

УДК 004.42

## РАЗРАБОТКА ПРОГРАММНОГО СИМУЛЯТОРА ДВИЖЕНИЯ И РАСХОЖДЕНИЯ БЕЗЭКИПАЖНОГО СУДНА

**А.А. Дыда, И.И. Пушкарёв, А.Ф. Красавина\***

Морской государственный университет имени адмирала Г.И. Невельского  
Россия, 690003, Владивосток, ул. Верхнепортовая, 50а

E-mail: adyda@mail.ru, b\_r\_i\_g88@mail.ru, ankrasmal.ru

**Аннотация.** Рассмотрен вопрос о разработке программного симулятора движения и расхождения безэкипажного судна, применение которого позволяет упростить процесс создания и тестирования безэкипажных судов. Приведены основные требования, предъявляемые к симулятору, его назначению, структуре и возможностям. Описана алгоритмическая основа разрабатываемого симулятора: подсистема удержания траектории использует алгоритм градиента вспомогательной функции, подсистема обхода препятствий реализована на методе скоростных препятствий, доработанного с учетом МППСС-72. Приведена общая логическая схема программы, которая содержит базовые подсистемы, элементы, обеспечивающие гибкость настройки симулятора и элементы для анализа результатов его работы. Рассмотрены наиболее распространенные в области робототехники языки программирования, выбран подходящий для разработки программного симулятора язык программирования C++. Показан графический интерфейс разработанной программы и основные его элементы. Для проверки работоспособности программного симулятора было проведено моделирование основных ситуаций расхождения: обгон, расхождение на встречных курсах и расхождение при пересечении курсов (препятствие справа или слева от БЭС). Представлены результаты данного моделирования. Созданный программный симулятор движения и расхождения безэкипажного судна соответствует поставленным к нему в работе требованиям, позволяет моделировать движение БЭС и при угрозе столкновения производить расхождение с препятствием по правилам МППСС-72. Реализация подсистем движения и расхождения как отдельных программных элементов дает возможность использовать их без значительных доработок в бортовых системах управления движением безэкипажных судов.

**Ключевые слова:** безэкипажное судно, расхождение, МППСС-72, программный симулятор, моделирование, язык программирования, система управления.



© Автор(ы), 2023

\* Александр Александрович Дыда, доктор технических наук, профессор кафедры «Автоматические и информационные системы».

Игорь Игоревич Пушкарёв, аспирант кафедры «Автоматические и информационные системы».

Анна Фёдоровна Красавина, аспирант кафедры «Автоматические и информационные системы».

## **Введение**

В современном мире безэкипажные суда (БЭС) становятся все более популярными, что обусловлено возможностью их применения в широком спектре задач в различных сферах. Они используются для мониторинга состояния окружающей среды, выполнения научных исследований, обеспечивают проведение подводных работ, выполняют спасательные и военные функции [1]. Перспективным направлением являются морские безэкипажные грузоперевозки [2, 3].

Однако разработка безэкипажных судов является сложной и дорогостоящей задачей. Она включает в себя проектирование судна, его изготовление, установку и настройку электроники и программного обеспечения, а также тестирование на воде. Все это требует больших затрат времени и ресурсов.

Для упрощения процесса разработки и тестирования безэкипажных судов программные симуляторы могут быть использованы для создания виртуальных моделей судов и их движения. Эти симуляторы могут быть настроены для воспроизведения различных маршрутов движения, погодных условий, морских течений, волнения и других факторов, которые могут повлиять на движение судна [4, 5].

С помощью программного симулятора разработчики могут тестировать различные сценарии без риска потери судна или нанесения ущерба окружающей среде. Они могут настроить различные параметры судна и его движения, чтобы определить оптимальные настройки для конкретных задач. Также использование программных симуляторов позволяет сократить затраты на разработку и тестирование безэкипажных судов, тем самым снизив их стоимость и время вывода в эксплуатацию.

Задача автоматического расхождения безэкипажного судна с подвижными или неподвижными препятствиями в соответствии с «Международными правилами предупреждения столкновения судов в море (МППСС-72)» является одной из важнейших [6, 7]. Разработка программного симулятора движения и расхождения позволяет оценить работоспособность и эффективность применяемых алгоритмов и является важным шагом в разработке безопасных и эффективных безэкипажных судов, которые могут успешно выполнять задачи в различных областях.

## **Основные требования**

Разрабатываемый симулятор предназначен в первую очередь для проверки алгоритма движения по заданной траектории и расхождения с представляющими опасность препятствиями в соответствии с МППСС-72. К моделированию динамических характеристик БЭС и подвижных объектов не предъявляется особых требований, и оно допускается в упрощенном виде.

Программный симулятор движения и расхождения безэкипажного судна является сложным инструментом, который должен соответствовать определенным требованиям для обеспечения возможности тестирования различных сценариев движения.

