

Article

Safe Control of an Autonomous Ship in a Port Based on Ant Colony Optimization and Linear Matrix Inequalities

Monika Rybczak ^{*,†}  and Agnieszka Lazarowska ^{*,†} 

Department of Ship Automation, Gdynia Maritime University, 81-87 Morska St., 81-225 Gdynia, Poland

* Correspondence: m.rybczak@we.umg.edu.pl (M.R.); a.lazarowska@we.umg.edu.pl (A.L.)

† These authors contributed equally to this work.

Abstract: The autonomous operation of a device or a system is one of the many vital tasks that need to be achieved in many areas of industry. This is also true for maritime transport. This paper introduces an approach developed in order to achieve the autonomous operation of a ship in a port. A safe trajectory was calculated with the use of the Ant Colony Optimization (ACO) algorithm. The ship motion control was based on two controllers: the master overriding trajectory controller (OTC) and the slave low speed controller based on the Linear Matrix Inequalities (LMI) method. The control object was the model of a Very Large Crude Carrier *Blue Lady*. The results of our simulation tests, which show the safe trajectories calculated by the ACO algorithm and executed by the ship using the designed controllers (OTC and LMI), are presented in this paper. The results present maneuvers executed by the *Blue Lady* ship when at port. The area where the tests were conducted is located in the Ship Handling, Research and Training Center of the Foundation for Shipping Safety and Environmental Protection on the Lake Silm in Kamionka, Poland.

Keywords: autonomous ship; ant colony optimization; linear matrix inequalities; ship motion control; safe navigation



Citation: Rybczak, M.; Lazarowska, A. Safe Control of an Autonomous Ship in a Port Based on Ant Colony Optimization and Linear Matrix Inequalities. *Electronics* **2024**, *13*, 3525. <https://doi.org/10.3390/electronics13173525>

Academic Editors: Luis Pires, Ricardo Santos and Joaquim Monteiro

Received: 2 August 2024

Revised: 26 August 2024

Accepted: 28 August 2024

Published: 5 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A device or a system is characterized by its autonomy when it is able to make decisions and take actions independently without any external control. Recent advances in the development of methods and algorithms for the autonomous operation of a ship include a proposal for a visual–acoustic navigation system for Unmanned Surface Vehicles (USVs) maneuvering in port and waterway areas [1] as well as a solution for creating a map of open water areas using electro-optical sensors and a convolutional neural network [2]. Other recent approaches include the development of a navigation system within the AN-CHOR project. The system proposes routes. It is also equipped with a collision warning system [3]. In [4], the authors introduced Denmark’s first autonomous harbor bus, called the Greenhopper, by applying the collision and grounding avoidance scheme, which they named the Short Horizon Planner (SHP). In [5], a path planning method for a ship called an elastic force contraction algorithm (EFCA) was proposed. The researchers also used the minimum set method (MSM) in order to remove redundant nodes in the initial path planned by the EFCA and further optimize the solution.

Other recent path planning methods for ships include a virtual potential field (VPF)-based method [6,7] and a method based on dynamic time–space network trees for solving encounters with multiple ships [8]. In [9], the authors introduced a dynamic adaptive decision-making method for application in traffic separation scheme (TSS) waters. The researchers in [10] present a survey of recent path planning methods for Maritime Autonomous Surface Ships (MASS). The above work presents a number of solutions proposed in the literature related to autonomous ship control. The following papers address the autonomous trajectory planning of maritime objects, including a review of various such

methods [11–15]. The reviews presented are broadly developed in terms of research areas, such as simulations or experiments. They concern a breakdown of autonomous trajectory methods or calculations related to precision ship control, but also present review research based on swarm behaviour, among other things.

LMI techniques were applied for solving a large number of control problems like estimation or stability analysis, formulated as convex problems [16]. In [17] the authors stated that an LMI solution of the gain assignment for feedback linear time-invariant systems is well-suited in the design of PID controllers for autopilot functions. Similarly, the solution was used in the research presented in [18]. The authors of this paper solved the yaw control problem in a ship's steering capacity by using a complex fourth-order Nomoto model. Linear matrix inequality-based robust control was also applied in this study. The authors improved the tracking ability and robust stability of H_∞ control with the use of an LMI. They proved the robust performance of the control system in the presence of disturbances and measurement noise. This type of control was also justified for use in controlling a multidimensional maritime object in [19,20].

In the area of designing controllers for autonomous ships, recent works also include a human-like constraint-adaptive model predictive control (MPC), with risk-tunable control barrier functions (CBF), proposed in [21], and a sliding mode control, which was applied to control the movement at low speeds of a Very Large Crude Carriage (VLCC) in port maneuvers [22].

This paper introduces an approach applied in order to achieve the autonomous operation of a ship in a port. A safe trajectory is planned with the use of the Ant Colony Optimization algorithm, considering static navigational obstacles such as the quay, shallows, and buoys. The ship's movement along the calculated trajectory is carried out with the use of a master overriding trajectory controller (OTC) and a slave low speed controller, designed using the Linear Matrix Inequalities (LMI) method.

The main contributions of this paper include:

- The development of a control system for the autonomous operation of a ship in a port, composed of two controllers: the master overriding trajectory controller and the slave low speed controller based on Linear Matrix Inequalities;
- The development of a method for safe trajectory planning in a port based on the Ant Colony Optimization;
- Simulation tests carried out using a model of a Very Large Crude Carrier *Blue Lady*, covering maneuvers in the port, located in the Ship Handling, Research and Training Center of the Foundation for Shipping Safety and Environmental Protection on Lake Słom in Kamionka, Poland.

The rest of this paper is organized as follows. In Section 2, the control object—the model of the *Blue Lady* ship—is described in details. In Section 3, the safe trajectory planning algorithm based on the Ant Colony Optimization is introduced. The pseudocodes of different procedures applied in the algorithm are also presented in this section. Section 4 describes the developed multi-dimensional ship control system, composed of the master overriding trajectory controller (OTC) and the slave low speed controller, designed using the Linear Matrix Inequalities (LMI). The results of our simulation tests are presented and discussed in detail in Section 5. The summary of the research and future research directions are given in the Conclusions Section 6.

2. The Control Object—Ship *Blue Lady*

This work focuses on trajectories made in a computer simulation. The ship itself is a simulated model of a training ship called the *Blue Lady*. Studies of its dynamics and a lot of research was carried out by several researchers from the Maritime University in Gdynia. Most importantly, works concerning parameters of the ship [23], kinematics of the ship [24,25], and the inclusion of the effects of wind on the ship [26,27] were carried out.

The ship model is located in Poland, specifically in Iława Kamionka, where the Foundation for Shipping Safety and Environmental Protection is based. The Foundation is

recommended by IMO as a center that provides training for navigators to improve the safety of ship navigation in various maneuvering conditions, including in ports. The training center has seven different ship models at its disposal, including the VLCC-type tanker training vessel shown in Figure 1.



Figure 1. Silhouette of the actual training ship in the harbor of Lake Silm in Iława Kamionka.

The research presented in this article is based on simulations and focuses on a control object, a model of the ship the *Blue Lady* made at a scale of 1:24 against a real VLCC (Very Large Crude Carrier) class tanker. The computer simulations of its kinematics and dynamics were built and presented in [23,28–30]. Parameters of the object, i.e., the training model of the *Blue Lady* ship are as follows:

- Length—13.75 m;
- Width—2.38 m;
- Speed—3.1 kn;

The ship's propulsion system, also called the ship's control system, is composed of the following elements:

- The main drive, marked as *ng*, operating within the range of (−200; +480) rpm;
- The fin rudder, operating in the range of (−35; +35) deg;
- Four thrusters to simulate the operation of the tugs; these are written below:
 - Two tunnel thrusters (bow, stern), designated as *sstd*, *sstr*, operating in the range of (−1; +1);
 - Two rotary thrusters (bow, stern), designated as *ssod*, *ssor*, operating in the ranges of (0; +1), where the swing angle is in the range of (−120; +120) deg for the bow and (+60; +300) deg for the stern.

