electronics

Article

Maximum Principle in Autonomous Multi-Object Safe Trajectory Optimization

Józef Andrzej Lisowski

Faculty of Electrical Engineering, Gdynia Maritime University, 81-225 Gdynia, Poland; j.lisowski@we.umg.edu.pl; Tel.: +48-694-458-333

Abstract: The following article presents the task of optimizing the control of an autonomous object within a group of other passing objects using Pontryagin's bounded maximum principle. The basis of this principle is a multidimensional nonlinear model of the control process, with state constraints reflecting the motion of passing objects. The analytical synthesis of optimal multi-object control became the basis for the algorithm for determining the optimal and safe object trajectory. Simulation tests of the algorithm on the example of real navigation situations with various numbers of objects illustrate their safe trajectories in changing environmental conditions. The optimal object trajectory obtained using Pontryagin's maximum principle was compared with the trajectory calculated using the Bellman dynamic programming method. The analysis of the research allowed for the formulation of valuable conclusions and a plan for further research in the field of autonomous vehicle control optimization. The maximum principle algorithm allows one to take into account a larger number of objects whose data are derived from ARPA anti-collision radar systems.

Keywords: multi-object control; maximum principle; computer simulation

Citation: Lisowski, J.A. Maximum Principle in Autonomous Multi-Object Safe Trajectory Optimization. *Electronics* **2024**, *13*, 1144. https://doi.org/10.3390/electronics13061144

Academic Editor: Felipe Jiménez

Received: 22 February 2024
Revised: 15 March 2024
Accepted: 18 March 2024
Published: 20 March 2024

1. Introduction

Dynamic optimization is a decision-making process that involves mathematical models of differential and algebraic equations to formulate the best solutions based on predictions of future outcomes. Representative applications address problems such as the control of moving objects—airplanes, ships, and cars; the safe operational control of batch reactors; the pyrolysis of oil shale; or the process of launching rockets.

There are numerous direct dynamic optimization methods, including Euler's calculus of variations, Bellman's optimality principle, the simple and conjugate gradient methods, variable metric methods, and the second variation; indirect methods, including Pontryagin's maximum principle, Newton's principle in state space, and the Newton–Raphson method in conjugate space; and special space methods, including those of Neustadt, Gilbert, Barr, Balakrishnan, and Findeisen.

Among these methods, Pontryagin's maximum principle is of particular importance because of the way in which it takes into account the constraints of the actual control process. The development of applications of the maximum principle for the dynamic optimization of various real control tasks will be presented below using examples.

1.1. State of Knowledge

The basis for the considerations is the work of Pontryagin [1], who presented the mathematical foundations of process optimization. In the years following Pontryagin's initial study, both the theory and applications of the maximum principle were developed.

Zhou [2] presents an extension of the known relationships between the maximum principle and dynamic programming for unconstrained problems to constrained cases.

Hartl et al. [3] present a study of various forms of Pontryagin's maximum principle for optimal control problems with state variable constraints in the form of inequalities.

An original approach to the application of the maximum principle is presented by Vasilieva in their study [4], with the author separating the initial and final conditions of the optimization task and proposing unifying the initial conditions with the dynamic system of the control process and treating the final conditions as a limitation in the search for the optimal solution to the problem.

The application of the maximum principle to the synthesis of the optimal control of one object such as a ship is presented by Abramowski et al. in their study [5]. The optimization task consists of finding the optimal ship route in a region influenced by environmental factors affecting the ship's movement, such as the impact of wind, waves, and sea currents, which are previously given in the form of forecasts of short-, medium-, and long-term sea wave fields. The state of the wave field reduces the ship's speed and increases fuel consumption as it travels to its destination port and constitutes a limitation to the task of optimizing the ship's weather route.

However, Danet [6] addresses extending the applications of the classical maximum principle to elliptic operators. The maximum principle is one of the most useful and best-known tools used in the study of partial differential equations describing the model of optimization problems.

Another interesting application of the maximum principle is proposed by Udin et al. in their study [7] for parametric, real-time identification of a mathematical ship model.

Mardanov et al. [8] proved the validity of Pontryagin's maximum principle for optimal control problems of the described ordinary differential equations with multi-point boundary conditions.

To aid the development of unmanned warships, Sands in [9] proposes combining physical, noisy algorithms and computational models to provide additional information about system states, baseline data, and parameters, emphasizing deterministic rather than stochastic selection, using for this purpose Pontryagin's maximum principle.

Parsian [10] analyzes in their study the extension of the maximum principle to the following aspects of its application: the uniqueness of the solution to elliptic boundary value problems, the property of extreme values for quadratic forms, the existence of an optimal solution in operations research, the largest distance between a point and a compact set, and inverse square laws.

The maximum rule iterative procedure for optimizing a problem described by a nonlinear Poisson equation with Dirichlet boundary conditions is presented by Cortissoz in [11]. In their study, Buldaev and Kazmin [12] solve the problem of optimizing quantum systems, equivalent to the maximum principle, using operator forms of optimality conditions. This approach ensures computational stability, in contrast to standard methods for solving the boundary value problem of the maximum principle; the nonlocality of successive control approximations; the absence of a laboratory procedure of needle or convex variation of the control in a small neighborhood of the considered approximation, which is typical for gradient methods; and the numerical solution of the Cauchy problems with a continuous and uniquely defined right-hand side.

The extension of the vessel theory to the Pontryagin space system was formulated by Alpay et al. in [13] by developing an undefined version of the de Branges–Rovnyak theory on real compact Riemann surfaces. In their study, Lu et al. [14] used the maximum principle for optimal engine control, ensuring minimum fuel consumption.

An algorithm for determining the optimal trajectory of an autonomous drone powered by DC motors, using the maximum principle supported by continuous deterministic artificial intelligence, was proposed by Xu and Sands in their study [15].

