

Планирование траектории автономного мини-корабля

Р.В. Федоренко, Б.В. Гуренко

Южный федеральный университет, Таганрог

Аннотация: В работе рассматривается процесс планирования траектории движения мини-корабля в среде с препятствиями. Приведена структура аппаратного обеспечения системы управления, архитектура программного обеспечения, построенного на базе клиент-серверной модели взаимодействия со слабой связанностью компонентов, в программной среде ROS. Глобальный планировщик разработан с использованием метода диаграмм Вороного. Локальный планировщик (регулятор) предложено реализовать с использованием неустойчивых режимов управления для обхода препятствий вблизи мини-корабля. Описаны результаты моделирования, показавшие работоспособность предложенных подходов.

Ключевые слова: мини-корабль, управление, планирование, диаграмма Вороного, ROS

Введение

Автономные мини-корабли могут быть полезны для выполнения широкого круга задач, включая мониторинг окружающей среды, работу на загрязненных участках, поисковые и спасательные операции.

Схема использования автономного мини-корабля заключается в следующем. Сначала оператор пункта управления визуально на карте строит миссию. Система управления осуществляет картографирование, обнаружение препятствий и планирование траектории для движения между контрольными точками миссии в обход препятствий.

Предложенная авторами в работе [1 – 5] система автоматического управления позволяет организовать автоматическое движение катера вдоль заданной траектории. В данной статье, являющейся продолжением работы авторов, рассматривается процесс планирования траектории движения катера в среде с препятствиями.

В работе приведена структура аппаратного обеспечения системы управления, архитектура программного обеспечения, построенного на базе клиент-серверной модели взаимодействия со слабой связанностью

компонентов, описаны результаты моделирования, показавшие работоспособность предложенных подходов.

Для разработки программного обеспечения авторы используют среду ROS представляющую стандарт де-факто в разработке ПО для робототехники. Данная среда имеет в своем составе стек планирования движения, который лег в основу описанной ниже программной архитектуры системы управления мини-корабля[6]. Однако, для адаптации к объекту управления – мини-кораблю авторами в рамках данной архитектуры разработаны новые программные пакеты в части планирования движения.

Планирование движения является одной из фундаментальных проблем в робототехнике. Существует ряд принципиально разных подходов, применимых в различных задачах, таких как метод семплирования, вероятностный метод, методы поиска на графах, метод обобщенных диаграмм Вороного. Наиболее часто используемыми являются методы поиска на графах, такие как A*. Такой метод реализован и в стандартном глобальном планировщике стека планирования программной среды ROS, который будет рассмотрен подробнее ниже. Данный планировщик строит кратчайший путь в целевую точку в обход препятствий. Однако, в ряде задач, в том числе и при планировании движения мини-корабля длина пути при движении в среде с препятствиями может не являться главным критерием при выборе метода планирования. Важной задачей является построение траектории движения максимально удаленной от препятствий для уменьшения вероятности столкновения с ними. Поэтому в данной работе была выполнена разработка программного пакета планирования глобальной траектории на базе обобщенных диаграмм Вороного для среды ROS, реализация которого в данной среде ранее не существовало.

Аппаратная реализация системы автоматического управления автономного катера

Структура аппаратного обеспечения системы управления автономного мини-корабля показана на рис.1. При разработке использована парадигма разделения вычислительной части на высокоуровневую (компьютер) и низкоуровневую (микроконтроллерный блок).

