

Раздел I. Алгоритмы обработки информации

УДК 004.896:681.5

DOI 10.18522/2311-3103-2023-6-8-24

Д.О. Бросалин, Б.В. Гуренко, М.Ю. Медведев

ИНТЕГРАЦИЯ ЛОКАЛЬНОГО И ГЛОБАЛЬНОГО ПЛАНИРОВЩИКОВ В СИСТЕМУ УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ*

Исследуется проблема интеграции методов локального и глобального планирования движения в системе управления роботом. Современный уровень технологического развития позволяет мобильным роботам не только следовать заранее заданным координатам, но и независимо от оператора принимать решения в реальном времени, реагируя на изменения в окружающей обстановке. Однако, динамическая природа окружающего пространства и ограничения по времени планирования пути, а также высокие скорости мобильных роботов, усложняют задачи, решаемые алгоритмами планирования. В данной работе рассмотрены некоторые методы планирования движения, основанные на клеточной декомпозиции (такие как A^ , D^* и Wavefront) и процедурах случайного поиска на графах (такие как быстро растущие случайные деревья RRT и вероятностные дорожные карты PRM), интегрированные с алгоритмом предсказания траектории движения (DWA). Проведены исследование особенностей работы каждого из перечисленных алгоритмов, а также серия численных и натуральных экспериментов по анализу влияния топологии карты на время выполнения и использование памяти алгоритмами планирования. Исследовано влияние скорости работы локального и глобального планирования при различных конфигурациях внешней среды. Для подтверждения эффективности исследуемых алгоритмов планирования пути в реальных условиях создано программное обеспечение для мобильного робота на базе колесного шасси. В статье приведены структурные и функциональные схемы взаимодействия реализованных модулей планирования и управления движением мобильного робота и окружающей среды. Также представлена математическая модель колесной платформы, для которой, на основе рассмотренных методов, разработаны алгоритмы планирования движения. В данной работе проведена оценка количественных показателей, включая время расчета алгоритма планирования движения и объем используемой алгоритмами памяти при различных картах среды. Рассмотрены как среда с произвольно расположенными препятствиями, так и различные виды лабиринтов. Также описана реализация разработанных алгоритмов в среде ROS-2. Показано, что реализованная система обеспечивает управление и планирование движения мобильного робота в реальном времени.*

Поиск пути; планирование движением; DWA A^ ; D^* ; Wave Front; PRM; ROS2.*

D.O. Brosalin, B.V. Gurenko, M.Y. Medvedev

INTEGRATION OF LOCAL AND GLOBAL SCHEDULER INTO A MOBILE ROBOT CONTROL SYSTEM

This paper investigates the problem of integrating local and global motion planning methods in a robot control system. The current level of technological development allows mobile robots not only to follow predetermined coordinates, but also to make real-time decisions independently of the operator, reacting to changes in the environment. However, the dynamic nature of the envi-

* Исследование выполнено за счет гранта Российского научного фонда №22-29-00533, <https://rscf.ru/project/22-29-00533/>, выполняемого в АО "НКБ РИСУ", г. Таганрог, Ростовская обл.

ronment and the constraints on planning time, as well as the high speeds of mobile robots, complicate the problems solved by planning algorithms. In this paper, some motion planning methods based on cellular decomposition (such as A*, D* and Wavefront) and random search procedures on graphs (such as fast growing random RRT trees and probabilistic roadmaps PRM) integrated with a motion trajectory prediction algorithm (DWA) are reviewed. A study of the performance characteristics of each of the above algorithms has been conducted, as well as a series of numerical and in-situ experiments to analyze the effect of map topology on the execution time and memory usage of the algorithms. The effect of the speed of local and global planning under different configurations of the external environment was investigated. To confirm the effectiveness of the investigated algorithms in real conditions, software for a mobile robot based on a wheeled chassis has been created. The paper presents structural and functional schemes of interaction between the implemented modules of planning and motion control of the mobile robot and the environment. It also presents a mathematical model of a wheeled platform, for which, based on the considered methods, motion planning algorithms are developed. In this paper, quantitative measures including the computation time of the motion planning algorithm and the amount of memory used by the algorithms under different environment maps are evaluated. Both environments with randomly placed obstacles and different types of mazes are considered. The implementation of the developed algorithms in the ROS-2 environment is also described. It is shown that the implemented system provides real-time control and motion planning of the mobile robot.

Pathfinding; motion planning; DWA A; D*; Wave Front; PRM; ROS2.*

Введение. В современном контексте развития робототехники, значительный интерес уделяется реализации роботов, способных эффективно функционировать в динамичных средах, адаптируясь к переменным условиям окружающего пространства. Одним из ключевых факторов, определяющих успешное функционирование мобильных роботов, является структура и функциональная схема, которые обеспечивают оптимальное взаимодействие между различными алгоритмами планирования и контроллером предсказания траектории движения.

В данной статье проводится исследование методов, основанных на клеточной декомпозиции и случайном поиске на графах. Также осуществляется интеграция рассмотренных алгоритмов с алгоритмом динамического окна (DWA), для создания комплексной системы управления роботом. Рациональное проектирование структуры и функциональной схемы является фундаментом для эффективного внедрения разработанных алгоритмов в реальных условиях и обеспечивает достижение оптимальных результатов в различных сценариях функционирования робототехнических систем.

В данной работе исследуются некоторые алгоритмы планирования, рассмотренные в обзоре [4]. Среди методов, базирующихся на клеточной декомпозиции, исследованы A* [5, 6], D* [7] и волновой алгоритм Wavefront [8]. Также исследованы алгоритмы, базирующиеся на быстрорастущих случайных деревьях [1, 9, 11] и вероятностных дорожных картах [10].

В данной работе поставлена задача разработать программное обеспечение, реализующее данные алгоритмы, и исследовать их эффективность, используя симулятор и натурные испытания.

Разработка структуры и функциональной схемы мобильного робота. Общая структурная схема рассматриваемого мобильного робототехнического комплекса колесного типа представлена на рис. 1.