Важнейшим требованием к симулятору является разработка такой структуры программы, базовые элементы которой без изменений возможно использовать в программном обеспечении бортовых систем управления безэкипажных судов. К таким элементам можно отнести программные модули и функции, реализующие алгоритмическую часть движения и принятия решений при маневрах расхождения. Язык программирования также должен быть использован тот, кото-

рый далее планируется использовать в бортовой системе управления БЭС, или наиболее подходящий и распространенный в данном применении, в том числе позволяющий обрабатывать значительный объем данных и вычислений при ограниченных аппаратных ресурсах.

Симулятор должен обеспечивать гибкость в настройке параметров, таких как размеры и характеристики движения судов, маршруты движения и условия на море, чтобы пользователи могли создавать разнообразные ситуации для тестирования.

Возможности анализа результатов моделирования также являются базовым требованием. К таким возможностям можно отнести отображение траекторий движения и способ отображения или визуализации решений при выполнении маневров расхождения. Это поможет разработчикам и экспертам оценить работоспособность применяемых алгоритмов, доработать и улучшить их производительность.

### Реализация

В качестве алгоритмической основы разрабатываемого симулятора используются способы движения и расхождения, предложенные в статье «Система управления движением и расхождением безэкипажного судна в соответствии с МППСС-72» [8].

В симуляторе реализована подсистема удержания траектории, основанная на алгоритме градиента вспомогательной функции [9]. Суть этого алгоритма заключается в формировании векторного поля – набора векторов, показывающих требуемое направления движения БЭС в конкретной точке. Ниже приведены обобщенные формулы для расчета требуемого курса БЭС при движении по заданному участку маршрута:

$$\begin{aligned}\psi_{set} &= \arctg 2 \left( \frac{dF_i}{dy_i} + \alpha w_{yi}, \frac{dF_i}{dx_i} + \alpha w_{xi} \right), \\ \text{grad} F_i &= (-a(ax_{usv} + by_{usv} + c), -b(ax_{usv} + by_{usv} + c)), \\ a &= (y_i - y_{i+1})\mu, \quad b = (x_i - x_{i+1})\mu, \quad c = (x_i y_{i+1} - x_{i+1} y_i)\mu, \\ \mu &= \pm \frac{1}{\sqrt{(y_i - y_{i+1})^2 + (x_{i+1} - x_i)^2}}, \\ w_{xi} &= \frac{x_{i+1} - x_i}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}, \\ w_{yi} &= \frac{y_{i+1} - y_i}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}},\end{aligned}$$

где  $\psi_{set}$  – требуемый курс;  $F_i$  – функция, достигающая экстремума на  $i$ -ом участке маршрута [9];  $(w_{xi}, w_{yi})$  – вспомогательный вектор единичной длины, направленный вдоль траектории движения БЭС;  $\alpha$  – весовой коэффициент, определяющий степень схождения векторного поля к заданному участку маршрута ( $\alpha > 0$ );  $a, b, c$  – некоторые действительные числа;  $(x_{usv}, y_{usv})$  – текущие

координаты БЭС;  $(x_i, y_i)$  – путевые точки маршрута;  $\mu$  – нормирующий множитель (берется по знаку, противоположному знаку  $c$ ).

Необходимый маршрут движения БЭС задается путем создания набора путевых точек, последовательно соединенных прямыми линиями. Таким образом создается кусочно-линейный маршрут движения БЭС.

Подсистема обхода препятствий, используемая в симуляторе, обеспечивает вычисление безопасного курса и скорости БЭС для предотвращения столкновения с угрожающими препятствиями и соблюдения при этом правил МППСС-72. В ней применяется метод скоростных препятствий [10], доработанный с учетом требований МППСС-72 [8]. Для каждого возможного варианта курса и скорости рассчитывается стоимостная функция:

$$C = k_1 U_{i,j}^T U_{i,j} + k_2 F_{collision}(V_i, \psi_j) + k_3 F_{COLREG}(V_i, \psi_j),$$

$$U_{i,j} = \begin{bmatrix} V_{set} \cos(\psi_{set}) - V_i \cos(\psi_j) \\ V_{set} \sin(\psi_{set}) - V_i \sin(\psi_j) \end{bmatrix},$$

где  $k_1, k_2, k_3$  – постоянные коэффициенты;  $F_{collision}$  – логическая функция опасной ситуации [8];  $F_{COLREG}$  – логическая функция МППСС-72 [8];  $V_i, \psi_j$  – текущая скорость и курс БЭС;  $U_{i,j}$  – вектор ошибки;  $V_{set}, \psi_{set}$  – заданная пользователем скорость и рассчитанный подсистемой удержания траектории курс.

Далее выбирается вариант курса и скорости с минимальным значением стоимостной функции, что будет соответствовать курсу и скорости БЭС, максимально близким к курсу подсистемы удержания траектории и заданной скорости и при этом позволяющим разойтись с препятствием, соблюдая правила МППСС-72.

Общая логическая схема программного симулятора показана на рис. 1.

Пользователь программного симулятора в полях настроек вводит параметры для расчета симуляции, включая коэффициенты, необходимые для работы подсистемы удержания траектории и подсистемы обхода препятствий, радиусы, описывающие БЭС и препятствия, требуемую скорость движения БЭС и заданные курсы и скорости движения препятствий. Маршрут движения БЭС задается интерактивным способом: пользователь активирует на панели инструментов режим создания путевых точек и отмечает их на виртуальной карте, точкам автоматически присваиваются порядковые номера, и они последовательно соединяются прямыми линиями, тем самым формируя маршрут движения.