The simulation model of the training vessel the *Blue Lady*, presented in the works of [24,31], is based on a 3DOF model. This model allows for the presentation of motion in three areas without taking into account lateral heel, longitudinal heel, and nuisance when performing tests on the lake. The dynamic assumptions are of the form

$$m(\dot{u} - rv) = X_D, \quad (1)$$

$$m(\dot{v} - ru) = Y_D, \quad (2)$$

$$I_z \dot{r} = N_D. \quad (3)$$

where m represents ship mass, $\dot{u}$ represents the longitudinal component of velocity, r represents the rotational component of velocity, v represents the lateral component of velocity, I_z represents the mass moment of inertia, and the forces X_D , Y_D and moments N_D represent the hydrodynamics and the main propulsion system, the rudder blade, and the individual propellers of the vessel. With the above assumptions, a block diagram of the mathematical simulation model of the training vessel the *Blue Lady* was determined and is shown in Figure 2.

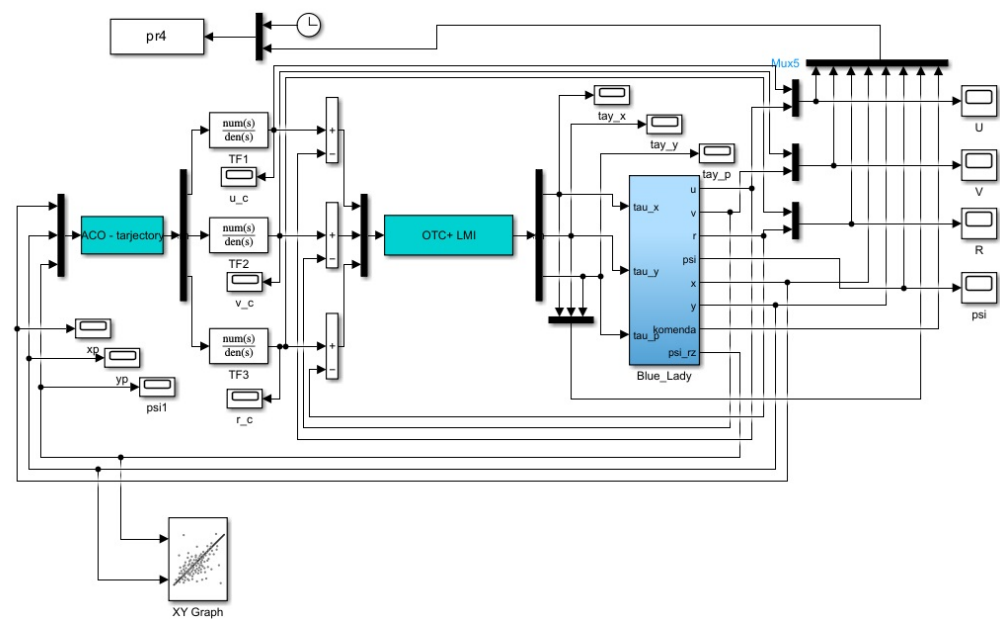


Figure 2. A general block diagram of the mathematical simulation model of the ship *Blue Lady*.

According to the principle presented in [23], the set signals on the rudder devices, such as main propulsion or thrusters, are fed into the block modeling the dynamics of the individual devices, i.e., the rudder blade, main propulsion, and thrusters, hereinafter referred to as the allocation system. It should be noted that the reference signals are components of the input signal vectors for the blocks where the forces and moments occurring on the right-hand side of the Equations (1)–(3) are calculated. In the presented scheme, the components of the output vectors from the hull dynamic block and the ship kinematics model are the values of the three ship velocities, such as:

- Longitudinal velocity: u ;
- Lateral velocity: v ;
- Rotational velocity: r .

On the other hand, when modeling the ship dynamics, the object had to be linearized around the operating point for controller synthesis. For the power distribution system, a conversion had to be made for the three components of the vector:

$$u = [\tau_x, \tau_y, \tau_p], \quad (4)$$

into a vector T of eight propulsion equipment command signals for the main propulsion, thrusters, and rudder blade. With τ_x being the desired force in the longitudinal axis of the vessel, τ_y being the force in the transverse axis of the vessel, and τ_p the desired torque. Once the object identification was performed, the object model was placed in the state space. Therefore, the state equations of the model together with the output equations are presented below:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} a_{uu} & 0 & 0 \\ 0 & a_{vv} & a_{vr} \\ a_{ru} & a_{rv} & a_{rr} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_{uu} & 0 & 0 \\ 0 & b_{vv} & b_{vr} \\ b_{ru} & b_{rv} & b_{rr} \end{bmatrix} \cdot \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_p \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} u \\ v \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (6)$$

where a_{uu} to b_{rr} is the averaged numerical values of the Blue Lady ship model coefficients obtained during object identification, which are shown below and recorded in matrices A, B, C and D:

$$A = \begin{bmatrix} -3.36 \cdot 10^{-5} & 0 & 0 \\ 0 & -9.0 \cdot 10^{-3} & -2.0 \cdot 10^{-4} \\ -3.0 \cdot 10^{-3} & -1.0 \cdot 10^{-2} & -7.75 \cdot 10^{-3} \end{bmatrix} \quad (7)$$

$$B = \begin{bmatrix} 3.62 \cdot 10^{-3} & 0 & 0 \\ 0 & 2.06 \cdot 10^{-3} & -1.28 \cdot 10^{-5} \\ 3.00 \cdot 10^{-5} & 1.15 \cdot 10^{-5} & 8.00 \cdot 10^{-3} \end{bmatrix} \quad (8)$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (9)$$

These were presented in papers describing the object *Blue Lady* [24,31]. The non-linear control object finally has a form represented in terms of the state variables proposed in Equations (5) and (6), where the simulation model was created in a Matlab and Simulink environment based on, among other things, the s-function. On the other hand, the ship's hull model was created in the Simulink environment in a library called 'Simulink 3D Animation/Virtual Reality', as presented in Figure 3.

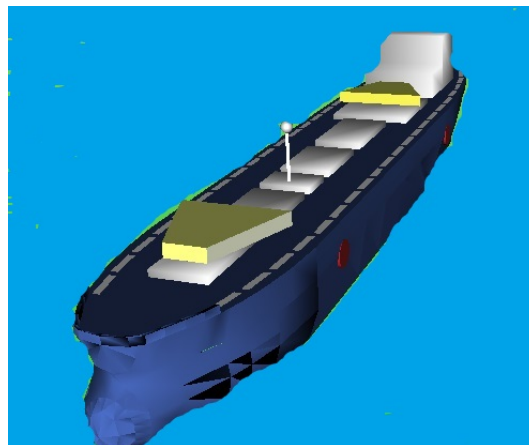


Figure 3. Simplified hull model of the training ship *Blue Lady* made in the Simulink environment based on the Simulink 3D Animation/Virtual Reality library.

3. Safe Trajectory Planning Based on Ant Colony Optimization Algorithm

A pseudocode of an algorithm used for calculating the ship's safe trajectory in the port, based on the Ant Colony Optimization (ACO), is shown as Algorithm 1.

ACO is one of the most popular Swarm Intelligence (SI) methods. The term Swarm Intelligence was introduced by Beni and Wang [32] in relation to cellular robotic systems. Bonabeau, Dorigo, and Theraulaz defined SI as any attempt to build an algorithm inspired by the collective behavior of insect colonies or other animal communities [33].

The insect colony, e.g., an ant colony, can be regarded as a decentralized problem-solving system, composed of many relatively simple individuals interacting with each other, characterized by self-organization, flexibility, and robustness. The flexibility of a colony allows it to adapt to changing environments. The robustness means that a colony continues to function even if some individuals do not fulfill their tasks. Insect colonies can solve problems such as finding food or building or expanding the nest by effectively dividing the work between individuals. Many of these problems have their equivalents in engineering and computer science. Observations of insect colony behavior and the discovery of factors influencing their functioning have enabled the application of this knowledge in the field of intelligent system design.