Previous studies have omitted the application of the maximum principle to a larger number of control objects, in particular, to determine the optimal trajectory of a moving object when safely passing other moving objects, with this omission prompting the conception of this study.

This approach to the problem will allow solving, among others, the tasks of the optimal and safe control of both maritime autonomous surface ships (MASS) and other autonomous surface vehicles (ASVs) and autonomous underwater vehicles (AUVs), as well as unmanned aerial vehicles (UAVs) in the form of drones. The result of this approach is appropriate algorithms for controlling autonomous objects that determine both the optimality of the task being performed and, above all, the safety of movement of a group of autonomous objects.

1.2. Research Objectives

The essence of this article is as follows:

- An analytical synthesis of safe multi-object control using Pontryagin's maximum principle;
- The development of an algorithm for determining the optimal and safe trajectory of an object in the event of excessive proximity to other objects;
- An experimental comparative analysis of safe object trajectories in various environmental conditions.

A comparison of Pontryagin's maximum principle used in this article to determine the optimal trajectories of one's own object in the situation of safely passing other encountered objects, with the Bellman dynamic programming method, demonstrates that the advantages and disadvantages of one method are the advantages and disadvantages of another. The advantage of the maximum principle is its usefulness for a larger number of state variables describing the dynamics of an object, while its disadvantage is the extended computation time required to take into account the motion of other objects in the form of process state constraints.

The significant results of this research will allow for, in the following stages, improvement of the control algorithms of autonomous objects along with the development of the technology involved in their driving and power supply devices.

1.3. Article Content

First, the control process model of a multi-object control process is presented. In the subsequent section, the synthesis of optimal control by Pontryagin's maximum principle and the appropriate algorithm for calculating a safe trajectory are presented. The simulation studies presented highlight safe object trajectories in several real navigation situations, with them differing in the number of passed objects. The analysis of the research results and the scope of further research are included in the Conclusions.

2. Control Process Model

The model of the Object 0 control process during the safe passing of *j* Objects for the purpose of its optimization can be formulated in the form of nonlinear state equations related to Object 0 movement and the nonlinear constraint process state related to *j* Objects, as shown in Figure 1.

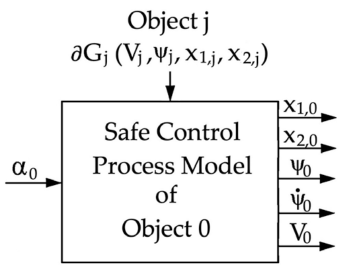

Figure 1. Input and output values of the process model for the safe traffic control of Object 0; α_0 is its rudder angle; ($x_{1,0}$, $x_{2,0}$) are coordinates of its position; ψ_0 is its course; $\dot{\psi}_0$ is the angular velocity of its direction; V_0 is its speed; and ∂G_j is the safe area boundary, assigned to *j* Object, moving with speed V_j and course ψ_j, located at position ($x_{1,j}$, $x_{2,j}$).

In accordance with the actual processes of controlling autonomous objects, the basic control system is used to change the course of the object by changing the rudder angle at a constant speed in relation to the object's movement.

The state variables of the object control process are the coordinates of its position measured by the GPS system, the heading and angular speed of its turn measured by the gyrocompass and speed gyroscope, and its linear speed measured by the log.

However, the encountered objects constitute moving constraints of the process state, the parameters of which, such as their course, speed, and position, are measured by the ARPA anti-collision radar system.

The process model presented in this way, in one regard, adequately reflects reality, and, conversely, is the basis for implementing the task of its optimization using one of the selected optimization methods.

2.1. State Equations

The safe control of Object 0 in relation to the encountered *j* Objects involves changing the course ψ_0 so that the smallest approach distance $D_{j,min} \geq D_s$ is greater than the safe distance D_s, which is illustrated in Figure 2.

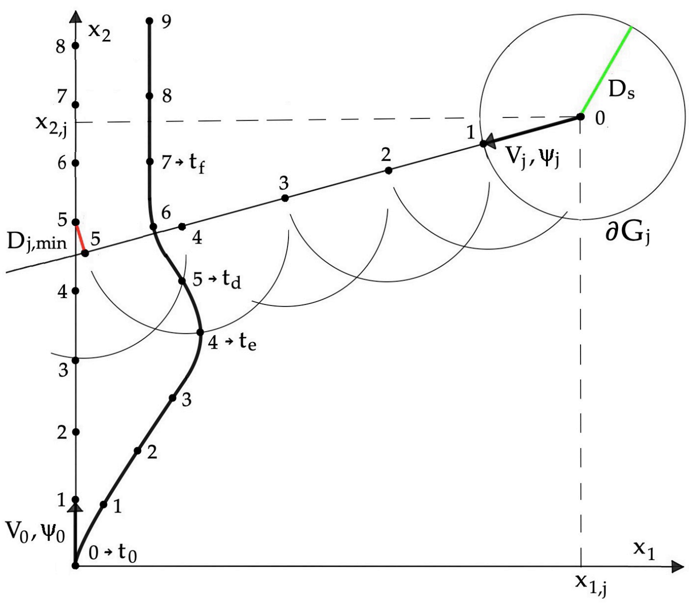

Figure 2. Geometric construction of the optimal trajectory of Object 0 moving with speed V_0 and course ψ_0 while safely passing Object j moving with speed V_j and course ψ_j; $D_{j,min}$ is the closest approach distance of objects; D_s is the safe distance; t_0 is the starting time; t_e is the time of entry; t_d is time of descent from the border of the restricted area ∂G_j; and t_f is the finish time of anti-collision maneuvering.

