

ЭКСПЛУАТАЦИЯ ВОДНОГО ТРАНСПОРТА, СУДОВОЖДЕНИЕ

DOI: 10.21821/2309-5180-2020-12-3-423-435

A GENERALIZED MODEL OF A VESSEL DEAD RECKONING BASED ON NEURAL NETWORKS

V. V. Deryabin

Admiral Makarov State University of Maritime and Inland Shipping,
St. Petersburg, Russian Federation

The task of predicting the vessel kinematic parameters arises in the dead reckoning mode of its track calculation. This task is solved in the framework of the traditional approach based on using the ordinary differential equations. As a rule, while constructing these equations, the difficulties arise. The difficulties result from selection of the algorithms which are used for the certain forces calculation. These algorithms are not universal, and their working ability could not be guaranteed for all sailing conditions. The task of predicting the vessel kinematic parameters can be performed as the task of approximation of multiple variables functions. Neural networks are known as universal algorithms of such approximation. In the paper, a generalized model of the vessel dead reckoning on the basis of neural networks is performed. An algorithm of its operation is performed as well. The base of the model is formed with deep neural nets which are constructed through a cascade connection between two-layered (shallow) networks. There are two types of neural networks in the structure of the model: main and auxiliary nets. The main nets predict kinematic parameters using the information about force influences on the vessel. The auxiliary nets use the information about only the ship kinematic history. The model configuration allows two modes of operation. The first mode is used when dead reckoning is performed in the usual way. The second mode is used when the reliable information from a log or (and) gyrocompass is unavailable. In the case of both sensors failure the model reproduces the vessel dynamics in the horizontal plane (with three degree of freedom) thus substituting the system of corresponding differential equations. The input signals of the main nets are formed in accordance with the configuration vector of the model. The capabilities of learning the model and its using in practice are also analysed.

Keywords: vessel dead reckoning, deep neural networks, differential equations, machine learning, cascade connection.

For citation:

Deryabin, Victor V. "A generalized model of a vessel dead reckoning based on neural networks." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S. O. Makarova* 12.3 (2020): 423–435. DOI: 10.21821/2309-5180-2020-12-3-423-435.

УДК 656.61.052:527.61:004.032.26

ОБОБЩЕННАЯ НЕЙРОСЕТЕВАЯ МОДЕЛЬ СЧИСЛЕНИЯ ПУТИ СУДНА

В. В. Дерябин

ФГБОУ ВО «ГУМРФ имени адмирала С. О. Макарова»,
Санкт-Петербург, Российская Федерация

Рассмотрена задача прогноза кинематических параметров судна, возникающая при определении его места в режиме счисления, решаемая в рамках традиционного подхода на основе обыкновенных дифференциальных уравнений движения. Отмечается, что при составлении данных уравнений, как правило, возникают трудности с выбором алгоритмов, по которым рассчитываются отдельные силы: не универсальны и работоспособность их в любых условиях плавания не может быть гарантирована. Задача прогноза кинематических параметров судна может быть представлена в виде аппроксимации (приближения) функции многих переменных, поскольку нейронные сети являются универсальными алгоритмами такой аппроксимации. В работе предлагается обобщенная модель счисления пути судна на основе нейронных сетей, а также алгоритм ее функционирования. Основу модели составляют глубокие нейронные сети,

полученные каскадным соединением двухслойных (мелких) сетей. В структуре модели два типа нейронных сетей: основные и дополнительные. Основные сети прогнозируют кинематические параметры на основе информации о силовых воздействиях на судно, вспомогательные — только на основе его кинематической истории. Конфигурация модели предусматривает режим обычного и аварийного счисления, когда отсутствует достоверная информация, поступающая от однокомпонентного относительного лага или / и гирокомпаса. При неисправности обоих датчиков модель счисления воспроизводит динамику движения судна в горизонтальной плоскости (с тремя степенями свободы), заменяя систему соответствующих дифференциальных уравнений. При этом входные сигналы основных нейронных сетей формируются в соответствии с вектором конфигурации модели. Проанализированы возможности настройки модели и ее использования на практике.

Ключевые слова: счисление пути судна, глубокие нейронные сети, дифференциальные уравнения, машинное обучение, каскадное соединение.

Для цитирования:

Дерябин В. В. Обобщенная нейросетевая модель счисления пути судна / В. В. Дерябин // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2020. — Т. 12. — № 3. — С. 423–435. DOI: 10.21821/2309-5180-2020-12-3-423-435.

Введение (Introduction)

Задача определения координат места судна автономными методами (счисления) является, как известно, актуальной научно-технической проблемой, так как счисление в ряде случаев может оказаться единственным источником навигационной информации. На большинстве судов торгового флота в качестве навигационных датчиков для счисления используются относительный лаг, измеряющий продольную скорость, а также гирокомпас. При такой комплектации основная проблема счисления заключается в адекватности учета внешних факторов (ветер, течение, волнение), для решения которой могут использоваться различные методы, основанные, как правило, на применении знаний из области обыкновенных дифференциальных уравнений [1]–[3].

При составлении дифференциальных уравнений движения судна всегда возникают проблемы, связанные с выбором методов учета силовых воздействий на корпус судна. Как правило, указанные методы имеют ограниченную область работоспособности, т. е. они не универсальны. Известно (см., например, [4]), что движение судна можно представить математически как решение задачи аппроксимации (приближения) функций многих переменных. В качестве инструментов универсальной аппроксимации, выступают искусственные нейронные сети (см., например, [5], [6]).

Идеи использования нейронных сетей для моделирования движения судна нашли широкое воплощение в современных исследованиях (см., например, [7]–[9]). При этом нейронные сети, как правило, применяются в задачах управления движением судна. Подобные задачи близки к задаче счисления, но все же не эквивалентны ей. Например, при построении прогнозирующей модели на основе нейронной сети требуется, чтобы данная модель была адекватной лишь на всем горизонте прогнозирования (как правило, речь идет о минутах и десятках минут). Счисление же может осуществляться сутками.

Целью настоящего исследования является разработка обобщенной нейросетевой модели счисления пути судна, а также алгоритма работы указанной модели. Для достижения данной обозначенной цели необходимо сначала записать дифференциальные уравнения движения судна, привести их к дискретному (по времени) виду, а затем показать, что решение данных уравнений может быть приближенно представлено как решение задачи аппроксимации функции многих переменных, для чего и следует использовать соответствующие нейронные сети.

Методы и материалы (Methods and Materials)

Математическая формулировка задачи. Дифференциальные уравнения движения судна в горизонтальной плоскости могут быть записаны следующим образом:

$$\begin{aligned} \frac{dV_x}{dt} &= f_1(V_x, V_y, \omega, p_{x1}, p_{x2}, \dots, p_{xN_x}); \\ \frac{dV_y}{dt} &= f_2(V_x, V_y, \omega, p_{y1}, p_{y2}, \dots, p_{yN_y}); \\ \frac{d\omega}{dt} &= f_3(V_x, V_y, \omega, p_{\omega 1}, p_{\omega 2}, \dots, p_{\omega N_\omega}), \end{aligned} \quad (1)$$

где V_x — продольная составляющая скорости судна относительно воды; V_y — поперечная составляющая скорости судна относительно воды; ω — угловая скорость рыскания (поворота вокруг вертикальной оси); $p_{x1}, p_{x2}, \dots, p_{xN_x}$ — набор из N_x величин, характеризующих силовые воздействия в продольном направлении; $p_{y1}, p_{y2}, \dots, p_{yN_y}$ — набор из N_y величин, характеризующих силовые воздействия в поперечном направлении; $p_{\omega 1}, p_{\omega 2}, \dots, p_{\omega N_\omega}$ — набор из N_ω величин, характеризующих силовые воздействия вокруг вертикальной оси; f_1, f_2, f_3 — непрерывные функции.