Algorithm 1 A pseudocode of the ACO-based algorithm for ship's safe trajectory planning in the port

```

1: DO:
2:   Receive data concerning static navigational constraints.
3:   Build a graph of possible ship's waypoints.
4:   Receive data concerning current ship's position and course.
5:   Initialize ACO parameters.
6:   for  $it = 1; it \leq max\_it; it++$  do
7:     Build solution by artificial ants.
8:     Update pheromone trail.
9:     Trajectory smoothing procedure.
10:  end for
11: Present the shortest safe trajectory  $path\_best$  in a graphical and numerical form.
  
```

Observation of ant colonies allowed us to find out that ants communicate with each other and with their surroundings using a chemical substance they secrete. This chemical substance is called a pheromone. Ants transmit information to other ants using the pheromone trail they leave on the ground. Ants leave pheromone trails along the path they travel between the nest and the food source. In this way the ants show others individuals in the colony a way to the food source. This kind of indirect communication, in which an individual's behavior modifies the environment, which is what influences the behavior of another individual in the colony, is called stigmergy.

ACO uses a positive feedback mechanism based on ant colonies' behavior of leaving and following pheromone trails. This mechanism is based on strengthening parts of good solutions or strengthening good solutions as a whole. For this purpose, the so-called virtual pheromone trail is applied. With the use of this mechanism good solutions are stored in the memory, so they can be used to obtain better solutions in the future.

However, reinforcing good but not optimal solutions can lead to premature convergence of the algorithm to a local minimum, resulting in what is known as stagnation. To prevent this, a negative feedback mechanism also needs to be applied, known as pheromone evaporation.

The first proposed ACO algorithm was the Ant System (AS). Initially, three versions of AS were proposed: the ant-density algorithm, the ant-quantity algorithm, and the ant-cycle algorithm. In the first two versions, ants update their pheromone trail directly after movement. In the ant-cycle algorithm the pheromone trail is updated after all ants in a given iteration have finished their movement. This version of AS was the inspiration for the development of the algorithm presented in this work for a ship's safe trajectory calculation. More information concerning ACO can be found in [34,35].

In the process of planning a ship's safe trajectory, an optimization algorithm calculates the safe trajectory P from the current position of the ship (wp_0) to the defined final position (wp_e), as defined by Equation (10). Every waypoint is defined using the coordinates of the ship's position (x and y) and the ship's course ψ . Algorithm 2 shows a pseudocode for the procedure applied to the algorithm for building a graph representing possible waypoints of the ship's path.

$$P = \{wp_0 = [x_0, y_0, \psi_0], wp_1 = [...], ..., wp_e = [x_e, y_e, \psi_e]\} \quad (10)$$

In calculating a safe trajectory for a ship in the port, static navigational constraints such as waterfront areas, shallows, and buoys must be considered. These restrictions are modeled in the form of polygons.

Algorithm 2 A pseudocode of an algorithm for building a graph of possible waypoints of the ship's path

```

1: DO:
2: Create a graph of possible waypoints of the ship's path with assumed step size.
3: Place static navigational constraints on the graph.
4: for  $j = 1; j \leq \text{number\_of\_obstacles}; j++$  do
5:   for  $n = 1; n \leq \text{number\_of\_possible\_waypoints}; n++$  do
6:     if Waypoint  $n$  exceeds the constraints (is placed inside or on an edge of an obstacle)
       then
7:       Save waypoint  $n$  on the list of waypoints exceeding the constraints.
8:     end if
9:   end for
10: end for
11: Remove waypoints exceeding the constraints from the graph.

```

3.1. ACO Data Initialization

The following ACO algorithm parameters are defined in the data initialization procedure:

- Initial value of the pheromone trail τ_0 on all possible waypoints (vertices) on the graph;
- α and β , applied in the formula for calculating the probability of the next ant's move;
- The pheromone evaporation rate ($0 < \rho \leq 1$);
- Maximum number of ants— max_ants ;
- Maximum number of ant's steps— max_steps ;
- Number of iterations— max_it .

3.2. ACO Algorithm Procedure

Every ant starts constructing its path from the starting waypoint (vertex) wp_0 with coordinates (x_0, y_0) , which is the current position of the ship.

Then, every ant constructs its path until it reaches the final waypoint wp_e with coordinates (x_e, y_e) or reaches a specified maximum number of steps max_steps . At every step, the ant chooses an action using the action choice rule. It probabilistically selects the next waypoint (vertex of the graph) among neighboring waypoints (vertices). The choice of the next waypoint (vertex) depends on the amount of the pheromone trail $\tau_j(t)$ on vertex j adjacent to the current vertex i and on some heuristic information η_{ij} . In the developed algorithm visibility was applied as an inverse of the distance between the current vertex i and the neighboring vertex j . The probabilistic selection of the next vertex works analogously to the roulette wheel selection procedure used in evolutionary algorithms [36].

This process is composed of the following stages:

1. Summing the probabilities of selecting neighboring waypoints (vertices) sum_p ;
2. Drawing a random number r from the range $[0; \text{sum}_p]$;
3. Traversing subsequent adjacent vertices until the sum of probabilities of selecting the vertices visited so far will be greater than or equal to r ; meeting this condition means selecting a given vertex.

In the formula for calculating the next vertex selection probability (Equation (11)) there are two coefficients. If coefficient $\alpha = 0$, it is most likely to select the nearest neighboring vertex, if $\beta = 0$, only pheromone reinforcement effects are seen, which results in rather unsatisfactory solutions.

$$P_{wp_{ij}}^{ant}(t) = \frac{[\tau_{wp_j}(t)]^\alpha \cdot [\eta_{wp_{ij}}]^\beta}{\sum_{l \in wp_i^{ant}} [\tau_{wp_l}(t)]^\alpha \cdot [\eta_{wp_{il}}]^\beta} \quad (11)$$

Algorithm 3 shows a pseudocode for the solution construction procedure, and Algorithm 4 shows a pseudocode for the action choice rule procedure.

Algorithm 3 A pseudocode of an algorithm implementing the solution construction procedure

```

1: for  $ant = 1; ant \leq max\_ants; ant++$  do
2:   for  $step = 1; step \leq max\_steps; step++$  do
3:     if  $wp_i$  is not equal to  $wp_e$  then
4:       Execute the action choice rule.
5:     end if
6:   end for
7:   if  $wp_i$  is equal to  $wp_e$  then
8:     Save the ant's path.
9:     Calculate the ant's path length.
10:    if Ant's path length is shorter than the shortest path then
11:      Save ant's path length as the shortest path length.
12:    end if
13:  end if
14: end for

```

Algorithm 4 A pseudocode of an algorithm implementing the action choice rule procedure

```

1: DO:
2: Initialize  $sum\_p = 0$ .
3: for  $j = 1; j \leq number\_of\_neighboring\_vertices; j++$  do
4:   Sum the probabilities of selecting neighboring waypoints (vertices).
5: end for
6: Draw a random number  $r$  from the range  $[0; sum\_p]$ .
7: for  $j = 1; j \leq number\_of\_neighboring\_vertices; j++$  do
8:   Calculate the probability of visiting the next neighboring vertex.
9:   if The sum the probabilities of selecting the vertices visited so far  $\geq r$  then
10:    Choose the currently considered vertex and end the action choice rule.
11:   end if
12: end for

```

$$\tau_{wp_j}(t+1) = (1 - \rho) \cdot \tau_{wp_j}(t) + \sum_{ant=1}^{max_ants} \Delta\tau_{wp_j}^{ant}(t) \quad (12)$$

3.3. Updating of the Pheromone Trail

When all of the ants in a given iteration of the algorithm have finished constructing their paths, the process of updating the pheromone trail is carried out, which is composed of two stages, defined by Equation (12).

1. In the first stage, the pheromone evaporation procedure is implemented. It involves reducing the pheromone trail amount on all vertices of the graph by a certain constant value. The pheromone evaporation is a mechanism that allows the ants to “forget” their bad decisions.
2. In the second stage, the pheromone deposit is carried out, during which a certain value of the pheromone trail is added to all vertices belonging to the paths constructed by ants in a given iteration. Thanks to this mechanism, vertices that constitute parts of the paths chosen by many ants, and which are part of the shortest paths, receive more pheromone trails and thus the probability of their selection by ants in subsequent iterations increases.

Algorithm 5 shows a pseudocode for the pheromone trail update procedure, composed of the pheromone trail evaporation (the first part of Equation (12)) and the pheromone trail deposit (the second part of Equation (12)).