Начальные положения БЭС и препятствий также задаются пользователем на карте интерактивно. Динамика безэкипажного судна (динамика изменения курса и скорости) в симуляторе реализована в упрощенной форме. Переход судна с одной скорости на другую определяется ускорением, задаваемым пользователем. Также задается угловая скорость для реализации изменения курса БЭС. При нажатии на кнопку начала симуляции производится выполнение алгоритмов подсистемы удержания траектории и подсистемы обхода препятствий, расчет и отрисовка перемещения безэкипажного судна и препятствий на карте.

Для анализа работы алгоритмов движения и расхождения в симуляторе предусмотрена функция отображения пройденной траектории БЭС и подвижных препятствий в течение заданного интервала времени. Также в отдельном окне

возможен вывод сетки вариантов курсов и скоростей с весовыми коэффициентами в графическом виде. Это позволяет оценить корректность работы алгоритма расхождения с угрожаемыми подвижными препятствиями в соответствии с МППСС-72 в реальном времени. Дополнительно возможен вывод рассчитываемых в реальном времени параметров, интересующих пользователя. Например, вывод дистанции и времени кратчайшего сближения БЭС с подвижным препятствием [11].

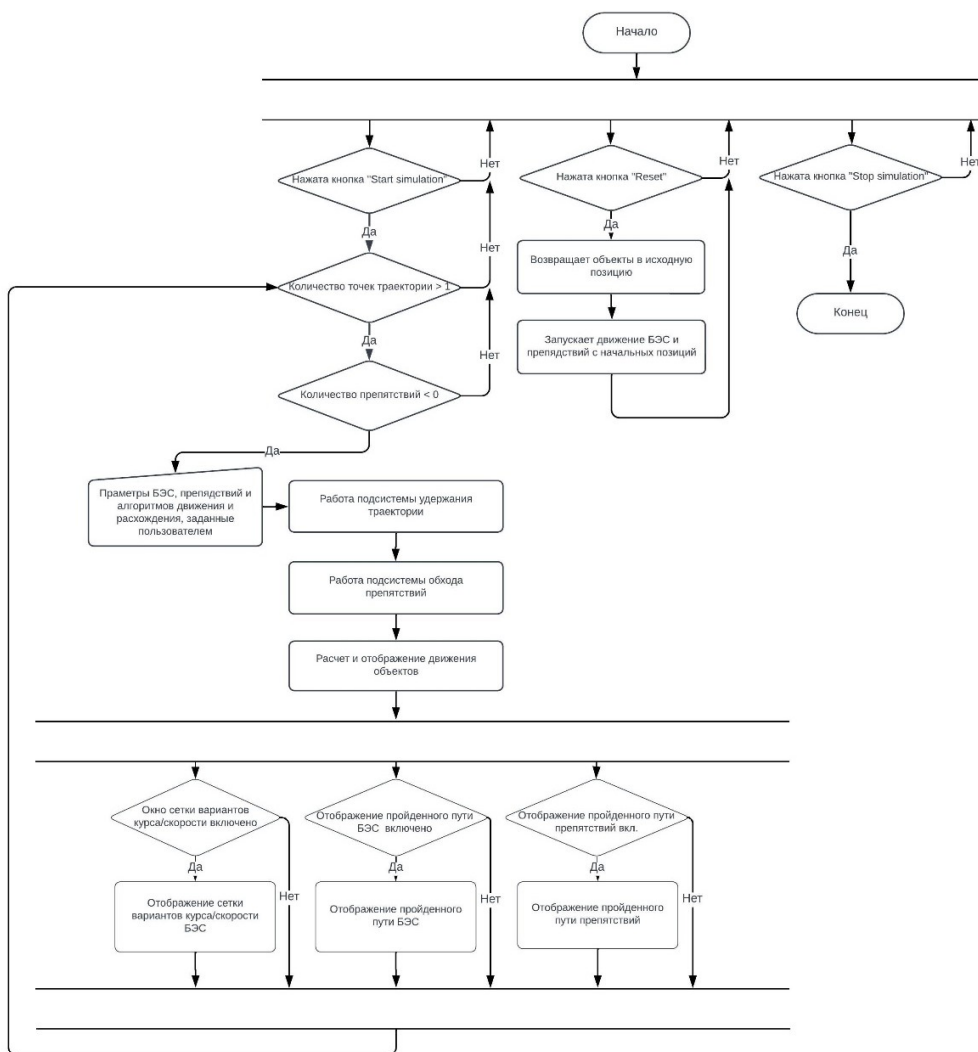


Рис. 1. Логическая схема программного симулятора движения и расхождения безэкипажного судна

Наиболее распространенными языками программирования в робототехнике, к которой в том числе можно отнести безэкипажные суда, являются MATLAB, Python, C++ [12, 13].

MATLAB [14] – это язык программирования, который часто используется в робототехнике для моделирования и анализа динамических систем. Он имеет богатую библиотеку инструментов для решения различных задач. MATLAB

больше ориентирован на математические расчеты и имеет ограниченные возможности для создания сложных систем управления. Также он может быть очень ресурсоемким при работе с большими объемами данных и вычислений, что может привести к проблемам с памятью и производительностью.

Python [15] широко используется в робототехнике для создания высокоуровневых систем управления, обработки изображений, анализа данных и машинного обучения. Python отличается простотой и легкостью в изучении и использовании. Основным его недостатком является невысокая скорость работы. Он является интерпретируемым языком, что означает, что его скорость выполнения ниже, чем у компилируемых языков. Это может быть проблемой при выполнении задач, требующих высокой производительности.

C++ [16] – один из наиболее популярных языков в робототехнике. C++ позволяет написать очень эффективный и быстрый код, что может быть важно для систем управления роботами с высокими требованиями к скорости и производительности. C++ является стандартным языком программирования для различных аппаратных платформ, что обеспечивает широкую совместимость и переносимость кода между различными системами. Недостатком C++ является его сложность. У него более сложный синтаксис и более сложные концепции, такие как управление памятью и указатели, что может затруднить разработку программ.

Исходным языком программирования для программного симулятора движения и расхождения безэкипажного судна выбран C++ как наиболее подходящий под требования к разработке. Для разработки графического интерфейса программы (рис. 2) используется Qt Framework [17] – кроссплатформенная библиотека разработки GUI на C++.

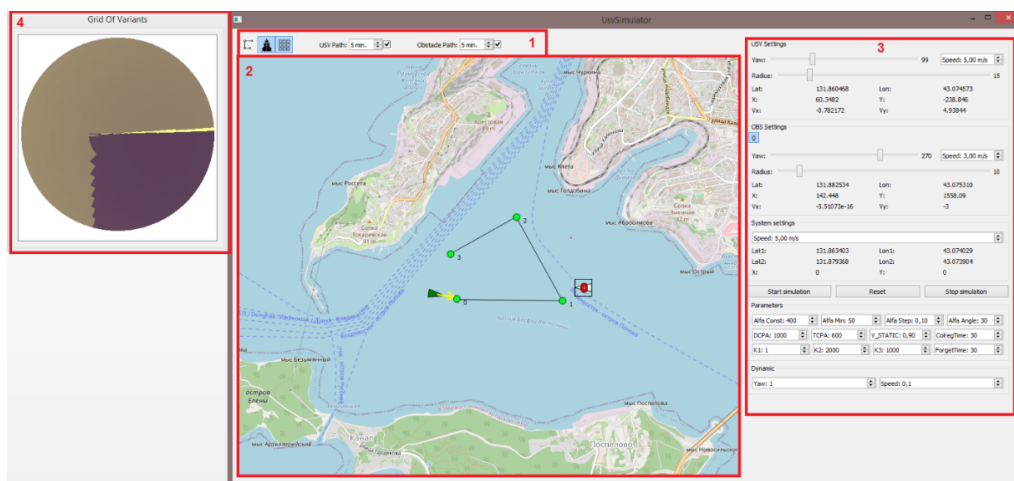


Рис. 2. Графический интерфейс программного симулятора (1 – панель инструментов; 2 – виртуальная карта с расположенными на ней БЭС, препятствием и заданным маршрутом движения БЭС; 3 – панель ввода параметров БЭС, препятствий и алгоритмов движения и расхождения; 4 – графическое отображение стоимостной функции сетки вариантов курсов и скоростей БЭС)

На рис. 3 более подробно продемонстрирована сетка вариантов, показывающая, какой курс и какую скорость выбирает БЭС в текущий момент времени.

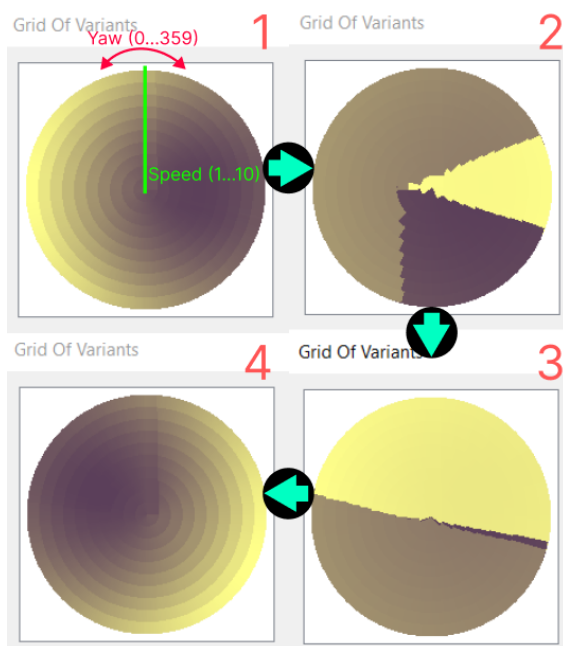


Рис. 3. Сетка вариантов курсов и скоростей безэкипажного судна

Представленная сетка вариантов показывает, какие области движения являются безопасными, а какие приводят к угрозе столкновения. Она представляет собой окружность, где градусная мера – курс движения судна, а радиус – скорость движения судна, как показано в области 1. Фиолетовые оттенки показывают области, которые являются приоритетными для БЭК, а желтые оттенки показывают области, которые приводят к угрозе столкновения, к нарушению правил МППСС-72 или к отклонению от траектории движения. Цвета определяются значением стоимостной функции, где наименьшее значение показывает наиболее подходящие курс и скорость движения безэкипажного судна.

Подсистемы удержания траектории и обхода препятствий реализованы в виде отдельных классов и функций, что позволяет использовать эти элементы программного обеспечения, отлаженного и оптимизированного в симуляторе, в дальнейшем в бортовом программном обеспечении систем управления безэкипажных судов.

### Тестирование

Проверка работоспособности программного симулятора проводилась при моделировании основных ситуаций расхождения, регламентированных МППСС-72. К ним относятся: обгон, расхождение на встречных курсах и пересечение курсов [18]. Использовались следующие коэффициенты стоимостной функции:  $k_1 = 1$ ,  $k_2 = 2000$ ,  $k_3 = 1000$ .

На рис. 4 показана смоделированная ситуация расхождения БЭС с препятствием на встречных курсах. БЭС обходит подвижное препятствие слева, соблюдая МППСС-72 и учитывая радиусы, описывающие БЭС и препятствие.

Моделирование ситуации, при которой БЭС совершает обгон препятствия, показано на рис. 5. Здесь БЭС совершает обгон, оставляя препятствие справа.

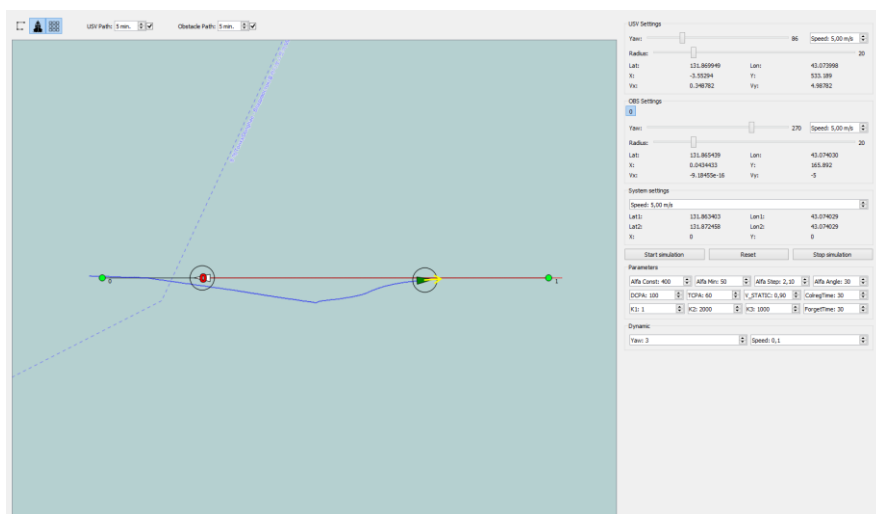


Рис. 4. Движение и расхождение БЭС с препятствием на встречных курсах (траектория движения БЭС – синяя, траектория движения препятствия – красная, путевые точки заданного маршрута БЭС – зеленые, длина заданного маршрута = 550 м)

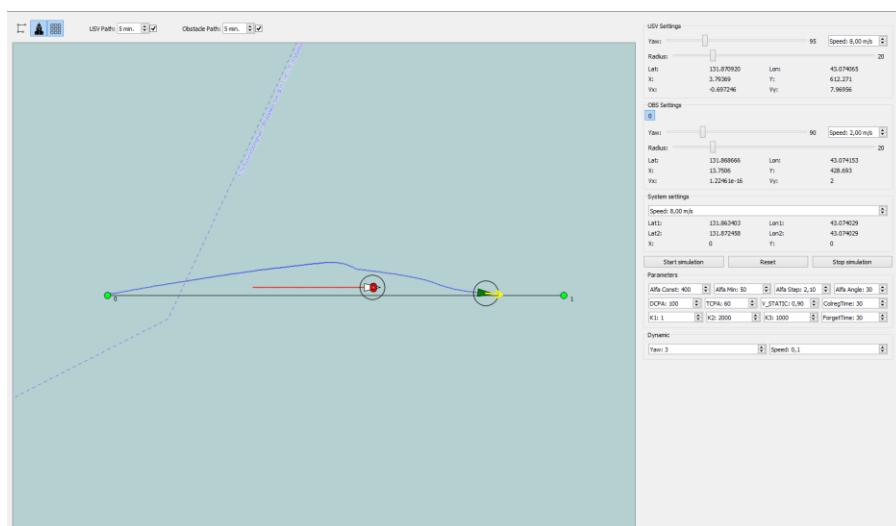


Рис. 5. Движение и расхождение БЭС с препятствием при обгоне (траектория движения БЭС – синяя, траектория движения препятствия – красная, путевые точки заданного маршрута БЭС – зеленые, длина заданного маршрута = 550 м)

Расхождение БЭС в ситуации пересечения курсов, когда препятствие находится справа, показано на рис. 6. При движении препятствия справа БЭС сбавляет скорость и меняет курс, уступая дорогу и обходя препятствие через корму, соблюдая МППСС-72 и учитывая радиусы, описывающие БЭС и препятствие.