Применив к каждой части каждого уравнения системы (1) операцию интегрирования по времени на отрезке $[0; \tau]$, получим:

$$\begin{aligned} V_x(\tau) &= V_x(0) + \int_0^\tau f_1(V_x, V_y, \omega, p_{x1}, p_{x2}, \dots, p_{xN_x}) dt; \\ V_y(\tau) &= V_y(0) + \int_0^\tau f_2(V_x, V_y, \omega, p_{y1}, p_{y2}, \dots, p_{yN_y}) dt; \\ \omega(\tau) &= \omega(0) + \int_0^\tau f_3(V_x, V_y, \omega, p_{\omega 1}, p_{\omega 2}, \dots, p_{\omega N_\omega}) dt. \end{aligned} \quad (2)$$

Значения интегралов, стоящих в правых частях соотношений (2), определяются видом функций f_1, f_2, f_3 на отрезке времени $[0; \tau]$. В условиях использования цифровой техники вид указанных функций приближенно представляется последовательностью их значений в опорных равноотстоящих точках отрезка $[0; \tau]$: $t_0 = 0, t_1, t_2, \dots, t_i, \dots, t_n = \tau$, где n — число элементарных интервалов интегрирования; $i = 0 \div n$. Шаг интегрирования $\Delta t = t_1 - t_0, \dots, t_n - t_{n-1}$. Очевидно, что $\tau = \Delta t n$. При переходе к такой дискретной постановке задачи получаем, что интегралы в соотношениях (2) определяются последовательностями функций вида $f_l(t_0 = 0), f_l(t_1), \dots, f_l(t_n = \tau)$, где нижний индекс $l = 1, 2, 3$. В свою очередь, значение функции f_l в любой опорной точке определяется значениями ее аргументов в данной точке. Например, функция f_1 определяется последовательностями вида $\{V_x(t_i)\}_{i=0 \div n}; \{V_y(t_i)\}_{i=0 \div n}; \{\omega(t_i)\}_{i=0 \div n}; \{p_{x1}(t_i)\}_{i=0 \div n}; \{p_{x2}(t_i)\}_{i=0 \div n}; \dots; \{p_{xN_x}(t_i)\}_{i=0 \div n}$. Аналогичные рассуждения могут быть использованы и для функций f_2 и f_3 .

С учетом ранее изложенного, уравнения (2) могут быть заменены уравнениями следующего вида:

$$\begin{aligned} V_x(\tau = t_n) &\approx V_x(0) + \\ &+ f_x(\{V_x(t_i)\}_{i=0 \div n}, \{V_y(t_i)\}_{i=0 \div n}, \{\omega(t_i)\}_{i=0 \div n}, \{p_{x1}(t_i)\}_{i=0 \div n}, \dots, \{p_{xN_x}(t_i)\}_{i=0 \div n}); \\ V_y(\tau = t_n) &\approx V_y(0) + \\ &+ f_y(\{V_x(t_i)\}_{i=0 \div n}, \{V_y(t_i)\}_{i=0 \div n}, \{\omega(t_i)\}_{i=0 \div n}, \{p_{y1}(t_i)\}_{i=0 \div n}, \dots, \{p_{yN_y}(t_i)\}_{i=0 \div n}); \\ \omega(\tau = t_n) &\approx \omega(0) + \\ &+ f_\omega(\{V_x(t_i)\}_{i=0 \div n}, \{V_y(t_i)\}_{i=0 \div n}, \{\omega(t_i)\}_{i=0 \div n}, \{p_{\omega 1}(t_i)\}_{i=0 \div n}, \dots, \{p_{\omega N_\omega}(t_i)\}_{i=0 \div n}). \end{aligned} \quad (3)$$

где f_x, f_y, f_ω — некоторые непрерывные функции.

Общие черты модели счисления. Соотношения (3) представляют собой уравнения счисления для дискретного времени. Второе уравнение используется в том случае, когда необходимо определить скорость дрейфа судна (обычный режим счисления). Другие уравнения могут использоваться для «аварийного счисления» в случае, когда отсутствуют достоверные данные от лага или / и гирокомпа. Более подробно схема использования уравнений системы (3) выглядит следующим образом:

1) обычное счисление: используется только второе уравнение;

- 2) неисправен лаг (используются первое и второе уравнения);
- 3) неисправен гирокомпас (используется второе и третье уравнения);
- 4) неисправны лаг и гирокомпас (используются все три уравнения системы).

В режиме «аварийного счисления» используется два или три уравнения в определенном порядке. Например, в случае неисправности относительного лага можно использовать вначале первое уравнение, а затем второе или наоборот. Последовательность использования уравнений, возможно, имеет значение, так как алгоритмы, реализующие преобразования f_x, f_y, f_ω , могут иметь различную степень адекватности. Поэтому уравнения (3) должны использоваться в порядке ухудшения адекватности приведенных ранее алгоритмов. При этом меньшая адекватность одного алгоритма может быть скомпенсирована более точным определением его входных данных на основе другого более адекватного алгоритма. Вообще, возможные варианты использования системы уравнений (3) могут быть представлены в виде вектора конфигурации $K = (k_1 \ k_2 \ k_3)$, где его ненулевые элементы могут принимать неодинаковые значения от одного до трех. Например, $K = (0 \ 2 \ 0)$ соответствует варианту обычного счисления, $K = (1 \ 2 \ 0)$, $K = (2 \ 1 \ 0)$ — варианту, когда неисправен относительный лаг, $K = (0 \ 2 \ 3)$, $K = (0 \ 3 \ 2)$ — варианту, когда неисправен гирокомпас. В случае неисправности обоих датчиков варианты вектора конфигурации следующие: $K = (1 \ 2 \ 3)$, $K = (1 \ 3 \ 2)$, $K = (2 \ 1 \ 3)$, $K = (2 \ 3 \ 1)$, $K = (3 \ 1 \ 2)$, $K = (3 \ 2 \ 1)$.

Оптимальные (по точности) значения вектора K для режима «аварийного счисления» могут быть получены на основе большого объема эмпирических данных в ходе тестирования модели. Таким образом, для каждого варианта счисления может быть определен оптимальный вектор конфигурации K . Использование системы (3) с выбранным вектором конфигурации K позволяет организовать итерационный процесс Пикара. Например, после однократного вычисления V_x, V_y, ω можно повторить вычисления, но уже с уточненными входными данными. Цикл следует остановить, когда модули разностей значений величин V_x, V_y, ω для двух последовательных итераций будут одновременно менее установленных значений $\varepsilon_x, \varepsilon_y, \varepsilon_\omega$ соответственно. При использовании только одной итерации ограничения не устанавливаются. Таким образом, параметрами модели счисления выступают вектор конфигурации K и ограничения $\varepsilon_x, \varepsilon_y, \varepsilon_\omega$. Указанные параметры выбираются на этапе использования модели.

Реализация модели на основе нейронной сети. Функции f_x, f_y, f_ω могут быть реализованы при помощи нейронных сетей, из теории которых известно, что нейронная сеть прямого распространения с одним скрытым слоем способна аппроксимировать (равномерно приблизить) любую непрерывную функцию нескольких переменных с любой наперед заданной точностью [10]. Скрытые нейроны такой сети имеют функции активации, которые не являются многочленами и при этом они локально интегрируемые (по Риману). Выходной нейрон имеет при этом тождественную функцию активации. Общий вид такой сети представлен на рис. 1.