Algorithm 5 A pseudocode of an algorithm implementing the pheromone trail update procedure

```

1: for  $j = 1; j \leq \text{number\_of\_vertices}; j++$  do
2:   Reduce the pheromone trail amount on all vertices of the graph by a certain constant value.
3: end for
4: for  $\text{ant} = 1; \text{ant} \leq \text{max\_ants}; \text{ant}++$  do
5:   for  $j = 1; j \leq \text{number\_of\_vertices}; j++$  do
6:     if  $j$  is a part of the path of the currently considered ant then
7:       Add a certain value of the pheromone trail to vertex  $j$ .
8:     end if
9:   end for
10: end for

```

Before starting the next iteration of calculations, the shortest path found so far is saved. Afterwards, the process of smoothing out the best path determined in a given iteration is carried out. The trajectory smoothing procedure removes unnecessary vertices, further optimizing the path found by the ants. The details of this procedure are presented in Algorithm 6. It is based on evaluating the possibility of connecting two waypoints of the path, marked as p_1 and p_2 , omitting that an intermediate waypoints exists when it does not cause a collision with any of the line segments, marked as (p_3, p_4) , constituting edges of obstacles occurring in the navigational environment. When the currently considered line segment (p_1, p_2) of an optimized path does not intersect with any edge of any navigational obstacle, then the possible path is saved. Such a procedure is repeated until no more reductions are possible in the safe path.

Algorithm 6 A pseudocode of an algorithm implementing the trajectory smoothing procedure

```

1: for  $i = 1; i \leq \text{number\_of\_waypoints}; i++$  do
2:   if The number of waypoints is greater than 3 then
3:     Assign point  $p_1$  as the waypoint  $i$  and point  $p_2$  as the waypoint  $i+2$ .
4:     for  $\text{obs} = 1; \text{obs} \leq \text{number\_of\_obstacles}; \text{obs}++$  do
5:       for  $j = 1; j \leq \text{number\_of\_obstacle\_vertices}; j++$  do
6:         Assign point  $p_3$  as the obstacle vertex  $i$  and point  $p_4$  as the obstacle vertex  $i+1$ .
7:         if The line segment  $(p_1, p_2)$  intersect the line segment  $(p_3, p_4)$  then
8:           Stop checking the possibility of connecting the points  $p_1$  and  $p_2$ .
9:           Do not connect the points  $p_1$  and  $p_2$ .
10:        else
11:          Continue checking the possible intersection between the line segment  $(p_1, p_2)$  and another line segment constituting an edge of the currently considered obstacle.
12:        end if
13:      end for
14:    end for
15:    Connect the points  $p_1$  and  $p_2$ , and save new optimized trajectory.
16:  end if
17: end for

```

The termination condition of the algorithm is the maximum number of iterations and the best solution is the shortest trajectory.

4. Multivariable Ship Control

4.1. Slave Controller—LMI

Multidimensional ship control is complicated insofar as it concerns the control of several channels simultaneously. It must be assumed that the object matrix $\mathbf{A}$ satisfies the derivative relation of the state variable as written below:

$$\frac{dx(t)}{dt} = \mathbf{A}x(t) = \dot{x} = \mathbf{A}x \quad (13)$$

When considering the ship model object described earlier, the first step is to verify that the system is stable. It is necessary to find a matrix $\mathbf{X}_x$ that is symmetric and positively defined for the inequality shown below:

$$\mathbf{A}^T \mathbf{X}_x + \mathbf{X}_x \mathbf{A} < 0 \quad (14)$$

whereby matrix $\mathbf{A}$ is the main matrix of the linear system under study (linearized ship model) and matrix $\mathbf{X}_x$ is the search matrix and positively defined, which means that:

$$\mathbf{X}_x = \mathbf{X}_x^T, \mathbf{X}_x \succ 0 \quad (15)$$

The search for the $\mathbf{X}_x$ matrix is called the existence problem [16,37]. The multidimensional LMI controller is described in [29,38]. In a simplified manner, the condition for the search for the stability of the system considering the pole position areas in the left half-plane of the composite variables will be presented below. During the synthesis of the controller, the area of the vertical chord and the cone of the dilation angle $\frac{\pi}{4}$ were chosen, where the vertical chords were placed on the axis of the composite variables $a_1 = -0.6, a_2 = -4.5$. This is how the region of the sector with a dilation angle of ψ was obtained, assuming that $\text{Re}(s) = x, \text{Im}(s) = y$, then the condition is:

$$\mathbf{Z}_1 = \left\{ s; y^2 < x^2 \tan^2(\psi), x \tan(\psi) < 0 \right\}, \quad (16)$$

The vertical strip is defined as follows:

$$\mathbf{Z}_2 = \left\{ s; a_1 < \frac{1}{2}(s + \bar{s}) < a_2 \right\}, \quad (17)$$

After taking into account the constraints in the drawing (Figure 4):

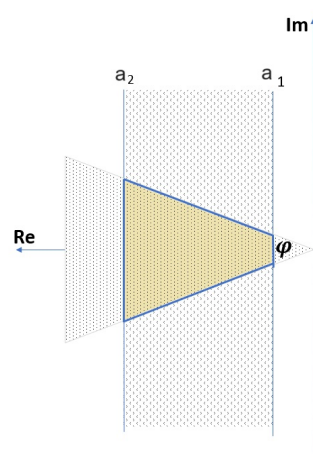


Figure 4. Zone is a sector with an angle of $\phi = \frac{\pi}{4}$ and two vertical strips $a_1 = -0.6$ and $a_2 = -4.5$.

The author has assumed that $\mathbf{L} = 0$,

$$\mathbf{M} = \begin{bmatrix} \sin(\psi) & \cos(\psi) \\ -\cos(\psi) & \sin(\psi) \end{bmatrix} \prec 0 \quad (18)$$

Finally, a symmetric and positively defined matrix $\mathbf{X}_{zone}$ is sought, in which case the first condition of the LMI has the following form:

$$\mathbf{Z}_{zone}(\mathbf{A}, \mathbf{X}_{zone}) = \mathbf{L} \otimes \mathbf{X}_{zone} + \mathbf{M} \otimes (\mathbf{A}\mathbf{X}_{zone}) + \mathbf{M}^T \otimes (\mathbf{A}\mathbf{X}_{zone})^T \prec 0 \quad (19)$$

is satisfied if and only if there is a symmetric and positively defined matrix $\mathbf{X}_{zone}$, where $\mathbf{A}$ is the matrix of the ship the *Blue Lady*; $\mathbf{X}_{zone}$ is the symmetric and positively defined matrix that is the sought after Lyapunov; and $\mathbf{L}, \mathbf{M}$ is the corresponding matrices, which have been defined by the user. Assuming that matrices $\mathbf{A}$ and $\mathbf{B}$ are the matrices of a closed system search for a system of equations, the state space has the following form:

$$\dot{x} = (\mathbf{A} + \mathbf{B}\mathbf{K})x \quad (20)$$

is sought for the controller matrix $\mathbf{K}$, computed after transforming [29,31] as follows:

$$\mathbf{K}_{zone} = \mathbf{Y}_{zone} \cdot \mathbf{X}_{zone}^{-1} \quad (21)$$

In the further synthesis of the controller, the norm H_∞ needed to determine the control criterion in the case of overshooting is taken into account. It relates to minimizing the scalar value of γ_∞ , i.e., finding the smallest scalar value, and estimates the norm from above. Based on the Riccati equation (algebraic inequality proof in [16,37]):

$$(\mathbf{X}\mathbf{x}_\infty \mathbf{A} + \mathbf{A}^T \mathbf{X}\mathbf{x}_\infty + \mathbf{C}^T \mathbf{C}) + (\mathbf{X}\mathbf{x}_\infty \mathbf{B} + \mathbf{C}^T \mathbf{D})(\gamma_\infty^2 \mathbf{I} - \mathbf{D}^T \mathbf{D})^{-1}(\mathbf{B}^T \mathbf{X}\mathbf{x}_\infty + \mathbf{D}^T \mathbf{C}) \prec 0 \quad (22)$$

using the correlation 5.18 in [16]:

$$\gamma_\infty^2 \mathbf{I} - \mathbf{D}^T \mathbf{D} \prec 0 \quad (23)$$

In the literature, there are three possible conditions for the H_∞ norm in this example study based on Theorem 5.4 according to [37]

$$\begin{bmatrix} \mathbf{A}\mathbf{X}\mathbf{x}_\infty + \mathbf{X}\mathbf{x}_\infty \mathbf{A}^T & \mathbf{B} & \mathbf{X}\mathbf{C}^T \\ \mathbf{B}^T & -\gamma_\infty^2 \mathbf{I} & \mathbf{D}^T \\ \mathbf{C}\mathbf{X}\mathbf{x}_\infty & \mathbf{D} & -\mathbf{I} \end{bmatrix} \prec 0, \quad (24)$$