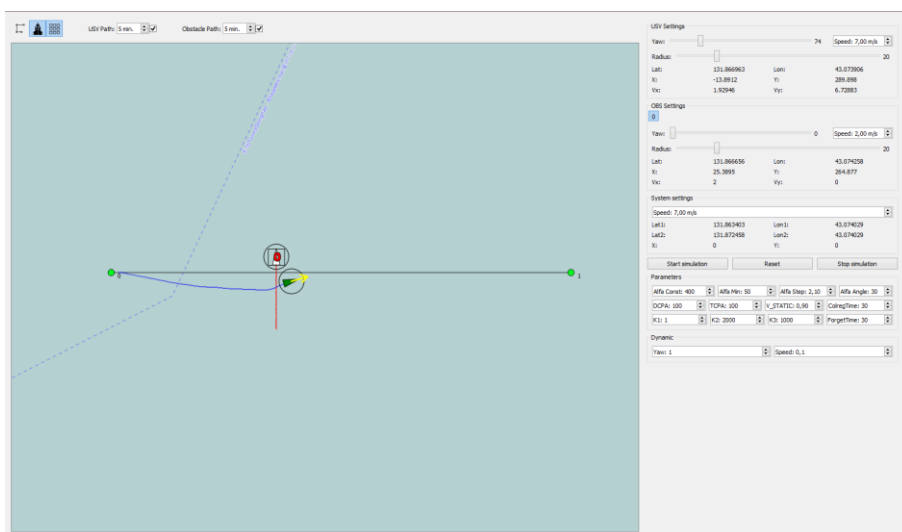


Рис. 6. Движение и расхождение БЭС с препятствием при пересечении курсов, препятствие справа (траектория движения БЭС – синяя, траектория движения препятствия – красная, путевые точки заданного маршрута БЭС – зеленые, длина заданного маршрута = 550 м)

На рис. 7 показано моделирование ситуации пересечения курсов, когда препятствие находится слева от БЭС. В данном случае согласно правилам уступить дорогу должно препятствие, что не реализовано в его программе движения. Безэкипажное судно пытается пройти, не уступая препятствию, но при этом незначительно меняет курс для уклонения от столкновения.

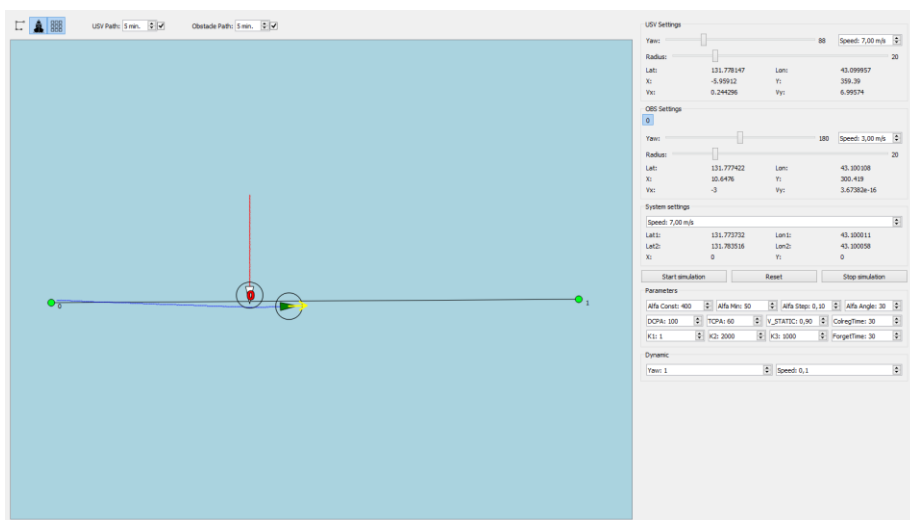


Рис. 7. Движение и расхождение БЭС с препятствием при пересечении курсов, препятствие слева (траектория движения БЭС – синяя, траектория движения препятствия – красная, путевые точки заданного маршрута БЭС – зеленые, длина заданного маршрута = 550 м)

## Заключение

Разработанный программный симулятор движения и расхождения БЭС позволяет оценить корректность работы алгоритмов движения с учетом правил МППСС-72. Функционал симулятора дает возможность смоделировать различные сценарии движения, включая заданный маршрут движения и требуемую скорость БЭС, курс и скорость движения препятствия, начальные позиции участников движения и их размеры. Возможно введение пользователем произвольных коэффициентов для расчета используемых алгоритмов движения и расхождения.

Отображение траектории движения БЭС и препятствия, а также графическая визуализация стоимостной функции сетки вариантов курсов и скоростей позволяют проанализировать работу применяемых алгоритмов.

Описание подсистем движения и расхождения в виде отдельных программных элементов на языке C++ после их окончательного тестирования и отладки позволяет применить их в бортовых системах управления безэкипажных судов практически без изменений.