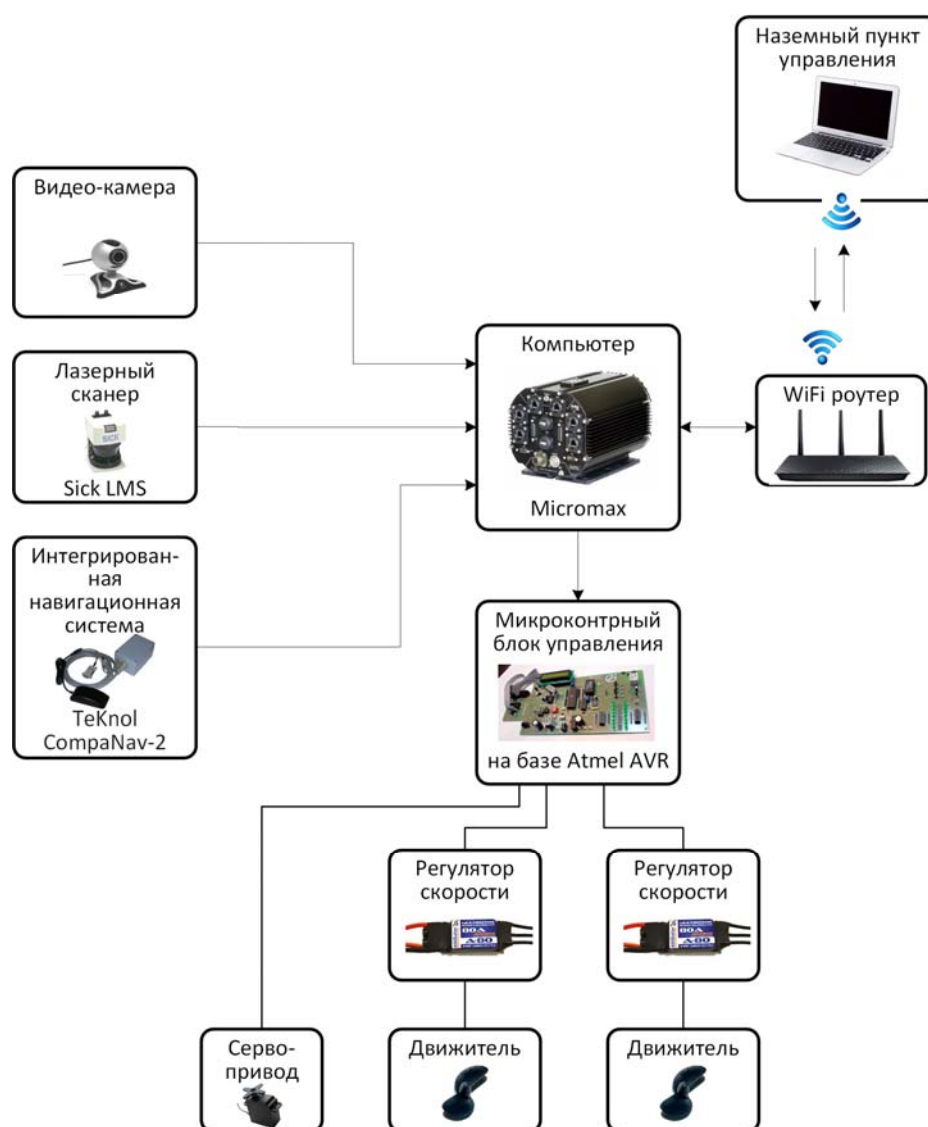


Рис.1– Аппаратная реализация системы управления мини-корабля

Микроконтроллерный блок получает данные от компьютера или пульта дистанционного управления и генерирует ШИМ-сигнал для управления двигателями и сервоприводами. Компьютер вычисляет требуемые

управляющие воздействия в соответствии с алгоритмом управления, выполняет программное обеспечение навигационной системы, и обменивается данными с наземным пунктом управления.

Архитектура программного обеспечения

В качестве операционной системы бортового компьютера используется Ubuntu Linux. Программное обеспечение построено на базе Robot Operating System, что позволяет реализовать слабую связанность компонентов и клиент-серверный шаблон их взаимодействия.

В качестве основы программной архитектуры системы планирования был использован стек ROSNavigation[3], который был сконфигурирован для мини-корабля и существенно переработан в части дополнения модулем планировщика глобальной траектории и локальным планировщиком (регулятором).

Структура программного обеспечения системы управления показана на рис.2.

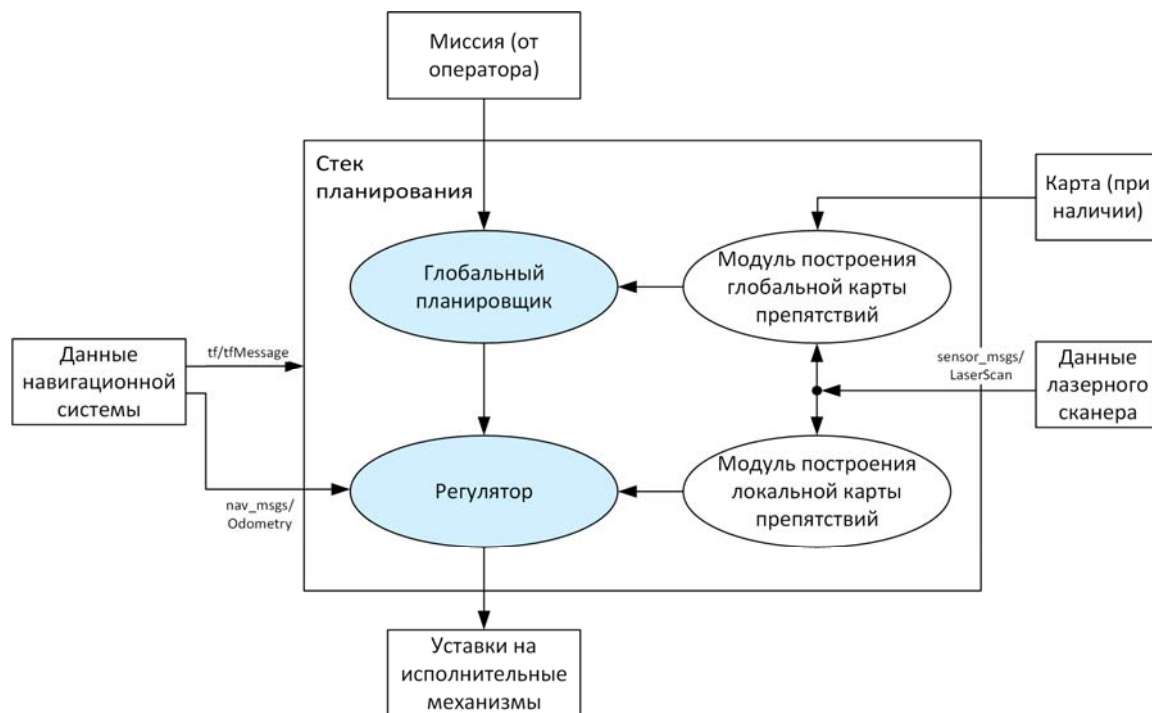


Рис.2–Структура программного обеспечения

В стеке используются две карты препятствий, которые строятся на основе данных лазерного сканера и навигационной системы, модулем `costmap_2di` отличаются назначением и, соответственно, размером и разрешением. Глобальная карта имеет большой размер и предназначена для планирования глобальной траектории движения, локальная карта препятствий имеет меньший размер, большее разрешение и предназначена для обхода близко расположенных препятствий. Глобальная карта используется глобальным планировщиком, локальная – локальным (или регулятором). Глобальный планировщик выполняет поиск плана движения на глобальной карте и передает этот план (в виде последовательности точек) локальному планировщику (регулятору), который рассчитывает управляющие воздействия на исполнительные механизмы для выполнения выработанного плана и одновременно выполняет обход близко расположенных препятствий, даже если это не было предусмотрено глобальным планом (подвижные или неизвестные препятствия).

Описание глобального планировщика приведено ниже. В качестве теоретической базы для реализации регулятора с функцией локального планировщика на базе неустойчивых режимов управления была использована работа [7].