For the condition for the norm of H_∞ presented in this way, the synthesis problem of the regulator K_∞ is defined as follows.

$$\mathbf{K}_\infty = \mathbf{Y}_\infty \cdot \mathbf{X}\mathbf{x}_\infty^{-1} \quad (25)$$

where Matrix $\mathbf{X}_\infty$ is symmetric and positively defined, see relation (15). Finally, it is assumed according to statement [37]:

$$\mathbf{X}\mathbf{x} = \mathbf{X}\mathbf{x}_{zone} = \mathbf{X}\mathbf{x}_\infty \quad (26)$$

assuming that all Lyapunov matrices are equal in all variants and that both conditions are treated as a single constraint allows the state regulator to be synthesized. Algorithm 7 presents a pseudocode for the determination of the matrix $\mathbf{K}$ using the LMI method. In this case, the state regulator matrix $\mathbf{K}$ is described according to the assumption:

$$\mathbf{K} = \mathbf{X}\mathbf{x}^{-1} \cdot \mathbf{Y} \quad (27)$$

After the above steps, finally, the content of the state regulator matrix is as follows:

$$\mathbf{K} = \begin{bmatrix} 1812.7 & 0.1000 & 0.1000 & -840.500 & -0.000 & 0.100 \\ -0.100 & 1981.7 & -0.430 & 0.000 & -979.8 & 0.210 \\ -7.000 & -0.330 & 518.300 & 0.360 & 0.170 & -255.7 \end{bmatrix} \quad (28)$$

Algorithm 7 A pseudocode of an algorithm implementing the LMI control

- 1: DO:
 - 2: Input the ship model Blue Lady matrix A,B,C,D.
 - 3: The first condition of the LMI is satisfied only if there is a symmetric and positively defined matrix X_{xzone} .
 - 4: **for** $gam = 1$; $gam \leq max_gamma[0.....1]$; $gam++$ **do**
 - 5: Solutions give Pareto resolution.
 - 6: $F = [Fzone, kron(L, Xxzone) + kron(M, AXxzone) + kron(M', AXxzone')] \leq 0$.
 - 7: $F = [F, [AXxinf + XxinfA' BXxC'; B' - gam^2 * 1D'; CXD - gam * 1] \leq 0]$.
 - 8: **end for**
 - 9: The second condition of the LMI is satisfied only if there is a symmetric and positively defined matrix X_{inf} .
 - 10: Compare $K_{zone} = K_{inf}$.
 - 11: Solution K matrix is: $K = (Xx)^{-1} * Y$.
-

4.2. Master Controller—Overriding Trajectory

Another multidimensional controller is the ship trajectory controller. The general concept is given in [38]. The general scheme of the controllers is shown below in Figure 5.

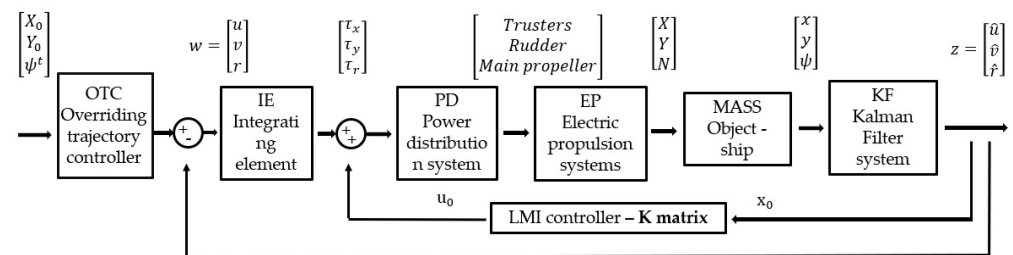


Figure 5. A concept of operation for two OTC controllers—the overriding trajectory and the LMI controller. In addition, the diagram includes system components such as the power distribution system, electric propulsion systems, and KF Kalmann filter system.

The trajectory coordinates X_0, Y_0, ψ_0 are calculated as functions of s on the setpoints for longitudinal velocity u , transverse velocity v , and angular velocity r . Due to the extensive nature of the issues not covered in this work, detailed information on this subject can be found in [23,29].

5. Results

This work involved the synthesis of an LMI controller for the non-linear ship model *Blue Lady* and the verification of the ACO algorithm. In order to verify the results, three different scenarios for the ship's trajectory were presented and tests were carried out at low speeds in the port. The computer simulations were performed in Matlab and Simulink 2023b environments. The Yalmip [39] and SeDuMi [40] libraries were used during the state controller synthesis.

The following values for the ACO algorithm's parameters were applied in the simulation tests: $\alpha = 0.5$, $\beta = 0.5$, $\tau_0 = 0.001$, $\rho = 0.5$, $max_ants = 300$, $max_steps = 500$ and $max_it = 20$.

Figures 6–8 present a comparison of trajectories calculated using the ACO-based algorithm with trajectories executed by the *Blue Lady* ship model using the developed

controllers. The areas filled with black represent the static navigational obstacles, such as the quay of the port on Lake *Silm* and the buoys, which are modeled as rectangles. The area marked using white is the lake area, where the ship model can realize its movement.

The executed trajectory also relates to the verification of the LMI controller's operation; hence, the graph below presents three values in Figure 9 of the set and regulated resting velocities for longitudinal u , lateral v , and rotational velocity r .

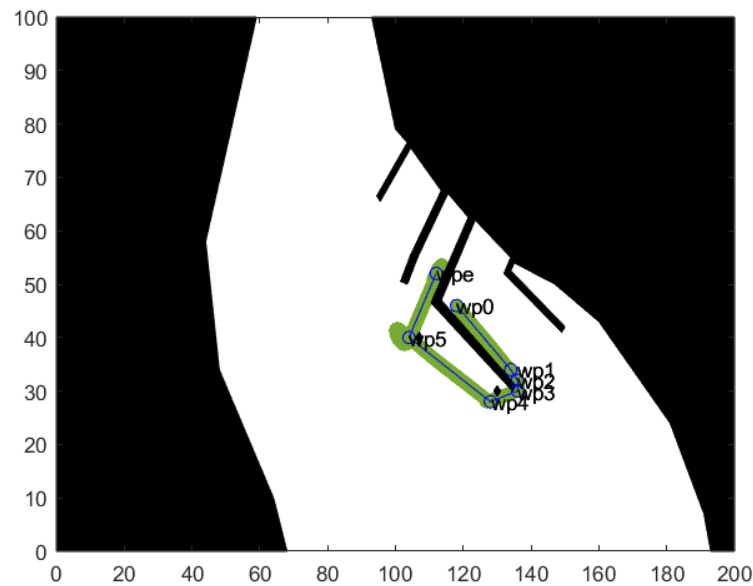


Figure 6. Scenario I—blue represents the trajectory set by the ACO, green represents the trajectory made by the vessel based on LMI control. In addition, start points $wp0$ and end points wpe are marked by six points.