В дальнейшем планируется усовершенствование разработанного программного симулятора путем введения возможностей формирования более разнообразных сценариев движения и расхождения с использованием группы подвижных препятствий со сложными траекториями. Возможно моделирование воздействий условий внешней среды – ветра и течений, проведение исследований быстродействия используемых алгоритмов и изучение возможности их оптимизации.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Liu Z. et al.* Unmanned surface vehicles: An overview of developments and challenges // *Annual Reviews in Control*. 2016. V. 41. Pp. 71–93.
2. *Gu Y., Wallace S.W.* Operational benefits of autonomous vessels in logistics – A case of autonomous water-taxi in Bergen // *Transportation Research Part E: Logistics and Transportation Review*. 2021. V. 154. P. 102456.
3. *Kretschmann L., Burmeister H.C., Jahn C.* Analyzing the economic benefit of unmanned autonomous ships: An exploratory cost-comparison between an autonomous and a conventional bulk carrier // *Research in transportation business & management*. 2017. V. 25. Pp. 76–86.
4. *Paravisi M. et al.* Unmanned surface vehicle simulator with realistic environmental disturbances // *Sensors*. 2019. V. 19. № 5. P. 1068.
5. *Lee H.W. et al.* Integrated Simulation of Virtual Prototypes and Control Algorithms of Unmanned Surface Vehicles Based on a Robot Operating System // *Journal of Marine Science and Technology*. 2021. V. 29. № 4. P. 2.
6. *Ma Y. et al.* Collision-avoidance under COLREGS for unmanned surface vehicles via deep reinforcement learning // *Maritime Policy & Management*. 2020. V. 47. № 5. Pp. 665–686.
7. *Deng F. et al.* COLREGs: Compliant Dynamic Obstacle Avoidance of USVs Based on the Dynamic Navigation Ship Domain // *Journal of Marine Science and Engineering*. 2021. V. 9. № 8. P. 837.
8. *Пушкарёв И.И.* Система управления движением и расхождением безэкипажного судна в соответствии с МППСС-72 // *Вестник Государственного университета морского и речного флота имени адмирала С.О. Макарова*. 2022. Т. 14. № 6. С. 837–848.
9. *Дыда А.А., Чумакова К.Н., Пушкарёв И.И.* Подход к управлению судном по траектории на основе градиента вспомогательной функции // *Научные проблемы водного транспорта*. 2020. № 65. С. 27–36.
10. *Fiorini P., Shiller Z.* Motion Planning in Dynamic Environments Using Velocity Obstacles // *International Journal of Robotics Research*. 1998. Vol. 17. Is. 7. Pp. 760–772. DOI: 10.1177/027836499801700706.

11. Бужбецкий Р.Ю. Зависимость дистанции кратчайшего сближения от курса маневрирующего судна // Эксплуатация морского транспорта. 2017. № 3. С. 69–72.
12. Joseph L., Johny A. Fundamentals of C for Robotics Programming // Robot Operating System (ROS) for Absolute Beginners: Robotics Programming Made Easy. Berkeley, CA: Apress, 2022. Pp. 53–91.
13. Егоров Д.В., Бобков В.Д. Анализ языков программирования роботов // OPEN INNOVATION. 2018. С. 45–47.
14. Moler C., Little J. A history of MATLAB // Proceedings of the ACM on Programming Languages. 2020. Т. 4. №. HOPL. P. 1–67.
15. Python W. Python // Python Releases Wind. 2021. Vol. 24.
16. Josuttis N. M. The C++ standard library: a tutorial and reference. 2012.
17. Lazar G., Penea R. Mastering Qt 5: Create stunning cross-platform applications using C++ with Qt Widgets and QML with Qt Quick. Packt Publishing Ltd, 2018.
18. Шарлай Г.Н. МППСС-72 с комментариями: учеб. пособие. М.: Моркнига, 2017. 136 с.

*Статья поступила в редакцию 27 марта 2023 г.*

# DEVELOPMENT OF SOFTWARE SIMULATOR FOR THE MOTION AND COLREG OBSTACLE AVOIDANCE OF UNMANNED SHIP

**A.A. Duda, I.I. Pushkarev, A.F. Krasavina \***

Maritime State University named after admiral G.I. Nevelskoy  
50a, Verhneportovaya st., Vladivostok, 690003, Russian Federation

E-mail: adyda@mail.ru, b\_r\_i\_g88@mail.ru, ankras2004@mail.ru

**Abstract.** *The article discusses the development of a software simulator for the motion and COLREG obstacle avoidance of unmanned ship, the use of which simplifies the process of creating and testing unmanned vessels. The main requirements for the simulator, its purpose, structure, and capabilities are presented. The algorithmic basis of the developed simulator is described: the trajectory tracking subsystem uses the gradient algorithm of the auxiliary function, and the obstacle avoidance subsystem is implemented using the velocity obstacle method adapted for the COLREG. The general logic diagram of the program is presented, containing basic subsystems, elements that provide flexibility in configuring the simulator, and elements for analyzing the results of its work. The most common programming languages in the field of robotics are considered, and the suitable language for developing the software simulator – the C++ programming language – is chosen. The graphical interface of the developed program and its main elements are presented. To test the functionality of the software simulator, modeling of the main obstacle avoidance situations was carried out: head-on, crossing (obstacle moves to the left or right), and overtaking. The results of this modeling are presented. The created software simulator for the motion and obstacle avoidance of unmanned ship meets the requirements set for it in the work, allows modeling of the motion of unmanned ship and to perform avoidance according to COLREG. The implementation of trajectory tracking and obstacle avoidance subsystems as separate software elements makes it possible to use them without significant modifications in the onboard control systems of unmanned ships.*