The kinematics of Object 0's motion are described by the following equations:

$$\dot{x}_{1,0} = V_0 \cos \psi_0 \tag{1}$$

$$\dot{x}_{2,0} = V_0 \sin \psi_0 \tag{2}$$

The nonlinear second-order Nomoto model can be used to describe the dynamics of Object 0:

$$T\ddot{\psi}_0 + \dot{\psi}_0 + a\dot{\psi}_0^2 = k\alpha_0 \tag{3}$$

where T is the time constant of the change in the course of Object 0 as a result of the rudder deflection α_0, and a and k are the coefficients of the nonlinear static characteristic [16].

The state equations of the Object 0 motion control process will take the following form:

$$\begin{gathered}\dot{x}_i = f_i(x,\ u,\ t) \\ \dot{x}_{1,0} = V_0 \cos x_{3,0} \\ \dot{x}_{2,0} = V_0 \sin x_{3,0} \\ \dot{x}_{3,0} = x_{4,0} \\ \dot{x}_{4,0} = -\frac{1}{T} x_{4,0} - \frac{a}{T} x_{4,0}^2 + \frac{k}{T} u_0 \\ \dot{x}_{5,0} = 1\end{gathered} \tag{4}$$

where ($x_{1,0}$,$x_{2,0}$) are the coordinates' Object 0 position; $x_{3,0}$ is Object 0 course ψ_0; $x_{4,0}$ is the angular velocity of Object 0 turn $\dot{\psi}_0$; $x_{5,0}$ is time; and u_0 is the rudder angle α_0 [17].

2.2. Constraints

The need to maintain a safe distance D_s when passing j Object, according to (1), imposes the following state variables constraint, illustrated by the circle ∂Gj in Figure 2:

$$g_j(x) = D_s^2 - \left(x_{1,0} - x_{1,j} - V_j x_{5,0} \sin \psi_j\ \right)^2 - \left(x_{2,0} - x_{2,j} - V_j x_{5,0} \cos \psi_j\ \right)^2 \leq 0 \tag{5}$$

The possibility of deflecting the rudder no greater than the maximum rudder angle $\alpha_{0,\max}$ imposes the following control constraint:

$$h(u) = u_0 \leq |\propto_{0,max}| \tag{6}$$

2.3. Control Quality Index

Knowing the process model described by Equation (4), state constraints (5), and control constraints (6), it is possible to solve the task of dynamic optimization of the control of this process, with the required optimality control index in the form of the control objective functional f_0.

$$Q(x, u) = \int_{t_0}^{t_f} f_0(x,\ u,\ t) dt \tag{7}$$

where t_0 is the start time of the process control and t_f is the final step of this process when Object 0 safely passes all j Objects and returns to its reference operating course ψ_0.

In the process of anti-collision control of the movement of Object 0, the control objective function is the smallest ds path loss $f_0 = ds = V_0 \cdot dt$ for safely passing j Objects, which at a constant operating speed V_0 = const leads to time-optimal control.

The presented model of the optimal multi-object control process applies to mobile robots and land and air vehicles.

3. Optimization

The task of optimizing the control process described above can be solved using many dynamic optimization methods. To solve the problem of optimal multi-object control, the Bellman dynamic programming method described in [18] and Pontryagin's maximum principle are most suitable.

For further considerations, Pontryagin's maximum principle was adopted, in which optimal control minimizing the control quality index Q maximizes a certain function called the Hamiltonian [19,20].

3.1. Hamiltonian

For the control process model formulated above, the Hamiltonian will take the following form:

$$\begin{gathered}H(\lambda,\ x,\ u) = -f_0(x,\ u,\ t) + \sum_{i=1}^{5} \lambda_i f_i(x,\ u,\ t) = \\ \lambda_0 + \lambda_1 V_0 \cos x_{3,0} + \lambda_2 V_0 \sin x_{3,0} + \lambda_3 x_{4,0} + \lambda_4 \left(-\frac{a}{T} x_{4,0} |x_{4,0}| - \frac{1}{T} x_{4,0} + \frac{k}{T} u_0\right) + \lambda_5\end{gathered} \tag{8}$$

where $\lambda_0 = -1$ and $\lambda_i,\ i = 1,\ 2, \ldots 5$ are conjugate variables.

The conjugate equations will take the following form:

$$\begin{gathered}\dot{\lambda}_i = \frac{\partial H}{\partial x_i} \\ \dot{\lambda}_1 = 0 \\ \dot{\lambda}_2 = 0 \\ \dot{\lambda}_3 = -\lambda_1 V_0 \cos x_{3,0} + \lambda_2 V_0 \sin x_{3,0} \\ \dot{\lambda}_4 = -\lambda_3 + \lambda_4 \left(\frac{1}{T} + \frac{2a}{T} |x_{4,0}| \right) \\ \dot{\lambda}_5 = 0\end{gathered} \tag{9}$$

According to the maximum principle, the optimal control of $u^*(t)$ provides the maximum of the Hamiltonian:

$$H[x_i^*(t), u_0^*(t), \lambda_i(t)] = \max_{u_0(t) \in U} H[x_i(t), u_0(t), \lambda_i(t)] = 0 \tag{10}$$

where $t \in < t_0, t_f >$.