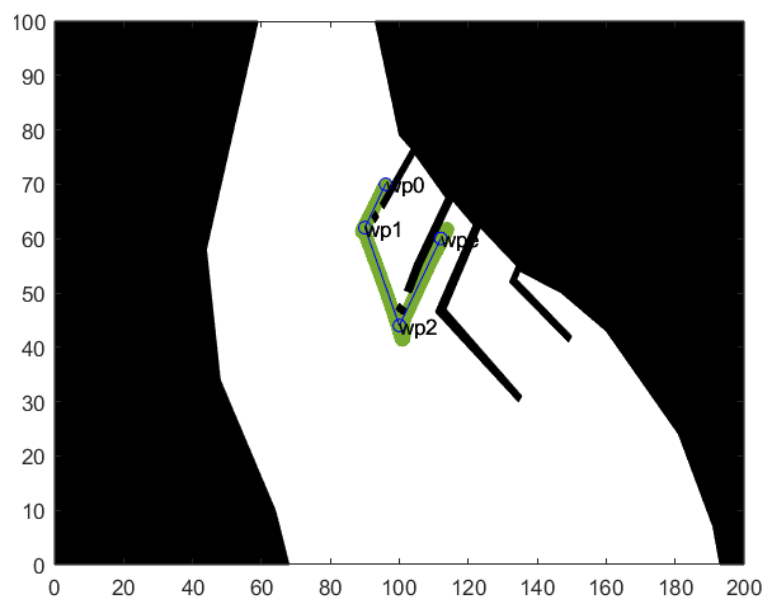


Figure 7. Scenario II—blue represents the trajectory set by the ACO, green represents the trajectory made by the ship model based on LMI control. In addition, start points $wp0$ and end points wpe are marked by four points.

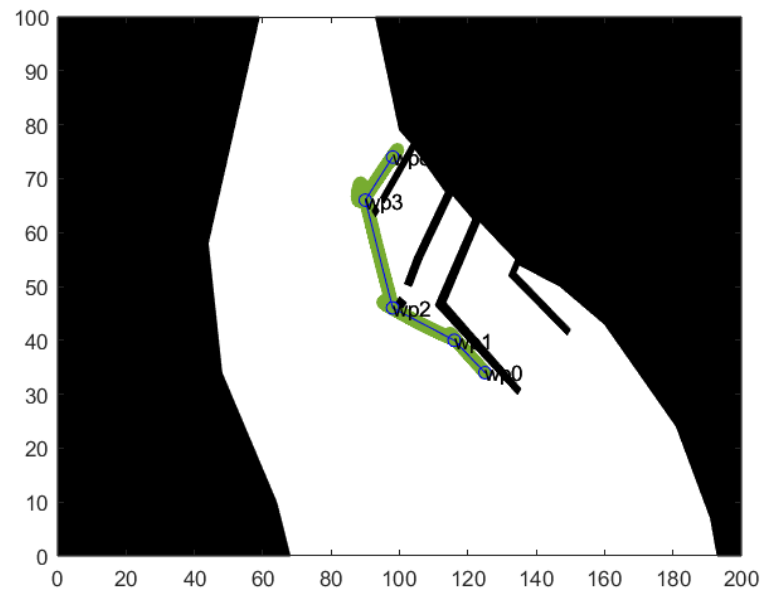


Figure 8. Scenario III—blue represents the trajectory set by the ACO, green represents the trajectory made by the ship model based on LMI control. In addition, start points $wp0$ and end points wpe are marked by five points.

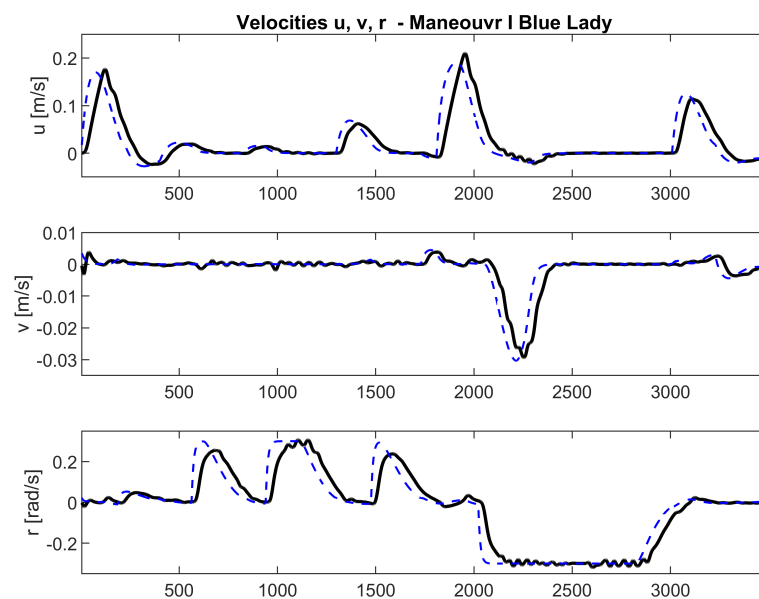


Figure 9. Scenario I—the blue colour represents the values of the velocity targets (output signals of the trajectory controller) and the black colour represents the values of the LMI control signal.

As can be seen in Figures 6–8, the three scenarios differ in the initial waypoint of the path, marked as wp_e . Every scenario also differs in the details concerning the navigational environment. The difference lies in the placement of small rectangular obstacles, which model buoys. For every scenario they are placed differently in order to ensure a proper safe distance of the vessel from the quay during maneuvering. The assumed initial and final waypoints and the placement of the buoys influence the complexity and effectiveness of the algorithm. In all scenarios, the algorithm was able to calculate a safe, optimized path.

The direction along the vertical axis towards the top of the figure is assumed to be 0 degrees. The value of the ship's course at the consecutive line segments composing the safe path were calculated regarding the above-mentioned assumption and the positions of the consecutive waypoints of the safe path calculated by the ACO algorithm.

The trajectory calculated in the first test case, presented in Figure 6, is composed of seven waypoints. The solution had to consider a part of the quay area and two buoys. The following course values were determined for the ship in this situation: 127, 135, 180, 256, 297, and 34 degrees. Figure 9 shows the control signals during the execution of the planned maneuvers for this test case. The blue dashed lines represent the outputs of the Overriding Trajectory Controller and the black lines represent the control signal determined by the LMI controller.

The trajectory calculated in the second test case, presented in Figure 7, is composed of four waypoints. The solution also had to consider a part of the quay area and two buoys, but in a different location than in the first test case. The following course values were determined for the ship in this situation: 217, 151, and 37 degrees. Figure 10 shows the control signals achieved during the execution of the planned maneuvers for this test case.

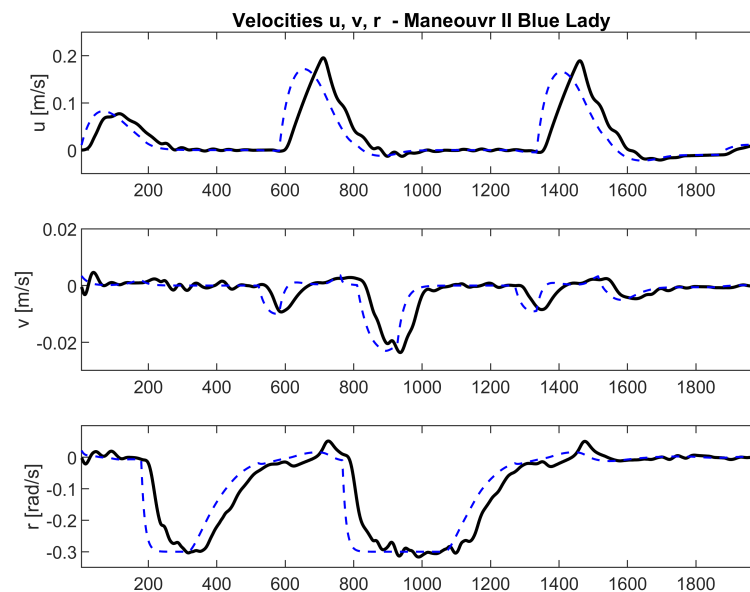


Figure 10. Scenario II—the blue colour represents the values of the velocity targets (output signals of the trajectory controller) and the black colour represents the values of the LMI control signal.

Figure 8 shows the trajectory calculated by the ACO-based algorithm for the third test case. This solution is composed of five waypoints. The safe trajectory also had to consider a part of the quay area and two buoys. The following course values were determined for the ship in this situation: 304, 288, 338, and 45 degrees. Figure 11 presents the control signals reached during the execution of the planned maneuvers for this test case.

The ship effectively executed the determined trajectories using the designed controllers, what can be observed in Figures 6–8. These figures present comparisons between calculated and executed trajectories, which confirm the successful implementation of the developed methods.

Research on the LMI controller, conducted over several years, focuses on the parameter settings of both the pole position on the left half-plane of the s -combined variable and the influence of the parameters calculated for the optimization of the H_2 and H_∞ norms responsible for the energy of the control signal and the deviation of the regulated signal in relation to the set values.

The synthesis of the controller with the support of LMI tools proves that it is possible to operate a multidimensional controller for several reference signals without taking into account cross-coupling, which was proven by presenting the values of the signals of individual ship speeds. As can be seen in Figures 9–11 in the simulations, a small deviation from the setpoint has no effect on the maneuver being performed, which is a promising assumption for carrying out the same simulations in real conditions on Lake Silm in Ilawa. In none of these examples is there any cross-coupling, and the controller is able to cope

even with changes in trajectory points. For example, in Figure 9, three velocity values had to be set between 1800 and 2100 s, which did not interfere with the controller and the final maneuver. A similar property can be observed in Figure 10 in the interval between seconds (1300 to 1600) and in Figure 11 in the interval between seconds (1000 to 1500).

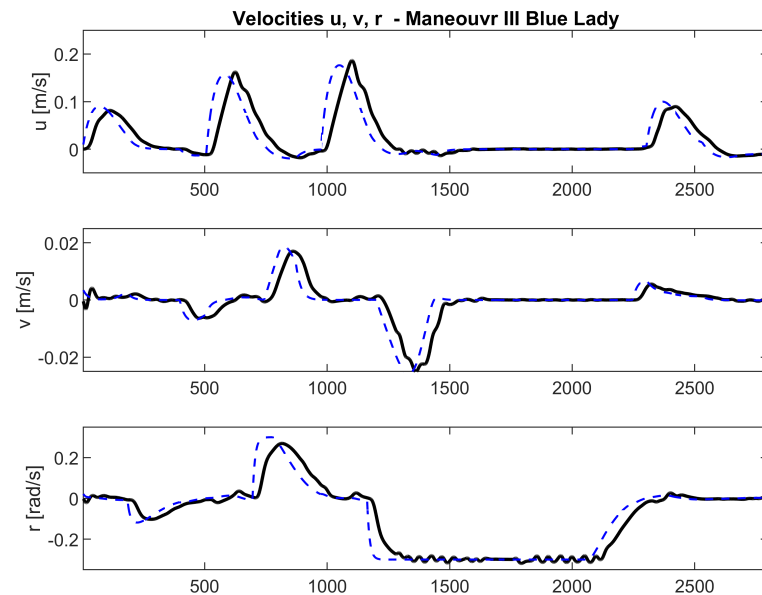


Figure 11. Scenario III—the blue colour represents the values of the velocity targets (output signals of the trajectory controller) and the black colour represents the values of the LMI control signal.

The results of the simulations tests presented in Section 5 allow us to state that the developed ACO-based algorithm calculated a safe trajectory between different waypoints considering static navigational obstacles. The successful operation of the ACO-based algorithm does not depend on the placement of the initial (wp_0) and final waypoints (wp_e).

According to the authors, the use of an LMI controller is justified, especially when testing maneuvers when in the port, i.e., when steering the vessel at low speeds. Multi-dimensional control for this type of task, where three setpoints are set, must be precise and there must be no cross-coupling. Furthermore, the proposed optimization in the LMI concerns both the control to minimize the deviation of the speed setpoint and the energy loss needed to perform the maneuver.

In the present study, it was decided to take the research a step further by taking into account the ant algorithm that predicts the optimal routes for a maneuvering vessel. This interpretation of the research proved that it is possible to steer a ship autonomously with respect to the IMO directive described in relation to the Maritime Autonomous Surface Ship [41].

6. Conclusions

This paper proposed a system for the autonomous operation of MASS or USVs in a port. It utilized the Ant Colony Optimization algorithm for path planning and Linear Matrix Inequalities for low speed controller design. An Overriding Trajectory Controller was also applied in order to achieve the task. The developed concept was confirmed by simulation tests with the use of a model of a VLCC, the *Blue Lady*. The results of a few test cases were presented and discussed.

This paper demonstrates that several design steps are necessary for building autonomous models. Experimental trajectories were created based on the real dimensions of the Iława Kamionka harbour at the Foundation for Navigation Safety and Environmental Protection. What is more, the proposed ACO algorithm made it possible to test vessel control in restricted waters where obstacles are static. Optimization of this trajectory affected the ability to further control a multidimensional, non-linear object. The master controller

included in the calculation was responsible for converting the trajectory into velocity component values. Then, by means of an LMI controller that did not exhibit cross-coupling, it was possible to control the forces that were applied to the allocation system needed to control the individual propellers of the ship in order to perform the maneuver.

The proposed approach makes it possible to consider autonomous ship control using a multi-stage approach consisting of a controller synthesis and autonomously determined trajectory, which is in line with IMO assumptions, i.e., no human intervention in the realized ship trajectory (including control of the ship). The ACO, OTC, and LMI algorithms proposed by the authors have shown that safe autonomous trajectory and autonomous control of an object in a port is possible.

Future research directions might include real-life experiments on Lake Silm and the application of other optimization methods in the development of a safe trajectory planning algorithm and the design of controllers. This phase could be followed by a comparative analysis of results using different algorithms.

Author Contributions: Conceptualization, M.R. and A.L.; methodology, M.R. and A.L.; software, M.R. and A.L.; validation, M.R. and A.L.; formal analysis, M.R. and A.L.; investigation, M.R. and A.L.; resources, M.R. and A.L.; data curation, M.R. and A.L.; writing—original draft preparation, M.R. and A.L.; writing—review and editing, M.R. and A.L.; visualization, M.R. and A.L.; supervision, M.R. and A.L.; project administration, M.R. and A.L.; funding acquisition, M.R. and A.L.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the research projects no. WE/2024/PZ/02 and WE/2024/PZ/03, of the Electrical Engineering Faculty, Gdynia Maritime University, Poland.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACO	Ant Colony Optimization
CBF	Control Barrier Functions
EFCA	Elastic Force Contraction Algorithm
LMI	Linear Matrix Inequalities
MASS	Maritime Autonomous Surface Ship
MPC	Model Predictive Control
MSM	Minimum Set Method
OTC	Overriding Trajectory Controller
SHP	Short Horizon Planner
TSS	Traffic Separation Scheme
USV	Unmanned Surface Vehicle
VLCC	Very Large Crude Carrier
VPF	Virtual Potential Field

References

1. Volden, Ø.; Cabecinhas, D.; Pascoal, A.; Fossen, T.I. Development and experimental evaluation of visual-acoustic navigation for safe maneuvering of unmanned surface vehicles in harbor and waterway areas. *Ocean Eng.* **2023**, *280*, 114675. [\[CrossRef\]](#)
2. Plenge-Feidenhans'l, M.K.; Blanke, M. Open Water Detection for Autonomous In-harbor Navigation Using a Classification Network. *IFAC Pap. Online* **2021**, *54*, 30–36. [\[CrossRef\]](#)
3. Trzuskowsky, A.; Hoelper, C.; Abel, D. ANCHOR: Navigation, Routing and Collision Warning during Operations in Harbors. *IFAC Pap. Online* **2016**, *49*, 220–225. [\[CrossRef\]](#)
4. Enevoldsen, T.T.; Blanke, M.; Galeazzi, R. Autonomy for Ferries and Harbour Buses: A Collision Avoidance Perspective. *IFAC Pap. Online* **2023**, *56*, 5735–5740. [\[CrossRef\]](#)
5. Wang, Z.; Yang, P.; Gao, D.; Bao, C. Path-planning algorithm based on elastic force contractions for autonomous navigation of unmanned container ships in waterborne transportation. *Ocean Eng.* **2024**, *310*, 118646. [\[CrossRef\]](#)

