of a colony allows it to adapt to changing environments, and robustness means functioning of a colony even when some individuals do not perform their tasks. Insect colonies solve problems such as searching for food or building and expanding their nest, effectively dividing the work between individuals. Many of these problems have their analogies in engineering and computer science. Observation of the behaviour of insect colonies and the discovery of factors affecting their functioning enables the use of this knowledge in the field of intelligent system design.

Ant Colony Optimization algorithms were inspired by the observation of ant colonies behaviour. It was discovered that some species of ants have a very limited sense of sight. Certain species are even completely blind. Therefore, how these insects are functioning in an environment and in a complex community with a hierarchical structure, such as an ant colony? It was found out that ants communicate with each other and with the surrounding environment using chemical substances, which they produce. These substances are called pheromones. Ants by a pheromone trail left by them on the ground provide information to other ants.

For example ants leave a pheromone trail on their route between their nest and the food source. In this way ants show the way to the food source to other individuals in the colony. This kind of indirect communication, in which the behaviour of an individual modifies the environment, which in turn later affects the behaviour of another individual in the colony, is called stigmergy.

The first problem to which the Ant Colony Optimization algorithms were used, was the Travelling Salesman Problem (TSP), which is the task of optimizing the transition path.

Ant Colony Optimization algorithms use the mechanism of positive feedback based on analogy to trail-lying and trail-following behaviour observed in ant colonies. This mechanism is based upon strengthening fragments of good solutions that affect the quality of these solutions or strengthening good solutions as the whole. For this purpose, the so-called virtual pheromone trail is applied. By using this mechanism good solutions are kept in memory and can be used to obtain even better solutions in the future.

However, care has to be taken while strengthening good, but not very good solutions, because that can lead to stagnation - premature convergence of the algorithm to the local minimum. In order to avoid this, a certain type of negative feedback is used, called the pheromone evaporation.

In Ant Colony Optimization (ACO) algorithms, the optimization problem, for example the problem of determining the shortest route, is not defined by determining possible solutions and their cost functions, but in the form of a weighted graph.

The first proposed ACO algorithm was called the Ant System. Three basic versions of the Ant System are distinguished:

- ant-density,
- ant-quantity,
- ant-cycle.

In the first two versions, ants update the pheromone trail immediately after the movement, while in the ant-cycle algorithm the pheromone trail is updated after all ants have passed in a given iteration. Due to poor results, the first two versions are not developed. Currently, the term Ant System is the ant-cycle version of the algorithm. This version was the inspiration and base for the development of the ACO algorithm presented in this work for ship's safe trajectory planning.

The structure of the ACO-based ship's safe trajectory planning algorithm is shown in Fig. 2.1. The steps of the algorithm are described below.

The first stage of the algorithm is the reception of data defining the current navigational situation, such as:

- own ship course Ψ in degrees,
- own ship speed V in knots,
- jth target ship course Ψ_j in degrees,
- jth target ship speed V_j in knots,
- bearing on jth target ship N_j in degrees,
- distance of jth target ship from an own ship D_j in nautical miles,
- data concerning the position of static navigational obstacles.

Based upon the input data the relative course Ψ_{rj}, relative bearing N_{rj} and relative speed V_{rj} of *jth* target ship is then calculated.

Determination of dangerous objects is based upon checking for every encountered object, whether it is a dangerous object, that means the object whose course intersects with the course of an own ship. If a target ship will be classified as dangerous, the distance of an own ship from the point of trajectories intersection is calculated and the time after which an own ship will be present at the point of trajectories intersection. Next, the position of target ship at the moment, when an own ship will be present at the point of trajectories intersection, is calculated.

Then a graph composed of admissible own ship positions (waypoints), is built. At this stage, the ship's trajectory is divided into *k* phases from the start point wp_0 to the final waypoint wp_e and on this basis a graph composed of possible own ship waypoints is created. The permissible space is an area in the range from wp_0 to wp_e in front of the bow of an own ship and a distance of at least 5 nautical miles to the left and right side of an own ship.

The next stage is placement of constraints in the form of hexagon domains on the constructed graph, representing the area around the target ships, which cannot be exceeded by an own ship. These restrictions are placed in the positions, where target ships will be found, when an own ship will be at the point of trajectories intersection. Next, the vertices that are placed inside the constraints or at their edges are removed from the graph.

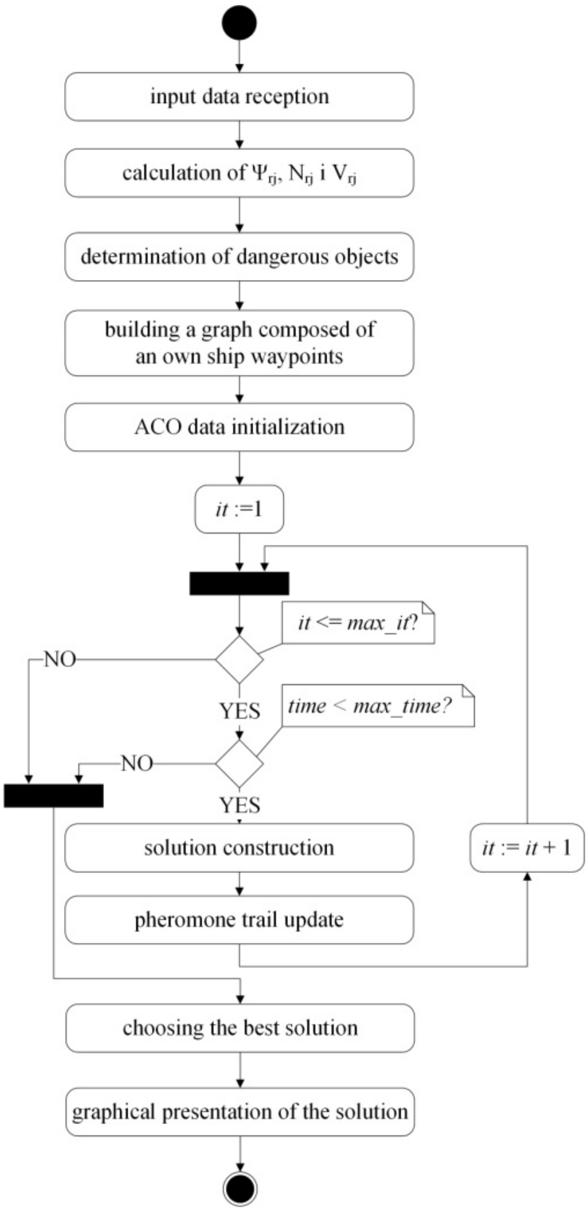


Fig. 2.1. Flowchart of the ACO-based ship's safe trajectory planning algorithm

Taking into account static constraints

Land areas, shallows and other static constraints are included in a manner analogous to the method of taking into account dynamic constraints. The first stage of static constraints consideration includes placing these areas on the graph of possible own ship waypoints. Next, the vertices of the graph that are inside or on the edges of static navigational constraints are removed.

The ACO calculations

Calculation of ship's safe trajectory using the ACO algorithm consists of three main stages:

- ACO data initialization,
- solution construction,
- pheromone trail update.

ACO data initialization

In the ACO data initialization procedure, the following parameters of the algorithm are defined:

- initial value of pheromone trail τ_0 on all vertices, set to the same value for all vertices,
- α and β coefficients appearing in the formula for the probability of the ant's next move,
- pheromone evaporation rate ρ ($0 < \rho <= 1$),
- number of ants - *ant_num*,
- maximum number of ant's steps - *max_steps*,
- maximum number of iterations - *max_it*.

Solution construction

1. Each ant begins constructing its path from the start waypoint wp_0 with coordinates (x_0, y_0), which is the current position of an own ship.

2. Then, each ant constructs its path until it reaches the final waypoint with coordinates (x_e, y_e) or it reaches the specified maximum number of steps - *max_steps* (Fig. 2.2). At every step the ant using the action choice rule in a probabilistic manner selects the next vertex of the graph from the neighbouring vertices (Fig. 2.3). The choice of the next vertex depends on the value of the pheromone trace at the vertex j adjacent to the current vertex i - $\tau_j(t)$ and on the heuristic information η_{ij}. In the developed algorithm the heuristic information is the inverse of the distance between the current vertex i and the adjacent vertex j. The probabilistic selection of the next vertex works similarly to the roulette wheel selection procedure used in evolutionary algorithms [9].

This process takes place in the following stages:

- adding the probabilities of particular neighbouring vertices selection - *sum_p*,
- random choice of the number r from the range [0; *sum_p*],
- traversing the following neighbouring vertices until the sum of the probabilities of choosing the vertices visited so far will be greater than or equal to the number of r; fulfilling this condition means choosing a given vertex.

In the formula for the probability of selecting the following vertex - equation (7) there are two coefficients. If $\alpha = 0$, the nearest neighbouring vertex is most likely to be chosen, whereas if $\beta = 0$, only pheromone enhancement works, which results in rather unsatisfactory solutions.

$$P_{wp_{ij}}^{ant}(t) = \frac{\left[\tau_{wp_j}(t)\right]^{\alpha} \cdot \left[\eta_{wp_{ij}}\right]^{\beta}}{\sum_{l \in wp_i^{ant}} \left[\tau_{wp_l}(t)\right]^{\alpha} \cdot \left[\eta_{wp_{il}}\right]^{\beta}} \quad (7)$$

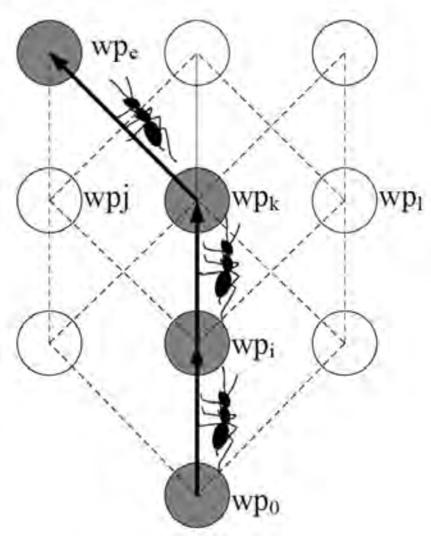


Fig. 2.2. An exemplary path constructed by an ant

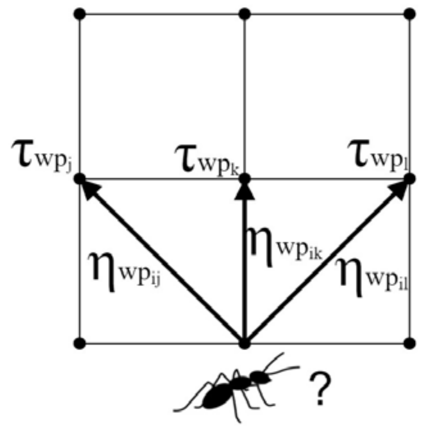


Fig. 2.3. A diagram explaining the process of selecting an ant's next move

Pheromone trail update

When all the ants in the given iteration of the algorithm finish constructing their paths, the process of updating the pheromone trail takes place, which is composed of two stages - equation (8).

1. In the first stage, the pheromone evaporation takes place, in which reduction of the pheromone trail on all vertices of the graph by a certain constant value is carried out. Reducing the pheromone trail is a mechanism that allows the ants to "forget" the wrong decisions they have made.

2. Next, the pheromone deposit is carried out, during which a certain value of the pheromone trail is added to all vertices of the graph belonging to the paths constructed by the ants in a given iteration. Due to that mechanism, the vertices, which constitute parts of the paths chosen by many ants and which constitute parts of the shortest paths, receive more pheromone trial. In this way the probability of their selection by ants in subsequent iterations increases.

$$\tau_{wp_j}(t+1) = (1-\rho)\cdot\tau_{wp_j}(t) + \sum_{ant=1}^{ant\_num} \Delta\tau_{wp_j}^{ant}(t) \tag{8}$$

Before starting the next iteration, the best solution found so far - the trajectory with the shortest length, is saved. Then the process of smoothing the best trajectory determined in a given iteration is executed.

The termination condition for the algorithm is meeting at least one of the following conditions:

- achievement of the maximum number of iterations,
- reaching the maximum calculation time.

Selection of the best solution considering the dynamic properties of an own ship. One best trajectory is calculated for each iteration. The control quality evaluation criterion is defined in the form of control quality index *I* - equation (9), where $u(t) \in U$, $x(t) \in X$, *U* is a set of admissible controls, and *X* is the admissible state space. Optimization task in the process of avoiding collisions at sea is to find such controls *u(t)*, defining the optimal trajectory, so that the control quality index *I* will reach the minimum value, while meeting the limitations. The limitations result from the conditions of an own ship safe passage in accordance with COLREGs in the encounter situation with target ships and static navigation restrictions.

$$I = \min_{U,X} \int_{t_0}^{t_k} f[x(t), u(t), t]dt \tag{9}$$

The value of the cost function *f(S)* of a given admissible trajectory *S* is its length *dist(S)* - equation (10). The optimization task is finding the minimum value of the cost function for the problem under consideration, while meeting the limitations.

The optimal trajectory S_{opt}, a trajectory with the shortest length not exceeding static and dynamic restrictions, belonging to the set of admissible trajectories, constitutes the solution to the considered problem.

$$I = dist(S) = \sum_{i=0}^{k-1} \sqrt{(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2} \rightarrow min \tag{10}$$

After the algorithm reaches the maximum number of iterations, it is checked which of the best trajectories meet the limitations taking into account the manoeuvre time. This process is realized by the division of the trajectory into small fragments of the same length, and then checking with respect to the manoeuvre time, whether each point is located outside restrictions by moving an own ship and simultaneously moving the domains of the target ships. After rejecting the trajectories that exceed the restrictions, the shortest trajectory is selected from the remaining trajectories.

The output of the algorithm include:

- the course of an own ship at every line segment of the calculated trajectory in degrees,
- an own ship travelling time from the initial waypoint wp_0 to the end waypoint wp_e in minutes,
- the length of the trajectory in nautical miles,
- graphical presentation of the calculated trajectory.

Discrete Artificial Potential Field algorithm

The Artificial Potential Field (APF) method was introduced by Khatib [10] in 1986. It’s operation principle is based upon the concept of potential forces. This concept was previously applied to solve the mobile robot path planning problem. There are two types of potential forces distinguished: an attractive potential force and a repulsive potential force. An attractive potential force is applied around the goal position in order to push the robot towards this position. In the same manner a repulsive potential force is applied around the positions of obstacles in order to repel the robot from these places. This concept is shown in Fig. 2.4.

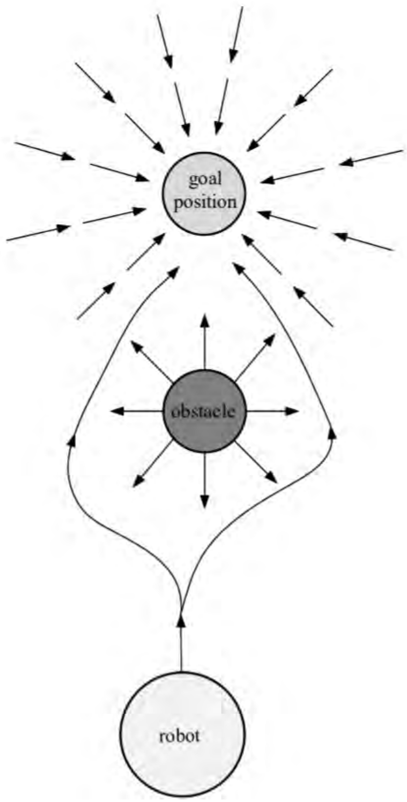


Fig. 2.4. The concept of the APF method

In the APF approach a continuous potential field is created with the use of functions describing the attractive and repulsive potentials. In the Discrete Artificial Potential Field (DAPF) approach a potential field is discrete. It is created with the use of a set of rules. The DAPF approach belongs to the deterministic methods. Therefore the algorithm is characterized by the achievement of repeatable solutions for every run of the algorithm with the same input data.

The model of the process applied for calculation of ship's safe trajectory with the use of the DAPF algorithm is defined by equations (6).

Target ships are assumed not to change their courses and speeds. Dynamic properties of an own ship are taken into account by the use of the manoeuvre time parameter. The manoeuvre time is the time of the ship movement from the beginning of the manoeuvre to the moment, when the value of the new course has been reached.

$$c_{xy} = \begin{cases} c_{00} - 10 \cdot y, & x = 0, \quad 0 \le y \le m-1 \\ c_{0y} + x, & 1 \le x \le \dfrac{n-1}{2} \\ c_{0y} - x + 0.5\,, & \dfrac{-n-1}{2} \le x \le -1 \end{cases} \tag{11}$$

Environment Representation

In the DAPF algorithm a discrete environment is applied. It is represented as a grid-based map composed of *m x n* cells, where *m* is the number of horizontal cells and *n* is the number of vertical cells. The following types of cells are distinguished: free cells, obstacle cells (cells occupied by obstacles), a start cell and a goal cell. Every cell is described with the use of the following parameters: the position of its centre (*x* and *y* coordinates) and a potential.

The following rules are applied in order to assign a proper value of a potential to every cell:

- a goal cell has a potential equal to zero,
- obstacle cells have a potential equal to infinity,
- free cells are assigned with increasing potentials from the goal cell to the start cell according to equation (11) (*x* and *y* are cell coordinates),
- cells on the right side from the line segment connecting the start and the goal cell have lower potentials than these on the left side in order to enforce fulfilment of COLREGs rules 14 and 15.

An exemplary environment representation for the DAPF algorithm is shown in Fig. 2.5.

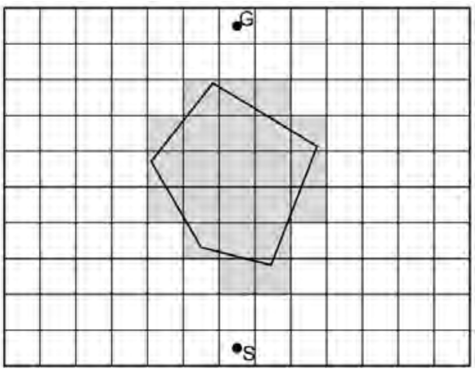


Fig. 2.5. A grid-based map of an environment

Constraints Consideration

The DAPF algorithm takes into account both static and dynamic obstacles. Static obstacles are modelled as polygons. Fig. 2.5 presents how a static obstacle is modelled in the DAPF algorithm. Obstacle cells are marked by grey colour. These include all cells intersected by an edge of an obstacle. Dynamic obstacles (target ships) are modelled with the use of a hexagon domain. It is also possible to apply other shapes of the target ship domain, e.g. in a form of an ellipse, a circle or a parabola. A hexagon target ship domain is applied in order to assure a proper distance between an own ship and target ships during manoeuvres. A proper shape and size of the target ship domain is applied in order to achieve the COLREGs fulfilment by the calculated own ship trajectory.

The Search Algorithm

A block diagram of the DAPF search algorithm is presented in Fig. 2.6. It is assumed that each cell uses an eight-connected mapping, as presented in Fig. 2.7. It means that from each cell there are eight possibilities of movement. The DAPF search algorithm calculated a safe own ship trajectory by choosing at every step of calculations a neighbouring cell with the lowest value of its potential. The process starts from the start cell and continues until the goal cell has been reached. When a cell is chosen, the value of its potential is set to infinity. This mechanism is applied in order to prevent the algorithm from generating loops in the calculated trajectory. An exemplary collision-free trajectory calculated by the DAFP algorithm is shown in Fig. 2.8.

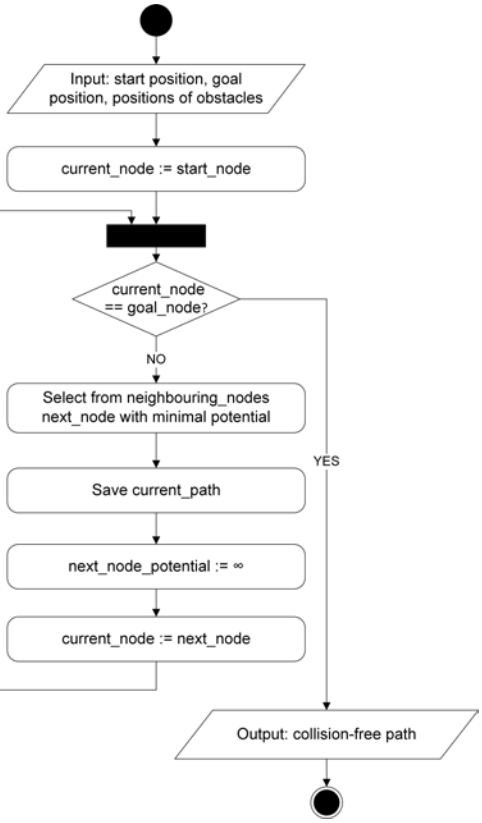


Fig. 2.6. A block diagram of the DAPF algorithm

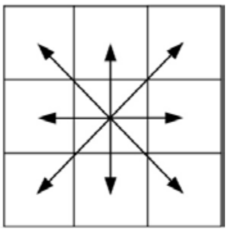


Fig. 2.7. An eight-connected mapping

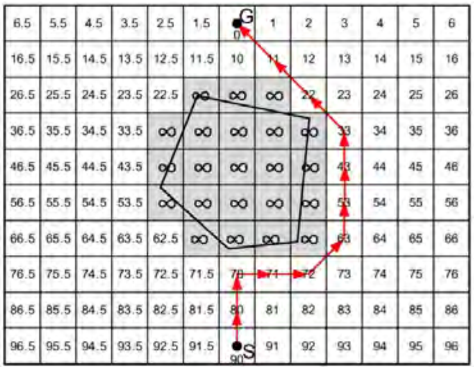


Fig. 2.8. A collision-free trajectory (red arrows)

Path Optimization

In order to optimize the safe trajectory a Path Optimization Algorithm (POA) is applied along with the DAFP algorithm. A block diagram of the POA is presented in Fig. 2.10.

In the POA procedure an intersection check between the line segment constituting a part of a trajectory and all of line segments constituting edges of static and dynamic obstacles is carried out. After an intersection is detected, calculations are terminated for the checked nodes and the nodes connecting this line segment will not be connected. Otherwise (if an intersection was not detected) the checked line segment becomes a part of the smoothened trajectory. Fig. 2.9 shows an exemplary optimized collision-free trajectory.

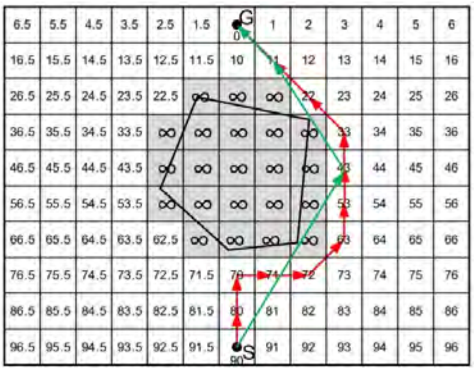


Fig. 2.9. A collision-free optimized trajectory (green arrows)

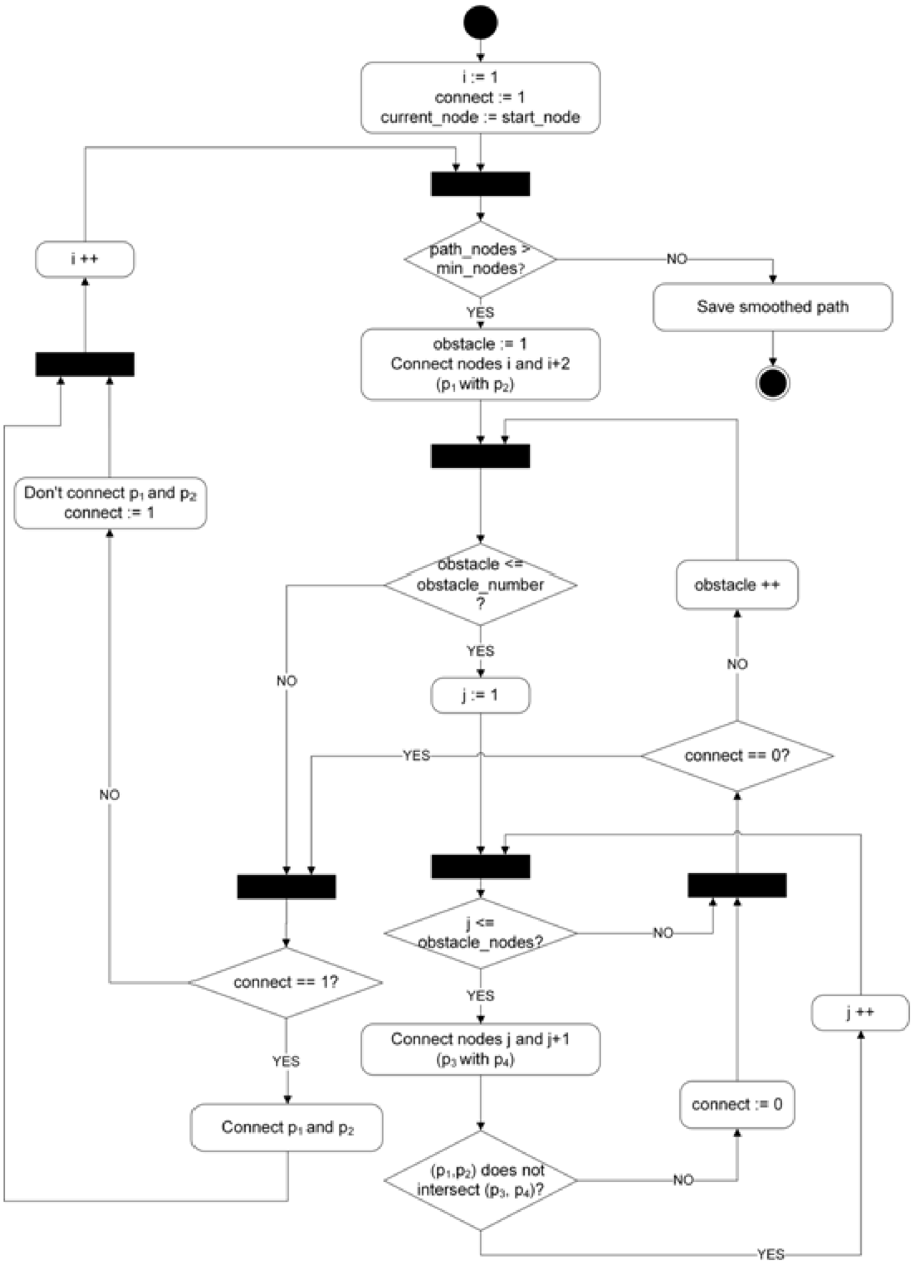


Fig. 2.10. A block diagram of the POA

Trajectory Base Algorithm

The Trajectory Base Algorithm (TBA) is a deterministic method. The algorithm determines a ship's safe trajectory by searching a base of trajectories in order to find the best solution to the currently considered navigational situation. The best solution is defined as a safe trajectory with the minimal value of its fitness function. The fitness function can be applied to optimize one criterion or multiple criteria. The following optimization criteria can be used in ship's trajectory planning:

- the shortest length of a trajectory,
- the shortest time of arrival at the final waypoint,
- the smoothest path.

The trajectories stored in the base of trajectories constitute candidate solutions to the considered problem of ship's trajectory planning. The search algorithm evaluates the candidate solutions and a trajectory with the lowest value of its fitness function is chosen as the final solution for the considered encounter situation.

The navigational environment is modelled as a discrete solution space, where a set of trajectories based on different rules are generated and saved in the base. The generated trajectories, based upon an exemplary rule, are shown in Fig. 2.11.

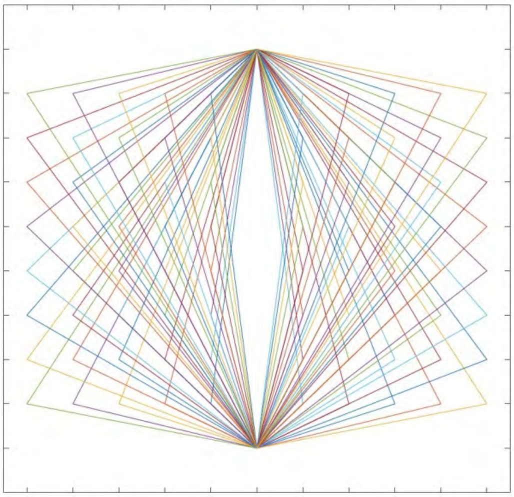


Fig. 2.11. An exemplary set of candidate trajectories

Assumptions

During the problem definition process it was assumed that:

- target ships maintain their motion parameters,
- a kinematic model of ship's motion is used to describe the process - equations (6),
- dynamic properties of an own ship are taken into account by the time of manoeuvre parameter,
- return to a predefined final waypoint will be carried out (important for restricted waters).

The Trajectory Base Search Algorithm

A block diagram of the TBA is presented in Fig. 2.12.

The input data to the algorithm, describing the current navigational situation include:

- an own ship course and speed,
- target ships' courses, speeds, bearings and distances from an own ship,
- static obstacles placement (lands, islands, buoys, fairways, canals, shallows, etc.).

These data are obtained with the use of navigational equipment such as radars with ARPA, AIS, ECDIS, GPS, echo sounders, a gyrocompasses and speed logs with the use of the NMEA standard.

After the data reception, the relative courses, speeds and bearings of target ships are calculated. Then, the first trajectory is retrieved from the base of trajectories for evaluation.

The trajectory under evaluation is first divide into a number of steps. After that, for every step the algorithm checks whether instantaneous positions of an own ship and target ships do not collide. An instantaneous position of an own ship is calculated taking the dynamic properties of a vessel into account by an application of the time of manoeuvre, When a collision is detected, the candidate trajectory is rejected. Then, the next trajectory is retrieved from the base for evaluation.
The applied fitness function was defined as the length of a trajectory and is calculated as a sum of the lengths of the k-1 line segments, where k is the number of trajectory waypoints, x and y are the waypoints coordinates - equation (10). Trajectories in the base are sorted in a descending order according to their fitness function value. A trajectory with the minimal value of its fitness function constitutes the best trajectory. After the algorithm has found a trajectory not exceeding the constraints, is stops the selection process and presents the solution.

The output data include:

- own ship courses calculated for every line segment of the trajectory,
- the length of a trajectory,
- time taken for an own ship to reach the final waypoint.

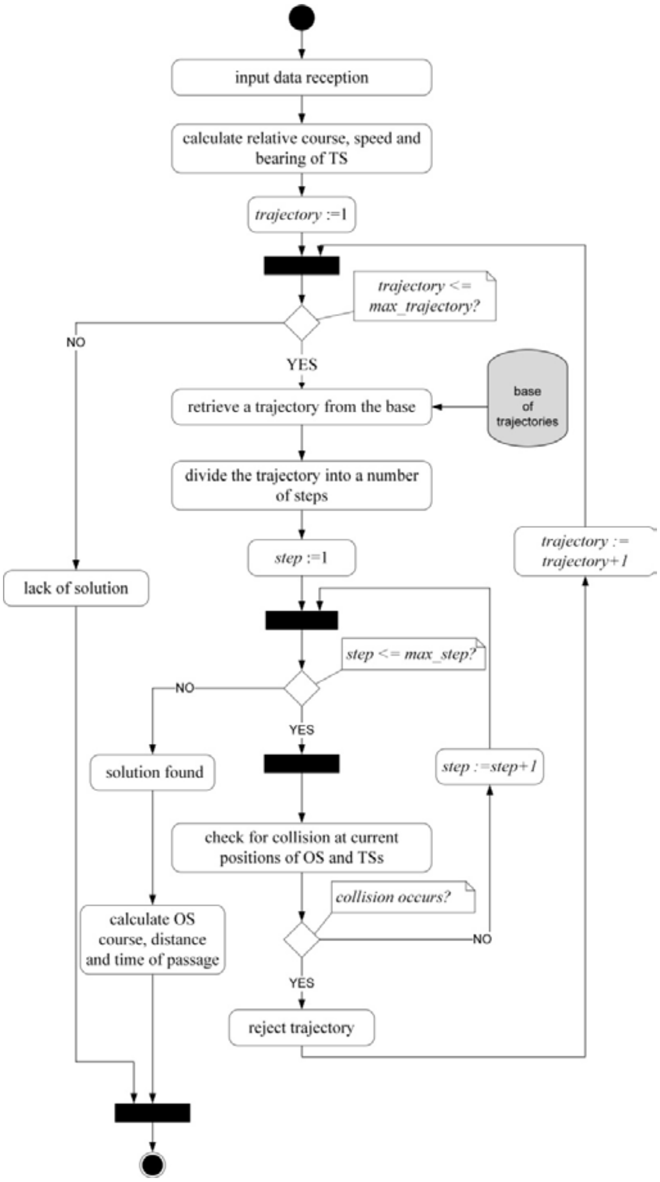


Fig. 2.12. A block diagram of the TBA

2.4 Conclusions

The developed algorithms include one heuristic approach based upon the Ant Colony Optimization and two deterministic methods, utilizing the concept of Artificial Potential Fields and the Trajectory Base. The environment representation, static and dynamic obstacles modelling and the model of the process was unified for all algorithms in order to facilitate the process of solutions comparison.

3. Stage II – simulation studies

3.1 Aim

The aim of this stage of the project was to verify the algorithms by simulation studies with the use of simple encounter situations described in COLREGs.

3.2 Methodology

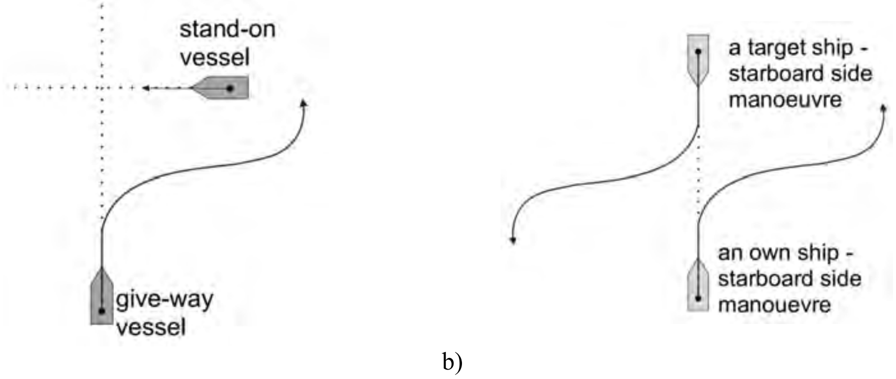
The algorithms were implemented in the MATLAB programming language due to build-in plotting functions that allow for an easier graphical presentation of results.

The main criterion in the evaluation of the algorithms was the safety of obtained solution. The second criterion is the COLREGs compliance of the calculated trajectory. The criteria evaluating the quality of a solution include: the run time of the algorithms, the length of the safe trajectory and calculated course alteration manoeuvres.

The algorithms were tested for different simple encounter situations defined in COLREGs. The environment description and obstacles modelling was unified for all of the developed algorithms in order to allow for a meaningful comparison of the methods. In all algorithms target ships were modelled with the use of hexagon shape ship's domains. The following dimensions of the target ship domain were assumed for the calculations: distance towards the bow - 1.3 NM, distance of amidships - 0.6 NM, distance towards the stern - 0.5 NM, distance towards the port side - 0.5 NM and distance towards the starboard side - 0.6 NM. For the ACO algorithm the following parameters were used in calculations: $\tau_0 = 1$, $\rho = 0.1$, $\alpha = 1$, $\beta = 2$, $max\_it = 20$ and $ant\_num = 10$.

Below results of exemplary tests are presented. They include two simple scenarios with one target ship classified as a crossing situation and a head-on situation and an encounter situation with two target ships. Presented results include a comparative analysis of solutions obtained with the use of three developed algorithms. The algorithm with the best performance was evaluated in terms of the feasibility of trajectories calculated for the same navigational situation from the perspective of different ships. Therefore a verification of solutions achieved by the TBA from the perspective of different ships is also presented. Calculations were carried out on a PC with i7-8550U 1,9 GHz processor, 8 GB RAM, 64-bit Windows 10 system.

3.3 Results



a) b)

Fig. 3.1. COLREGs rules for a) a crossing situation – rule 15 b) a head-on situation – rule 14

Test case 1 – a head-on situation

This test case is a simple encounter situation with one target ship, classified as a head-on situation by COLREGs. Such situation is specified by rule 14 (Fig. 3.1b). In such situation an own ship is a give-way vessel and is obliged to alter course to starboard side. However, the target ship is also a give-way vessel in this case and also has to change her course to starboard.

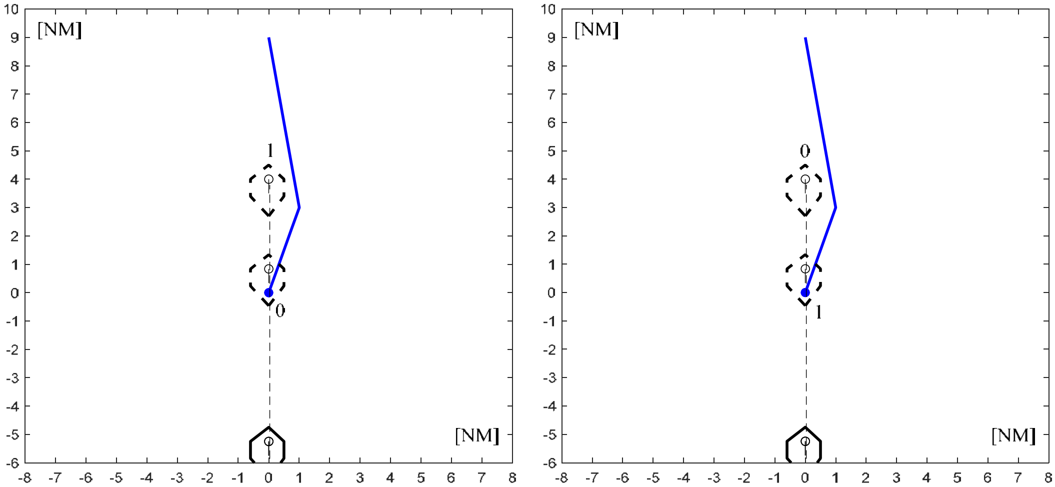
Input data for this test case are listed in Tab. 3.1. The trajectory calculated for an own ship by the TBA is presented in Fig. 3.2a). It can be noticed that the TBA calculated a starboard side manoeuvre for an own ship. Such manoeuvre fulfils rule 14 of COLREGs. A trajectory calculated by the TBA from the target ship perspective is shown in Fig. 3.2b). As it can be seen, it is also a starboard side manoeuvre, compliant with COLREGs. The course change manoeuvres calculated to an own ship and a target ship also fulfil rule 8b of COLREGs, stating that the manoeuvres should be large enough to be apparent for other ships. All manoeuvres are larger than 10 degrees, what can be noticed in Tab. 3.2, presenting results obtained for this test case by different algorithms.

A graphical presentation of trajectories calculated by the TBA from the perspectives of both ships is shown in Fig. 3.3a). The trajectory of an own ship does not intersect the trajectory of a target ship, what means that the solutions returned by the TBA allow for a safe passage of both ships.

A comparison of own ship trajectories calculated by different algorithms is presented in Fig. 3.3b). ACO and TBA calculated the same solution, but the run time of the TBA reached about 0,6 second and the run time of ACO algorithm was much longer (about 50 seconds). The trajectory calculated by the DAPF for this test case was about 0,2 nautical mile longer, but obtained in about 1 second. To sum up, the TBA obtained for this test case the best result in terms of both run time and quality of a solution.

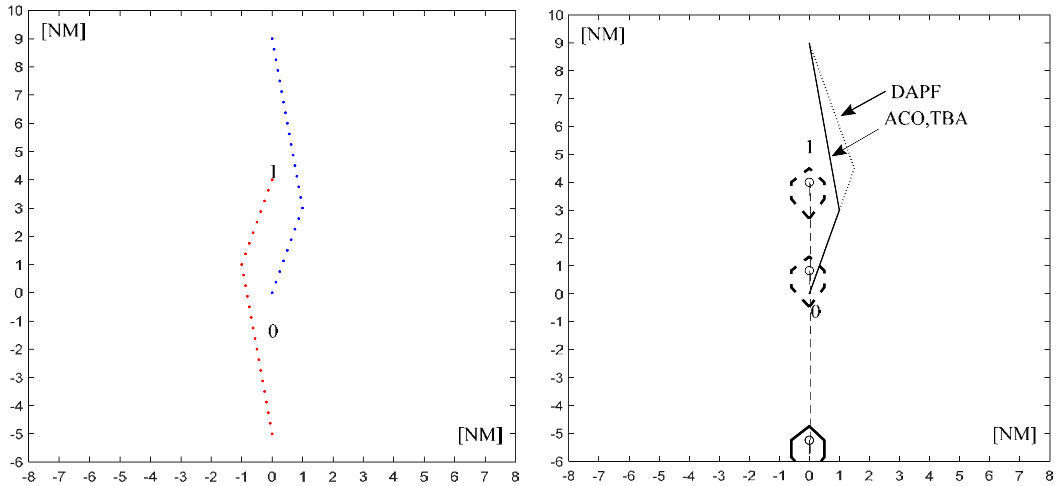
Tab. 3.1. Input data for test case 1

Ship	Course [º]	Speed [kn]	Distance [nm]	Bearing [º]
0	0	10	-	-
1	180	10	4	0



a) b)

Fig. 3.2. Ship's safe trajectory for test case 1. Solution calculated from a) an own ship perspective b) a target ship perspective



a) b)

Fig. 3.3. Ship's safe trajectory for test case 1. Solution calculated a) from both ships perspective b) by different algorithms

Tab. 3.2. Solutions for test case 1

Algorithm	Course [°]	Distance [nm]	Run time [s]
ACO	18, 351	9.25	49.67
DAPF	18, 342	9.48	1.09
TBA	18, 351	9.25	0.62

Test case 2 – a crossing situation

This test case is an encounter situation with one target ship, classified by COLREGs as a crossing situation, defined by rule 15 (Fig. 3.1a). In such situation an own ship is a give-way vessel and is obliged to keep out of the way of the target ship. If possible an own ship should avoid crossing ahead of the target ship.

Input data for test case 1 are given in Tab. 3.3. In Fig. 3.4a) an own ship trajectory calculated by the TBA is presented. An own ship is a give-way vessel in this situation, therefore it should take a collision avoidance action. The algorithm calculated an own ship manoeuvre to the starboard side. Such course change fulfils rule 15 of COLREGs. It is also compliant with rule 8b of COLREGS, stating that manoeuvres should be large enough to be easily noticeable by other vessels.

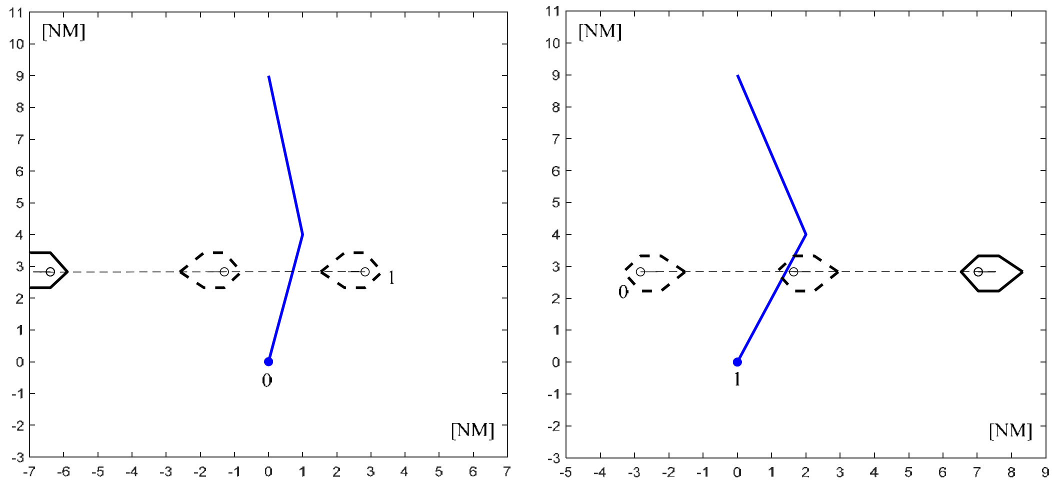
Fig. 3.4b) shows a trajectory calculated by the TBA from the target ship perspective. The target ship is a stand-on vessel, so generally according to rule 15 of COLREGs should maintain her course and speed. According to COLREGs, a stand-on vessel may take action to avoid collision, if a give-way ship is not acting properly. It is also stated in COLREGs that a stand-on vessel should avoid turning to port side in a crossing situation. In this situation the TBA calculated a starboard side manoeuvre for the target ship.

Figure 3.5a) shows trajectories calculated by the TBA from the perspectives of both ships. It can be noticed that obtained trajectories allow for a safe passage of the ships. Figure 3.5b) presents a comparison of trajectories calculated by three developed algorithms, while numerical results obtained by the application of different algorithms are listed in Tab. 3.4.

It was verified that all algorithms returned a feasible solution for this test case. The trajectories returned by the compared algorithms are very similar in terms of the length of the trajectory and the values of course alterations. They are characterized by insignificant differences. More meaningful differences are observed as comparing the run time of the algorithms. The shortest run time was obtained by the TBA (about 0,3 second). The DAPF calculated the solution in 0,5 second, while the ACO algorithm in about 11 seconds. Comparison of the results allows to state that the best solution both with reference to the quality of returned trajectory and the run time of calculations was obtained by the TBA.

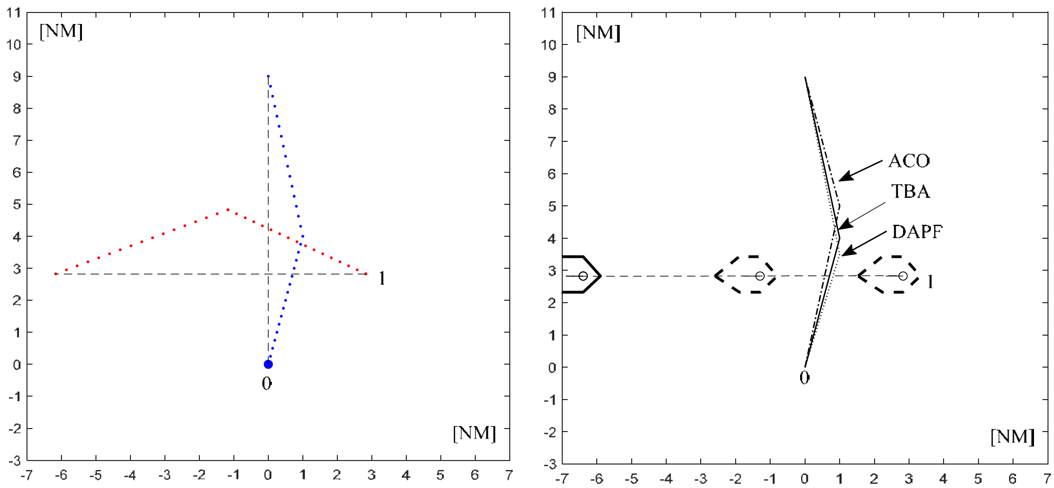
Tab. 3.3. Input data for test case 2

Ship	Course [°]	Speed [kn]	Distance [nm]	Bearing [°]
0	0	12	-	-
1	270	12	4	45



a) b)

Fig. 3.4. Ship's safe trajectory for test case 2. Solution calculated from a) an own ship perspective b) a target ship perspective



a) b)

Fig. 3.5. Ship's safe trajectory for test case 2. Solution calculated a) from both ships perspective b) by different algorithms

Tab. 3.4. Solutions for test case 2

Algorithm	Course [°]	Distance [nm]	Run time [s]
ACO	11, 346	9.22	11.4
DAPF	16, 350	9.23	0.48
TBA	14, 349	9.22	0.28

Test case 3 – an encounter situation with two target ships

This test case is an encounter with two target ships. Input data for this test case are presented in Tab. 3.5. In this situation an own ship is a give-way vessel and is obliged to keep out of the way of both target ships. Moreover, if the circumstances of the case admit, an own ship should alter course to starboard. Target ship 1 is a stand-on vessel in this case and does not have to alter her course. Target ship 2 is a stand-on vessel with reference to an own ship and is a give-way ship in relation to target ship 1.

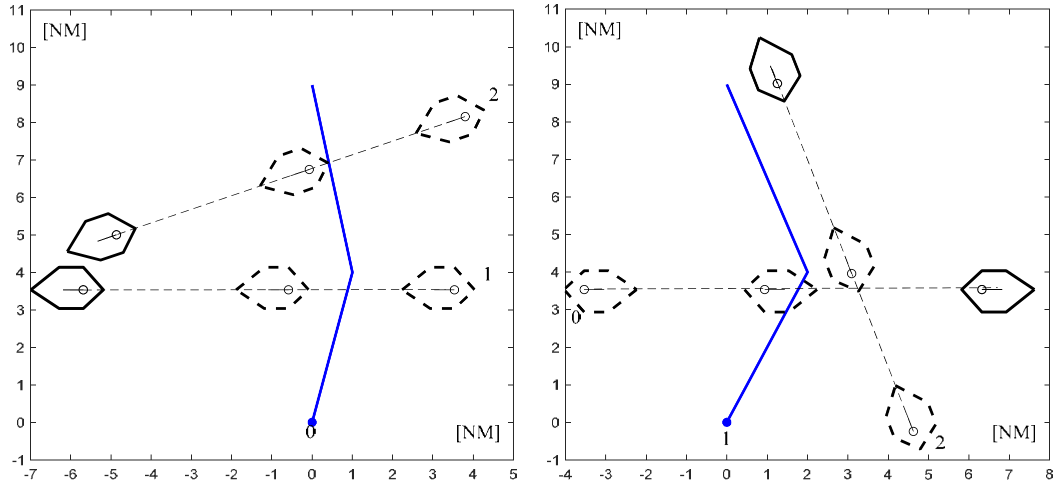
An own ship trajectory calculated for this test case by the TBA is shown in Fig. 3.6a). Obtained solution covers an own ship manoeuvre to starboard side, giving way to both target ships and avoiding crossing ahead of the target ships, as it should be solved according to COLREGs. The course alteration manoeuvres are also large enough to be readily apparent for other vessels. In Fig. 3.6b) a trajectory calculated by the TBA from the target ship 1 perspective is presented. As it can be noticed, the algorithm calculated a starboard side manoeuvre for the target ship 1. According to COLREGs, such behaviour is acceptable, if .an own ship does not execute an appropriate manoeuvre in a proper time.

In Fig. 3.7a) trajectories calculated by the TBA from the perspectives of all ships are shown. It can be seen that, as it was calculated by the algorithms, target ship 2 maintains her course and speed. It can also be noticed that trajectories of all ships taking part in the considered situation allow for their safe passage.

In Fig. 3.7b) a comparison of solutions obtained for this test case by different developed algorithms in presented. Numerical results are compared in Tab. 3.6. The TBA and DAPF returned the same solution. The ACO calculated a trajectory with the same length, but slightly different course changes. The run time of the TBA reached 0,5 second, the DAPF calculated the trajectory in about 1,5 second, while the ACO algorithm in about 30 seconds. Therefore, the best solution in the shortest run time was obtained by the TBA.

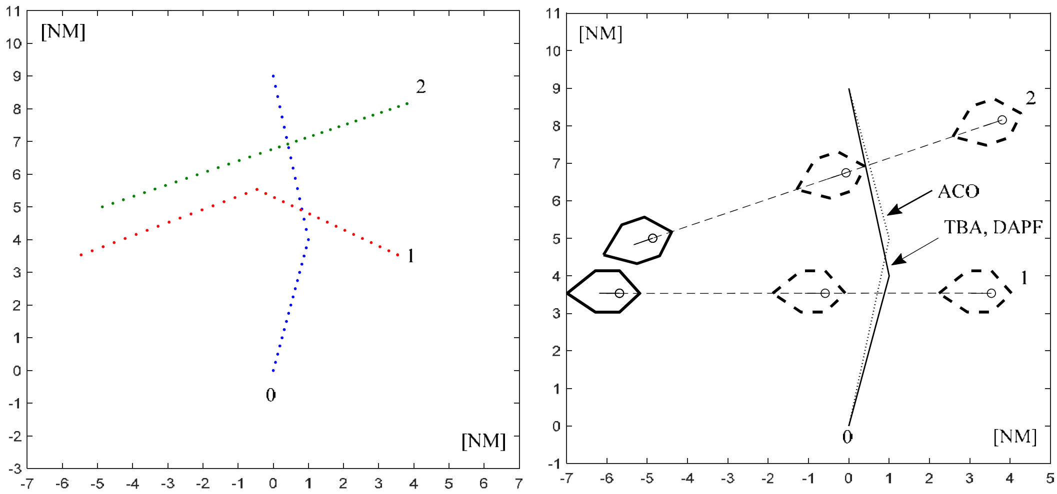
Tab. 3.5. Input data for test case 3

Ship	Course [°]	Speed [kn]	Distance [nm]	Bearing [º]
0	0	12	-	-
1	270	12	5	45
2	250	12	9	25



a) b)

Fig. 3.6. Ship's safe trajectory for test case 3. Solution calculated from a) an own ship perspective b) the target ship 1 perspective



a) b)

Fig. 3.7. Ship's safe trajectory for test case 3. Solution calculated a) from both ships perspective b) by different algorithms

Tab. 3.6. Solutions for test case 3

Algorithm	Course [°]	Distance [nm]	Run time [s]
ACO	11, 346	9.22	33.26
DAPF	14, 349	9.22	1.4
TBA	14, 349	9.22	0.48

3.4. Conclusions

Analysis of results allows to state the following concluding remarks:

- all algorithms return feasible solutions for simple encounter situations in the assumed amount of time (less than sixty seconds);
- the best results in terms of the quality of solutions are achieved by the TBA (the smallest course changes and the shortest run time);
- the TBA analysis from the perspective of different ships enable to state that the algorithm returns trajectories that do not intersect.

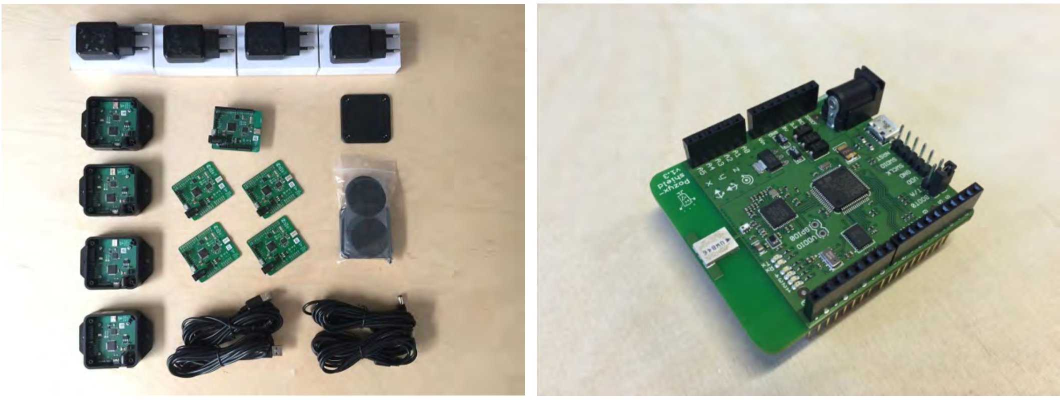
The above statements prove that the developed algorithms are applicable for use in a Decision Support Systems on-board a ship or in Guidance, Navigation and Control systems of autonomous vessels for trajectory planning. The solutions calculated by the developed algorithms fulfil COLREGs. A safe trajectory is determined to the specified end point, what is particularly important for restricted waters. Results prove the algorithm's feasibility and near-real time operation. Results of the TBA demonstrate that the trajectories calculated from the perspectives of different ships for the same encounter situation do not lead to a collision situation between the vessels.

4. Stage III – experimental studies (mobile platforms)

4.1 Aim

The aim of this stage of the project was to verify the developed algorithms on real objects in the laboratory conditions. It was assumed that the test will be carried out with the use of mobile platforms as mobile robots path planning constitutes a similar task to ship's safe trajectory planning.

4.2 Methodology



a) b)

Fig. 4.1. a) The Indoor Positioning System from Pozyx labs [11], b) Pozyx tag [11]

Tab. 4.1. Technical specification of the IPS tag

Parameter	Value
Microcontroller	STM32F4
Serial communication	I2C
Tag-size	6 cm x 5.3 cm
Tag-weight	12 g
Arduino board support	Arduino Uno R3, Mega and Nano
General purpose LEDs	4
LEDs for UWB connectivity	2
Optional GPIO pins	4
Onboard 3.3 V regulator:	Power selection: battery, Arduino or USB
Firmware updates	Micro USB

Tab. 4.2. Technical specification of the IPS Ultra-wideband technology

Parameter	Value
Transceiver	Decawave DW1000
Ranging accuracy	10 cm
Range	Up to 100 m
Frequency range	3.5 - 6.5 GHz
Communication speed	Up to 6.8 Mbps

Tab. 4.3. Technical specification of the IPS motion sensors

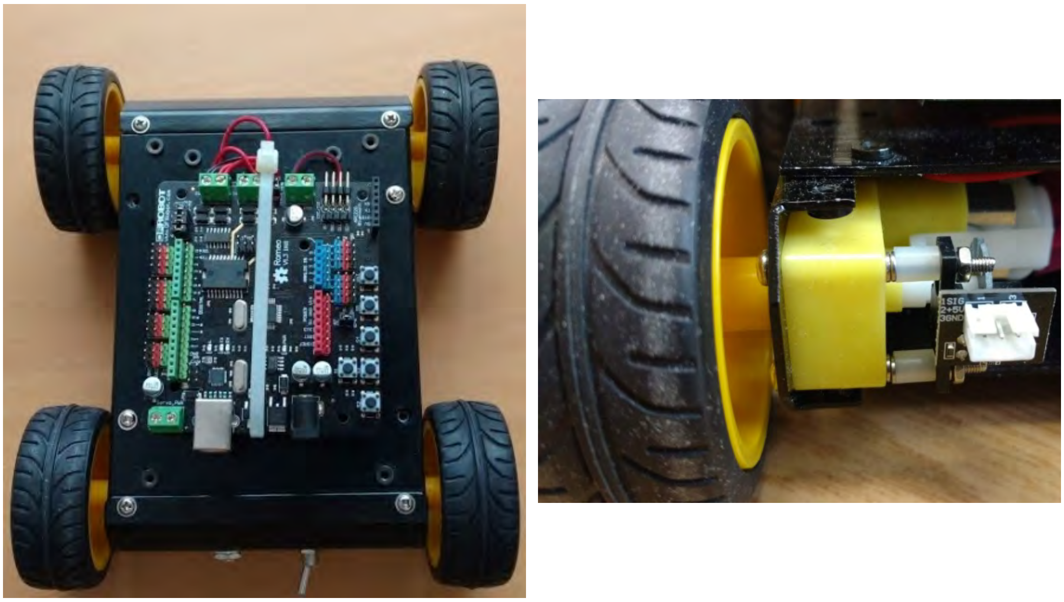
Parameter	Value
Chip	BNO-055
Sensor1	3-axis accelerometer
Sensor2	3-axis gyroscope
Sensor3	3-axis compass
Pressure sensor	MPL3115A2

Tab. 4.4. Technical specification of the mobile platform

Parameter	Value
Motors	3 ~ 7.5V DC
Speed	90 cm/s
Dimensions	200 x 170 x 105 mm

Tab. 4.5. Technical specification of the wheel encoders

Parameter	Value
Voltage	5 V
Current	< 20 mA
Resolution	20 PPR
Weight	20 g



a) b)

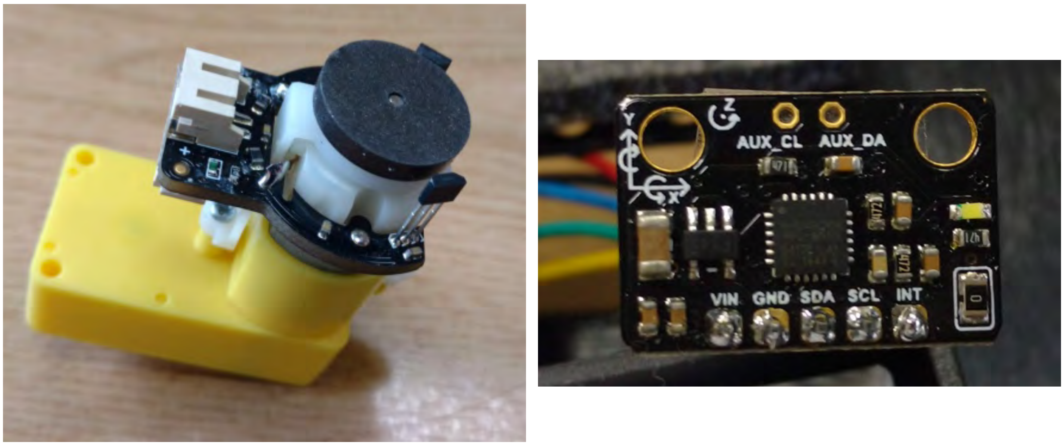
Fig. 4.2. a) The DFRobot Pirate-4WD b) Gravity TT wheel encoders

The system for experimental studies with the use of mobile platforms is composed of:

- the Indoor Positioning System (IPS) from Pozyx labs (Fig. 4.1a);
- mobile platforms DFRobot Pirate-4WD (Fig. 4.2a);
- additional sensors for a mobile platform (encoders, gyroscopes);
- a workstation for data registration.

Indoor Positioning System (IPS)

The Indoor Positioning System (IPS) from Pozyx labs is used for localization of mobile objects. The IPS is composed of tags placed on mobile robots and four nodes (anchors) with known positions. Tags (Fig. 4.1b) provide the position and direction of movement of the mobile object. Tags contain an ultra-wideband transceiver and an inertial measurement unit DWM1000 from DecaWave for measurement of the object's orientation. Accelerometers, gyroscopes and magnetometers are included in the inertial measurement unit. Technical specification of the IPS tag is given in Tab. 4.1. The main parameters of the IPS Ultra-wideband technology are listed in Tab. 4.2. In Tab. 4.3 technical specification of the IPS motion sensors is presented.



a) b)

Fig. 4.3. a) The DFRobot TT Micro DC geared motors with encoders b) the MPU 6050 sensor

Mobile platforms DFRobot Pirate-4WD

The mobile platform DFRobot Pirate-4WD has dimensions of 200 x 170 x 105 mm (length x width x height), speed of up to 0.9 m/s, and is equipped with 4 DC motors [12]. It is driven by an ATMega328 microcontroller with an integrated DC motor driver. It is a differential-drive platform, what means that the wheels on the left and right side of the mobile robot are controlled independently. Technical specification of the mobile platform is presented in Tab. 4.4.

At first the Gravity TT Motor Encoders (Fig. 4.2b) were applied in order to achieve a straight line movement of a mobile platform. Technical specification of the wheel encoders is given in Tab. 4.5. Due to unsatisfactory results obtained with the use of this approach, DFRobot TT Micro DC geared motors with encoders (Fig. 4.3a) and the MPU 6050 sensor (Fig. 4.3b) were implemented instead. Technical specification of the DFRobot TT Micro DC geared motors with encoders is presented in Tab. 4.6. MPU 6050 is a 3-axis accelerometer and gyroscope, so it is applied to measure acceleration and angular velocity in three axes. Its main parameters are listed in Tab. 4.7.

Tab. 4.6. Technical specification of the DFRobot TT Micro DC geared motors with encoders

Parameter	Value
Gear ratio	120:1
No-load speed at 6 V	160 rpm
No-load speed at 3 V	60 rpm
No-load current at 6 V	0.17 A
No-load current at 3 V	0.14 A
Encoder operating voltage	4.5~7.5 V
Motor operating voltage	3~7.5 V (Rated voltage 6V)
Operating ambient temperature	-10~+60 ºC
Weight	50 g

Tab. 4.7. Technical specification of the MPU 6050 sensor

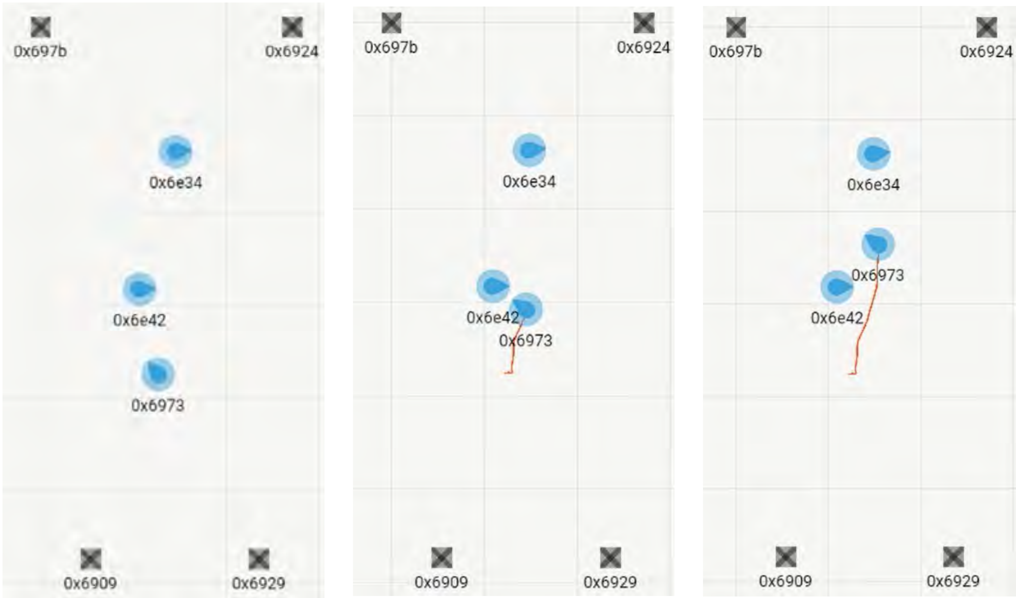
Parameter	Value
Supply voltage	3 to 5 V
Current consumption	approximately 5mA
Communication interface	I2C (TWI) - 400 kHz
Resolution	16-bits for each axis
Board dimensions	21 x 14 mm
Weight	0.9 g

4.3 Results

Real experiments included execution of a trajectory calculated by the developed algorithms for both static and dynamic obstacles in the environment.

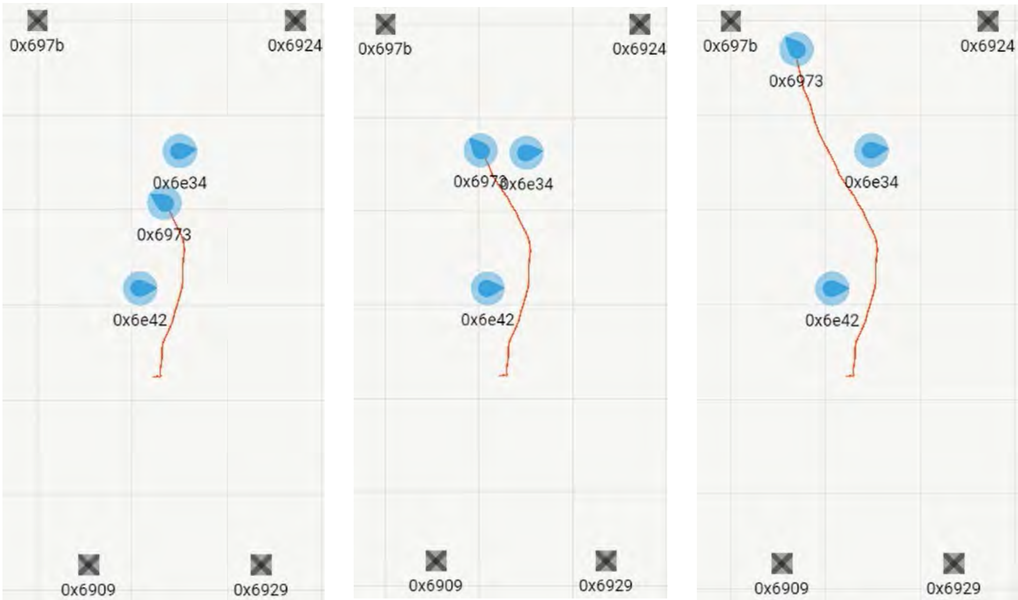
Results of an exemplary experimental test for situation with two static obstacles in the environment are shown in Fig. 4.4-4.5. The mobile platform is marked as 0x6e42 and two static obstacles are marked as 0x6973 and 0x6e34. The objects marked by 0x6909, 0x6929, 0x6924 and 0x697b are the anchors of the IPS. The results confirmed that the trajectory generated by the DAPF algorithm and executed by a mobile platform is feasible and ensures a collision-free movement between the start and final waypoint.

Results of an exemplary experimental test for situation with one dynamic obstacle in the environment are shown in Fig. 4.6-4.7. The mobile platform is marked as 0x6973 and the dynamic obstacle is marked as 0x6e42. Obtained results confirm that the trajectory calculated by the algorithm is a safe trajectory and that its execution will not cause a collision with an obstacle moving in the environment.



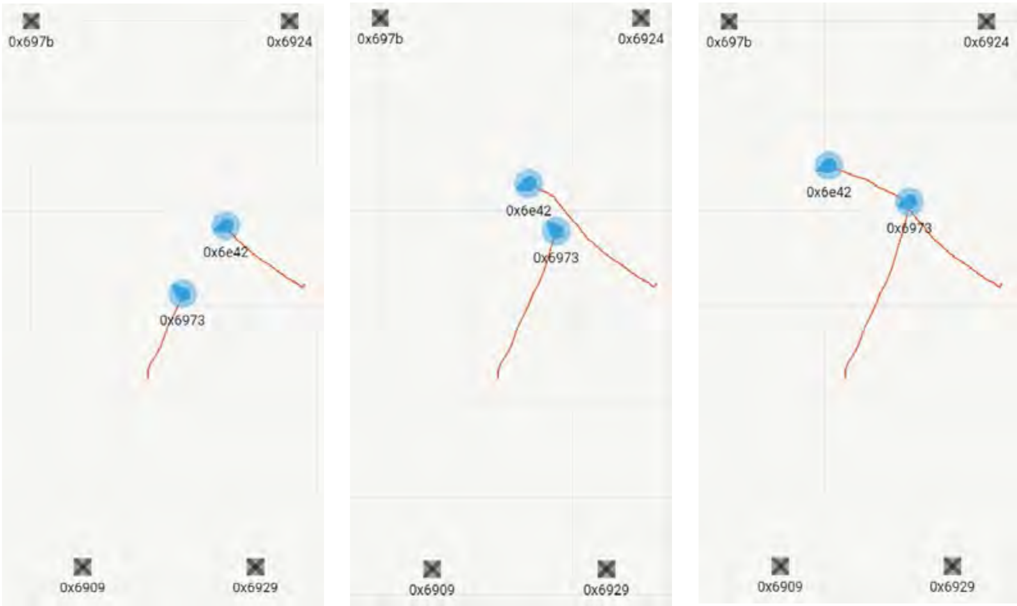
a) b) c)

Fig. 4.4. Exemplary results of real experiments with static obstacles



a) b) c)

Fig. 4.5. Exemplary results of real experiments with static obstacles



a) b) c)

Fig. 4.6. Exemplary results of real experiments with a dynamic obstacle

a) b) c)

Fig. 4.7. Exemplary results of real experiments with a dynamic obstacle

4.4 Conclusions

Conducted experiments allow to state that the trajectories calculated by the developed algorithms are feasible and enable the mobile platform to reach the final waypoint avoiding collision with static and dynamic obstacles encountered in the environment.

5. Stage IV – experimental studies (vessel *M/V Horyzont II*)

5.1 Aim

The aim of this stage of the project was to verify the developed algorithms for ship's safe trajectory planning with the use of real navigational situations registered on board a ship.

5.2 Methodology

The data, describing current navigational situation, necessary for the ship's trajectory planning algorithms include at least:

- an own ship course and speed;
- courses, speeds, bearings and distances of target ships.

The input data to the algorithms can be obtained from the radar system with ARPA in the NMEA standard. The NMEA standard is a serial asynchronous data transmission protocol used for communication between marine electronic equipment and external devices. The standard defines the data frame structure such as one start bit, eight data bits, no parity bit and one stop bit, and the transmission speed of 4800 bits per second. It also defines the structures of transmitted sentences.

```
$RAOSD,a.a,A,c.c,X,x.x,Y,y.y,z.z,N*hh<CR><LF>
RA - talker identifier: RA - radar
OSD - sentence formatter: OSD - own ship data
a.a - heading, degrees true
A - heading status: a - valid, V - invalid
c.c - vessel course, degrees true
X - course reference: W - water-referenced
x.x - vessel speed
Y - speed reference: W - water-referenced
y.y - vessel set, degrees true
z.z - vessel drift (speed)
N - speed units: N - knots

$RATTM,aa,b.b,c.c,T,t.t,x.x,T,y.y,z.z,X,n,Y,R,hhmmss. ss,M*hh<CR><LF>
RA - talker identifier: RA - radar
TTM - sentence formatter: TTM - tracked target message
aa - target number: 00 to 99
b.b - target distance from own ship
c.c - bearing from own ship
T - degrees true
t.t - target speed
x.x - target course
y.y - DCPA - distance at closest point of approach
z.z - TCPA - time to the closest point of approach [min]
X - speed, distance units
n - user data (e.g. target name)
Y - target status: L - lost, Q - query (target in the process of acquisition), T - tracking
R - reference target (used to determine own ship speed)
hhmmss.ss - time of data (UTC) M - type of acquisition: a - automatic, M - manual,
R - reported (eg.AIS)
```

a)

```
14-Jul-2018 06:40:04
$RAOSD,132.2,A,132.2,W,10.8,W,,,N*7F
$RATTM,45,5.15,169.3,T,16.32,311.9,R,3.13,15.1,N,,Q,,,M*13
$RATTM,41,1.42,345.5,T,3.50,137.8,R,0.66,21.6,N,,T,,,M*28
$RATTM,44,3.41,109.2,T,10.90,315.6,R,1.52,16.9,N,,Q,,,M*1B
$RATTM,42,1.57,108.8,T,5.02,145.7,R,0.94,-15.0,N,,T,,,M*03

14-Jul-2018 11:22:03
$RAOSD,161.4,A,161.4,W,11.6,W,,,N*70
$RATTM,70,8.29,169.3,T,16.74,340.4,R,1.28,29.4,N,,T,,,M*19
$RATTM,71,3.68,238.5,T,17.45,350.6,R,3.41,4.8,N,,T,,,M*29
$RATTM,72,2.51,249.6,T,17.67,353.2,R,2.44,2.0,N,,T,,,M*29
$RATTM,73,3.92,181.9,T,4.18,101.5,R,3.87,-9.6,N,,T,,,M*38
$RATTM,74,4.93,211.2,T,17.20,352.4,R,3.09,13.4,N,,T,,,M*16
$RATTM,75,5.21,213.4,T,17.30,354.8,R,3.25,14.1,N,,T,,,M*1C
$RATTM,76,6.78,205.9,T,17.36,352.8,R,3.71,19.6,N,,T,,,M*11
$RATTM,77,8.94,201.5,T,17.71,350.5,R,4.60,26.0,N,,T,,,M*15
$RATTM,79,2.20,206.5,T,14.38,35.2,R,0.33,9.1,N,,Q,,,M*1B
$RATTM,80,2.04,190.5,T,15.41,311.6,R,1.76,4.1,N,,Q,,,M*24
$RATTM,81,1.86,176.1,T,22.01,291.1,R,1.68,2.1,N,,Q,,,M*27
$RATTM,82,6.00,180.5,T,12.66,342.8,R,1.82,27.1,N,,Q,,,M*17
$RATTM,50,1.71,335.9,T,6.96,274.6,R,1.50,-7.1,N,,T,,,M*3C
$RATTM,64,5.16,258.6,T,8.09,314.7,R,4.28,-21.4,N,,T,,,M*0E
$RATTM,65,2.93,14.0,T,22.25,341.8,R,1.56,-6.7,N,,T,,,M*3A
$RATTM,66,2.20,67.6,T,19.68,341.6,R,2.20,-0.5,N,,T,,,M*3A
$RATTM,67,1.94,10.3,T,11.10,339.1,R,1.00,-9.0,N,,T,,,M*30
$RATTM,68,4.52,15.5,T,10.76,341.7,R,2.51,-21.0,N,,T,,,M*06
```

b)

Fig. 5.1. a) OSD and TTM sentences structure b) exemplary OSD and TTM sentences registered on board *M/V Horyzont II*

Sentences marked as OSD (Own Ship Data) and TTM (Tracked Target Message) are necessary for ship's trajectory planning. OSD and TTM sentences structures are given in Fig. 5.1a). OSD and TTM sentences of an exemplary navigational situation registered on board *M/V Horyzont II* are shown in Fig. 5.1b).

In order to obtain real navigational situations, a system for data registration was developed and installed on board the vessel *Horyzont II*. This is a Research/Training ship owned by Gdynia Maritime University. The vessel is presented in Fig. 5.2a). The vessel's main functions cover conducting research and students trainings and transporting equipment of Polish Academy of Sciences to the Polish scientific bases on Spitsbergen. The main technical parameters of *Horyzont II* are listed in Tab. 5.1. The bridge of *Horyzont II* is shown in Fig. 5.2b).

Tab. 5.1. Technical specification of Research/Training Ship *M/V Horyzont II*

Parameter	Value
Length 56.34 m	56.34 m
Breadth	11.36 m
Designed draft	3.90 m (5.33 m together with the keel)
Deadweight	288 t
Gross Tonnage	1321 BRT
Speed	12 knots
Main engine power	1280 kW
Controllable Pitch Propeller CPP	CP 65 WARTSILA , D = 2.1 m
Bow thruster	STT 10 LK SCHOTTEL - power: 125 kW
Build year and place	2000, Gdańsk
IMO Number	9231925

a)

b)

Fig. 5.2. a) Research/Training ship *M/V Horyzont II*, b) the bridge interior of the ship *M/V Horyzont II*

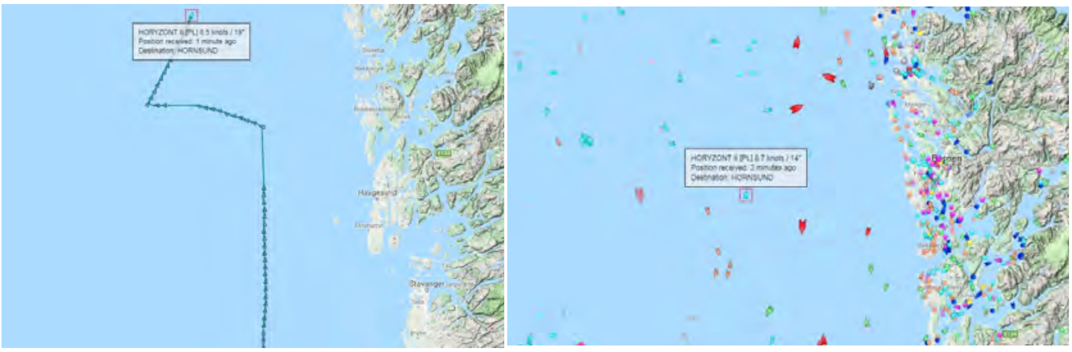
Fig. 5.3b presents the system for navigational data registration from the Furuno radar (Fig. 5.3a) with ARPA.

Real navigational data describing the current situation at sea were registered during XLI (20.06.2018 – 20.07.2018) and XLII (31.08.2018 – 30.09.2018) voyages of a ship *M/V Horyzont II* to Spitsbergen.

The track of *M/V Horyzont II* registered by Marine Traffic on the 12th of September 2018 is shown in Fig. 5.4a. An exemplary navigational situation, with marked *M/V Horyzont II* position, is presented in Fig. 5.4b.

a) b)

Fig. 5.3. a) The Furuno radar with ARPA on board the ship *M/V Horyzont II* b) the system for data registration



a) b)

Fig. 5.4. a) *M/V Horyzont II* track registered by Marine Traffic on the 12th of September 2018 b) *M/V Horyzont II* position registered by Marine Traffic on the 12th of September 2018

It should be mentioned that during analysis of the data registered on board *M/V Horyzont II* during its XLI voyage to Spitsbergen, it was noticed that some data frames real are incomplete (Fig. 5.5). Therefore there it is necessary to add a data filtering mechanism before these data can be fed into the ship's trajectory planning algorithm.

```
14-Jul-2018 05:12:18
$RAOSD,088.7,A,088.5,W,11.4,W,,,N*70
$RATTM,25,1.99,37.2,T,22.33,268.4,R,1.55,3.4,N,,Q,,,M*18
$RATTM,29,5.06,46.8,T,,,R,1.36,,N,,Q,,,M*3E
$RATTM,26,7.77,99.2,T,21.10,228.7,R,6.00,14.1,N,,Q,,,M*28
$RATTM,27,9.78,95.5,T,30.37,279.1,R,0.62,19.3,N,,Q,,,M*29
$RATTM,28,4.21,130.2,T,8.73,276.0,R,2.36,23.9,N,,Q,,,M*2B
$RATTM,30,5.61,295.5,T,,,R,4.90,,N,,Q,,,M*0F
$RATTM,24,2.93,305.2,T,21.10,269.9,R,1.69,-6.8,N,,Q,,,M*07
```

Fig. 5.5. An example of incomplete data registered on board *M/V Horyzont II*

5.3 Results

The ship's safe trajectory planning algorithms described in Section 2 were tested with the use of navigational data registered on board the ship *M/V Horyzont II*. Two exemplary situations were chosen for the presentation in this report. These are encounters with four and fourteen target ships. The dimensions of the target ship domain used in the calculations were: distance towards the bow - 1.05 NM, distance of amidships - 0.65 NM, distance towards the stern - 0.4 NM, distance towards the port side - 0.4 NM and distance towards the starboard side - 0.65 NM. The following parameters of ACO algorithm were used: $\tau_0 = 1$, $\rho = 0.1$, $\alpha = 1$, $\beta = 2$, *max_it = 20* and *ant_num* = 10.

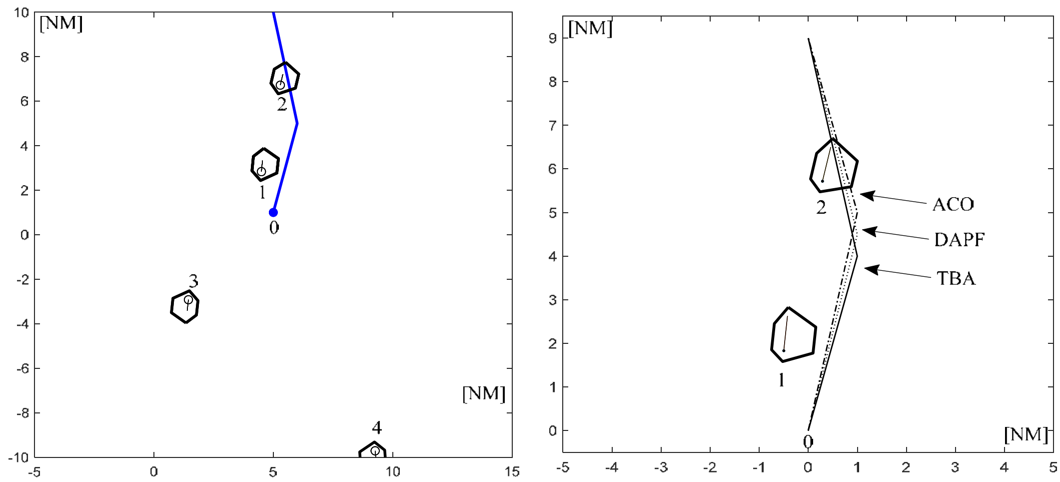
Real navigational situation with 4 target ships

Tab. 5.2. Input data for an encounter with 4 target ships

Ship	Course [º]	Speed [kn]	Distance [nm]	Bearing [º]
0	132.2	10.7	-	-
1	137.5	3.48	1.39	346.1
2	144.3	5.02	1.61	109.7
3	318	19.04	12.55	123.5
4	307.8	17.02	5.02	170.9

Tab. 5.3. Input data for an encounter with 14 target ships

Ship	Course [º]	Speed [kn]	Distance [nm]	Bearing [º]
0	160.9	11.6	-	-
1	273.6	7.62	1.67	338.1
2	315.2	7.87	5.1	257.8
3	340.9	22.43	2.73	16.4
4	341.2	19.44	2.2	73.1
5	337.7	11.09	1.84	12.1
6	340.8	10.71	4.43	16.3
7	340.1	16.69	8.44	169.1
8	350.2	17.47	3.74	236.3
9	353.1	17.53	2.55	246.1
10	134.2	5.37	3.93	182.4
11	352.1	17.22	5.05	210.1
12	354.8	17.25	5.33	212.4
13	353.5	17.16	6.92	205.1
14	353.4	17.34	9.08	200.8

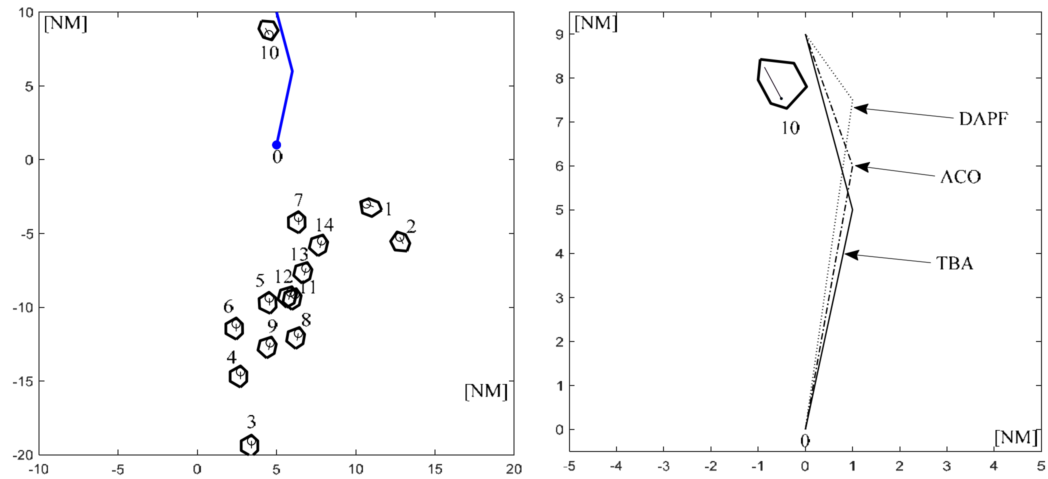


a) b)

Fig. 5.6. Ship's safe trajectory for an encounter with 4 target ships. Solution calculated by a) the TBA b) three different developed algorithms

Tab. 5.4. Solutions for an encounter with 4 target ships

Algorithm	Course [°]	Distance [nm]	Run time [s]
ACO	144, 118	9.22	13.99
DAPF	145, 120	9.22	0.99
TBA	146, 121	9.22	0.27



a) b)

Fig. 5.7. Ship's safe trajectory for an encounter with 14 target ships. Solution calculated by a) the TBA b) three different developed algorithms

Tab. 5.5. Solutions for an encounter with 14 target ships

Algorithm	Course [°]	Distance [nm]	Run time [s]
ACO	170, 142	9.25	20.91
DAPF	168, 127	9.37	0.71
TBA	172, 147	9.22	0.32

The first described test case is an encounter situation between an own ship and four target ships. Input data for this navigational situation are shown in Tab. 5.2. The comparison of numerical results for different developed algorithms including the length of the safe trajectory in nautical miles, calculated course of an own ship at consecutive stages of its movement along the determined trajectory in degrees and the run time of the algorithm in seconds are listed in Tab. 5.4.

An own ship safe trajectory obtained for the TBA, for which the shortest trajectory was calculated, are shown in Fig. 5.6a) along with the instantaneous positions of the target ships. Trajectories calculated by different algorithms are compared graphically in Fig. 5.6b). A safe trajectory was determined by all of the considered algorithms. Solutions obtained by different algorithms do not differ significantly. They are characterized by the same length of an own ship trajectory with slightly different course changes. The shortest run time was achieved by the TBA (about 0.3 second).

Real navigational situation with 14 target ships

The second presented exemplary situation is an encounter between an own ship and fourteen target ships.

Input data for this navigational situation are shown in Tab. 5.3. Numerical results are listed in Tab. 5.5. Graphical results obtained by the TBA are presented in Fig. 5.7a). In Fig. 5.7b) a comparison of trajectories obtained by different algorithms is shown.

All algorithms have found an own ship safe trajectory for this navigational situation. The solutions calculated by different algorithm also do not vary considerably. For this test case TBA also reached the shortest run time (about 0.3 second) and achieved the shortest trajectory.

5.4 Conclusions

Analysis of obtained results allow to formulate the following concluding remarks:

- the TBA algorithm proved to be more effective in terms of the optimality of solution and the run time of the algorithm for most of the test cases, a heuristic algorithm (ACO) could be useful for very complex situations, where the deterministic approach would have difficulties in finding a solution in a reasonable amount of time;
- carried out simulation studies with real navigational data registered on board the ship *M/V Horyzont II* confirmed applicability of implemented algorithms for solving the ship's trajectory planning problem;
- tests in real conditions carried out on board the ship *M/V Horyzont II* showed that application of a hard drive of SSD type is required due to particularly difficult weather conditions during storms, which caused a damage to a hard drive of a HDD type used for data storage during the XLII voyage of *M/V Horyzont II* to Spitsbergen;
- real data registration from the radar with ARPA proved that it would be needed to apply also data from AIS and ENC;
- real data registration revealed also that an additional data filtering mechanism is needed, because sometimes some part of data can be incomplete;
- analysis of achieved results of the algorithms verification (solutions feasibility, run time of the algorithms) enables to state that the developed ship's trajectory planning algorithms are suitable for application in a Decision Support System or a GNC system of an autonomous vessel.

References

[1] Allianz Global Corporate and Specialty SE, "Safety and shipping review", (2018), [viewed 7 May 2019], available from: https://www.agcs.allianz.com/

[2] Kongsberg Maritime AS, "Autonomous ship project", (2018), Kongsberg, Norway, [viewed 7 May 2019], available from: https://kmdoc.kongsberg.com/

[3] China Classification Society, Zhuhai Municipal Government, Wuhan University of Technology and Zhuhai Yunzhou Smart Co., "Wanshan marine test field", (2018), China, [viewed 7 May 2019], available from: https://www.marineinsight.com/

[4] Fossen T.I., *"Handbook of Marine Craft Hydrodynamics and Motion Control"*, John Wiley and Sons Ltd., (2011).

[5] Goodwin E.M., "A Statistical Study of Ship Domains", *Journal of Navigation*, Vol. 28, (1975), pp 328–344.

[6] Cockcroft A. and Lameijer J., "*A Guide to the Collision Avoidance Rules*", Butterworth-Heinemann, (2012).

[7] Beni G. and Wang J., "Theoretical problems for the realization of distributed robotic systems", *Proc. IEEE International Conference on Robotics and Automation*, Sacramento, USA, Vol. 3, (1991), pp 1914–1920.

[8] Bonabeau E., Dorigo M. and Theraulaz G., *"Swarm Intelligence: From Natural to Artificial Systems"*, Oxford University Press Inc., (1999).

[9] Goldberg D. E., *"Genetic Algorithms in Search, Optimization and Machine Learning"*, Addison-Wesley Longman Publishing Co. Inc., (1989).

[10] Khatib O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", *The International Journal of Robotics Research*, Vol. 5, No. 1, (1986), pp 90–98.

[11] Pozyx labs BVBA, Pozyx labs indoor positioning system, https://www.pozyx.io, accessed 4 December 2018.

[12] Zhiwei Robotics Corp, DFRobot Pirate-4WD mobile platform, https://www.dfrobot.com, accessed 4 December 2018.

[13] Szlapczynski R., "A new method of ship routing on raster grids, with turn penalties and collision avoidance", *Journal of Navigation,* Vol. 59, (2006), pp 27–42.

[14] Mohamed-Seghir M., "The branch-and-bound method and genetic algorithm in avoidance of ships collisions in fuzzy environment", *Polish Maritime Research*, Vol. 19, (2012), pp 45–49.

[15] Tam C. and Bucknall R., "Cooperative path planning algorithm for marine surface vessels", *Ocean Engineering*, Vol. 57 , (2013), pp 25–33.

[16] Lisowski J., "Computational intelligence methods of a safe ship control", *Procedia Computer Science*, Vol. 35, (2014), pp 634–643.

[17] Liu Y. and Bucknall R., "Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment", *Ocean Engineering*, Vol. 97, (2015), pp 126–144.

[18] Lisowski J., "The sensitivity of state differential game vessel traffic model", *Polish Maritime Research*, Vol. 23, Issue 2, (2016), pp 14–18.

[19] Lisowski J., "Sensitivity of Safe Trajectory in a Game Environment on Inaccuracy of Radar Data in Autonomous Navigation", *Sensors*, Vol. 19(8), (2019).

[20] Lisowski J., "Multi-criteria optimization of multi-step matrix game in collision avoidance of ships", *TransNav – The International Journal on Marine Navigation and Safety of Sea Transportation*, Vol. 13(1), (2019), pp 125–131.

[21] Tsou M., Kao S. and Su C., "Decision support from genetic algorithms for ship collision avoidance route planning and alerts", *Journal of Navigation*, Vol. 63, (2010), pp 167–182.

[22] Perera L., Carvalho J. and Soares C. G., "Bayesian network based sequential collision avoidance action execution for an ocean navigational system", *IFAC Proceedings Volumes*, Vol. 43, (2010), pp 266–271.

[23] Lisowski J. and Mohamed-Seghir M., "Comparison of Computational Intelligence Methods Based on Fuzzy Sets and Game Theory in the Synthesis of Safe Ship Control Based on Information from a Radar ARPA System", *Remote Sensing*, Vol. 11(1), (2019).

[24] Szlapczynski R. and Szlapczynska J., "Customized crossover in evolutionary sets of safe ship trajectories", *International Journal of Applied Mathematics and Computer Science,* Vol. 22 (4), (2012), pp 999–1009.

[25] Kuczkowski L. and Smierzchalski R., "Comparison of single and multi-population evolutionary algorithm for path planning in navigation situation", *Solid State Phenomena*, Vol. 210, (2013), pp 166–177.

[26] Tam C. and Bucknall R., "Path-planning algorithm for ships in close-range encounters", *Journal of Marine Science and Technology*, Vol. 15 , (2010), pp 395–407.

[27] Tsou M. and Hsueh C., "The study of ship collision avoidance route planning by ant colony algorithm", *Journal of Marine Science and Technology-TAIWAN*, Vol. 18 , (2010), pp 746–756.

[28] Li-Jia C. and Li-Wen H., "Ship Collision Avoidance Path Planning by PSO Based on Maneuvering Equation", In: Zhang Y. (eds) Future Wireless Networks and Information Systems, *Lecture Notes in Electrical Engineering*, Vol. 144., (2012), Springer, Berlin, Heidelberg, pp 675–682.

[29] Xue Y., Clelland D., Lee B. and Han D., "Automatic simulation of ship navigation", *Ocean Engineering*, Vol. 38 , (2011), pp 2290–2305.

[30] Naeem W., Henrique S.C. and Hu L., "A Reactive COLREGs-Compliant Navigation Strategy for Autonomous Maritime Navigation", *IFAC-PapersOnLine*, Vol. 49(23), (2016), pp 207–213.

[31] Naeem W., Irwin G. and Yang A., "Colregs-based collision avoidance strategies for unmanned surface vehicles", *Mechatronics*, Vol. 22 , (2012), pp 669–678.

[32] Lee T., Kim H., Chung H., Bang Y. and Myung H., "Energy efficient path planning for a marine surface vehicle considering heading angle", *Ocean Engineering*, Vol. 107, (2015), pp 118–131.

[33] Simsir U., Amasyal M.F., Bal M., Celebi U.B. and Ertugrul S., "Decision support system for collision avoidance of vessels", *Applied Soft Computing* Vol. 25, (2014), pp 369–378.

[34] Statheros T., Howells G. and Maier K.M., "Autonomous ship collision avoidance navigation concepts, technologies and techniques", *Journal of Navigation,* Vol. 61(1), (2008), pp 129-142.

[35] Tam C., Bucknall R. and Greig A., "Review of collision avoidance and path planning methods for ships in close range encounters", *Journal of Navigation*, Vol. 62(3), (2009), pp 455-476.

[36] Campbell S., Naeem W. and Irwin G., "A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres", *Annual Reviews in Control*, Vol. 36(2), (2012), pp 267-283.

Attachment

Attachment: IAMU AGA 2018 presentation

Research on New Obstacle Avoidance Algorithms for Ships (NOAA) – interim report

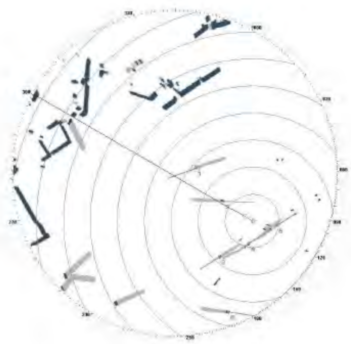
Agnieszka Lazarowska
Gdynia Maritime University, Poland

19th Annual General Assembly of the International Association of Maritime Universities
IAMU AGA 2018

Barcelona, Spain
18 October 2018

Contents

- What?
- Why?
- How?
- The concept validation
- Summary and future research directions …



2

The research project

Research project for Young Academic Staff in FY2018

Research Title: Research on New Obstacle Avoidance Algorithms for Ships (NOAA)

Duration: 1 May 2018 – 31 May 2019

Main goal: development of new solutions
to achieve safer shipping and
development of autonomous navigation

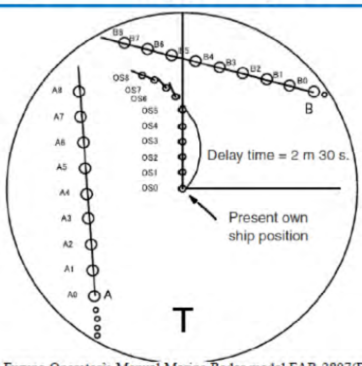
3

The problem definition

Ship's trajectory planning

Decision Support System

Autonomous Vessel

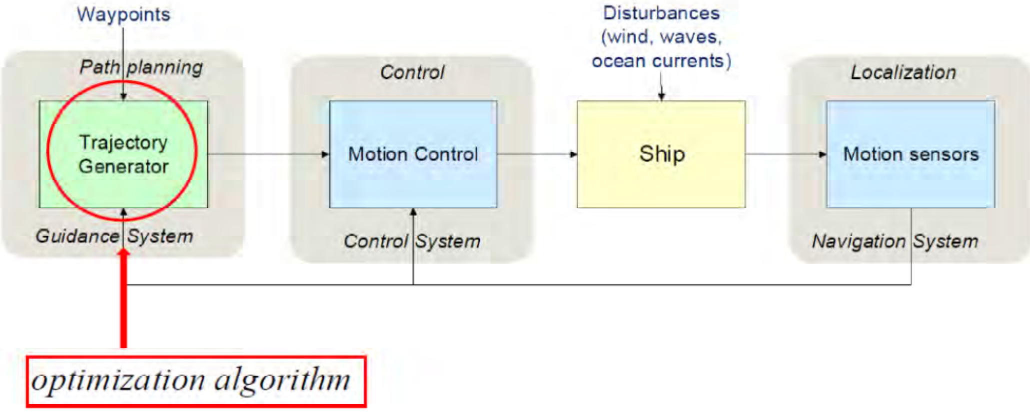


source: Furuno Operator's Manual Marine Radar model FAR-2807(D) Series FAR-2107(-BB,-D) Series, 2011.

source: https://www.rolls-royce.com/products-and-services/marine/ship-intelligence.aspx

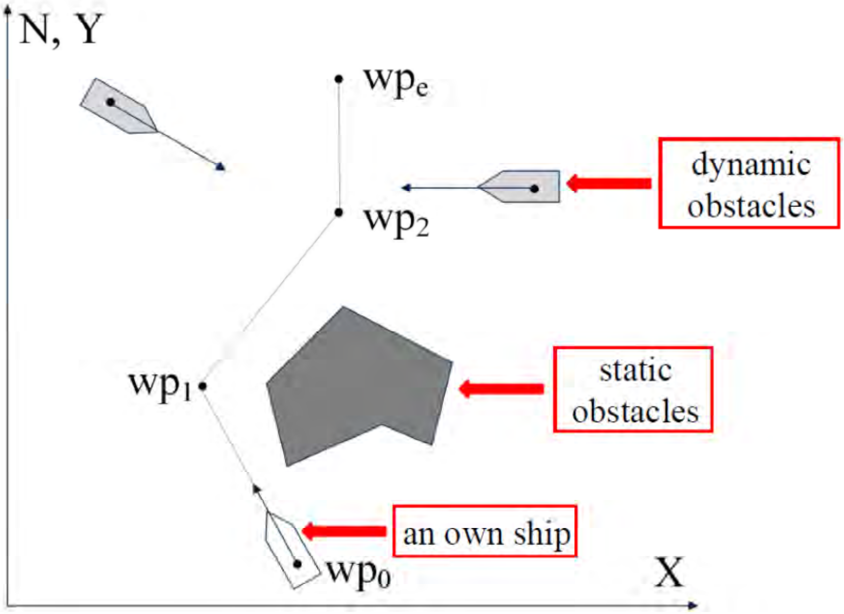
4

The problem definition



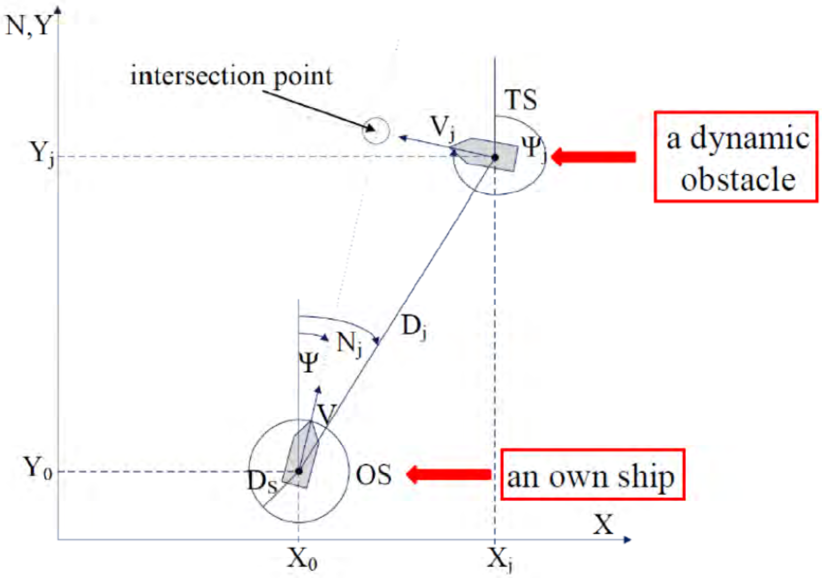
5

The problem definition



6

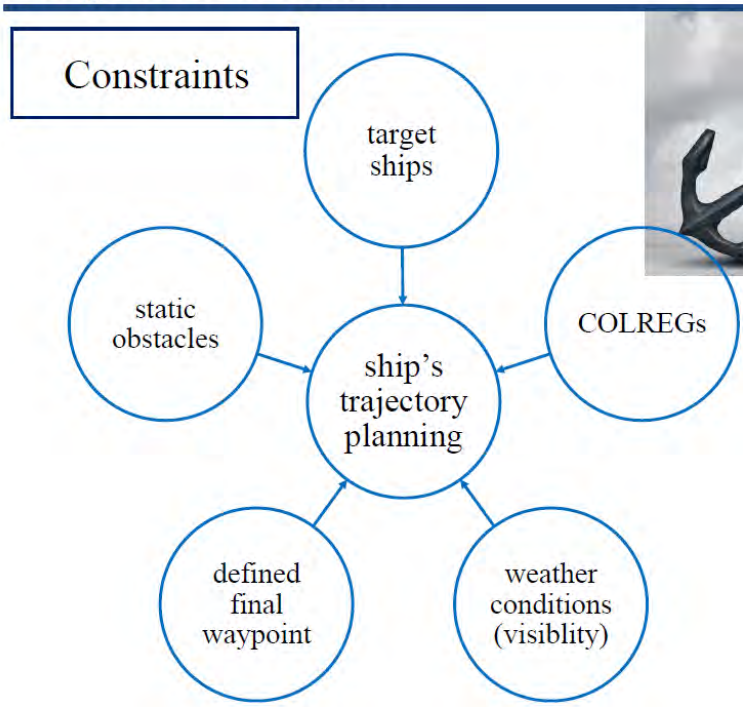
The problem definition



$q_{OS} = (X_0, Y_0, \Psi, V)$ $q_{TSj} = (X_j, Y_j, \Psi_j, V_j, N_j, D_j)$

7

The problem definition



8

Ship's trajectory planning algorithm

The method requirements:

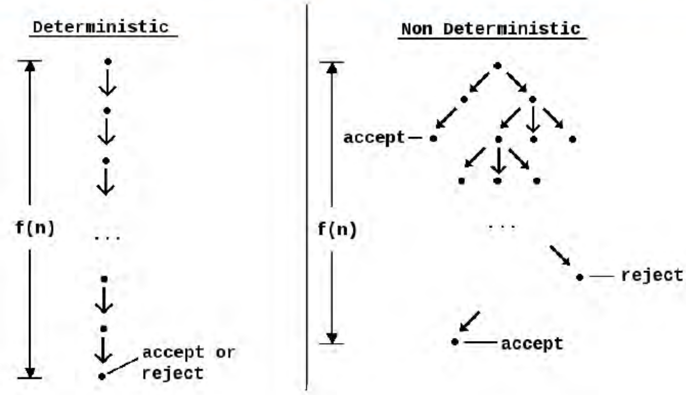
- safety
- efficiency
- reliability
- near-real run time
- simplicity

9

Ship's trajectory planning algorithm

Proposed methods

- deterministic approach
- heuristic approach

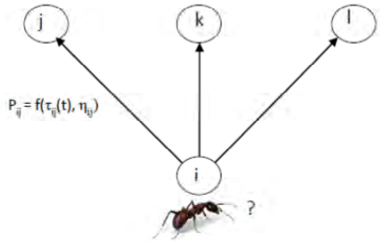


10

Ship's trajectory planning algorithm

Proposed heuristic method – Ant Colony Optimization (ACO)

Probability of ant's next move

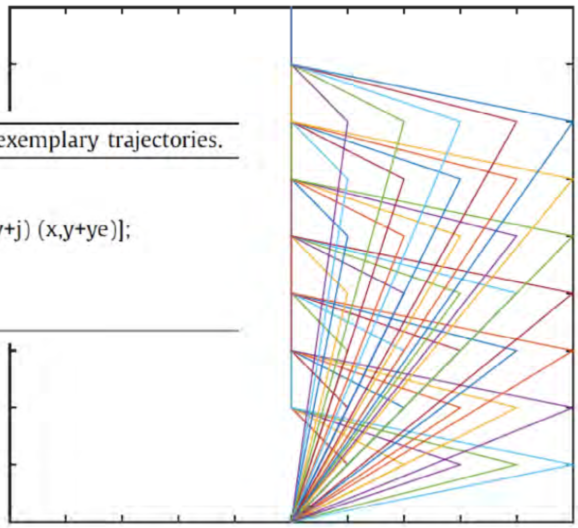


$$P_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}}$$

11

Ship's trajectory planning algorithm

Proposed deterministic method – Trajectory Base Algorithm (TBA)



Algorithm 1 Rule for generation of exemplary trajectories.

```
for j:=2 to 8 do
  for k:=1 to 5 do
    trajectory(i)=[(x,y), (x+k,j) (x,y+j) (x,y+ye)];
    i:=i+1;
  end for
end for
```

12

Validation of the proposed concept

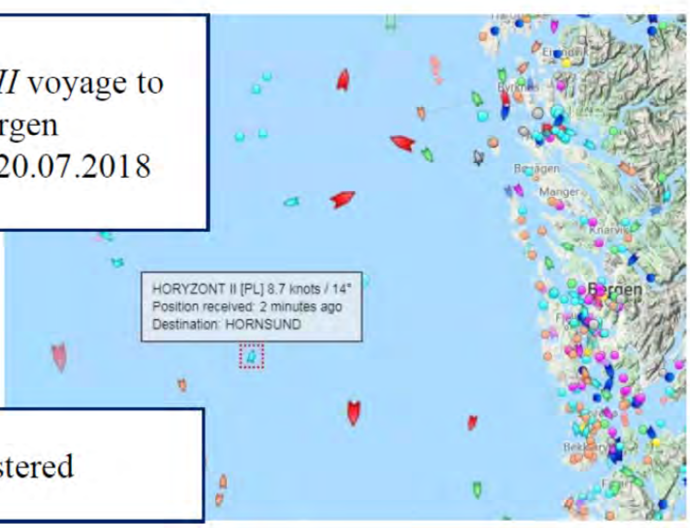
Real data registration

13

Validation of the proposed concept

Real data registration

XLI *Horyzont II* voyage to Spitsbergen
20.06.2018 – 20.07.2018



data registered

source: https://www.marinetraffic.com

14

Validation of the proposed concept

Real data registration - exemplary results

14-Jul-2018 09:36:53
$RADSD,160.5,A,160.5,W,11.7,W,,,N*71
$RATTM,50,1.06,348.2,T,0.59,331.8,R,0.30,-103.6,N,,T,,,M*33
$RATTM,60,4.87,92.2,T,17.41,349.2,R,4.74,3.8,N,,T,,,M*14
$RATTM,49,14.73,160.1,T,7.21,149.1,R,2.80,-120.3,N,,T,,,M*07
$RATTM,57,5.72,141.9,T,20.38,339.6,R,1.73,16.0,N,,T,,,M*18
$RATTM,58,9.69,143.6,T,22.48,338.9,R,2.56,25.0,N,,T,,,M*13

14-Jul-2018 09:37:28
$RADSD,160.7,A,160.7,W,11.7,W,,,N*71
$RATTM,50,1.07,348.3,T,0.62,343.5,R,0.09,-103.1,N,,T,,,M*3E
$RATTM,60,4.83,90.4,T,17.15,349.4,R,4.75,3.2,N,,T,,,M*18
$RATTM,49,14.80,160.1,T,7.16,152.2,R,2.02,-122.9,N,,T,,,M*04
$RATTM,57,5.53,141.3,T,20.39,339.4,R,1.72,15.5,N,,T,,,M*15
$RATTM,58,9.48,143.3,T,22.30,338.2,R,2.44,24.7,N,,T,,,M*14

14-Jul-2018 09:38:04
$RADSD,160.4,A,160.4,W,11.7,W,,,N*71
$RATTM,57,5.34,140.7,T,20.38,339.1,R,1.69,14.9,N,,T,,,M*12
$RATTM,58,9.27,143.0,T,22.30,337.9,R,2.38,24.1,N,,T,,,M*17
$RATTM,60,4.81,88.5,T,16.99,350.2,R,4.76,2.5,N,,T,,,M*1C
$RATTM,49,14.87,160.0,T,7.22,150.7,R,2.41,-121.9,N,,T,,,M*06
$RATTM,50,1.08,348.5,T,0.60,4.5,R,0.30,-102.6,N,,T,,,M*39
$RATTM,60,4.81,88.3,T,16.96,350.1,R,4.76,2.5,N,,T,,,M*16
$RATTM,49,14.88,160.0,T,7.22,150.8,R,2.39,-122.0,N,,T,,,M*03

14-Jul-2018 09:38:40
$RADSD,160.4,A,160.4,W,11.7,W,,,N*71
$RATTM,58,9.06,142.7,T,22.40,337.4,R,2.30,23.5,N,,T,,,M*13
$RATTM,50,1.08,348.9,T,0.50,5.0,R,0.30,-124.4,N,,T,,,M*34
$RATTM,60,4.79,86.5,T,16.64,350.1,R,4.76,1.9,N,,T,,,M*1B
$RATTM,49,14.95,160.0,T,7.25,151.7,R,2.16,-122.4,N,,T,,,M*0F
$RATTM,57,5.13,140.0,T,20.36,338.8,R,1.66,14.3,N,,T,,,M*13

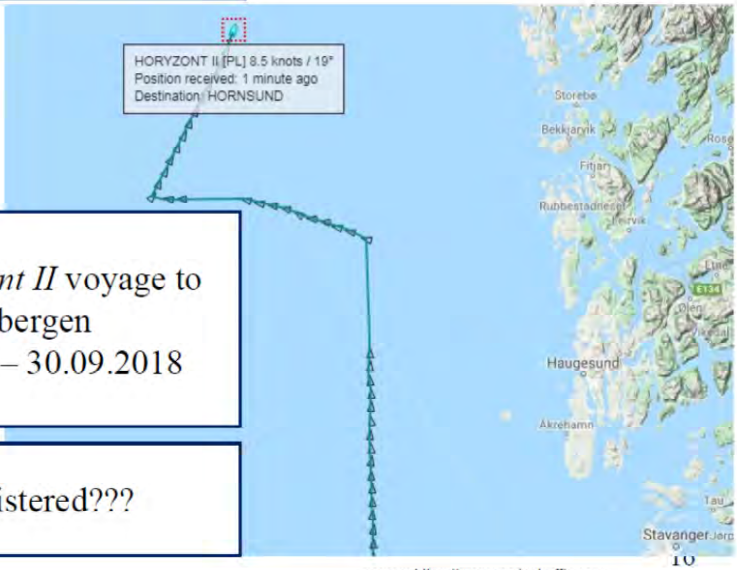
14-Jul-2018 09:39:17
$RADSD,160.4,A,160.4,W,11.6,W,,,N*70
$RATTM,50,1.08,348.4,T,0.42,351.9,R,0.07,-155.2,N,,T,,,M*35
$RATTM,60,4.78,84.5,T,16.56,350.9,R,4.78,1.1,N,,T,,,M*17
$RATTM,49,15.01,159.9,T,7.34,149.4,R,2.73,-120.6,N,,T,,,M*09
$RATTM,57,4.95,139.2,T,20.34,339.1,R,1.69,13.7,N,,T,,,M*16
$RATTM,58,8.83,142.2,T,22.37,337.5,R,2.32,22.9,N,,T,,,M*14
$RATTM,49,15.02,159.9,T,7.35,149.3,R,2.77,-120.5,N,,T,,,M*0B

14-Jul-2018 09:39:53
$RADSD,161.2,A,161.2,W,11.6,W,,,N*70
$RATTM,49,15.09,159.8,T,7.29,148.7,R,2.91,-121.9,N,,T,,,M*0C
$RATTM,57,4.76,138.4,T,20.31,339.4,R,1.70,13.1,N,,T,,,M*12
$RATTM,58,8.61,141.9,T,22.44,337.5,R,2.31,22.2,N,,T,,,M*1C
$RATTM,50,1.09,348.6,T,0.42,1.2,R,0.24,-151.1,N,,T,,,M*3D
$RATTM,57,4.75,138.3,T,20.27,339.3,R,1.70,13.1,N,,T,,,M*16
$RATTM,58,8.60,141.9,T,22.40,337.4,R,2.31,22.2,N,,T,,,M*18
$RATTM,60,4.78,82.5,T,16.52,351.9,R,4.78,0.2,N,,T,,,M*16

15

Validation of the proposed concept

Real data registration



XLII *Horyzont II* voyage to Spitsbergen 31.08.2018 – 30.09.2018

data registered???

16

source: https://www.marinetraffic.com

Validation of the proposed concept

Real experiments using mobile platforms

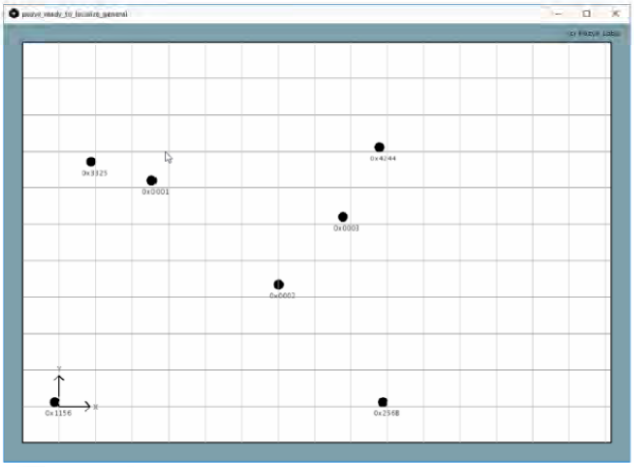
- Indoor Positioning System (IPS) for localization of the moving objects from Pozyx labs
- mobile platforms DFRobot Pirate-4WD

source: https://www.pozyx.io

17

Validation of the proposed concept

Real experiments using mobile platforms



source: https://www.pozyx.io

18

Validation of the proposed concept

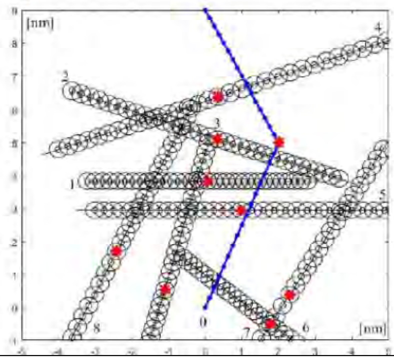
Real experiments using mobile platforms

19

Conclusions

The most significant advantages of the methods:

- simplicity
- handling static and dynamic obstacles
- computational time < 1minute
- repeatability of solution
- consideration of the user's preferences (size of the ship's domain)



20

Conclusions

Things done:

- simulation tests of the algorithms in Matlab environment
- construction of a system for real experiments with mobile platforms
- preliminary tests on mobile platforms
- real data registration on-board *Horyzont II*

21

Conclusions

Things in progress:

- analysis of data registered on-board *Horyzont II*
- simulation tests of the algorithms in Matlab environment with data registered on-board *Horyzont II*
- development of a system for real experiments with mobile platforms
- extensive tests on mobile platforms
- preparation for implementation of the system on-board *Horyzont II*


```
14-Jul-2018 12:15:04
$RAOSD,159.7,A,160.0,W,11.5,W,,,N*7E
$RATTM,95,10.17,154.9,T,22.58,339.7,R,0.85,26.9,N,,T,,,M*22
$RATTM,96,11.37,197.5,T,10.69,337.3,R,7.34,48.7,N,,T,,,M*23
$RATTM,85,4.20,308.8,T,15.99,354.5,R,3.00,-11.0,N,,T,,,M*36
$RATTM,86,4.38,299.5,T,15.90,351.2,R,3.44,-10.3,N,,T,,,M*31
$RATTM,87,3.85,180.3,T,9.21,315.7,R,2.70,17.9,N,,T,,,M*28
$RATTM,90,2.39,83.4,T,17.57,345.9,R,2.37,1.1,N,,T,,,M*14
$RATTM,93,2.35,161.2,T,14.31,4.7,R,0.94,9.1,N,,T,,,M*28
```


22

Conclusions

Things planned to be done:

- implementation of the system on-board *Horyzont II*
- real experiments on-board *Horyzont II* during voyages in 2019
- presentation of results on a maritime conference

23

Conclusions

Contributions of the presented research:

- improved safety of shipping
- improved efficiency of ship transport
- development of intelligent ship concept
- development of autonomous navigation systems

source: https://www.rolls-royce.com/products-and-services/marine/ship-intelligence.aspx

24

Thank you for your attention

Gdynia Maritime University
81 – 87 Morska Str.
81 – 225 Gdynia Poland

a.lazarowska@we.umg.edu.pl
www.umg.edu.pl

International Association of Maritime Universities

Meiwa Building 8F, 1-15-10 Toranomon, Minato-ku, Tokyo 105-0001, Japan

Tel : 81-3-6257-1812   E-mail : info@iamu-edu.org   URL : http://www.iamu-edu.org

ISBN No. 978-4-907408-30-5