From Equations (9) and (11), it follows that

$$u_0^*(t) = sign\ \lambda_4 \tag{11}$$

3.2. Control on Boundary ∂G_j

In the process of safely passing Object 0 with Object j, its trajectory coincides with the boundary ∂Gj of a circular area with a radius equal to the safe approach distance D_s. Then, Leitmann's bounded maximum principle theorem [21] can be used to synthesize the optimal trajectory. For this purpose, the following scalar function p is introduced:

$$\begin{gathered}p(x, u) = \nabla g(x) \cdot f(x, u, t) = \\ 2\left[\left(x_{1,0} - V_j x_{5,0} \sin \psi_j - x_{1,j} \right) \left(V_j \sin \psi_j - V_0 \sin x_{3,0} \right) + \left(x_{2,0} - V_j x_{5,0} \cos \psi_j - x_{2,j} \right) \left(V_j \cos \psi_j - V_o \cos x_{3,0} \right) \right]\end{gathered} \tag{12}$$

$$\begin{gathered}\nabla g(x) = \left(0,\ \frac{\partial g}{\partial x_{1,0}},\ \ldots,\ \frac{\partial g}{\partial x_{5,0}} \right) = \\ \left[0, -2\left(x_{1,0} - V_j x_{5,0} sin\psi_j - x_{1,j} \right), -2\left(x_{2,0} - V_j x_{5,0} cos\psi_j - x_{2,j} \right), 0,\ 0,\ 2V_j sin\psi_j \left(x_{1,0} - V_j x_{5,0} sin\psi_j - x_{1,j} \right) \right. \\ \left. +2V_j cos\psi_j \left(x_{2,0} - V_j x_{5,0} cos\psi_j - x_{2,j} \right) \right]\end{gathered} \tag{13}$$

which for the optimal trajectory section on the $x^*(t) \in \partial G$ constraint will take the following value:

$$p[x^*(t), u^*(t), t] = 0, \quad t \in [t_e, t_d] \tag{14}$$

where t_e is the time of entry and t_d is the time of descent from the border of the restricted area ∂G_j.

Then, the optimal control u^* will be

$$u^*(t) = \frac{\dfrac{\dfrac{\dfrac{2\left(V_0^2 + V_j^2\right) - 2V_0\left(x_{2,0} - x_{2,j}\right) x_{4,0} \sin x_{3,0} + 2V_0\left(x_{1,0} - x_{1,j}\right) x_{4,0} \cos x_{3,0} -}{V_0 x_{5,0}\left[\left(x_{2,0} - x_{2,j}\right) \sin x_{3,0} - \left(x_{1,0} - x_{1,j}\right) \cos x_{3,0} + V_j x_{5,0} \sin\left(\psi_j - x_{3,0}\right)\right]} - V_0 V_j x_{4,0} x_{5,0} \sin\left(\psi_j - x_{3,0}\right) - 4V_0 V_j \cos\left(\psi_j - x_{3,0}\right) -}{\frac{k}{T}} - V_0 V_j x_{4,0}\left[3 \sin\left(\psi_j - x_{3,0}\right) - x_{4,0} x_{5,0} \cos\left(\psi_j - x_{3,0}\right)\right] -}{1} - V_0 x_{4,0}^2 x_{5,0}\left[\left(x_{1,0} - x_{1,j}\right) \sin x_{3,0} + \left(x_{2,0} - x_{2,j}\right) \cos x_{3,0}\right]}{1} + \frac{x_{4,0} + a x_{4,0}^2}{k} \tag{15}$$

3.3. Initial and Final Conditions

The initial state of the control process is determined by the following vector:

$$\vec{x}_0 = \begin{bmatrix} x^0_{1,0} \\ x^0_{2,0} \\ x^0_{3,0} \\ x^0_{4,0} = 0 \\ x^0_{5,0} \end{bmatrix} \quad (16)$$

and the final state of the control process determines the return of Object 0 to the initial course and is determined by the vector

$$\vec{x}_f = \begin{bmatrix} x^f_{1,0} \\ x^f_{2,0} \\ x^f_{3,0} = x^0_{3,0} \\ x^f_{4,0} = 0 \\ x^f_{5,0} \end{bmatrix} \quad (17)$$

The final conditions for conjugate variables λ_i can be selected using numerical methods. The final value of the *i*-th conjugate variable in the *k*-th iteration will be

$$\delta x_i^{(k)} = x_{i,f}^{(k)} - x_{i,f},\ \ i = 1,\ 2,\ \ldots,\ 5 \quad (18)$$

Expanding the expression for the conjugate variable λ_i into a Taylor series and omitting the second- and higher-order components, we obtain

$$\lambda_i^{(k)} = \lambda_{i,0} + \sum_{i=1}^{5} \frac{\delta \lambda_{i,0}}{\delta x_{i,k}} \delta x_i^{(k)},\quad \lambda_{i,0} = \lambda_i(t_0) \quad (19)$$

Assuming initial values for $i = 1, 2, \ldots, 5$ and $k = 1, 2, \ldots, 6$ and applying the Runge–Kutta integration method, we obtain

$$\begin{bmatrix} 1 & \delta x_1^{(1)} & \delta x_2^{(1)} & \delta x_4^{(1)} \\ 1 & \delta x_1^{(2)} & \delta x_2^{(2)} & \delta x_4^{(2)} \\ 1 & \delta x_1^{(3)} & \delta x_2^{(3)} & \delta x_4^{(3)} \\ 1 & \delta x_1^{(4)} & \delta x_2^{(4)} & \delta x_4^{(4)} \\ 1 & \delta x_1^{(5)} & \delta x_2^{(5)} & \delta x_4^{(5)} \\ 1 & \delta x_1^{(6)} & \delta x_2^{(6)} & \delta x_4^{(6)} \end{bmatrix} \begin{bmatrix} \lambda_{1,0} \ldots\ldots & \lambda_{5,0} \\ \frac{\delta \lambda_{1,0}}{\delta x^f_{1,0}} \ldots & \frac{\delta \lambda_{5,0}}{\delta x^f_{1,0}} \\ \frac{\delta \lambda_{1,0}}{\delta x^f_{2,0}} \ldots & \frac{\delta \lambda_{5,0}}{\delta x^f_{2,0}} \\ \frac{\delta \lambda_{1,0}}{\delta x^f_{4,0}} \ldots & \frac{\delta \lambda_{5,0}}{\delta x^f_{4,0}} \end{bmatrix} = \begin{bmatrix} \lambda_1^{(1)} \ldots \lambda_5^{(1)} \\ \lambda_1^{(2)} \ldots \lambda_5^{(2)} \\ \lambda_1^{(3)} \ldots \lambda_5^{(3)} \\ \lambda_1^{(4)} \ldots \lambda_5^{(4)} \\ \lambda_1^{(5)} \ldots \lambda_5^{(5)} \\ \lambda_1^{(6)} \ldots \lambda_5^{(6)} \end{bmatrix} \quad (20)$$

To assess the accuracy of calculations, the mean square error index is used:

$$E_k = \sum_{i=1}^{5} \left(\delta x_i^{(k)} \right)^2 \quad (21)$$

The derived mathematical relationships for the optimal control of Object 0 in the situation of safely passing other objects became the basis for the synthesis of the algorithm for determining the trajectory that meets these conditions.

3.4. Algorithm

Based on the relationships presented in the previous section, an Algorithm 1 for determining the safe optimal trajectory of Object 0 using the maximum principle was developed, which was written in Matlab/Simulink version 2023 software (Mathworks, Natick, MA, USA) and is presented below.

Algorithm 1: Safe trajectory optimization via the maximum principle.

BEGIN
1. Read Data: $x_{1,0}$, $x_{2,0}, x_{3,0}, x_{4,0}, x_{5,0}, a, k, T, V_0, \psi_0, x_{1,j}, x_{2,j}, V_j, \psi_j, D_s$
2. Input of initial conditions $\lambda_i^{(k)}$
$i = 1, 2, \ldots, 5; k = 1, 2, \ldots, 6$
3. Integration according to equations:
(9) and (11) or (15),
4. Calculation of $\delta x_i^{(k)}$ values (18),
5. **IF** not end of calculation
THEN
Calculation of values E_k (21) and finding max E_k,
Calculation of new λ_i values (21), and **GOTO** point 3,
ELSE IF Change of speed is true,
THEN
Calculation of the ship's new speed, and **GOTO** point 2,
ELSE Print trajectory: $(x_{1,0}, x_{2,0}, t)$,
END

The essence of the algorithm is its multi-stage nature, the identification of the most dangerous Object j at the following stage, and then solving the optimization task according to the maximum principle and ending the optimization task when all encountered Objects j are safely passed and there is a return to the initial reference travel course ψ_0 of Object 0. The necessary input comes from measurement devices, such as GPS, radar, gyrocompasses, and log devices. The results of the algorithm are displayed in the form of the movement trajectory of Object 0 in the subsequent moments of time. It is important to perform calculations of the optimal trajectory with the appropriate lead time necessary to implement the programmed control of the object.

4. Computer Simulation

Simulation tests of the algorithm were carried out in Matlab/Simulink software version 2023b on the example of navigation situations recorded on the research and training ship r/v HORYZONT II travelling from the Baltic Sea through the Danish Straits to Spitsbergen. Situations differing in the number of passed j Objects were adopted for comparative analysis.

In the recorded navigation situations, the object moved safely with low values of the safe passing distance D_s. For the purposes of simulating the tested algorithm, in order to artificially create a collision situation, the value of the safe passing distance D_s was increased to 1.0 nm.

The data from the first situation for j = 2 met Objects are presented in Table 1, and Figure 3 provides an illustration of the optimal trajectory of Object 0.

Table 1. Values, measured in the ARPA anti-collision system, of the state variables of the process of controlling the movement of Object 0 and j = 2 passing Objects.

Object j	Coordinate $x_{1,j}$ (nm)	Coordinate $x_{2,j}$ (nm)	Speed V_j (kn)	Course ψ_j (deg)
0	0	0	20	0
1	−2.8	2.4	20	95
2	2.7	2.5	10	275

The data from the second situation for j = 18 encountered Objects are presented in Table 2, and Figure 4 provides an illustration of the optimal trajectory of Object 0.

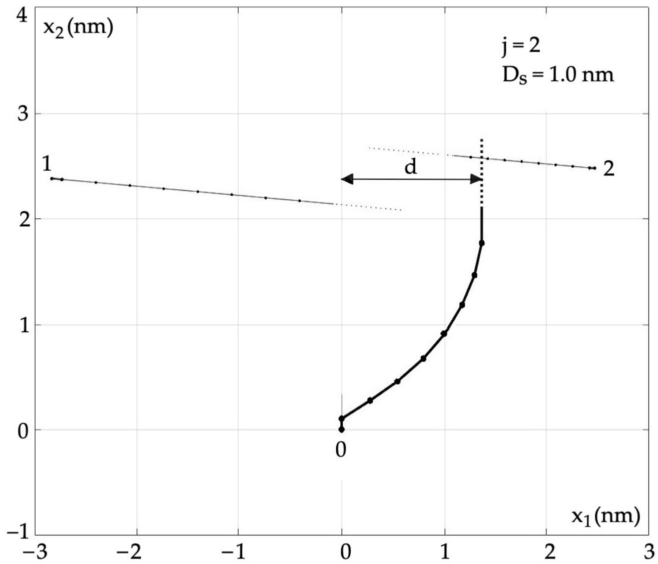

Figure 3. Optimal trajectory of Object 0 in the situation of safely passing j = 2 Objects.

Table 2. Values, measured in the ARPA anti-collision system, of the state variables of the process of controlling the movement of Object 0 and j = 18 passing Objects.

Object j	Coordinate $x_{1,j}$ (nm)	Coordinate $x_{2,j}$ (nm)	Speed V_j (kn)	Course ψ_j (deg)
0	0	0	13	80
1	2.6	−2.2	3	350
2	7.8	−4.3	4	330
3	2.2	2.1	5	220
4	−0.3	−0.9	5	100
5	3.7	−0.9	5	120
6	3.1	−3.9	7	58
7	4.2	4.0	5	150
8	3.0	0.2	1	85
9	8.7	−5.0	3	20
10	2.3	−3.1	2	350
11	3.9	−4.3	4	350
12	4.5	−6.3	7	0
13	1.8	−3.8	5	50
14	1.1	2.8	8	140
15	2.3	−7.7	6	50
16	0.9	5.1	15	150
17	−3.6	2.0	10	100
18	7.2	3.7	7	205

The data from the third situation for j = 34 encountered Objects are presented in Table 3, and Figure 5 provides an illustration of the optimal trajectory of Object 0.

The algorithm determines the optimal trajectory of Object 0 at constant speeds and courses of the encountered j Objects. Other j Objects can maneuver among themselves or in relation to Object 0, but if Object 0 detects the maneuver of another j Object, the calculation of the optimal safe trajectory of Object 0, which takes several seconds, must be repeated.

It is possible to take into account the maneuvering of other j Objects but by adopting a control model in the form of a positional or matrix game described in the works [22].

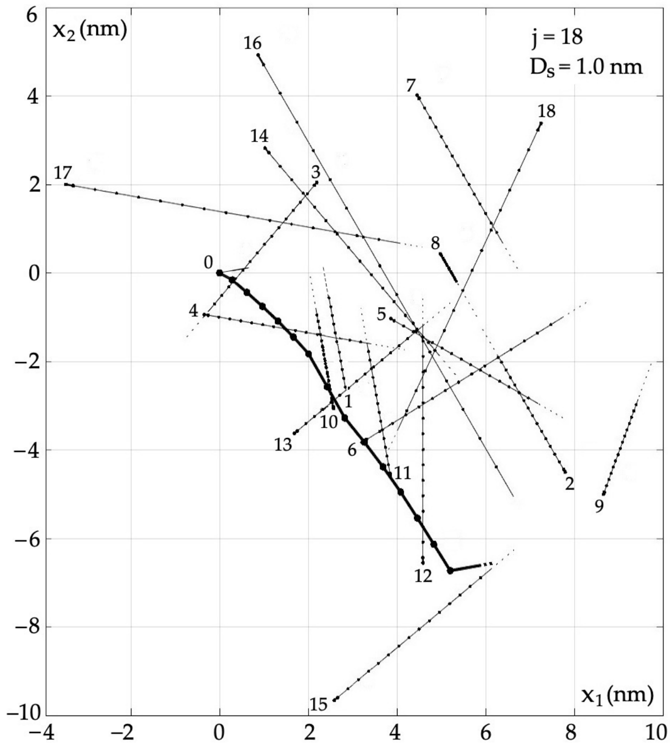

Figure 4. Optimal trajectory of Object 0 in the situation of safely passing j = 18 Objects.

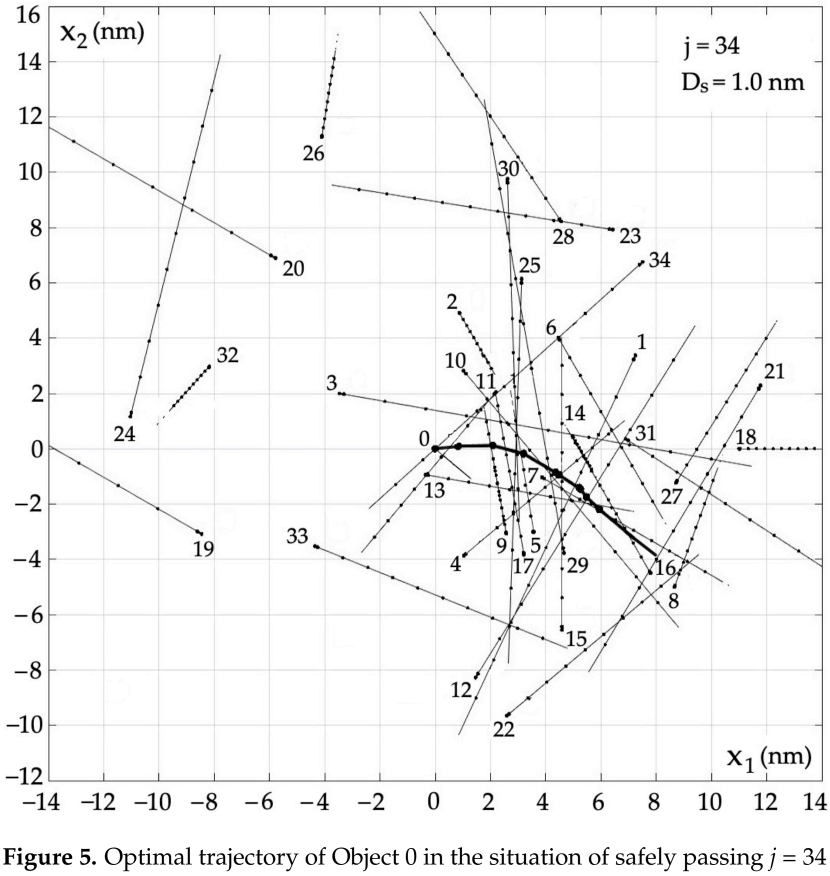

Figure 5. Optimal trajectory of Object 0 in the situation of safely passing j = 34 Objects.

Table 3. Values, measured in the ARPA anti-collision system, of the state variables of the process of controlling the movement of Object 0 and j = 34 passing Objects.