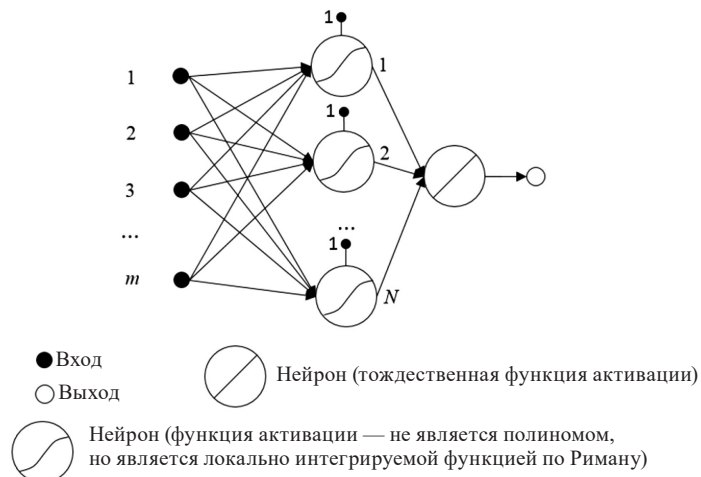


Рис 1. Общий вид архитектуры нейронной сети, используемой для универсальной аппроксимации функций

Основная проблема использования нейронных сетей с одним скрытым слоем заключается в том, что размер слоя (число нейронов) может быть очень большим для того, чтобы обеспечить требуемую точность приближения функции. В качестве возможного решения проблемы иногда имеет смысл перейти к «глубокой» архитектуре, т. е. использовать два и более скрытых слоя исходя из того, что суммарное число нейронов сети может быть сокращено при сохранении точности аппроксимации.

Наибольшую популярность приобрели глубокие сети прямого распространения, т. е. имеющие архитектуру, подобную приведенной на рис. 1, но между скрытым слоем и выходным у них располагаются дополнительные слои. Интуитивно идею использования дополнительных слоев можно объяснить следующим образом. Пусть имеется сеть с одним скрытым слоем («мелкая» нейронная сеть). Пусть она приближает с требуемой точностью непрерывную функцию m переменных $f = f(x_1, x_2, \dots, x_m)$. При добавлении еще одного слоя (ближе к выходу сети) на выходе каждого i -го нейрона первого скрытого слоя будет получаться функция $\tilde{f}_i = \tilde{f}_i(x_1, x_2, \dots, x_m)$, где $i = 1 \div N$. Впоследствии N таких функций представляют вход уже для обычной мелкой нейронной сети, гарантированно преобразующей их в функцию $f_1 = f_1(\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_N) = f_1(x_1, x_2, \dots, x_m)$. Подбором числа нейронов в каждом скрытом слое и настройкой параметров можно добиться того, что $f \approx f_1$. При этом вход для мелкой сети в виде функций $(\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_1, \dots, \tilde{f}_N)$ может потребовать для аппроксимации меньшее количество нейронов, чем если бы подавать на вход только (x_1, x_2, \dots, x_m) , а также общее количество нейронов может быть меньше для обеспечения такой же точности. Все ранее изложенное справедливо также в случае трех и более скрытых слоев. Основная идея заключается в том, что на вход мелкой сети (последнего скрытого слоя) поступает не вектор входных величин (x_1, x_2, \dots, x_m) , а вектор функций от указанных величин $(\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_N)$ таким образом сформированных, что требуемая точность аппроксимации достигается меньшим суммарным числом нейронов по сравнению с мелкой архитектурой.

Далее возникает вопрос о том, каким образом формируются указанные функции $(\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_N)$, представляющие вход для последнего скрытого слоя. Можно показать, что данные функции в структуре такой глубокой сети зависят от одинаковых весовых коэффициентов и порогов предшествующих скрытых слоев. Иными словами, регулировка любого коэффициента сети оказывает влияние на вид каждой функции из набора $(\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_N)$. Получается, что рассматриваемая глубокая структура как бы ограничивает возможности представления функций $(\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_N)$, подаваемых на вход последнего скрытого слоя, в чем и заключается ее недостаток. Для устранения выявленного ограничения можно использовать каскадное соединение мелких нейронных сетей, так что вид функций $(\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_N)$ регулируется независимо. Сеть работает следующим образом. Входной вектор $X = (x_1, x_2, \dots, x_m)$ подается одновременно на входы N_1 мелких нейронных сетей (см. рис. 2), имеющих архитектуру, представленную на рис. 1. Каждая такая сеть формирует N_1 функций переменных x_1, x_2, \dots, x_m . Каждая из таких функций подается вновь на входы N_2 мелких сетей, прогнозирующих другой набор функций: переменных x_1, x_2, \dots, x_m . Процесс повторяется до тех пор, пока не будут рассмотрены M слоев мелких сетей, причем $N_M \equiv 1$. Последняя сеть прогнозирует значение приближаемой функции. Общий вид архитектуры такой глубокой сети (ГС) представлен на рис. 2.

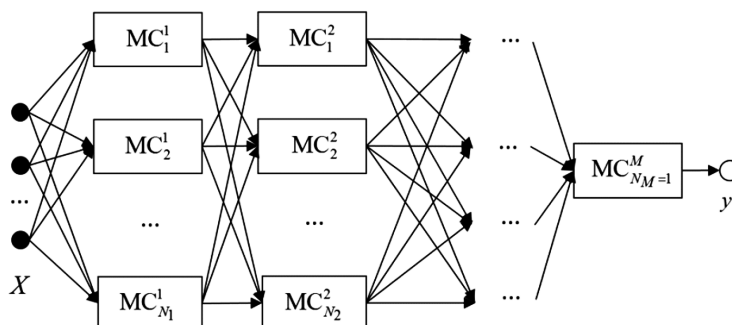


Рис 2. Общий вид архитектуры глубокой сети:
 МС — мелкая нейронная сеть; X — вектор входа; y — выходная величина

Представленная на рис. 2 сеть используется для нелинейного отображения $y = y(X)$, выступая в качестве «глубокого» аппроксиматора. Использование глубокой архитектуры позволит, возможно, добиться удовлетворительной точности аппроксимации меньшим числом скрытых нейронов по сравнению с мелкой архитектурой. Параметрами архитектуры ГС являются матрица архитектуры A . Каждый столбец такой матрицы соответствует слою скрытых сетей. Формирование строк матрицы начинается всегда с первой строки и заканчивается номером строки, который равен количеству сетей $N_{j=1 \div M}$ в данном скрытом слое. Элементы всех следующих строк принимаются нулевыми. Каждый ненулевой элемент матрицы архитектуры есть вектор, имеющий следующий вид: $A_{ij} = (s_{i=1 \div N_{j=1 \div M}}, \text{тип и параметры } \Phi A_{i=1 \div N_{j=1 \div M}})$, где s — число нейронов скрытого слоя мелкой нейронной сети $MS_{i=1 \div N_{j=1 \div M}}^{j=1 \div M}$, «тип ΦA » — вид функций активации нейронов ее скрытого слоя, «параметры ΦA » — значения соответствующих параметров. Матрицей архитектуры ГС (в сокращенном виде — \tilde{A}) будем называть матрицу, составленную из тех же элементов A_{ij} , но содержащих только число нейронов $s_{i=1 \div N_{j=1 \div M}}$. Рассмотрим следующую матрицу архитектуры ГС в сокращенном виде:

$$\tilde{A} = \begin{pmatrix} 10 & 15 & 5 & 20 \\ 10 & 15 & 5 & 0 \\ 10 & 0 & 5 & 0 \\ 0 & 0 & 5 & 0 \end{pmatrix}$$

Приведенная матрица задает ГС (см. рис. 2), первый слой которой содержит три мелкие сети, каждая из них содержит десять скрытых нейронов; второй слой содержит две мелкие сети, каждая из которых имеет по пятнадцать скрытых нейронов; третий слой содержит четыре мелкие сети с пятью скрытыми нейронами; четвертый слой представлен единственной сетью с двадцатью нейронами в скрытом слое. Таким образом, можно установить следующие свойства матрицы A (и \tilde{A}). Во-первых, число строк матрицы определяется наибольшим количеством «скрытых» сетей, используемых в каком-либо слое (в данном примере, третий слой имеет 4 сети, что и определяет число строк матрицы). Во-вторых, все элементы последнего столбца матрицы, за исключением первого в строке, всегда равны нулю. Указанное свойство объясняется скалярным характером выходного сигнала ГС.

Матрицу \tilde{A} имеет смысл использовать вместо матрицы A в том случае, когда тип и параметры функций активации определяются более или менее одинаковыми для всех мелких сетей, составляющих ГС. В таком случае, параметрами архитектуры ГС является матрица \tilde{A} и дополнительные характеристики функции активации.

Будем использовать ГС для аппроксимации зависимостей f_x, f_y, f_ω в выражении (3), прежде обозначив набор аргументов указанных функций через ${}_n X_x, {}_n X_y, {}_n X_\omega$. Например, в соответствии с соотношением (3), получим

$${}_n X_x = \begin{pmatrix} \{V_x(t_i)\}_{i=0 \div n}, \{V_y(t_i)\}_{i=0 \div n}, \{\omega(t_i)\}_{i=0 \div n}; \\ \{p_{x1}(t_i)\}_{i=0 \div n}, \dots, \{p_{xN_x}(t_i)\}_{i=0 \div n} \end{pmatrix}$$

Система уравнений (3) с учетом введенных обозначений примет вид:

$$\begin{aligned} V_x(\tau) &\approx V_x(0) + f_x({}_n X_x) \approx V_x^{\text{neuro}}(\tau) := V_x(0) + f_x^{\text{neuro}}({}_n X_x); \\ V_y(\tau) &\approx V_y(0) + f_y({}_n X_y) \approx V_y^{\text{neuro}}(\tau) := V_y(0) + f_y^{\text{neuro}}({}_n X_y); \\ \omega(\tau) &\approx \omega(0) + f_\omega({}_n X_\omega) \approx \omega^{\text{neuro}}(\tau) := \omega(0) + f_\omega^{\text{neuro}}({}_n X_\omega), \end{aligned} \quad (4)$$

где $f_x^{\text{neuro}}(\cdot), f_y^{\text{neuro}}(\cdot), f_\omega^{\text{neuro}}(\cdot)$ — преобразования, выполняемые глубокими нейронными сетями, $V_x^{\text{neuro}}(\tau), V_y^{\text{neuro}}(\tau), \omega^{\text{neuro}}(\tau)$ — нейросетевые оценки соответствующих кинематических параметров движения судна.

В дальнейшем нейронную сеть, реализующую одно из отображений $f_x^{\text{neuro}}(\cdot), f_y^{\text{neuro}}(\cdot), f_\omega^{\text{neuro}}(\cdot)$, будет называть *основной глубокой сетью* (ОГС) и для сокращения записей использовать следую-

щие обозначения: $ОГС_q^n$, где нижний индекс q принимает значения x, y или ω и показывает, какая величина прогнозируется; а верхний n , как и прежде, показывает число отрезков равной длины, образующих промежуток $[0; \tau]$. Например, символ « $ОГС_x^1$ » обозначает, что используется основная глубокая нейронная сеть, входной вектор которой ${}_1X_x$ (сеть прогнозирует $V_x^{neuro}(\tau)$), т. е. для его формирования необходимы данные только на концах отрезка $[0; \tau]$. В соответствии с соотношениями (4), нейронная сеть $ОГС_q^n$ прогнозирует приращение скорости на промежутке $[0; \tau]$. Однако, если начальную скорость подать на последний (выходной) нейрон через постоянный единичный вес, то получится, что $ОГС_q^n$ прогнозирует само значение скорости (например, $V_x^{neuro}(\tau)$). В дальнейшем будем считать, что такая операция выполнена, и сети $ОГС_q^n$ имеют последний нейрон, соединенный с начальной скоростью через ненастраиваемый коэффициент, равный единице.

Для любых вариантов конфигурации необходимо использовать *спрогнозированные* аналоги кинематических параметров $\hat{V}_x(\tau), \hat{V}_y(\tau), \hat{\omega}(\tau)$, полученные на основе их значений в предшествующие моменты времени. Например, при конфигурации $K = (0 \ 2 \ 0)$, в соответствии с соотношениями (4), для определения поперечной скорости как входной величины $V_y(\tau)$ необходимо (на первой итерации) использовать ее спрогнозированный аналог, т. е. значение $\hat{V}_y(\tau)$. В данном случае $V_y(\tau)$ является определяемой величиной и можно использовать лишь ее спрогнозированное значение $\hat{V}_y(\tau)$.

Необходимость использования спрогнозированных значений возникает также и в режимах «аварийного счисления». Например, при конфигурации $K = (1 \ 2 \ 0)$, в соответствии с соотношениями (4), для определения продольной скорости $V_x(\tau)$ необходимо знание поперечной для всех точек интервала, включая τ . Однако $V_y(\tau)$ определится лишь после решения второго уравнения. Поэтому в данном случае на первой итерации необходимо взять $V_y(\tau) \approx \hat{V}_y(\tau)$. Конечно, уже на второй итерации значение $V_y(\tau)$ берется как вычисленное на первой итерации. Таким образом, возникает задача прогноза временных рядов. Для прогноза (на шаг вперед) логично использовать глубокую нейронную сеть той же архитектуры (см. рис. 2), которую будем называть *вспомогательной глубокой сетью* (ВГС) и обозначать символом $ВГС_q^{lq}$, где $l_q \geq 1$ — количество интервалов, используемых для построения прогноза. Например, символ $ВГС_y^2$ обозначает нейронную сеть, которая прогнозирует скорость $V_y(t_3)$, если известны предыдущие значения $V_y(t_0 = 0), V_y(t_1), V_y(t_2)$, т. е. использует два интервала для построения прогноза.

Решение системы (4) может быть реализовано при различных значениях вектора конфигурации K , определяющего вид входных векторов: ${}_nX_x, {}_nX_y, {}_nX_\omega$. Как известно, существует одиннадцать вариантов вектора конфигурации. Прежде чем определять входные векторы для каждого варианта, уточним следующие обозначения. Символами V_x, V_y, ω обозначаются теоретические значения кинематических параметров движения судна, символами $\hat{V}_x, \hat{V}_y, \hat{\omega}$ — спрогнозированные значения на основе вспомогательных нейронных сетей $ВГС_q^{lq}$, $V_y^{neuro}, V_x^{neuro}, \omega^{neuro}$ — спрогнозированные значения при помощи основных нейронных сетей $ОГС_q^n$ на *текущей итерации*; $\tilde{V}_x^{neuro}, \tilde{V}_y^{neuro}, \tilde{\omega}^{neuro}$ — спрогнозированные значения при помощи основных нейронных сетей $ОГС_q^n$, взятые с *предыдущей итерации*; V_x^*, ω^* — значения параметров, полученные на основе измерений.

Вне зависимости от конфигурации, каждый из входных векторов ${}_nX_x, {}_nX_y, {}_nX_\omega$ содержит измеренные значения параметров $\{p_{q1}(t_i)\}_{i=0:n}, \dots, \{p_{qNq}(t_i)\}_{i=0:n}$, характеризующих силовые воздействия на судно.

Далее рассмотрим наиболее интересные для практики варианты конфигурации модели счисления, отличающиеся правилами формирования входных векторов ${}_nX_x, {}_nX_y, {}_nX_\omega$.

1. При векторе $K = (0 \ 2 \ 0)$ выполняется обычное счисление. Начальные значения кинематических параметров определяются следующим образом:

$$V_x(0), V_y(0), \omega(0) = \begin{cases} V_x^*(T=0), V_y(T=0), \omega(T=0), & \text{если } T / \tau = 1; \\ V_x^*(0), V_y^{neuro}(0), \omega^*(0), & \text{если } T / \tau > 1, \end{cases}$$

где T — общее время плавания.

Здесь и в дальнейшем предполагается, что угловая скорость поворота ω определяется численным дифференцированием курса судна, поэтому $\omega(0) = \omega(T = 0)$, если $T / \tau = 1$, а не $\omega(0) = \omega^*(T = 0)$.

Последовательности $\{V_x(t_i)\}_{i=1 \div n}$ и $\{\omega(t_i)\}_{i=1 \div n}$ заполняются измеренными значениями соответствующих величин, т. е. $\{V_x(t_i)\}_{i=1 \div n} = \{V_x^*(t_i)\}_{i=1 \div n}$ и $\{\omega(t_i)\}_{i=1 \div n} = \{\omega^*(t_i)\}_{i=1 \div n}$. Последовательность $\{V_y(t_i)\}_{i=1 \div n-1} = \{V_y^{\text{neuro}}(t_i)\}_{i=1 \div n-1}$. Естественно, что векторы ${}_n X_x, {}_n X_\omega$ не формируются.

В качестве значения $V_y(\tau)$ на первой итерации используется спрогнозированное вспомогательной нейронной сетью ВГС_y^{ly} значение $\hat{V}_y(\tau)$. При этом $l_y + 1$ должно быть не более общего количества имеющихся предыдущих значений скорости: $V_y(T = 0), V_y^{\text{neuro}}(T = 1\Delta t), V_y^{\text{neuro}}(T = 2\Delta t), \dots, V_y^{\text{neuro}}(t_{n-1})$. Для последующих итераций в качестве $V_y(\tau)$ берется значение, спрогнозированное основной сетью ОГС_yⁿ на предыдущей итерации, т. е. $\tilde{V}_y^{\text{neuro}}(\tau)$.

2. При векторе $K = (1 \ 2 \ 0)$ начальные значения кинематических параметров определяются следующим образом:

$$V_x(0), V_y(0), \omega(0) = \begin{cases} V_x(T = 0), V_y(T = 0), \omega(T = 0), & \text{если } T / \tau = 1; \\ V_x^{\text{neuro}}(0), V_y^{\text{neuro}}(0), \omega^*(0), & \text{если } T / \tau > 1. \end{cases}$$

Последовательность $\{\omega(t_i)\}_{i=1 \div n}$ заполняется измеренными значениями угловой скорости, т. е. $\{\omega(t_i)\}_{i=1 \div n} = \{\omega^*(t_i)\}_{i=1 \div n}$. Последовательность $\{V_x(t_i)\}_{i=1 \div n-1} = \{V_x^{\text{neuro}}(t_i)\}_{i=1 \div n-1}$ и $\{V_y(t_i)\}_{i=1 \div n-1} = \{V_y^{\text{neuro}}(t_i)\}_{i=1 \div n-1}$. Вектор ${}_n X_\omega$ не формируется.

Для сети ОГС_xⁿ конечная продольная скорость $V_x(t_n = \tau)$. На первой итерации используется значение $\hat{V}_x(\tau)$, спрогнозированное вспомогательной сетью ВГС_x^{lx}. При этом $l_x + 1$ должно быть не более общего количества имеющихся предыдущих значений скорости: $V_x(T = 0), V_x^{\text{neuro}}(T = 1\Delta t), V_x^{\text{neuro}}(T = 2\Delta t), \dots, V_x^{\text{neuro}}(t_{n-1})$. Для последующих итераций в качестве $V_x(\tau)$ берется спрогнозированное ОГС_xⁿ значение скорости с предыдущей итерации, т. е. $\tilde{V}_x^{\text{neuro}}(\tau)$.

Конечная поперечная скорость $V_y(t_n = \tau)$. На первой итерации используется значение $\hat{V}_y(\tau)$, спрогнозированное вспомогательной сетью ВГС_y^{ly}. Условие для l_y должно быть выполнено по аналогии с принципом, описанным ранее. Для последующих итераций в качестве $V_y(\tau)$ берется значение, спрогнозированное основной сетью ОГС_yⁿ на предыдущей итерации $\tilde{V}_y^{\text{neuro}}(\tau)$.

Для сети ОГС_yⁿ конечная продольная скорость $V_x(t_n = \tau)$. Используется значение $V_x^{\text{neuro}}(\tau)$, спрогнозированное ОГС_xⁿ для текущей итерации.

Конечная поперечная скорость $V_y(t_n = \tau)$. На первой итерации используется значение $\hat{V}_y(\tau)$, спрогнозированное вспомогательной сетью ВГС_y^{ly}. Для последующих итераций в качестве $V_y(\tau)$ берется значение, спрогнозированное основной сетью ОГС_yⁿ на предыдущей итерации $\tilde{V}_y^{\text{neuro}}(\tau)$.

3. При $K = (2 \ 1 \ 0)$ начальные значения кинематических параметров определяются следующим образом:

$$V_x(0), V_y(0), \omega(0) = \begin{cases} V_x(T = 0), V_y(T = 0), \omega(T = 0), & \text{если } T / \tau = 1; \\ V_x^{\text{neuro}}(0), V_y^{\text{neuro}}(0), \omega^*(0), & \text{если } T / \tau > 1. \end{cases}$$

Последовательность $\{\omega(t_i)\}_{i=1 \div n}$ заполняется измеренными значениями угловой скорости, т. е. $\{\omega(t_i)\}_{i=1 \div n} = \{\omega^*(t_i)\}_{i=1 \div n}$. Последовательность $\{V_x(t_i)\}_{i=1 \div n-1} = \{V_x^{\text{neuro}}(t_i)\}_{i=1 \div n-1}$ и $\{V_y(t_i)\}_{i=1 \div n-1} = \{V_y^{\text{neuro}}(t_i)\}_{i=1 \div n-1}$. Вектор ${}_n X_\omega$ не формируется.

Для сети ОГС_xⁿ конечная продольная скорость $V_x(t_n = \tau)$. На первой итерации используется значение $\hat{V}_x(\tau)$, спрогнозированное вспомогательной сетью ВГС_x^{lx}. Для последующих итераций в качестве $V_x(\tau)$ берется значение, спрогнозированное основной сетью ОГС_xⁿ на предыдущей итерации $\tilde{V}_x^{\text{neuro}}(\tau)$.

Конечная поперечная скорость $V_y(t_n = \tau)$. На первой итерации используется значение $\hat{V}_y(\tau)$, спрогнозированное вспомогательной сетью ВГС_y^{ly}. Для последующих итераций в качестве $V_y(\tau)$ берется спрогнозированное ОГС_yⁿ значение, спрогнозированное ОГС с предыдущей итерации, т. е. $\tilde{V}_y^{\text{neuro}}(\tau)$.

Для сети ОГС_xⁿ конечная продольная скорость $V_x(t_n = \tau)$. На первой итерации используется значение $\hat{V}_x(\tau)$, спрогнозированное вспомогательной сетью ВГС_x^{lx}. Для последующих итераций в качестве $V_x(\tau)$ берется значение, спрогнозированное основной сетью ОГС_xⁿ на предыдущей итерации $\tilde{V}_x^{\text{neuro}}(\tau)$.

Конечная поперечная скорость $V_y(t_n = \tau)$. Используется значение $V_y^{\text{neuro}}(\tau)$, спрогнозированное ОГС_yⁿ для текущей итерации.

4. При $K = (0 \ 2 \ 3)$ начальные значения кинематических параметров определяются следующим образом:

$$V_x(0), V_y(0), \omega(0) = \begin{cases} V_x^*(T=0), V_y(T=0), \omega(T=0), & \text{если } T/\tau = 1; \\ V_x^*(0), V_y^{\text{neuro}}(0), \omega^{\text{neuro}}(0), & \text{если } T/\tau > 1. \end{cases}$$

Последовательность $\{V_x(t_i)\}_{i=1 \div n}$ заполняется измеренными значениями продольной скорости, т. е. $\{V_x(t_i)\}_{i=1 \div n} = \{V_x^*(t_i)\}_{i=1 \div n}$. Последовательность $\{V_y(t_i)\}_{i=1 \div n-1} = \{V_y^{\text{neuro}}(t_i)\}_{i=1 \div n-1}$ и $\{\omega(t_i)\}_{i=1 \div n-1} = \{\omega^{\text{neuro}}(t_i)\}_{i=1 \div n-1}$. Вектор ${}_n X_x$ не формируется.

Для сети ОГС_yⁿ конечная поперечная скорость $V_y(t_n = \tau)$. На первой итерации используется значение $\hat{V}_y(\tau)$, спрогнозированное вспомогательной сетью ВГС_y^{ly}. Для последующих итераций в качестве $V_y(\tau)$ берется значение, спрогнозированное основной сетью ОГС_yⁿ на предыдущей итерации, т. е. $\tilde{V}_y^{\text{neuro}}(\tau)$.

Конечная угловая скорость $\omega(t_n = \tau)$. На первой итерации используется значение $\hat{\omega}(\tau)$, спрогнозированное вспомогательной сетью ВГС_ω^{lω}. Для последующих итераций в качестве $\omega(\tau)$ берется значение, спрогнозированное основной сетью ОГС_ωⁿ на предыдущей итерации, т. е. $\tilde{\omega}^{\text{neuro}}(\tau)$.

Для сети ОГС_ωⁿ конечная поперечная скорость $V_y(t_n = \tau)$. Берется значение, получаемое на выходе ОГС_yⁿ на текущей итерации, т. е. $V_y^{\text{neuro}}(\tau)$.

Конечная угловая скорость $\omega(t_n = \tau)$. На первой итерации используется значение $\hat{\omega}(\tau)$, спрогнозированное вспомогательной сетью ВГС_ω^{lω}. Для последующих итераций в качестве $\omega(\tau)$ берется значение, спрогнозированное основной сетью ОГС_ωⁿ на предыдущей итерации, т. е. $\tilde{\omega}^{\text{neuro}}(\tau)$.

5. При $K = (0 \ 3 \ 2)$ начальные значения кинематических параметров определяются следующим образом:

$$V_x(0), V_y(0), \omega(0) = \begin{cases} V_x^*(T=0), V_y(T=0), \omega(T=0), & \text{если } T/\tau = 1; \\ V_x^*(0), V_y^{\text{neuro}}(0), \omega^{\text{neuro}}(0), & \text{если } T/\tau > 1. \end{cases}$$

Последовательность $\{V_x(t_i)\}_{i=1 \div n}$ заполняется измеренными значениями продольной скорости, т. е. $\{V_x(t_i)\}_{i=1 \div n} = \{V_x^*(t_i)\}_{i=1 \div n}$. Последовательность $\{V_y(t_i)\}_{i=1 \div n-1} = \{V_y^{\text{neuro}}(t_i)\}_{i=1 \div n-1}$ и $\{\omega(t_i)\}_{i=1 \div n-1} = \{\omega^{\text{neuro}}(t_i)\}_{i=1 \div n-1}$. Вектор ${}_n X_x$ не формируется.

Для сети ОГС_ωⁿ конечная поперечная скорость $V_y(t_n = \tau)$. На первой итерации используется значение $\hat{V}_y(\tau)$, спрогнозированное вспомогательной сетью ВГС_y^{ly}. Для последующих итераций в качестве $V_y(\tau)$ берется значение, спрогнозированное основной сетью ОГС_yⁿ на предыдущей итерации, т. е. $\tilde{V}_y^{\text{neuro}}(\tau)$.

Конечная угловая скорость $\omega(t_n = \tau)$. На первой итерации используется значение $\hat{\omega}(\tau)$, спрогнозированное вспомогательной сетью ВГС_ω^{lω}. Для последующих итераций в качестве $\omega(\tau)$ берется значение, спрогнозированное основной сетью ОГС_ωⁿ на предыдущей итерации, т. е. $\tilde{\omega}^{\text{neuro}}(\tau)$.

Для сети ОГС_yⁿ конечная поперечная скорость $V_y(t_n = \tau)$. На первой итерации используется значение $\hat{V}_y(\tau)$, спрогнозированное вспомогательной сетью ВГС_y^{ly}. Для последующих итераций в качестве $V_y(\tau)$ берется значение, спрогнозированное основной сетью ОГС_yⁿ на предыдущей итерации, т. е. $\tilde{V}_y^{\text{neuro}}(\tau)$.

Конечная угловая скорость $\omega(t_n = \tau)$. Берется значение, получаемое на выходе ОГС $_\omega^n$, на текущей итерации, т. е. $\omega^{\text{neuro}}(\tau)$.

6. При $K = (1 \ 2 \ 3)$ начальные значения кинематических параметров определяются следующим образом:

$$V_x(0), V_y(0), \omega(0) = \begin{cases} V_x(T=0), V_y(T=0), \omega(T=0), & \text{если } T/\tau = 1; \\ V_x^{\text{neuro}}(0), V_y^{\text{neuro}}(0), \omega^{\text{neuro}}(0), & \text{если } T/\tau > 1. \end{cases}$$

Последовательности $\{V_x(t_i)\}_{i=1 \div n-1} = \{V_x^{\text{neuro}}(t_i)\}_{i=1 \div n-1}$, $\{V_y(t_i)\}_{i=1 \div n-1} = \{V_y^{\text{neuro}}(t_i)\}_{i=1 \div n-1}$ и $\{\omega(t_i)\}_{i=1 \div n-1} = \{\omega^{\text{neuro}}(t_i)\}_{i=1 \div n-1}$.

Для сети ОГС $_x^n$ конечная продольная скорость $V_x(t_n = \tau)$. На первой итерации используется значение $\hat{V}_x(\tau)$, спрогнозированное вспомогательной сетью ВГС $_x^{lx}$. Для последующих итераций в качестве $V_x(\tau)$ берется значение, спрогнозированное основной сетью ОГС $_x^n$ на предыдущей итерации, т. е. $\tilde{V}_x^{\text{neuro}}(\tau)$.

Конечная поперечная скорость $V_y(t_n = \tau)$. На первой итерации используется значение $\hat{V}_y(\tau)$, спрогнозированное вспомогательной сетью ВГС $_y^{ly}$. Для последующих итераций в качестве $V_y(\tau)$ берется значение, спрогнозированное основной сетью ОГС $_y^n$ на предыдущей итерации, т. е. $\tilde{V}_y^{\text{neuro}}(\tau)$.

Конечная угловая скорость $\omega(t_n = \tau)$. На первой итерации используется значение $\hat{\omega}(\tau)$, спрогнозированное вспомогательной сетью ВГС $_\omega^{lo}$. Для последующих итераций в качестве $\omega(\tau)$ берется значение, спрогнозированное основной сетью ОГС $_o^n$ на предыдущей итерации, т. е. $\tilde{\omega}^{\text{neuro}}(\tau)$.

Для сети ОГС $_y^n$ конечная продольная скорость $V_x(t_n = \tau) = V_x^{\text{neuro}}(\tau)$, т. е. берется непосредственно с выхода ОГС $_x^n$ на текущей итерации. Конечная поперечная скорость $V_y(t_n = \tau)$. На первой итерации используется значение $\hat{V}_y(\tau)$, спрогнозированное вспомогательной сетью ВГС $_y^{ly}$. Для последующих итераций в качестве $V_y(\tau)$ берется значение, спрогнозированное основной сетью ОГС $_y^n$ на предыдущей итерации, т. е. $\tilde{V}_y^{\text{neuro}}(\tau)$.

Конечная угловая скорость $\omega(t_n = \tau)$. На первой итерации используется значение $\hat{\omega}(\tau)$, спрогнозированное вспомогательной сетью ВГС $_o^{lo}$. Для последующих итераций в качестве $\omega(\tau)$ берется значение, спрогнозированное основной сетью ОГС $_o^n$ на предыдущей итерации, т. е. $\tilde{\omega}^{\text{neuro}}(\tau)$.

Для сети ОГС $_o^n$ конечная продольная скорость $V_x(t_n = \tau) = V_x^{\text{neuro}}(\tau)$, т. е. берется непосредственно с выхода ОГС $_x^n$ на текущей итерации. Конечная поперечная скорость $V_y(t_n = \tau) = V_y^{\text{neuro}}(\tau)$, т. е. берется значение, получаемое на выходе ОГС $_y^n$ на текущей итерации.

Конечная угловая скорость $\omega(t_n = \tau)$. На первой итерации используется значение $\hat{\omega}(\tau)$, спрогнозированное вспомогательной сетью ВГС $_o^{lo}$. Для последующих итераций в качестве $\omega(\tau)$ берется значение, спрогнозированное основной сетью ОГС $_o^n$ на предыдущей итерации, т. е. $\tilde{\omega}^{\text{neuro}}(\tau)$.

Оставшиеся пять вариантов вектора конфигурации K могут быть рассмотрены по аналогии с вариантами, рассмотренными ранее, и для сокращения записей не приводятся.

Результаты (Results)

Теперь можно определить параметры архитектуры нейросетевой модели счисления пути судна в виде следующего списка:

1. Вектор конфигурации K .
2. Итерационные ограничения $\varepsilon_x, \varepsilon_y, \varepsilon_\omega$ (при наличии).
3. Число шагов модели n и длительность временного интервала τ .
4. Матрицы архитектуры $A_{\text{огс}_x^i}, A_{\text{огс}_y^i}, A_{\text{огс}_o^i}$ (или сокращенные их версии $\tilde{A}_{\text{огс}_x^i}, \tilde{A}_{\text{огс}_y^i}, \tilde{A}_{\text{огс}_o^i}$ + характеристики функций активации) основных глубоких сетей ($i = 1 \div n$).
5. Матрицы архитектуры $A_{\text{вгс}_x^{lx}}, A_{\text{вгс}_y^{ly}}, A_{\text{вгс}_o^{lo}}$ (или сокращенные их версии $\tilde{A}_{\text{вгс}_x^{lx}}, \tilde{A}_{\text{вгс}_y^{ly}}, \tilde{A}_{\text{вгс}_o^{lo}}$ + характеристики функций активации) вспомогательных глубоких сетей.
6. Количества интервалов, используемых для построения прогноза вспомогательными сетями, l_x, l_y, l_ω (количества соответствующих точек, уменьшенные на единицу).

7. Наборы параметров $p_{x1}, p_{x2}, \dots, p_{xN_x}, p_{y1}, p_{y2}, \dots, p_{yN_y}, p_{\omega1}, p_{\omega2}, \dots, p_{\omega N_\omega}$, характеризующих управляющие и возмущающие воздействия в соответствии с выражением (1).

Алгоритм работы нейросетевой модели счисления:

1. Задаются параметры архитектуры модели.
2. Начинается определение кинематических параметров V_x, V_y, ω (в зависимости от вектора конфигурации K) на конец промежутка времени $[0; \tau]$:
 - 2.1. Прогноз значения на конец отрезка $[t_0 = 0; t_1]$.
 - 2.2. Прогноз значения на конец отрезка $[t_1; t_2]$.
 -
 2. n . Прогноз значения на конец отрезка $[t_{n-1}; t_n = \tau]$.

При выполнении указанных операций, описываемых п. 2.1–2. n , формируются соответствующие входные векторы ${}_n X_x, {}_n X_y, {}_n X_\omega$ в зависимости от вектора конфигурации K (с. 426), работают задействованные основные ОГС $_q^n$ и вспомогательные ВГС $_q^{lg}$ глубокие нейронные сети. При этом контролируется, чтобы $l_q \geq 1$, если данных недостаточно для выполнения прогноза с использованием установленного числа промежутков l_x, l_y, l_ω . В ходе каждой операции, описанной в пп. 2.1–2. n , реализуется в общем случае циклическая процедура, критерием выхода из которой является то, что изменение прогнозируемых величин от итерации к итерации не превосходит по модулю установленные значения $\epsilon_x, \epsilon_y, \epsilon_\omega$.

3. Границы промежутка $[0; \tau]$ сдвигаются на элементарный шаг Δt вперед (по мере поступления данных на момент $\tau + \Delta t$). Затем переход к пп. 2.

Обсуждение (Discussion)

Приведенный в проведенном исследовании алгоритм предполагает реализацию в режиме реального времени. Как только поступают данные на дискретный момент, алгоритм сразу начинает свою работу. Модель содержит основные и вспомогательные глубокие нейронные сети, которые обучаются, в общем случае, различным образом. Основные сети обучаются, как правило, на основе большого объема заранее накопленных данных, хотя теоретически нельзя исключать и настройку таких сетей в реальном времени (по мере поступления образцов). Их свободные параметры определяются физическими характеристиками конкретного судна. Вспомогательные сети обучаются только в режиме реального времени на выборках прогнозируемых параметров, т. е., как правило, в последовательном режиме. Их свободные параметры определяются непосредственно лишь предыдущими значениями прогнозируемых величин. Использование таких сетей необходимо только на первых итерациях циклов.

При малой временной дискретности Δt система должна «успеть» выполнить вычисления до следующего момента времени, что накладывает определенные ограничения, относящиеся к быстродействию навигационной вычислительной машины. Указанные ограничения не слишком критичны, так как основные сети обучаются заранее, либо последовательно (по мере поступления данных), а вспомогательные настраиваются обычно последовательно. В значительной степени быстродействие системы определяется выбранным вектором конфигурации K .

Алгоритм работы модели циклический. Во-первых, цикличность заключается в том, что для вычисления интересующих параметров на конце отрезка $[0; \tau]$, в общем случае, используется итерационная процедура. Во-вторых, цикличность выражается в том, что внутри любого отрезка $[0; \tau]$ необходимо последовательно рассмотреть n архитектур основных сетей, при этом индекс архитектуры пробегает значения от 1 до n .

Вектор конфигурации K для режимов аварийного счисления может быть подобран опытным путем на основе тестирования системы с использованием больших объемов данных, накапливаемых по мере повседневной эксплуатации судна. Можно подобрать такие конфигурации для режимов неисправности лага, гирокомпаса и обоих датчиков одновременно. Вектор конфигурации $K = (0 \ 2 \ 0)$ является единственным для режима обычного счисления. Таким образом, из одиннадцати возможных векторов конфигурации остаются четыре оптимальные.

Заключение (Conclusion)

В результате настоящего исследования разработана обобщенная нейросетевая модель счисления пути судна. При этом счисление понимается в широком смысле: традиционное и аварийное (когда неисправен один или более навигационных датчиков). В последнем случае нейросетевая модель фактически имитирует движение судна в горизонтальной плоскости, точнее динамику такого движения в условиях взаимодействия с окружающей средой. Модель прогнозирует продольную и поперечную составляющие линейной скорости судна, а также составляющую угловой скорости вокруг вертикальной (к плоскости горизонта) оси. Основу обобщенной модели составляют глубокие нейронные сети, которые выполняют задачи аппроксимации (приближения) функций многих переменных, возникающие, как показано, в процессе выработки счисляемых координат места судна. Глубокие сети представляют собой каскадные соединения обычных («мелких») двухслойных нейронных сетей прямого распространения, обладающих свойствами универсальной аппроксимации. Сформирован список параметров обобщенной модели счисления пути судна на основе нейронных сетей. Предложен алгоритм работы синтезированной модели счисления, реализация которого возможна в судовой навигационной аппаратуре.

Разработанная модель имеет достаточно сложную архитектуру с большим числом параметров, определяющих ее конечную точность. Следует отметить, что определение оптимального набора указанных параметров модели — задача сложная, требующая дополнительных исследований. Проблема выбора параметров усугубляется тем, что оптимальные их значения зависят в общем случае от условий плавания конкретного судна и его физических характеристик. В связи с этим получение теоретического решения в общем виде (для всех судов и условий плавания) является затруднительным.

Предлагаемая обобщенная нейросетевая модель счисления и алгоритм ее работы представляют собой всего лишь начальный этап общего пути построения работоспособной нейросетевой системы счисления. Следующими этапами являются процесс подготовки набора учебных данных (образцов), непосредственная настройка (обучение) системы и, наконец, ее тестирование на основе реальных данных, полученных в процессе повседневной эксплуатации судна. Разработка методов реализации указанных этапов и представляет собой задачу дальнейших исследований.

СПИСОК ЛИТЕРАТУРЫ

1. Дерябин В. В. Модель счисления пути судна в условиях воздействия внешних факторов / В. В. Дерябин // Эксплуатация морского транспорта. — 2011. — № 1 (63). — С. 33–39.
2. Дмитриев С. П. Задачи навигации и управления при стабилизации судна на траектории: научное издание / С. П. Дмитриев, А. Е. Пелевин. — СПб.: ЦНИИ «Электроприбор», 2002. — 160 с.
3. Юдин Ю. И. Идентификация математической модели контейнеровоза проекта Arctic Container Ship ACS 650 для различных этапов швартовки / Ю. И. Юдин, С. В. Пашенцев, Б. В. Дабижа, С. О. Петров // Эксплуатация морского транспорта. — 2019. — № 3 (92). — С. 89–107. DOI: 10.34046/aumsuomt92/15.
4. Дерябин В. В. Использование нейронных сетей для стабилизации судна на траектории / В. В. Дерябин // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2018. — Т. 10. — № 4. — С. 665–678. DOI: 10.21821/2309-5180-2018-10-4-665-678.
5. Haykin S. Neural Networks and Learning Machines / S. Haykin. — Third Edition. — New Jersey: Pearson, 2009. — 936 p.
6. Liu Z. Neural network as a function approximator and its application in solving differential equations / Z. Liu, Y. Yang, Q. Cai // Applied Mathematics and Mechanics. — 2019. — Vol. 40. — Is. 2. — Pp. 237–248. DOI: 10.1007/s10483-019-2429-8.
7. Дерябин В. В. Обзор исследований, посвященных использованию нейросетевых технологий в судовождении / В. В. Дерябин // Вестник государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2015. — № 6 (34). — С. 29–43. DOI: 10.21821/2309-5180-2015-7-6-29-43.
8. Подпорин С. А. Использование нейро-нечетких контроллеров в системах управления движением морских судов / С. А. Подпорин // Збірник наукових праць Харківського університету Повітряних Сил. — 2012. — № 4 (33). — С. 181–187.

9. Kula K. S. Model-based controller for ship track-keeping using neural network / K. S. Kula // 2015 IEEE 2nd International Conference on Cybernetics (CYBCONF). — IEEE, 2015. — Pp. 178–183. DOI: 10.1109/CYB-Conf.2015.7175928.

10. Hornik K. Some new results on neural network approximation / K. Hornik // Neural Networks. — 1993. — Vol. 6. — Is. 8. — Pp. 1069–1072. DOI: 10.1016/S0893-6080(09)80018-X.

REFERENCES

1. Deryabin, V. V. “Vessel’s dead reckoning model under external factors conditions.” *Ekspluatatsiya morskogo transporta* 1(63) (2011): 33–39.

2. Dmitriev, S. P., and Pelevin A. E. *Zadachi navigatsii i upravleniya pri stabilizatsii sudna na traektorii*. SPb.: Elektropribor, 2002.

3. Yudin, Yu. I., S. V. Pashentsev, B. V. Dabizha, and S. O. Petrov. “Arctic Container Ship ACS 650 mathematical model identification for various stages of mooring.” *Ekspluatatsiya morskogo transporta* 3(92) (2019): 89–107. DOI: 10.34046/aumsuomt92/15.

4. Deryabin, Victor V. “Neural networks implementation for a vessel track control.” *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S. O. Makarova* 10.4 (2018): 665–678. DOI: 10.21821/2309-5180-2018-10-4-665-678.

5. Haykin, Simon. *Neural Networks and Learning Machines*. Third Edition. New Jersey: Pearson, 2009.

6. Liu, Zeyu, Yantao Yang, and Qingdong Cai. “Neural network as a function approximator and its application in solving differential equations.” *Applied Mathematics and Mechanics* 40.2 (2019): 237–248. DOI: 10.1007/s10483-019-2429-8.

7. Deryabin, V. V. “Review of researches of neural network technologies implementation in navigation.” *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S. O. Makarova* 6(34) (2015): 29–43. DOI: 10.21821/2309-5180-2015-7-6-29-43.

8. Podporin, S.A. “Ispolzovanie neyro-nechetkih kontrollerov v sistemah upravleniya dvizheniem morskikh sudov.” *Sbornik naukovykh prac Harkivskogo universitety Povitryannyh Sil* 4(33) (2012): 181–187.

9. Kula, Krzysztof S. “Model-based controller for ship track-keeping using Neural Network.” *2015 IEEE 2nd International Conference on Cybernetics (CYBCONF)*. IEEE, 2015. 178–183. DOI: 10.1109/CYBConf.2015.7175928.

10. Hornik, Kurt. “Some new results on neural network approximation.” *Neural Networks* 6.8 (1993): 1069–1072. DOI: 10.1016/S0893-6080(09)80018-X.

ИНФОРМАЦИЯ ОБ АВТОРЕ

Дерябин Виктор Владимирович —
 кандидат технических наук, доцент
 ФГБОУ ВО «ГУМРФ имени адмирала
 С. О. Макарова»
 198035, Российская Федерация, Санкт-Петербург,
 ул. Двинская, 5/7
 e-mail: gmavitder@mail.ru, kaf_nav@gumrf.ru

INFORMATION ABOUT THE AUTHOR

Deryabin, Victor V. —
 PhD, associate professor.
 Admiral Makarov State University of Maritime
 and Inland Shipping
 5/7 Dvinskaya Str., St. Petersburg 198035,
 Russian Federation
 e-mail: gmavitder@mail.ru, kaf_nav@gumrf.ru

Статья поступила в редакцию 8 декабря 2019 г.
 Received: December 8, 2019.