**Keywords:** *unmanned ship, obstacle avoidance, COLREG, software simulator, modeling, programming language, control system.*

## REFERENCES

1. Liu Z. et al. Unmanned surface vehicles: An overview of developments and challenges // Annual Reviews in Control. 2016. V. 41. Pp. 71–93.
2. Gu Y., Wallace S.W. Operational benefits of autonomous vessels in logistics – A case of autonomous water-taxi in Bergen // Transportation Research Part E: Logistics and Transportation Review. 2021. V. 154. P. 102456.
3. Kretschmann L., Burmeister H.C., Jahn C. Analyzing the economic benefit of unmanned autonomous ships: An exploratory cost-comparison between an autonomous and a conventional bulk carrier // Research in transportation business & management. 2017. V. 25. Pp. 76–86.
4. Paravisi M. et al. Unmanned surface vehicle simulator with realistic environmental disturbances // Sensors. 2019. V. 19. № 5. P. 1068.



© The Author(s), 2023

\* Alexander A. Duda (Dr. Sci. (Techn.)), Professor.

Igor I. Pushkarev, Postgraduate Student.

Anna F. Krasavina, Postgraduate Student.

5. *Lee H.W. et al.* Integrated Simulation of Virtual Prototypes and Control Algorithms of Unmanned Surface Vehicles Based on a Robot Operating System // *Journal of Marine Science and Technology*. 2021. V. 29. № 4. P. 2.
6. *Ma Y. et al.* Collision-avoidance under COLREGS for unmanned surface vehicles via deep reinforcement learning // *Maritime Policy & Management*. 2020. V. 47. № 5. Pp. 665–686.
7. *Deng F. et al.* COLREGs: Compliant Dynamic Obstacle Avoidance of USVs Based on the Dynamic Navigation Ship Domain // *Journal of Marine Science and Engineering*. 2021. V. 9. № 8. P. 837.
8. *Pushkaryov I.I.* Sistema upravleniya dvizheniem i raskhozhdeniem bezekipazhnogo sudna v sootvetstvii s MPPSS-72 [A control system for movement and divergence of unmanned ship according to colreg] // *Vestnik Gosudarstvennogo universiteta morskogo i rech-nogo flota imeni admirala SO Makarova*. 2022. V. 14. № 6. Pp. 837–848. (In Russian)
9. *Dyda A.A., Chumakova K.N., Pushkarev I.I.* Podhod k upravleniyu sudnom po traekto-rii na osnove gradienta vspomogatel'noj funktsii [Auxiliary function gradient approach to marine vehicle path-following control] // *Nauchnye problemy vodnogo transporta*. 2020. № 65. Pp. 27–36. (In Russian)
10. *Fiorini P., Shiller Z.* Motion Planning in Dynamic Environments Using Velocity Obstacles // *International Journal of Robotics Research*. 1998. Vol. 17. Is. 7. Pp. 760–772. DOI: 10.1177/027836499801700706.
11. *Buzhbeckij R.Yu.* Zavisimost' distantsii krachajshhego sblizheniya ot kursa manevri-ruyushchego sudna [Dependence of the closest approach distance on the course of the maneu-ving vessel] // *Ekspluatatsiya morskogo transporta*. 2017. № 3. Pp. 69–72. (In Russian)
12. *Joseph L., Johnny A.* Fundamentals of C for Robotics Programming // *Robot Operating System (ROS) for Absolute Beginners: Robotics Programming Made Easy*. Berkeley, CA: Apress, 2022. Pp. 53–91.
13. *Egorov D.V., Bobkov V.D.* Analiz yazykov programmirovaniya robotov [Analysis of robot programming languages] // *OPEN INNOVATION*. 2018. Pp. 45–47.
14. *Moler C., Little J.* A history of MATLAB // *Proceedings of the ACM on Programming Lan- guages*. 2020. T. 4. № HOPL. Pp. 1–67.
15. *Python W.* Python // *Python Releases Wind*. 2021. Vol. 24.
16. *Josuttis N. M.* The C++ standard library: a tutorial and reference. 2012.
17. *Lazar G., Penea R.* Mastering Qt 5: Create stunning cross-platform applications using C++ with Qt Widgets and QML with Qt Quick. Packt Publishing Ltd, 2018.
18. *SHarlaj G.N.* MPPSS-72 s kommentariyami: uchebnoe posobie [COLREGs-72 with com- ments: textbook]. M.: Morkniga, 2017. 136 pp.

*Original article submitted 27.03.2023*