Object j	Coordinate $x_{1,j}$ (nm)	Coordinate $x_{2,j}$ (nm)	Speed V_j (kn)	Course ψ_j (deg)
0	0	0	11	130
1	7.2	3.3	10	205
2	0.8	5.0	2	150
3	−3.5	2.0	10	100
4	1.2	−3.9	5	50
5	3.9	2.9	4	350
6	4.3	4.0	5	150
7	3.9	−1.1	5	120
8	7.8	−4.9	3	20
9	2.7	3.1	2	350
10	1.1	2.9	8	140
11	2.2	2.1	5	220
12	2.8	−9.7	10	32
13	−0.4	−0.9	5	100
14	5.0	0.3	1	150
15	4.5	−6.5	7	0
16	7.9	−4.3	4	330
17	3.0	−3.9	3	350
18	11.2	0	2	90
19	−8.3	−3.0	11	300
20	−5.9	7.0	11	300
21	11.8	2.4	8	211
22	2.8	−9.7	6	50
23	6.3	8.0	7	279
24	−11.0	1.2	9	14
25	3.2	6.2	9	182
26	−4.0	11.5	2	9
27	8.9	−1.3	5	32
28	4.6	8.3	6	326
29	4.8	−3.9	11	350
30	2.6	9.9	8	178
31	7.0	0.5	7	123
32	−8.0	−3.0	3	222
33	−4.2	−3.7	7	112
34	7.7	6.7	9	228

By comparing the optimal trajectories of Object 0 in situations of safely passing different numbers of encountered Objects j, the following can be concluded:

- The final state of the optimization task, determined by the final deviation d of the safe trajectory from the initial course ψ_0, shown in Figure 3, depends not on the number of objects passed but rather the complexity of the navigation situation, measured by the risk of collision;
- The time needed to calculate the optimal trajectory, depending on the number of necessary maneuvering stages, also depends on the complexity of the navigation situation;
- However, the time needed to calculate the optimal trajectory depends very much on the number of objects passed.

In order to assess the quality of solving the task of optimizing the safe trajectory of Object 0 when passing other j Objects using Pontryagin's maximum principle, a comparison was made with the trajectory calculated using the Bellman dynamic programming method. The Bellman optimality principle, which is the basis for dynamic programming, was used. Minimum-time control was adopted as the optimality criterion. Passed Objects j were moving constraints of the process state in the form of a circle, hexagon, ellipse, or parabola and were generated by a previously trained artificial neural network. The domains of passing Objects j change their size depending on the risk of collisions and allow one to take into account the basic requirements of COLREGs rules. In conditions of good visibility, the principle of giving way to an object on the right and adopting the safe passing distance D_s in the range from 0.1 to 1.0 nm applies. However, in conditions of restricted visibility, only an increased value of the safe distance D_s in the range from 1.0 to 3.0 nm is used. The study, which was carried out on the example of the first scenario j = 2, is illustrated in Figure 6.

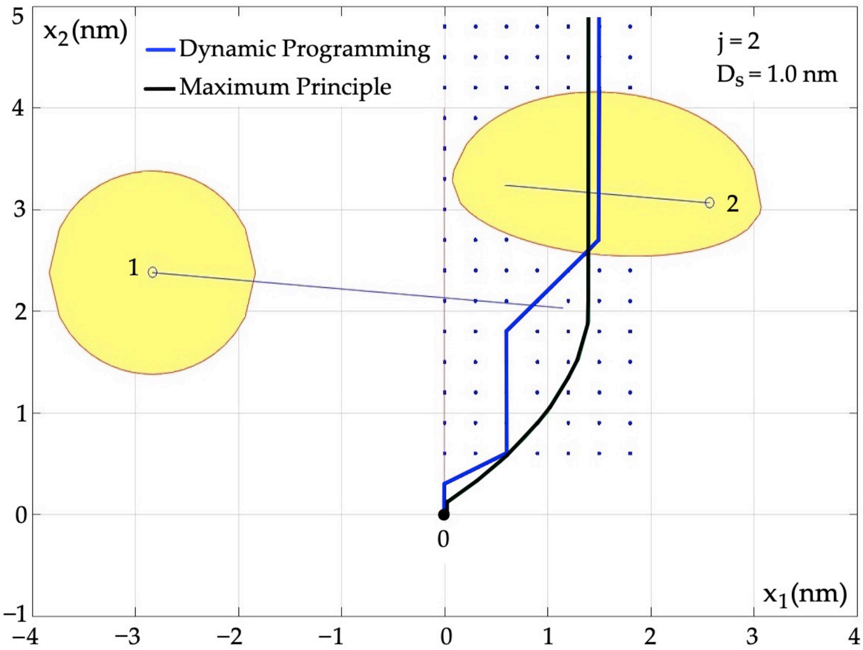

Figure 6. Comparison of the optimal trajectory of Object 0 when safely passing j = 2. Objects determined according to the maximum principle (black) and using the dynamic programming method (blue).

A comparison of the optimal trajectories of Object 0 in the situation of safely passing several Objects j, determined by the Bellman dynamic programming method and according to Pontryagin's maximum principle, shows the following:

- The advantages and disadvantages of one method are the disadvantages and advantages of another;
- The advantage of the maximum principle is its usefulness for a larger number of state variables describing the dynamics of Object 0, while the disadvantage is the extended computation time to take into account passing objects in the form of process state constraints;
- The advantage of dynamic programming is the ease of taking into account passed objects in the form of neural constraints on the state of the control process, which is reflected in the fact that the more constraints assigned to the encountered objects, the shorter the computation time of the optimal trajectory of Object 0. The disadvantage of dynamic programming, meanwhile, is the fact that the number of state variables is limited, with the number usually being limited to only a few variables;
- Both methods are equivalent and provide the same optimization result, in the form of the final deviation of the optimal trajectory of Object 0 from its initial heading;
- The safe trajectory determined according to the maximum principle, compared to the trajectory calculated using dynamic programming using a network of nodes, is smoother and easier to implement in practice when controlling Object 0.

5. Conclusions

The conducted synthesis of the analytical optimization of the multi-object safe control process according to Pontryagin's maximum principle and then its presentation in the form of an appropriate algorithm for calculating the safe Object 0 trajectory and its simulation studies allowed the following conclusions to be formulated:

- The presented study shows that it is possible to formulate an adequate model of the actual multi-object control process that will allow for its optimization while maintaining safe movement conditions;
- Optimization using Pontryagin's maximum principle on a multidimensional and nonlinear model of the control process correctly reproduces the complex dynamics of moving objects;
- The limitations of the maximum principle allow for an accurate representation of the essence of the actual control process;
- By appropriately formulating the final conditions of the optimization task, it is possible to synthesize the target algorithm for safe control of the object both back to the initial course and to a given point on its movement route;
- The maximum principle algorithm allows one to take into account a larger number of objects whose data come from ARPA anti-collision radar systems.

The following issues may be considered in future research on the presented topic:

- Application of the maximum principle with a penalty function to synthesize object control;
- Testing the sensitivity of optimal object control to the inaccuracy of information about the process state and to changes in environmental conditions, characterized by the safe proximity of objects;
- Comparison of the results of optimizing the object's trajectory according to the maximum principle with the optimization results obtained through the use of other static, dynamic, and game optimization methods.

Funding: This research was funded by the research project "Development of control and optimization methods for use in robotics and maritime transport" no. WE/2024/PZ, of the Electrical Engineering Faculty, Gdynia Maritime University, Poland.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Pontryagin, L.S. *Mathematical Theory of Optimal Processes*; Gordon and Breach Sciense Publishers: Great Britain, UK, 1987.
2. Zhou, X.Y. Maximum Principle, Dynamic Programming, and their Connection in Deterministic Control. *J. Optim. Theory Appl.* **1990**, *65*, 363–373. [CrossRef]
3. Hartl, R.F.; Sethi, S.P.; Vickson, R.G. A Survey of the Maximum Principles for Optimal Control Problems with State Constraints. *Soc. Ind. Appl. Math.* **1995**, *37*, 181–218. Available online: http://www.jstor.org/stable/2132823 (accessed on 8 January 2024). [CrossRef]
4. Vasilieva, O. Maximum principle and its extension for bounded control problems with boundary conditions. *Int. J. Math. Math. Sci.* **2004**, *2004*, 130179. [CrossRef]
5. Abramowski, P.; Abramowski, T.; Zwierzewicz, Z. Formal solution of ship weather routing problem via Pontryagin's maximum principle. *Sínt. Tecnol.* **2006**, *3*, 27–31. [CrossRef]
6. Danet, C.P. The classical maximum principle. Some of its extensions and applications. *Ann. Acad. Rom. Sci. Ser. Math. Its Appl.* **2011**, *3*, 2. Available online: https://api.semanticscholar.org/CorpusID:543385 (accessed on 10 January 2024).
7. Udin, U.; Pashentsev, S.; Petrov, S. Using Pontryagin maximum principle for parametrical identification of ship maneuvering mathematical model. *Transp. Probl.* **2014**, *9*, 11–18.
8. Mardanov, M.J.; Sharifov, Y.A. Pontryagin's Maximum Priciple for trhe Optimal Control Problems with Multipoint Boundary Conditions. *Abstr. Appl. Anal.* **2015**, *2015*, 428042. [CrossRef]
9. Sands, T. Virtual Sensoring of Motion Using Pontryagin's Treatment of Hamiltonian Systems. *Sensors* **2021**, *21*, 4603. [CrossRef]
10. Parsian, A. An extension of maximum principle with some applications. *Turk. J. Math.* **2021**, *45*, 66–80. [CrossRef]

11. Cortissoz, J.C.; Torres-Orozco, J. A note on the Maximum Principle and the Iteration Method for elliptic equations. *arXiv* **2021**, arXiv:2106.12650. [CrossRef]
12. Buldaev, A.; Kazmin, I. Operator Methods of the Maximum Principle in Problems of Optimization of Quantum Systems. *Mathematics* **2022**, *10*, 507. [CrossRef]
13. Alpay, D.; Pinhas, A.; Vinnikov, V. Commuting operators over Pontryagin spaces with applications to system theory. *J. Funct. Anal.* **2023**, *284*, 109864. [CrossRef]
14. Lu, L.; Zhao, H.; Liu, X.; Sun, C.; Zhang, X.; Yang, H. MPC-ECMS Energy Management of Extended-Range Vehicles Based on LSTM Multi-Signal Speed Prediction. *Electronics* **2023**, *12*, 2642. [CrossRef]
15. Xu, J.; Sands, T. Autonomous Drone Electronics Amplified with Pontryagin-Based Optimization. *Electronics* **2023**, *12*, 2541. [CrossRef]
16. Nomoto, K.; Taguchi, K.; Honda, K.; Hirano, S. On the steering qualities of ships. *Int. Shipbuild. Prog.* **1957**, *4*, 354–370. [CrossRef]
17. Das, S.; Talole, S.E. Nonlinear Control of Marine Surface Vessels. *J. Inst. Eng. Ser. C* **2019**, *100*, 385–400. [CrossRef]
18. Lisowski, J. Creating Autonomous Multi-Object Safe Control via Different Forms of Neural Constraints of Dynamic Programming. *Electronics* **2024**, *13*, 936. [CrossRef]
19. Leitmann, G. *The Calculus of Variations and Optimal Control Theory*; Plenum Press: New York, NY, USA, 1981; ISBN 978-0306407079.
20. Kahne, S. Note on two-point boundary value problems. *IEEE Trans. Autom. Control* **1963**, *8*, 257–258. [CrossRef]
21. Mazzia, M.; Cash, J.R.; Soetaert, K. Solving boundary value problems in the open source software R: Package bvpSolve. *Opusc. Math.* **2014**, *34*, 387–403. [CrossRef]
22. Lisowski, J. Artificial Intelligence Methods in Safe Ship Control Based on Marine Environment Remote Sensing. *Remote Sens.* **2023**, *15*, 203. [CrossRef]