Следует отметить, что на рис. 1 отсутствует модуль комплексирования данных системы навигации и системы технического зрения, которые являются частью сенсорного уровня. Этот факт обусловлен наличием встроенных вычислительных устройств, которые находятся в самих сенсорах [2, 3] и реализуют интеллектуальную обработку данных. Алгоритмы планирования движения выполняются в блоке планирования, который реализован на процессорном модуле Jetson Nano. Управление движением мобильного робота реализуется посредством регуляторов ис-

полнительных механизмами, которые относятся к нижнему уровню системы управления. Данный уровень включает в себя микроконтроллерные устройства и датчики вращения колес.

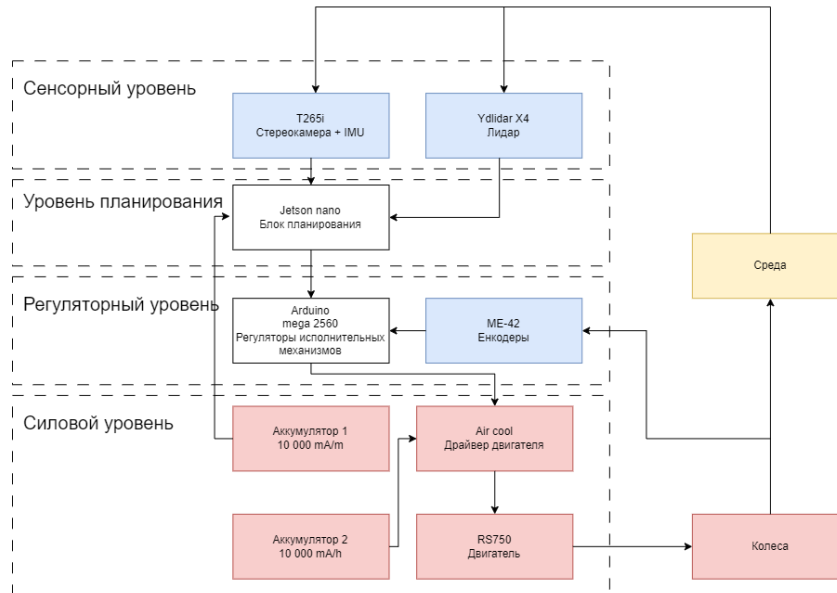


Рис. 1. Структурная схема робота

Функциональная схема программных модулей робототехнического комплекса представлена на рис. 2.

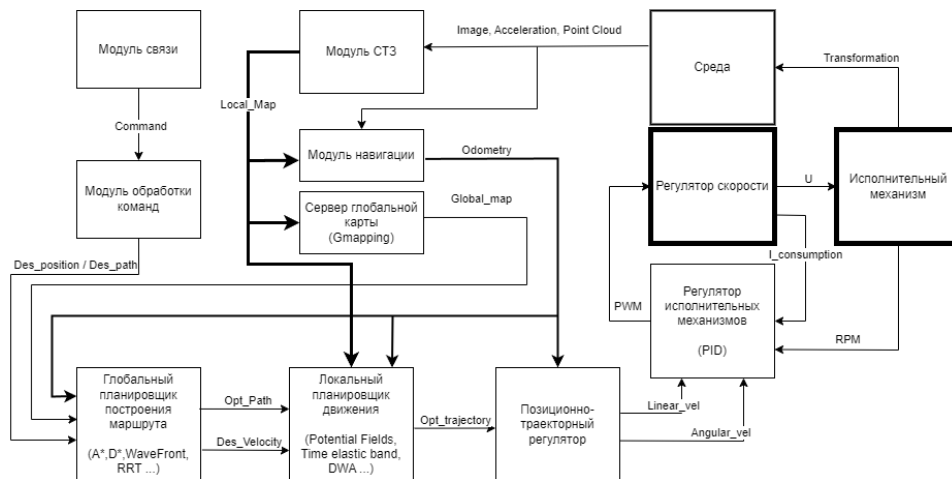


Рис. 2. Функциональная схема программных модулей робота

Схема, представленная на рис. 2, дает представление об иерархии и взаимодействии программных модулей системы управления. В схеме используется позиционно-траекторный регулятор [13, 15, 16], обеспечивает движение робота по траектории, которую выдал локальный планировщик движения.

Математическая модель мобильного робота. Модуль позиционно-траекторного регулятора базируется на многосвязной нелинейной модели кинематики и динамики мобильного робота [15, 16]. В этой связи рассмотрим соответствующие кинематическую и динамическую модель мобильного робота.

В качестве мобильной платформы рассматривается четырехколесный мобильный робот. Особенностью данного робота является то, что оси его колес не поворачиваются. Следовательно, управление направлением его перемещения реализуется с помощью изменения скорости вращения колес относительно друг друга. Для анализа модели подвижного объекта введем неподвижную систему координат OXY и связанную систему координат с объектом Oxy , как показано на рис. 3.

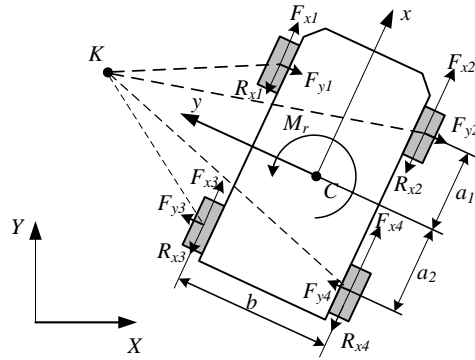


Рис. 3. Модель автономного мобильного робота

В неподвижной системе координат для линейных скоростей подвижного объекта можно записать

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} \dot{x} \cos(\phi) - \dot{y} \sin(\phi) \\ \dot{x} \sin(\phi) + \dot{y} \cos(\phi) \end{bmatrix} = R(\phi) \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, \quad (1)$$

где ϕ - угол ориентации подвижного объекта относительно оси Ox ; (x, y) - координаты робота в связанной системе; (X, Y) - координаты робота в неподвижной системе; $R(\phi)$ - матрица кинематики.