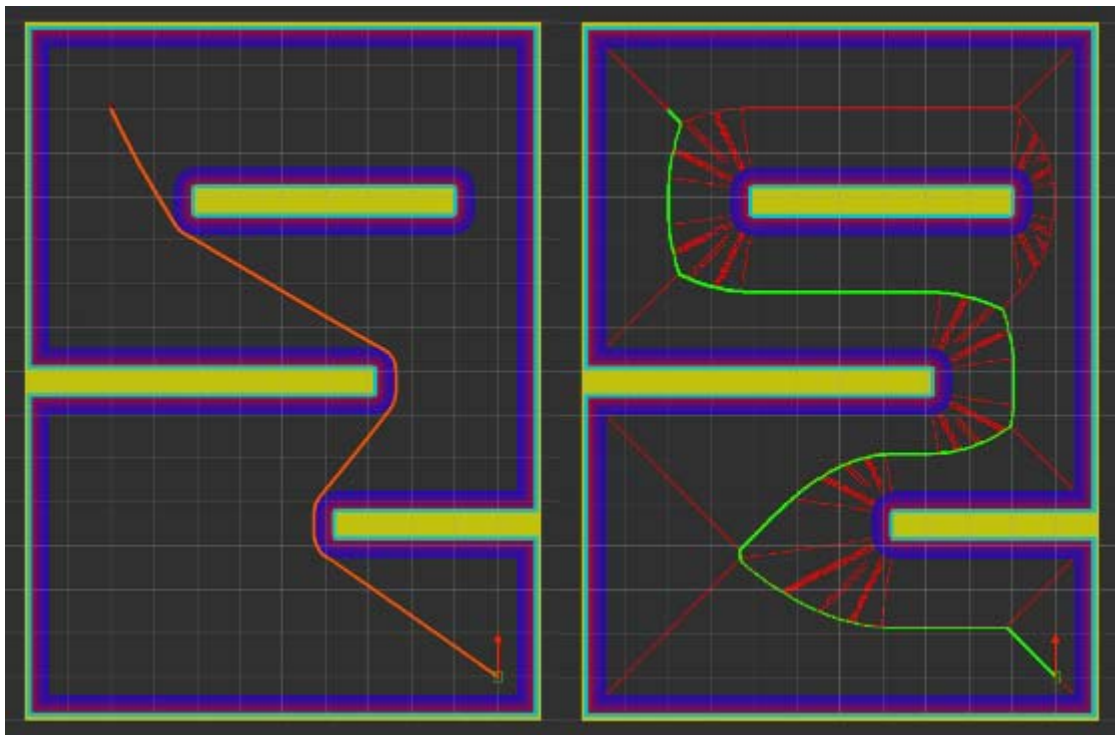
Планирование глобальной траектории

При планировании движения мини-корабля длина пути при движении в среде с препятствиями может не являться главным критерием при выборе метода планирования. Важной задачей является построение траектории движения максимально удаленной от препятствий для уменьшения вероятности столкновения с ними. Поэтому в данной работе была выполнена разработка программного пакета `voronoi_planner` [8] планирования глобальной траектории на базе обобщенных диаграмм Вороного для среды ROS, реализация которого в данной среде ранее не существовало.

Диаграмма Вороного множества точек S на плоскости представляет собой такое разбиение плоскости, при котором каждая область этого разбиения образует множество точек, более близких к одной из точек множества S , чем к любой другой точке этого множества. Таким образом, границы областей этого разбиения являются равноудаленными от ближайших двух точек. Обобщенные диаграммы Вороного строятся вокруг множества фигур на плоскости (вместо множества точек) и их границы представляют собой геометрическое место точек удаленных от этих фигур, что можно использовать для построения траектории, наиболее удаленной от препятствий. Существует ряд алгоритмов построения обобщенных диаграмм Вороного и их программных реализаций. В данной работе использована библиотека с открытым исходным кодом `dynamicvoronoi` [9, 10].

Пакет `voronoi_planner` реализован как плагин навигационного стека, реализующий интерфейс `nav_core::BaseGlobalPlanner`. Пакет читает глобальную карту препятствий, преобразует ее и передает в функцию построения обобщенной диаграммы Вороного библиотеки `dynamicvoronoi`. Далее выполняется поиск пути, включающий 3 этапа: поиск пути от текущей координаты мини-корабля до графа Вороного, поиск пути от целевой точки до графа Вороного и поиск пути на графе Вороного между двумя его точками, расположенными вблизи начального и целевого положения.

Сравнение траекторий, полученных планировщиком A^* и `voronoi_planner`, представлено на рис. 3, 4. Видно, что траектория, полученная планировщиком с использованием диаграмм Вороного, является наиболее удаленной от препятствий.



а)

б)

Рис.3–Сравнение траекторий, полученных планировщиком A* (а) и voronoi_planner (б)

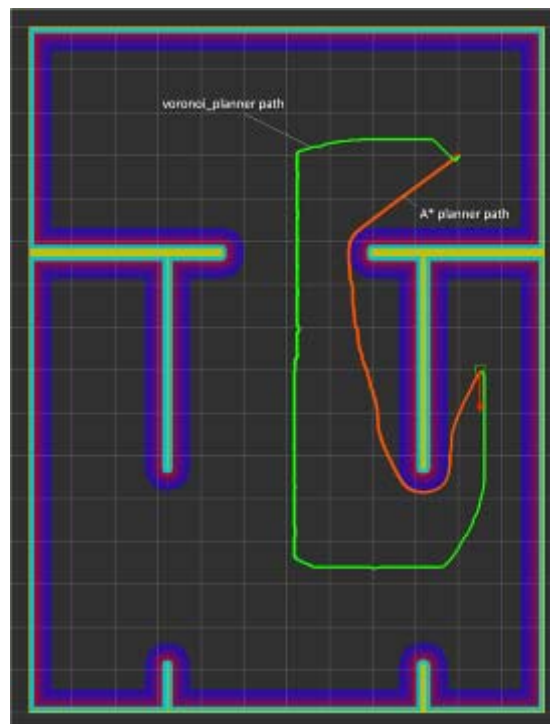


Рис.4–Сравнение траекторий, полученных планировщиком A* и voronoi_planner, пример 2

Недостатком планировщика `voronoi_planner` по сравнению с планировщиками на основе поиска на графах, такими как A^* , является, безусловно, быстроедействие. Однако, из данных оценки быстрогодействия планировщика `voronoi_planner`, представленных в таблице №1 и на рис. 5 видно, что при правильном выборе размера и разрешения карты такой планировщик вполне может использоваться для задачи глобального планирования в реальном времени. Измерение проводилось посредством команд получения отметки времени задания целевой точки и начала поиска `rostopicecho /move_base/goal/header/stamp` и отметки времени получения плана `rostopicecho /move_base/VoronoiPlanner/plan/header/stamp`.

Таблица №1

Время получения плана планировщиком `voronoi_planner` для карт различного размера

Размер карты	Среднее время получения плана, с
1200 X 900	2,8
800 X 600	0,8
600 X 450	0,5

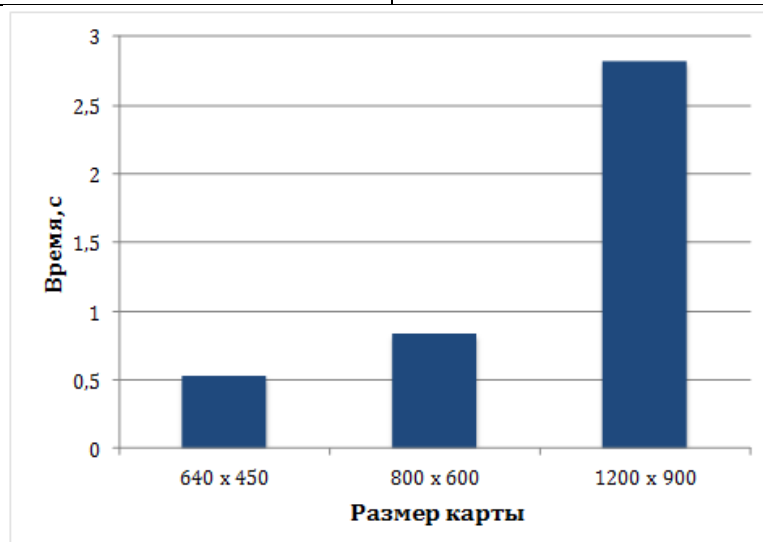


Рис. 4 – Время получения плана планировщиком `voronoi_planner` для карт различного размера

Результаты моделирования

Для подтверждения работоспособности предложенных решений в целом был разработан комплекс моделирования движения мини-корабля. Комплекс позволяет моделировать движение мини-корабля в среде с препятствиями, подавать уставки управления исполнительными механизмами, получать навигационную информацию и данные имитируемого лазерного сканера.

На рис. 5а представлена трехмерная сцена для моделирования задачи выхода мини-корабля из бухты, на рис. 5б – соответствующая карта, полученная посредством имитируемого лазерного сканера, и спланированная траектория движения мини-корабля.

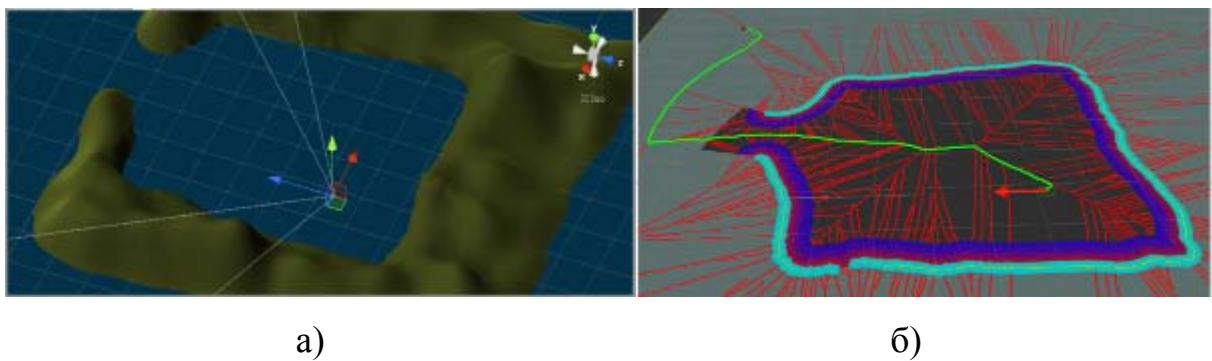


Рис. 5 –Моделирование задачи выхода мини-корабля из бухты

Использование комплекса моделирования позволило подтвердить работоспособность предложенных решений и перейти к подготовке натуральных испытаний.

Выводы

Предложенные в статье модификации программных компонентов описанной программной архитектуры (глобального планировщика и регулятора) в соответствии с результатами моделирования позволяют использовать предложенные решения для дальнейших натуральных экспериментов с мини-кораблем.

Благодарности

Работа поддержана Министерством образования и науки РФ, НИР No114041540005 «Теория и методы позиционно-траекторного управления морскими роботизированными системами в экстремальных режимах и условиях неопределенности среды» по государственному заданию ВУЗам и научным организациям в сфере научной деятельности, а также внутренним грантом Южного федерального университета «Теория и методы энергосберегающего управления распределенными системами генерации, транспортировки и потребления электроэнергии».

Литература

1. Boris Gurenko, Roman Fedorenko, Anatoly Nazarkin Autonomous surface vehicle control system // Applied Mechanics and Materials Journal.- 2014 (ISSN: 1660-9336). - Vol. 704. -p. 277.

2. Гуренко Б.В., Федоренко Р.В., Назаркин А.С. Система управления автономного надводного мини-корабля // Современные проблемы науки и образования. - 2014. - №5, URL: science-education.ru/119-14511 (дата обращения: 12.10.2015).

3. Пшихопов В.Х., Медведев М.Ю., Федоренко Р.В., Гуренко Б.В., Чуфистов В.М., Шевченко В.А. Алгоритмы многосвязного позиционно-траекторного управления подвижными объектами // Инженерный вестник Дона, 2014, №4 URL: ivdon.ru/ru/magazine/archive/n4y2014/2579.

4. Пшихопов В.Х., Гуренко Б.В. Синтез и исследование авторулевого надводного мини-корабля «Нептун» // Инженерный вестник Дона, 2013, №4 URL: ivdon.ru/ru/magazine/archive/n4y2013/1919.

5. Гуренко Б.В., Федоренко Р.В., Береснев М.А., Сапрыкин Р.В., Переверзев В.А. Разработка симулятора автономного необитаемого подводного аппарата // Инженерный вестник Дона, 2014, №3 URL: ivdon.ru/ru/magazine/archive/n3y2014/2504/.

6. Navigation - ROS Wiki // Robot Operating System URL: wiki.ros.org/navigation (accessed: 20.09.2015).

7. Пшихопов В.Х. Управление подвижными объектами в недетерминированных средах использованием неустойчивых режимов // Материалы конференции "Управление в технических, эргатических, организационных и сетевых системах". - Под редакцией С.Н. Васильева, И.А. Каляева, Д.А. Новикова, Г.Г. Себрякова. - 2012. - С. 786-789.

8. Voronoi_planner - ROS Wiki // Robot Operating System URL: wiki.ros.org/voronoi_planner (accessed: 20.09.2015).

9. B. Lau, C. Sprunk and W. Burgard, Improved Updating of Euclidean Distance Maps and Voronoi Diagrams // IEEE Intl. Conf. on Intelligent Robots and Systems (IROS). - Taipei, Taiwan. - 2010. - pp. 281 – 286.

10. Dynamicvoronoi - ROS Wiki // Robot Operating System URL: wiki.ros.org/dynamicvoronoi (accessed: 20.09.2015).

References

1. Boris Gurenko, Roman Fedorenko, Anatoly Nazarkin Autonomous surface vehicle control system Applied Mechanics and Materials Journal (ISSN: 1660-9336). 2014. Vol. 704.p. 277.

2. Gurenko B.V., Fedorenko R.V., Nazarkin A.S. Sovremennye problemy nauki i obrazovaniâ. 2014. №5, URL: science-education.ru/119-14511.

3. Pshikhopov V.Kh., Medvedev M.Yu., Fedorenko R.V., Gurenko B.V., Chufistov V.M., Shevchenko V.A. Inženernyj vestnik Dona (Rus), 2014, №4 URL: ivdon.ru/ru/magazine/archive/n4y2014/2579.

4. Pshikhopov V.Kh., Gurenko B.V. Inženernyj vestnik Dona (Rus), 2013, №4 URL: ivdon.ru/ru/magazine/archive/n4y2013/1919.

5. Gurenko B.V., Fedorenko R.V., Beresnev M.A., Saprykin R.V., Pereverzev V.A. Inženernyj vestnik Dona (Rus), 2014, №3 URL: ivdon.ru/ru/magazine/archive/n3y2014/2504/.



6. Navigation - ROS Wiki Robot Operating System URL:
wiki.ros.org/navigation (accessed: 20.09.2015).

7. Pshikhopov V.Kh. Materialy konferentsii "Upravlenie v tekhnicheskikh, ergaticheskikh, organizatsionnykh i setevykh sistemakh". Pod redaktsiey S.N. Vasil'eva, I.A. Kalyaeva, D.A. Novikova, G.G. Sebyakova. 2012.pp. 786-789.

8. Voronoi_planner - ROS Wiki Robot Operating System URL:
wiki.ros.org/voronoi_planner (accessed: 20.09.2015).

9. B. Lau, C. Sprunk and W. Burgard, Improved Updating of Euclidean Distance Maps and Voronoi Diagrams IEEE Intl. Conf. on Intelligent Robots and Systems (IROS). Taipei, Taiwan. 2010. pp. 281 – 286.

10. Dynamicvoronoi - ROS Wiki Robot Operating System URL:
wiki.ros.org/dynamicvoronoi (accessed: 20.09.2015).