6. He, Z.; Liu, C.; Chu, X.; Wu, W.; Zheng, M.; Zhang, D. Dynamic domain-based collision avoidance system for autonomous ships: Real experiments in coastal waters. *Expert Syst. Appl.* **2024**, *255*, 124805. [[CrossRef](#)]
7. Xue, Y.; Clelland, D.; Lee, B.S.; Han, D. Automatic simulation of ship navigation. *Ocean Eng.* **2011**, *38* (17–18), 2290–2305. [[CrossRef](#)]
8. Liu, Z.; Li, Y.; Zhang, Z.; Wang, J.; Du, Y. A dynamic topology analysis method for multi-ship encounters based on multi time-space network trees. *Ocean Eng.* **2024**, *307*, 118052. [[CrossRef](#)]
9. Yu, D.; He, Y.; Zhao, X.; Chen, J.; Liu, J.; Huang, L. Dynamic adaptive autonomous navigation decision-making method in traffic separation scheme waters: A case study for Chengshanjiao waters. *Ocean Eng.* **2023**, *285*, 115448. [[CrossRef](#)]
10. Öztürk, Ü.; Akdağ, M.; Ayabaka, T. A review of path planning algorithms in maritime autonomous surface ships: Navigation safety perspective. *Ocean Eng.* **2022**, *251*, 111010. [[CrossRef](#)]
11. Li, D.; Wang, P.; Du, L. Path planning technologies for autonomous underwater vehicles—a review. *IEEE Access* **2018**, *7*, 9745–9768. [[CrossRef](#)]
12. Vagale, A.; Oucheikh, R.; Bye, R.T.; Osen, O.L.; Fossen, T.I. Path planning and collision avoidance for autonomous surface vehicles I: A review. *J. Mar. Sci. Technol.* **2021**, *26*, 1292–1306. [[CrossRef](#)]
13. Vagale, A.; Bye, R.T.; Oucheikh, R.; Osen, O.L.; Fossen, T.I. Path planning and collision avoidance for autonomous surface vehicles II: A comparative study of algorithms. *J. Mar. Sci. Technol.* **2021**, *26*, 1307–1323. [[CrossRef](#)]
14. Geng, X.; Wang, Y.; Wang, P.; Zhang, B. Motion Plan of Maritime Autonomous Surface Ships by Dynamic Programming for Collision Avoidance and Speed Optimization. *Sensors* **2019**, *19*, 434. [[CrossRef](#)] [[PubMed](#)]
15. Wu, Y.; Wang, T.; Liu, S. A Review of Path Planning Methods for Marine Autonomous Surface Vehicles. *J. Mar. Sci. Eng.* **2024**, *12*, 833. [[CrossRef](#)]
16. Boyd, S.; El Ghaoui, L.; Feron, E.; Balakrishnan, V. *Linear Matrix Inequalities in System and Control Theory*; Siam Philadelphia: Philadelphia, PA, USA, 1994.
17. Alessandri, A.; Donnarumma, S.; Vignolo, S.; Figari, M.; Martelli, M.; Chiti, R.; Sebastiani, L. System control design of autopilot and speed pilot for a patrol vessel by using LMIs. In *Towards Green Marine Technology and Transport*; CRC Press: Boca Raton, FL, USA, 2015; pp. 577–583.
18. Chairinnas, J.E.; Gani, G.; Putri, N.K. Robust control with linear matrix inequality approach for ship steering problem. In Proceedings of the 2016 International Conference on Instrumentation, Control and Automation (ICA), Bandung, Indonesia, 29–31 August 2016; pp. 126–131. [[CrossRef](#)]
19. Nasuno, Y.; Shimizu, E.; Ito, M.; Yamamoto, I.; Tsukioka, S.; Yoshida, H.; Aoki, T. A controller design for autonomous underwater vehicle “MR-X1” using linear matrix inequalities. *Int. J. Control.* **2007**, *80*, 1125–1135. [[CrossRef](#)]
20. Nasuno, Y.; Shimizu, E.; Ito, M.; Yamamoto, I.; Tsukioka, S.; Yoshida, H.; Aoki, T. Design method for a new control system for an autonomous underwater vehicle using linear matrix inequalities. *Artif. Life Robot.* **2007**, *11*, 149–152. [[CrossRef](#)]
21. Xue, H.; Lai, Y.-H.; Sun, K. Human-like constraint-adaptive model predictive control with risk-tunable control barrier functions for autonomous ships. *Ocean Eng.* **2024**, *308*, 118219. [[CrossRef](#)]
22. Le, M.-D.; Nguyen, H.-N.; Nguye, K.-B. Control of large ship motions in harbor maneuvers by applying sliding mode control. *IFAC Proc. Vol.* **2004**, *37*, 309–314. [[CrossRef](#)]
23. Gierusz, W. *Synteza Wielowymiarowych Układów Sterowania Precyzyjnego Ruchem Statku z Wykorzystaniem Wybranych Metod Projektowania Układów Odpornych*; Akademia Morska w Gdyni: Gdynia, Poland, 2004.
24. Tomera, M. Dynamic positioning system design for “Blue Lady”. Simulation tests. *Pol. Marit. Res.* **2012**, *19*, 57–65. [[CrossRef](#)]
25. Tomera, M. Path Controller for Ships with Switching Approach. In *Advanced, Contemporary Control*; Springer: Cham, Switzerland, 2020; pp. 1519–1530.
26. Miller, A.; Rak, A. Measurement System for the Environmental Load Assessment of the Scale Ship Model. *Sensors* **2023**, *23*, 306. [[CrossRef](#)]
27. Miller, A.; Rak, A. A Measurement System for the Environmental Load Assessment of a Scale Ship Model—Part II. *Sensors* **2023**, *23*, 3415. [[CrossRef](#)] [[PubMed](#)]
28. Rybczak, M.; Rak, A. Prototyping and simulation environment of ship motion control system. *TransNav* **2020**, *14*, 367–374. [[CrossRef](#)]
29. Gierusz, W.; Rybczak, M. Effectiveness of multidimensional controllers designated to steering of the motions of ship at low speed. *Sensors* **2020**, *20*, 3533. [[CrossRef](#)] [[PubMed](#)]
30. Miller, A.; Rybczak, M.; Rak, A. Towards the autonomy: Control systems for the ship in confined and open waters. *Sensors* **2021**, *21*, 2286. [[CrossRef](#)] [[PubMed](#)]
31. Rybczak, M. *Synteza Elektronawigacyjnego Układu Wielowymiarowego Sterowania Statkiem Wykorzystująca Liniowe Nierówności Macierzowe*; Uniwersytet Morski w Gdyni: Gdynia, Poland, 2020.
32. Beni, G.; Wang, J. Theoretical problems for the realization of distributed robotic systems. In Proceedings of the IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, 9–11 April 1991; Volume 3, pp. 1914–1920.
33. Bonabeau, E.; Dorigo, M.; Theraulaz, G. *Swarm Intelligence: From Natural to Artificial Systems*; Oxford University Press, Inc.: Oxford, UK, 1999.
34. Dorigo, M.; Stützle, T. *Ant Colony Optimization*; The MIT Press, Inc.: Cambridge, MA, USA, 2004.
35. Solnon, C. *Ant Colony Optimization and Constraint Programming*; ISTE Ltd. and John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2010.

36. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 1989.
37. Duan, G.R.; Yu, H.H. *LMIs in Control Systems: Analysis, Design and Applications*; CRC Press: Boca Raton, FL, USA, 2013.
38. Koznowski, W.; Kula, K.; Lazarowska, A.; Lisowski, J.; Miller, A.; Rak, A.; Rybczak, M.; Mohamed-Seghir, M.; Tomera, M. Research on Synthesis of Multi-Layer Intelligent System for Optimal and Safe Control of Marine Autonomous Object. *Electronics* **2023**, *12*, 3299. [[CrossRef](#)]
39. Yalmip Library. Available online: <https://yalmip.github.io/download/> (accessed on 25 August 2022).
40. Sturm, J.F. Using Se DuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.* **1999**, *11*, 625–653. [[CrossRef](#)]
41. IMO London. Regulatory Scoping Exercise for the Use of Maritime Autonomous Surface Ships. MSC 2018, 12. Available online: <https://www.setfords.co.uk/wp-content/uploads/MSC9955.pdf> (accessed on 30 June 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.