Дифференцируя (1) получим

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \end{bmatrix} = R(\phi) \begin{bmatrix} \ddot{x} - \dot{y}\dot{\phi} \\ \ddot{y} + \dot{x}\dot{\phi} \end{bmatrix} = R(\phi) \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix}, \quad (2)$$

В каждый момент времени робот совершает движение вокруг точки C , которая является центром вращения и в которой линейные скорости равны нулю относительно системы координат Oxy . Ее координаты могут быть найдены по формуле

$$\begin{bmatrix} x_K \\ y_K \end{bmatrix} = \begin{bmatrix} -\dot{y}/\dot{\phi} \\ \dot{x}/\dot{\phi} \end{bmatrix}. \quad (3)$$

Продольные и поперечные компоненты скоростей колес могут быть найдены по формулам

$$\begin{aligned} \dot{x}_1 &= \dot{x}_4 = \dot{x} - 0.5b\dot{\phi}, \\ \dot{x}_2 &= \dot{x}_3 = \dot{x} + 0.5b\dot{\phi}, \\ \dot{y}_1 &= \dot{y}_2 = \dot{y} + a_1\dot{\phi}, \\ \dot{y}_3 &= \dot{y}_4 = \dot{y} - a_2\dot{\phi}. \end{aligned} \quad (4)$$

где b – ширина платформы подвижного объекта; a_1, a_2 – расстояние от центра масс до оси, соединяющей переднюю и заднюю пары колес соответственно.

Допустим, что скорости вращения колес каждого борта синхронизированы тогда уравнения движения подвижного объекта можно представить в виде

$$\begin{aligned} m\ddot{x} &= (F_{x1} + F_{x2}) + (F_{x3} + F_{x4}) - R_x, \\ m\ddot{y} &= -F_y, \\ J_z\ddot{\phi} &= b((F_{x1} + F_{x2}) - (F_{x3} + F_{x4})) - M_r. \end{aligned} \quad (5)$$

где m – масса подвижного объекта;

J_z – момент инерции подвижного объекта;

$F_{x1}, F_{x2}, F_{x3}, F_{x4}$ – продольная составляющая сил тяги;

F_y – равнодействующая поперечной составляющей сил тяги;

R_x – равнодействующая продольных сил сопротивления;

M_r – момент сопротивления.

Выражение для равнодействующей продольных сил сопротивления имеет вид

$$R_x = \sum_{i=1}^4 R_{xi} = \eta \frac{mg}{2} (\operatorname{sgn}(\dot{x}_1) + \operatorname{sgn}(\dot{x}_2)), \quad (6)$$

где η – коэффициент трения качения.

Выражение для равнодействующей поперечной составляющей сил тяги имеет вид

$$F_y = \sum_{i=1}^4 F_{yi} = \mu \frac{mg}{a_1 + a_2} (a_2 \operatorname{sgn}(\dot{y}_1) + a_1 \operatorname{sgn}(\dot{y}_3)), \quad (7)$$

где μ – коэффициент трения скольжения.

Выражение для момента сопротивления имеет вид

$$M_r = \mu \frac{a_1 a_2 mg}{a_1 + a_2} (\operatorname{sgn}(\dot{y}_1) - \operatorname{sgn}(\dot{y}_3)) + \eta \frac{bmg}{4} (\operatorname{sgn}(\dot{x}_2) - \operatorname{sgn}(\dot{x}_1)). \quad (8)$$

В общем виде модель динамики подвижного объекта с вектором состояния $q=(X, Y, \phi)$ в системе координат ОХУ имеет вид

$$M\ddot{q} + c(q, \dot{q}) = E(q)\tau, \quad (9)$$

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J_z \end{bmatrix}, \quad (10)$$

$$c(q, \dot{q}) = \begin{bmatrix} R_x \cos(\phi) - F_y \sin(\phi) \\ R_x \sin(\phi) + F_y \cos(\phi) \\ M_r \end{bmatrix}, \quad (11)$$

$$E(q) = \begin{bmatrix} \cos(\phi)/r & \cos(\phi)/r \\ \sin(\phi)/r & \sin(\phi)/r \\ 0.5b/r & -0.5b/r \end{bmatrix}, \quad (12)$$

(r – радиус колеса),

$$\tau_j = 2rF_{xj} (j = 1, 2). \quad (13)$$

Пусть $x_1 = X, x_2 = \dot{X}, x_3 = Y, x_4 = \dot{Y}, x_5 = \phi, x_6 = \dot{\phi}$, тогда полная модель ПО [2] в переменных состояния будет иметь вид:

$$\begin{aligned} \dot{x}_1 &= x_2; \\ \dot{x}_2 &= \frac{1}{m}(F_y \sin x_5 - R_x \cos x_5 + 2F_{x1} \cos x_5 + 2F_{x2} \cos x_5); \\ \dot{x}_3 &= x_4; \\ \dot{x}_4 &= \frac{1}{m}(-F_y \cos x_5 - R_x \sin x_5 + 2F_{x1} \sin x_5 + 2F_{x2} \sin x_5); \dot{x}_5 = x_6; \\ \dot{x}_6 &= \frac{1}{J_z}(-M_r + 0.5bF_{x1} - 0.5bF_{x2}), \end{aligned} \quad (14)$$

$$F_y = \mu \frac{mg}{a_1+a_2} (a_1 \operatorname{sgn}(x_4 + a_1x_6) + a_2 \operatorname{sgn}(x_4 - a_2x_6)), \quad (15)$$

$$R_x = \eta \frac{mg}{2} (\operatorname{sgn}(x_2) - 0.5bx_6) + \operatorname{sgn}(x_2 + 0.5bx_6), \quad (16)$$

$$M_r = \mu \frac{a_1a_2mg}{a_1+a_2} (\operatorname{sgn}(x_4 + a_1x_6) - \operatorname{sgn}(x_4 - a_2x_6)) + \eta \frac{bmg}{4} (\operatorname{sgn}(x_2 + 0.5bx_6) - \operatorname{sgn}(x_2 - 0.5bx_6)). \quad (17)$$

Алгоритмы управления исполнительными механизмами. Траектория реализуется посредством изменения линейной и угловой скорости во времени. Связь между желаемой линейной скоростью и угловой скоростью каждого колеса определяется уравнениями кинематики, которые для рассматриваемого мобильного робота имеют вид:

$$w_{fl} = \frac{v - \frac{w*L}{2}}{R}; w_{fr} = \frac{v + \frac{w*L}{2}}{R}; w_{rl} = \frac{v - \frac{w*L}{2}}{R}; w_{rr} = \frac{v + \frac{w*L}{2}}{R}; \quad (18)$$

где $w_{fl}, w_{fr}, w_{rl}, w_{rr}$ это угловые скорости каждого колеса (переднего левого, переднего правого, заднего левого, заднего правого);

w – угловая скорость робота;

v – линейная скорость робота.

Управление угловыми и линейными скоростями реализуется посредством ПИД-регуляторов, структура которых представлена на рис. 4.

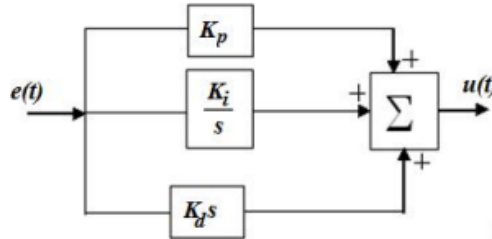


Рис. 4. Структура ПИД-регулятора

На рис. 4 $e(t)$ – ошибка управления; $u(t)$ – выходной управляющий сигнал.

Система управления мобильного робота включает ПИД-регуляторы скоростей вращения колес.

Регулятор линейной скорости движения мобильного робота описывается следующими выражениями:

$$e(t) = V_{desired} - V_{current}; \quad (19)$$

$$u(t) = K_p * e(t) + K_i \int e(t)dt + K_d * \frac{de(t)}{dt}, \quad (20)$$

здесь $e(t)$ – ошибка; $u(t)$ – управление; $V_{current}$ – текущая скорость робота; $V_{desired}$ – желаемая скорость робота; K_p, K_i, K_d – коэффициенты регулятора.

Аналогичный закон управления также применен к управлению угловыми скоростями для каждого колеса.

$$w_{wheel_current} = \frac{(\frac{i}{i_{circle}} * R * \pi * 2)}{dt}, \quad (21)$$

где $w_{wheel_current}$ – текущая угловая скорость колеса; i – количество сработанных прерываний контроллера в зависимости от положения энкодера; i_{circle} – количество полюсов энкодера; R – радиус колеса.

Локальный планировщик движения. В ходе движения робота по заранее известной карте и уже спланированным маршрутом может появиться некартографированное препятствие. В такой ситуации нецелесообразно пересчитывать всю траекторию движения, достаточно использовать локальный планировщик для обхода препятствий.

В данной работе для решения задачи обхода препятствий используется алгоритм, базирующийся на методе динамического окна – (Dynamic Window Approach), который представлен на рис. 5.

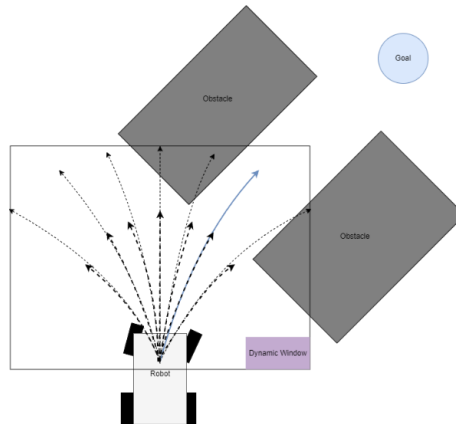


Рис. 5. Метод динамического окна (DWA)

На вход алгоритма подаются текущие координаты робота (скорости, ускорения, позиция) и целевая позиция. Метод генерирует возможные куски траектории при переборе ряда линейных и угловых скоростей, которые достижимы в ближайший момент времени. Эти траектории имеют определенные веса, отвечающие за желаемую скорость, угол ориентации робота к цели, пересечение с препятствиями. Недостатком алгоритма динамического окна является проблема локальных минимумов. Для устранения локальных минимумов используется процедура их детектирования. В случае обнаружения локального минимума запускается перепланирование глобального маршрута. Используются следующие условия попадания в локальный минимум: расстояние до цели больше заданной величины, а текущая линейная скорость равна 0 в течении некоторого времени.

Робот может достичь нулевой скорости при следующих ситуациях:

- ◆ Робот попал локального минимума.
- ◆ Робот достиг цели.
- ◆ Выполняется разворот на месте (если кинематика платформы позволяет).

Алгоритм динамического окна описывается следующим выражением:

$$\begin{aligned} f_{int} &= 1; \\ f_{vel} &= 0.3; \\ f_{dir} &= 0.5; \end{aligned}$$

$$\text{Trajectory}_{idx} = \min(F_{intersection} * f_{int} + F_{velocity} * f_{vel} + F_{direction} * f_{dir}). \quad (22)$$

Для увеличения плавности движения робота также добавляют коэффициент переключения с одной траектории на другую, чтобы не было резких скачков между максимальным и минимальными значениями скоростей в заданном диапазоне.

Коэффициенты f_{int} , f_{vel} и f_{dir} в (22) подбираются экспериментально.

При увеличении коэффициента f_{dir} , отвечающего за угол направления движения увеличивается шанс попадания в локальный минимум. При этом, если уменьшить этот коэффициент, то алгоритм обретает некую свободу и способен преодолевать препятствия более эффективно.

При увеличении коэффициента f_{vel} , отвечающего за скорость движения, увеличивается качество поддержания крейсерской скорости.

При увеличении коэффициента f_{int} , отвечающего за отдаление от препятствий, робот планирует траекторию на большем расстоянии от препятствий.

Применение метода динамического окна вместе с глобальным алгоритмом планирования позволяет проходить сложные участки карты.

Указанные выше весовые коэффициенты могут изменяться динамически, в следующих ситуациях:

- ◆ Если есть пересечение с препятствием.
- ◆ Если линейная скорость меньше заданной.
- ◆ Если угол к целевой точке достаточно велик.

Модуль планирования не выдает никаких траекторий движения, если выполняется хотя-бы одно из перечисленных ниже условий:

- ◆ Не работает сервер с готовой картой.
- ◆ Не работает модуль одометрии от робота.
- ◆ Обновление одометрии происходит довольно редко.
- ◆ Препятствие находится в критичной зоне для робота.

Глобальный планировщик движения. В данной работе исследуются следующие глобальные алгоритмы планирования траекторий движения: A^* , D^* , Wavefront, PRM, RRT.

Алгоритм A^* является информированным алгоритмом поиска пути, который используется для поиска кратчайшего пути от начальной до конечной точки на графе. Он использует комбинацию оценок стоимости, чтобы эффективно выбирать следующий узел для исследования. Формула A^* :

$$f(n) = g(n) + h(n). \quad (23)$$

где $f(n)$ – полная оценка стоимости прохождения через узел n ; $g(n)$ – стоимость пути от начальной вершины до узла n ; $h(n)$ – эвристическая оценка стоимости от узла n до целевого узла.

Сложность алгоритма оценивается следующим образом. В худшем случае: $O(b^d)$, где b – максимальное количество возможных действий, d – глубина решения. В среднем: $O(b^{(d/2)})$.

Алгоритм D^* – это алгоритм поиска пути, который представляет собой улучшенную версию алгоритма Dijkstra. Он использует информацию об изменениях в графе для более эффективного пересчета пути от начальной до конечной точки. Он также обеспечивает динамическое обновление пути при изменениях в графе. Сложность алгоритма в худшем случае оценивается как $O(|E| + |V|\log|V|)$, где $|E|$ – количество ребер, $|V|$ – количество вершин.

Алгоритм волнового фронта Wavefront является методом планирования движения в дискретной среде. Он начинает волну с выделения начальной точки и распространяет волны по карте, чтобы найти оптимальный путь к целевой точке. Данный алгоритм обычно используется в статических и малоразмерных средах. Сложность волнового алгоритма оценивается в худшем случае как $O(n^2)$, где n – количество ячеек на карте.

Алгоритм вероятностной дорожной карты PRM (Probabilistic Roadmap) – это вероятностный алгоритм планирования движения, который строит граф путей, основанный на случайно сгенерированных точках в пространстве состояний робота. Он используется для поиска путей в сложных и многомерных пространствах. Сложность алгоритма определяется в худшем случае как $O(n^2 * \log n)$, где n – количество сгенерированных точек.

Алгоритм быстро растущих случайных деревьев RRT (Rapidly-exploring Random Tree) – это алгоритм планирования движения, который строит дерево путей, добавляя случайные узлы и расширяясь в направлении этих узлов. Он используется для эффективного планирования путей в пространствах большой размерности, где структура окружения неизвестна заранее. Сложность алгоритма определяется в худшем случае как $O(n^2)$, где n – количество узлов в дереве.

Приведенные оценки сложности алгоритмов характеризуют худшие ситуации для планирования. На практике, при благоприятных входных данных, сложность перечисленных алгоритмов может быть меньше. Благоприятные данные могут способствовать более раннему достижению целевой точки ввиду принципа их работы.

Программное обеспечение управления и планирования перемещения мобильного робота. Перенос разработанных методов и алгоритмов из программной среды matlab в c++ или python. Среди многих фреймворков для робототехнических комплексов, был выбран ROS2 для реализации программных модулей [14]. Выбран данный фреймворк т.к. он имеет встроенный функционал по отображению и сбору данных. Так же при использовании Middleware DDS Cyclone, фреймворк позволяет создавать группы роботов, общающихся в одной сети. Такой подход может означать добавления в контур управления модуль сервера траекторий для обмена данными о планировании между участниками.

Концепция ROS2 предполагает, что каждая программа – это узел (node), который общается с другими узлами посредством тем (topic) как показано на рис. 6. У каждой темы существует свой формат сообщений. В частности, для работы с линейной алгеброй в Python используется numpy, а для C++ используется Eigen.

Существует две концепции обмена данными в ROS2:

- ◆ Подписчик, публикатор.
- ◆ Сервис, клиент.

Для более надежной архитектуры системы управления, используется микро сервисная архитектура, где каждый узел, решает отдельную задачу. В случае отказа одного из них, вся система продолжит работу. Отказавшие модули будут перезапущены.

Из рис. 6 видно, что модуль глобального планировщика (global_planner) выполняет следующие функции:

- ◆ Получает информацию из тем карты (/map), позиции робота(/robot_0/pose) и целевой точки (/goal_pose).
- ◆ Отдает информацию о траектории (/robot_0/trajectory) в модуль локального планировщика(/unstable_planner_new).

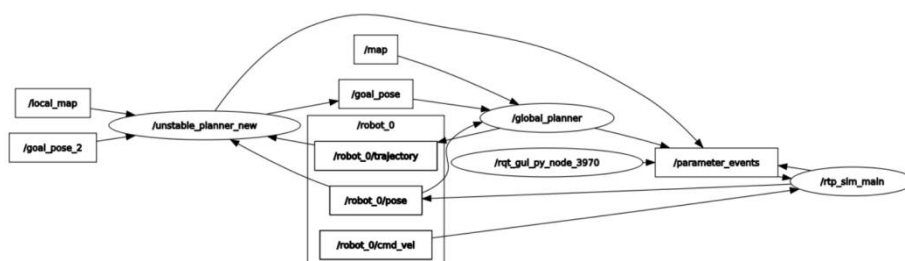


Рис. 6. Взаимодействие программных узлов в среде ROS2. (в прямоугольниках обозначены названия топиков, в овалах обозначены названия узлов)

Виртуальная среда для тестирования системы управления и планирования мобильного робота. Для простоты реализации карты в виртуальной среде выбран формат Occupancy Grid (Сетка занятости), который представлен на рис. 7. Данный формат имеет несколько основных полей, представленных в табл. 1, также данный формат использует фреймворк ROS2.

Таблица 1

Настройки Occupancy Grid

| Характеристика | Значение |
|---------------------|----------------------|
| Длина карты | Ко-во ячеек в длину |
| Ширина карты | Ко-во ячеек в ширину |
| Разрешение ячейки | Размер ячейки (м) |
| Точка отсчета карты | Координаты (x,y,z) |

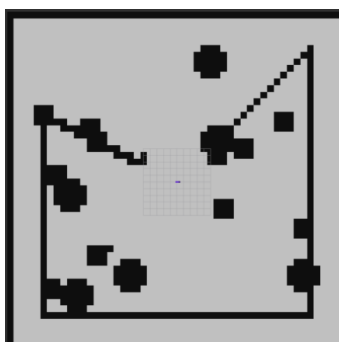


Рис. 7. Карта занятости (Occupancy Grid) в Rviz2

Цвет ячейки означает вероятность существования в ней препятствия. Чем темнее зона, тем больше вероятность того, что в ней находится препятствие. В случае, представленном на рис. 7, веса были бинаризованны до двух пороговых значений (0-100).

В качестве сложной среды был выбран лабиринт. Существует множество реализаций к походу создания лабиринта. В первую очередь лабиринты распределяются по размерности:

- ◆ 2D;
- ◆ 3D;
- ◆ размерности более высокого порядка (предусматривают искажения либо порталы в определенных точках маршрута).

Также лабиринты можно классифицировать по гипер-размерности. Этот параметр отвечает за размер непосредственного перемещающегося по лабиринту объекта. Обычно все лабиринты – это не гиперлабиринты, т.е. по ним перемещается определенная точка, а след движения образует линию маршрута. В гиперлабиринтах объект обладает дополнительными свойствами, и обычно данный подход реализуется в лабиринтах с большой размерностью пространства (например, 3D). Так же у лабиринта есть такая характеристика как тесселяция, которая отвечает за геометрию ячеек.

Тесселяция бывает: Ортогональной; Дельта; Сигма; Тетта; Ипсилон; Дзета; Омега; Скак; Фрактальной.

Так же немало важным параметром генерации лабиринта является маршрутизация. Данный параметр влияет на геометрию проходов в зависимости от тесселяции. Маршрутизация бывает следующих типов.

Лабиринт с идеальной маршрутизацией – лабиринт без петель или замкнутых цепей и без недостижимых областей. Также он называется лабиринтом с одиночным соединением. Из каждой точки существует ровно один путь к любой другой точке. Лабиринт имеет только одно решение. С точки зрения программирования такой лабиринт можно описать как дерево, связующее множество ячеек или вершин.

Лабиринт с плетеной маршрутизацией – это лабиринт, в котором нет тупиков. Также его называют лабиринтом с многократными соединениями. В таком лабиринте используются проходы, замыкающиеся и возвращающиеся друг к другу. Они заставляют тратить больше времени на ходьбу кругами вместо попадания в тупики.

Лабиринт с одно маршрутной маршрутизацией подразумевает лабиринт без развилок. Одно маршрутный лабиринт содержит один длинный извивающийся проход, который меняет направление на всём протяжении лабиринта. Сложность возникает только при случайном развороте в обратное направление.

Лабиринт с разреженной маршрутизацией – это лабиринт, в котором не прокладываются проходы через каждую ячейку, то есть некоторые из них не создаются. Это подразумевает наличие недостижимых областей, то есть он в некотором смысле противоположен плетёному лабиринту. Похожую концепцию можно применить и при добавлении стен, благодаря чему можно получить неравномерный лабиринт с широкими проходами и комнатами.

Лабиринт с частично плетеной маршрутизацией – это смешанный лабиринт, в котором есть и петли, и тупики. Слово «плетёный» можно использовать для количественной оценки, то есть «лабиринт с сильным плетением» – это лабиринт со множеством петель или убранных стен, а в «лабиринте со слабым плетением» их всего несколько.

В качестве простоты реализации были взяты лабиринты, которые представлены на рис. 8, со следующими характеристиками, представленными в табл. 2 и 3.

Таблица 2

Параметры лабиринта А

| Характеристика | Значение |
|----------------|-----------------|
| Размерность | 2D |
| Тесселяция | Ортогональная |
| Маршрутизация | одно маршрутная |

Таблица 3

Параметры лабиринта Б

| Характеристика | Значение |
|----------------|---------------|
| Размерность | 2D |
| Тесселяция | Ортогональная |
| Маршрутизация | Идеальный |

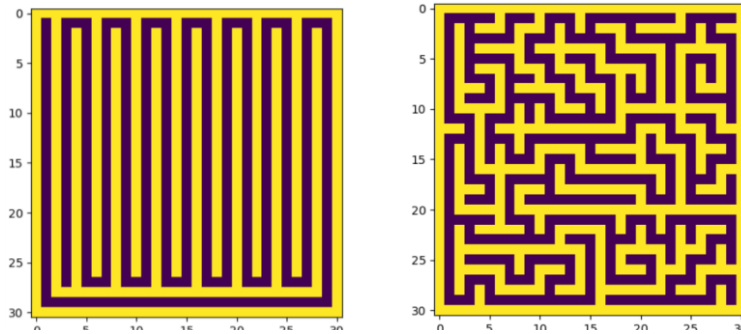


Рис. 8. Пример реализованных лабиринтов согласно табл. 2 и 3 (слева одномаршрутный А, справа идеальный Б)

Отладка и тестирование программного обеспечения мобильного робота в виртуальной среде и на мобильной платформе. Для проведения опытов использован робот WM4-1A, представленный на рис. 9. Данный робот был выбран, потому что его математическая модель совпадает с той, что реализована в симуляции.



Рис. 9. Платформа для испытаний

Для натурных испытаний использована карта среды, представленная на рис. 10. Данная карта получена на основе решения задачи SLAM в одном из помещений.

В табл. 4-6 представлены полученные оценки времени выполнения исследуемых алгоритмов локального планирования при разных конфигурациях среды. Следует отметить, что для улучшения полученных показателей по времени возможно применение следующих техник:

- ◆ Распараллеливание процессов для асинхронного вычисления и получения промежуточных результатов.
- ◆ Определение пересечений с препятствиями при помощи нейросетевых технологий по кластеризации набора облака точек.

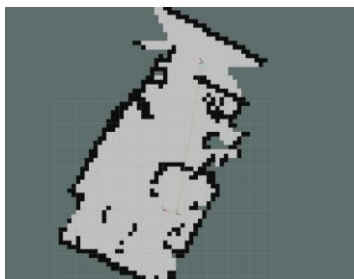


Рис. 10. Карта полученная от SLAM [50м,50м] с шагом 0.2 м

Таблица 4

**Эксперименты алгоритма планирования с использованием DWA
(прогнозирование 1с, диапазон скоростей: линейная [-0.5; 4] м/с,
угловая [-1,1] рад/с)**

| Номер конфигурации алгоритма планирования | Шаг по скорости (м/с) | Шаг по угловой скорости (рад/с) | Время выполнения (с) |
|---|-----------------------|---------------------------------|----------------------|
| 1 | 0.01 | 0.01 | 0.500 |
| 2 | 0.1 | 0.1 | 0.153 |
| 3 | 0.3 | 0.1 | 0.053 |
| 4 | 0.3 | 0.3 | 0.037 |
| 5 | 0.5 | 0.3 | 0.011 |

Таблица 5

**Эксперименты алгоритма планирования с использованием DWA
(прогнозирование 3с, диапазон скоростей: линейная [-0.5; 4], угловая [-1,1])**

| Номер конфигурации алгоритма планирования | Шаг по скорости (м/с) | Шаг по угловой скорости (рад/с) | Время выполнения (с) |
|---|-----------------------|---------------------------------|----------------------|
| 1 | 0.01 | 0.01 | 1.252 |
| 2 | 0.1 | 0.1 | 0.723 |
| 3 | 0.3 | 0.1 | 0.439 |
| 4 | 0.3 | 0.3 | 0.131 |
| 5 | 0.5 | 0.3 | 0.096 |

Таблица 6

**Эксперименты алгоритма планирования с использованием DWA
(прогнозирование 6с, диапазон скоростей: линейная [-0.5; 4], угловая [-1,1]).**

| Номер конфигурации алгоритма планирования | Шаг по скорости (м/с) | Шаг по угловой скорости (рад/с) | Время выполнения (с) |
|---|-----------------------|---------------------------------|----------------------|
| 1 | 0.01 | 0.01 | 1.852 |
| 2 | 0.1 | 0.1 | 1.323 |
| 3 | 0.3 | 0.1 | 0.651 |
| 4 | 0.3 | 0.3 | 0.234 |
| 5 | 0.5 | 0.3 | 0.142 |

В ходе проведения экспериментов в симуляторе с картой на рис. 10 были получены характеристики производительности исследованных алгоритмов планирования. Указанные данные позволяют провести сопоставление с результатами, которые были получены путем натуральных испытаний. Результаты исследования глобальных алгоритмов планирования в симуляторе и на натурном образце, представлены в табл. 7.

Из анализа данных табл. 7 следует, что время выполнения алгоритма планирования значительно зависит от плотности заполнения и формы препятствия. Алгоритм RRT, благодаря своей оптимальной загрузке памяти и высокой скорости планирования, подтверждает свою эффективность не только в качестве глобального планировщика, но и в роли модификации для локального планировщика.

Таблица 7

Эксперименты глобальных алгоритмов планирования при испытаниях в симуляторе и на натурном образце

| Метод | Параметры | Натурный эксперимент (среднее значение из 20 запусков) [50;50] 0.2м | Эксперимент в симуляторе (среднее значение из 20 запусков) [50;50] 1м |
|------------|--------------|---|---|
| A* | время | 4.56754 с | 0.463214253 с |
| | Объем памяти | 85 200 b | 49 916 b |
| D* | время | 8.5953 с | 0.5879719257354736 |
| | объем памяти | 561 036 b | 271 036 b |
| Wave front | время | 6.23589 с | 1.45563898541 |
| | объем памяти | 88184 b | 65184 b |
| PRM | Время | 9.025176525115967 | 9.02517652511597 |
| | объем памяти | 74708 b | 74708 b |
| RRT | Время | 2.009567610 | 0.367093334007 |
| | объем памяти | 63525 b | 66932 b |

Чтобы обеспечить оперативную реакцию локального планировщика на изменения в глобальном пути, критически важно правильно настроить параметры алгоритма динамического окна, обеспечив при этом превосходство по времени по сравнению с результатами глобального планирования. Кроме того, выбор параметров локального планировщика должен учитывать временной запас для адекватной реакции при достижении максимальной скорости. В случае необходимости следует корректировать максимальную скорость таким образом, чтобы обеспечить достаточное время для прогнозирования и улучшения производительности локального планировщика.

Заключение. В данной работе проведено исследование гибридных алгоритмов планирования движения мобильного робота. В качестве глобального алгоритма использованы различные модификации A*, D*, волновой алгоритм, алгоритм быстрорастущих случайных деревьев и алгоритм вероятностных дорожных карт. Проведены исследования с использованием симулятора и натуральных экспериментов с колесным роботом.

Получены оценки быстродействия и объема используемой памяти для различных карт. Исходя из полученных оценок можно сделать вывод, что для повышения скорости расчета алгоритмов планирования необходимо также учитывать геометрию местности. Нельзя пренебрегать таким свойством, как демпфированная сложность, так как применение алгоритмов по назначению сможет дать на выходе максимальную производительность. Обработка и учет препятствий очень важны при

быстродействию. Так, например RRT должен проверять свое пересечение с другими препятствиями на карте. Возможно использование дополнительного свойства поиска ближайшего соседа или методы кластеризации [17–18] для упрощения геометрии препятствий и их обработки. Также стоит рассмотреть использование нейронных сетей для кластеризации препятствий в простые геометрические фигуры [19]. В качестве алгоритмов локального планирования стоит рассмотреть способность их выхода из локального оптимума или методы, которые изначально лишены данного недостатка, но при этом могут быть затратны по памяти и производительности [20].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Hamdan N., Pshikhopov V., Medvedev M., Brosalin D., Vasileva M., Gurenko B.* Study of Path Planning Methods in Two-Dimensional Mapped Environments / In: Ronzhin, A., Sadigov, A., Meshcheryakov, R. (eds) Interactive Collaborative Robotics. ICR 2023 // Lecture Notes in Computer Science. Vol. 14214. – Springer, Cham, 2023. – https://doi.org/10.1007/978-3-031-43111-1_13.
2. *Lucet E., Grand C., Sallé D. and Bidaud P.* Dynamic yaw and velocity control of the 6WD skid-steering mobile robot RobuROC6 using sliding mode technique // 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 2009. – P. 4220-4225. – DOI: 10.1109/IROS.2009.5354373.
3. *Hausamann P., Sinnott C.B., Daumer M. et al.* Evaluation of the Intel RealSense T265 for tracking natural human head motion // Sci. Rep. – 2021. – 11. – 12486. – <https://doi.org/10.1038/s41598-021-91861-5>.
4. *Казakov К.А., Семенов В.А.* Обзор современных методов планирования пути // Тр. ИСП РАН. – 2016. – Т. 28 (4). – С. 241-294.
5. *Zhou R., Hansen E.A.* Memory-Bounded $\{A^*\}$ Graph Search // The Florida AI Research Society Conference – FLAIRS. – 2002. – P. 203-209.
6. *Hart P.E., Nilsson N.J., Raphael B.A.* Formal Basis for the Heuristic Determination of Minimum Cost Paths // IEEE Transactions on Systems Science and Cybernetics. – 1968. – Vol. 2. – P. 100-107.
7. *Stentz A.* Optimal and efficient path planning for partially known environments // In Intelligent Unmanned Ground Vehicles. – Springer, Boston, MA, USA, 1997. – P. 203-220.
8. *Murphy R.R.* Introduction to AI Robotics. – 1st ed. – MIT Press: Cambridge, MA, USA, 2000.
9. *LaValle S.M., Kuffner J.J.* Rapidly-exploring random trees: Progress and prospects // 2000 Workshop on the Algorithmic Foundations of Robotics. – 2000. – P. 293-308.
10. *Kavraki L.E., Svestka P., Latombe J.-C., Overmars M.H.* Probabilistic roadmaps for path planning in high-dimensional configuration spaces // IEEE Transactions on Robotics and Automation. – 1996. – Vol. 12 (4). – P. 566-580.
11. *Pshikhopov V., Medvedev M., Kostjukov V., Houssein F., Kadhim A.* Trajectory planning algorithms in two-dimensional environment with obstacles // Informatics and Automation. – 2022. – 21:3. – P. 459-492.
12. *Tianran Peng, Jun Qian, Bin Zi, Jiakui Liu, Xingwei Wang,* Mechanical Design and Control System of an Omni-directional Mobile Robot for Material Conveying // Procedia CIRP. – 2016. – Vol. 56. – P. 412-415. – ISSN 2212-8271. – <https://doi.org/10.1016/j.procir.2016.10.068>.
13. *Pshikhopov V.Kh. (Ed.), Beloglazov D., Finaev V., Guzik V., Kosenko E., Krukhmalev V., Medvedev M., Pereverzev V., Pyavchenko A., Saprykin R., Shapovalov I., Soloviev V.* Path Planning for Vehicles Operating in Uncertain 2D Environments. – Elsevier, Butterworth-Heinemann, 2017. – 312 p. – ISBN: 9780128123058.
14. *Macenski T. Foote, Gerkey B., Lalancette C. Woodall W.* Robot Operating System 2: Design, architecture, and uses in the wild // Science Robotics. – May 2022. – Vol. 7.
15. *Pshikhopov V.Kh., Medvedev M.Yu., Gaiduk A.R., Fedorenko R.V., Krukhmalev V.A., Gurenko B.V.* Position-Trajectory Control System for Unmanned Robotic Airship // IFAC Proceedings Volumes (IFAC-PapersOnline). – 2014. – P. 8953-8958.
16. *Pshikhopov V., Medvedev M.* Multi-Loop Adaptive Control of Mobile Objects in Solving Trajectory Tracking Tasks // Automation and Remote Control. – 2020. – Vol. 81 (11). – P. 2078-2093.

17. Bogoslavskiy I., & Stachniss C. Efficient Online Segmentation for Sparse 3D Laser Scans // PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science. – 2017. – P. 1-12. – Retrieved from <https://link.springer.com/article/10.1007%2Fs41064-016-0003-y>.
18. Liu C, Xie S, Ma X, Huang Y, Sui X, Guo N, Yang F, Yang X. A Hierarchical Clustering Obstacle Detection Method Applied to RGB-D Cameras // Electronics. – 2023. – 12 (10). – 2316. – <https://doi.org/10.3390/electronics12102316>.
19. Kang D, Wong A, Lee B, Kim J. Real-Time Semantic Segmentation of 3D Point Cloud for Autonomous Driving // Electronics. – 2021. – 10 (16). – 1960. – <https://doi.org/10.3390/electronics10161960>.
20. Phalgun Chintala, Rolf Dornberger, and Thomas Hanne. Robotic Path Planning by Q Learning and a Performance Comparison with Classical Path Finding Algorithms // International Journal of Mechanical Engineering and Robotics Research. – June 2022. – Vol. 11, No. 6. – P. 373-378. – DOI: 10.18178/ijmerr.11.6.373-378.

REFERENCES

1. Hamdan N., Pshikhopov V., Medvedev M., Brosalin D., Vasileva M., Gurenko B. Study of Path Planning Methods in Two-Dimensional Mapped Environments / In: Ronzhin, A., Sadigov, A., Meshcheryakov, R. (eds) Interactive Collaborative Robotics. ICR 2023, *Lecture Notes in Computer Science*. Vol. 14214. Springer, Cham, 2023. Available at: https://doi.org/10.1007/978-3-031-43111-1_13.
2. Lucet E., Grand C., Sallé D. and Bidaud P. Dynamic yaw and velocity control of the 6WD skid-steering mobile robot RobuROC6 using sliding mode technique, *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 2009*, pp. 4220-4225. DOI: 10.1109/IROS.2009.5354373.
3. Hausamann P., Sinnott C.B., Daumer M. et al. Evaluation of the Intel RealSense T265 for tracking natural human head motion, *Sci. Rep.*, 2021, 11, 12486. Available at: <https://doi.org/10.1038/s41598-021-91861-5>.
4. Kazakov K.A., Semenov V.A. Obzor sovremennykh metodov planirovaniya puti [Review of modern path planning methods], *Trudy ISP RAN [Proceedings of ISP RAS]*, 2016, Vol. 28 (4), pp. 241-294.
5. Zhou R., Hansen E.A. Memory-Bounded $\{A^*\}$ Graph Search, *The Florida AI Research Society Conference – FLAIRS*, 2002, pp. 203-209.
6. Hart P.E., Nilsson N.J., Raphael B.A. Formal Basis for the Heuristic Determination of Minimum Cost Paths, *IEEE Transactions on Systems Science and Cybernetics*, 1968, Vol. 2, pp. 100-107.
7. Stentz A. Optimal and efficient path planning for partially known environments, *In Intelligent Unmanned Ground Vehicles*. Springer, Boston, MA, USA, 1997, pp. 203-220.
8. Murphy R.R. Introduction to AI Robotics. 1st ed. MIT Press: Cambridge, MA, USA, 2000.
9. LaValle S.M., Kuffner J.J. Rapidly-exploring random trees: Progress and prospects, *2000 Workshop on the Algorithmic Foundations of Robotics*, 2000, pp. 293-308.
10. Kavraki L.E., Svestka P., Latombe J.-C., