Application of artificial neural networks in tasks to support safe maneuvering of the vessels in confined waters

Sergey Bidenko¹, Evgeniy Borodin², Aleksandr Yashin¹, Ivan Gorychev³, Andrey Degtyrev³

¹Inteltch, 8, Kantemirovskaya str., 197342, St. Petersburg, Russia
²State Maritime University named after Admiral F.F. Ushakova, 93, Lenin Ave., 353918, Novorossiysk, Russia
³Kerch State Marine Technological University, 82, Ordzhonikidze Str., Kerch, Russia

E-mail: sergiiblack@mail.ru

Abstract. Sailing in restricted areas (coastal zone, approaches to ports, narrows, and shallow waters) is the most difficult from the navigational point of view. There, the navigational situation changes dynamically due to the complexity of the bottom topography, shoals, the variability of hydrometeorological conditions (wind, currents, visibility, precipitation), the complex configuration of the fairway, the possible appearance of small vessels (high-speed boats and scooters, yachts, fishing boats) steering courses different from recommended ones, etc. An algorithm based on the artificial neural networks is presented for making recommendations on collision avoidance with dangerous objects and safe maneuvering. Limitations in the use of the algorithm are revealed and options for its practical application and further improvement are proposed.

1. Introduction
At the same time, the problem of changing the ship's course constantly arises. This task is being solved to avoid emerging dangers to navigation. Certain techniques, approaches, and methods are known to support the steering of a vessel maneuvering in severe navigation conditions [1-14]

These are artificial intelligence methods and systems, semantic networks, trajectory approaches, assessments and predictions of navigation situation, zone navigation, situational management, analysis of hierarchies, key indicators, etc. When using these support systems, it takes some time to perform calculations to make recommendations for avoidance of dangerous objects and safe maneuvering.

Less estimated time is required when using the techniques of artificial neural networks (ANNs), since they rely on a priori training samples.

ANNs are used here in the secondary processing of signals received from the radar for training in a special sample consisting of formal mathematical representations of possible recommended (safe) options for maneuvering the vessel.

Let us explain the gist of the method on the following example of the simplest situation regarding the ship’s steering.

Suppose that a vessel proceeds in a certain water area using a control element that has a certain radius of "vision" (observation). To simplify, we use the circle of vision, but we can also consider a real cone truncated by six intersecting planes. It is worth noting that the control element in our situation is a function by which external factors from the viewing radius are fed to the input, and the
vessel’s response to these factors is output. Due to the fact that there can be several such factors, i.e., several inputs and outputs, we will use a back propagation neural network, since it allows approximation of such functions. In fact, such a network is a multilayer perceptron in which the back propagation method of error trains all layers in one pass [15]. An example of the network architecture used is shown in Figure 1.

![Network Architecture](image)

**Figure 1.**

The advantage of this type of the network is its ability to training. The advantage of this network is also its learning ability, which will simplify the task of placing weights for neurons. To solve this problem, we will consider a neural network consisting of 6 layers. It was proved experimentally that sufficient accuracy is achieved with a minimum of 4 layers, however, with 6 layers, the calculation accuracy is higher, while a further increase in the number of layers does not give a noticeable increase in accuracy, while significantly increasing the program run time and the load on machine power. To simplify the model, we will place three neurons in the input layer, and two in the output layer. We will place 9 neurons in the intermediate layers (a further increase in their number does not make sense, not giving an increase in accuracy, but slowing down the system).

As a function of neuron activation, we will use the sigmoid function $S(x) = \frac{1}{1 + \exp(-x)}$ [16].

This function, as it is known, allows one to amplify weak signals, which makes it preferable for solving a specific problem. In addition, it can significantly reduce the computational complexity of the error back propagation method.
In order to reckon the position, the position \((x, y)\) of every object, the position \((x, y)\) of the vessel and the angle of the vehicle (Figure 2) are required. One also needs \(r\) (radius of a circle) and \(d_{\text{right}}, d_{\text{left}}\), i.e. vectors between the vessel and the lines \(L_{\text{right}}\) and \(L_{\text{left}}\) which are parallel to the direction of the vessel's movement. Both vectors are perpendicular to the lines. For the sake of simplicity, let us imagine that our pattern is two-dimensional, since the vessel cannot move in the third dimension, as she does not take off and does not dive. Thus, we note that reefs and underwater objects are not taken into account in this pattern. To account for these objects, it is necessary to complicate the pattern and introduce the third dimension.

For each object in the field of vision, we determine whether it is in the left or right field of view, or in the center. Array A is fed to the input to the neural network. Distances to the nearest obstacle to the left, in the center, and to the right of the vehicle will be stored in \(A[0]\), \(A[1]\) and \(A[2]\), respectively.

Let us calculate the equations for lines \(L_{\text{right}}\) and \(L_{\text{left}}\), which contributes to determining where the obstacle is located, i.e. to the right, to the left or in the centre from the watercraft.

\[
a_R x + b_R y + c_R = 0, \tag{1}
\]

where \(a_R = d_{x}^{\text{Right}}\); \(b_R = d_{y}^{\text{Right}}\).

Let us estimate the coordinates of the point on the line

\[
x_1 = V_x + d_{x}^{\text{Right}},
\]

\[
x_2 = V_y + d_{y}^{\text{Right}}. \tag{2}
\]

where \(V_x\) and \(V_y\) are the position of the vehicle. Let us estimate \(c_r\).
For a similar purpose we derive the line equation using the \( \mathbf{d}_{\text{eff}} \) vector. Next, it is required to calculate the center of the circle. Everything inside the circle will be visible to the control element. The center of the circle \( C(x, y) \) is at distance \( r \) from the position of the vessel \( V(x, y) \).

\[
C_x = V_x + r \cdot \sin(\alpha); \\
C_y = V_y + r \cdot \cos(\alpha),
\]

where: \( V_x, V_y \) is the position of the vessel; \( C_x, C_y \) is the centre of the circle. Then it should be checked on if the object is within the circle. If

\[
\sqrt{(O_x - C_x) \cdot (O_y - C_y)} \leq r,
\]

where \( O_x, O_y \) are the position of the obstacle, then the object is located within the circle.

For every object within the circle, we should check on whether it is located to the right, left, or center of the vessel. If \( a \mathbf{R} \cdot x + b \mathbf{R} \cdot y + c \mathbf{R} > 0 \), then the object is on the right side of the circle, otherwise if \( a \mathbf{L} \cdot x + b \mathbf{L} \cdot y + c \mathbf{L} \), then it is on the left side, otherwise it is in the center. Let us calculate the distance from the object to the vessel:

\[
\text{DIST}=\sqrt{(O_x - C_x) \cdot (O_y - C_y)}
\]

Now we save the distance in the respective area of the array (\( A[0] \), \( A[1] \) or \( A[2] \)), provided that the previously saved distance is greater than the just calculated one. Initially, array \( A \) must be initialized with \( 2r \) values. After checking every object, we have array \( A \) with distances to the nearest objects to the right, in the center and to the left of the vessel. If no objects were found in this field of vision, the array element will have a default value of 0, which means there are no objects in the radius of vision. Since the neural network uses a sigmoid function, the input data should be in the range from 0.0 to 1.0 [17]. 0.0 will mean that the object is touching the vehicle and 1.0 means that there are no objects within visibility. Since we set the maximum distance that the control can see, we can easily bring all distances to a range from 0.0 to 1.0.

\[
A[i]=A[i]/(2\cdot r).
\]

In the output, we have instructions for changing the speed of the vessel and course (heading). It can be acceleration, slowing down and steering angle. So we need two outputs; one will be the acceleration / slowing down value (sowing down is negative acceleration), and the other will indicate a change in the heading.

The result is between 0.0 and 1.0 for the same reason as the input. For acceleration, 0.0 means “stop engine”; 1.0 - “full ahead” and 0.5 - no stopping or acceleration. For steering, 0.0 means “hard to port”, 1.0 means “hard to starboard” and 0.5 means do not change the heading. In addition, the results should be transformed into values that can be used:

\[
\text{Acc}=(\text{Out}[0]-0.5)*2*\text{MAcc}; \\
\text{Str}=(\text{Out}[0]-0.5)*2*\text{MStr}.
\]

"Negative acceleration" will mean slowing down and stopping if the watercraft is proceeding forward, or moving in the reverse direction if the ship is at rest. In addition, “positive acceleration” means slowing down if the vehicle is proceeding in the reverse direction.

It is also worth noting that this neural network needs long-term training on a large number of sets [18, 19]. We briefly consider the network learning algorithm in this case:
1. Network d-drive
2. Calculation of the output element error
3. The calculation of the value of the adjustment of the link weights
4. Error detection of elements of the first hidden layer
5. Adjustment of the link weights
6. Estimation the amount of adjustment of the remaining weights. In fact, it is the completion of the network backward drive.

Note that in this case, the classical problems of the error back propagation algorithm associated with the capability of infinite learning are solved by the choice of the descent step, which was calculated empirically. Furthermore, a set of training data was formed on the basis of practical observations of the behavior of a neural network on a simulator. It integrates 500 training sets, that is, 500 vectors consisting of 5 values, i.e. three ones relevant for inputs and two vectors relevant for outputs.

In addition, a note should be made of a number of problems that arise while using this pattern. They arise due to the adopted simplifications in the pattern of space under consideration.

The ship can maintain her position for a while, because she is hesitant in deciding whether to go, i.e. left or right. It is not so easy to correct this, trying to adjust the weights of the neural network. The solution may be to add a coercive action in the event of stopping for more than the specified time.

The vehicle does not distinguish between a small gap between two reefs. Since the model did not initially incorporate a high level of accuracy in vision (only three positions: left, center, right), two objects located close to each other will look like a wall for artificial intelligence. To solve this problem, it is necessary to introduce more various options for the position of obstacles.

To accelerate the work of this network in situations with a more complex field of vision and greater accuracy of decision-making, it is possible to use graphics processors; however, this optimization is beyond the scope of this paper.

This neural network on its own has a fuzzy destination, however, when one adds a specific goal and connect a neural network that analyzes the situation in the near field, one can get a full-fledged formal techniques responsible for the movement of a vessel or other vehicle.

References
[1] Astrein V V 2011 Development of technologies for decisions to prevent collisions at sea, Extended abstract of Cand. Sci. (Eng.) Dissertation (Novorossiysk) p 24
[2] Vasiliev S N 2000 Intelligent control of dynamic systems (M.: Fizmatlit) p 352
[3] Vaguschenko L L, Vaguschenko A L 2010 Support for decisions to avoid collisions with vessels (Odessa: Fenix) p 229
[4] Vaskov A S, Mironenko A A Navigation 2003 Proceedings Novorossiysk NSMA 8 5
[5] Vaskov V A 2013 Proceedings Novorossiysk NSMA 3 15
[6] Dmitriev V I 2010 Modern navigation systems and navigation safety (M.: Morkniga) p 160
[7] Maltsev A S 2005 Maneuvering of vessels to avoid collisions (Odessa: CPAP) p 208
[8] Maltsev A N, Bernstein L S, Korovin S Ya 1990 Situational advisory systems with fuzzy logic (M.: Nauka)
[9] Rodionov A I 1992 Automation of navigation (M.: Transport) p 192
[10] Smolentsev S V 2016 Vestnik GUMRF 2(36) 7
[11] Smolentsev S V 2012 Operation of maritime transport 4(70) 11
[12] Smolentsev S V 2015 Bulletin of Admiral S.O. Makarov State University of the Sea and River Fleet 6(34) 23
[13] Subanov E E 2012 Development of models for the effective assessment of the danger of collisions when making decisions by analyzing hierarchies, Extended abstract of Cand. Sci. Dissertation (Novorossiysk) p 24
[14] Subanov E E, Mironov A V 2014 Operation of maritime transport 1 24
[15] Khaikin S 2006 Neural Networks: Complete Course 2nd Edition (M.: Williams Publishing
6

[16] Rogers J 1997 Object-Oriented Neural Network in C++ (San Diego, CA: Academic Press)
[17] Hagan M T, Demuth H B, Beale M H 1995 Neural Network Design (Boston, MA: PWS Publishing)
[18] Zhilenkov A, Chernyi S 2020 Automatic Documentation And Mathematical Linguistics 54(1) 36 doi: 10.3103/s0005105520010021
[19] Chernyi S G, Dorovskoy V A 2020 Autom. Doc. Math. Linguist. 54 73 https://doi.org/10.3103/S0005105520020028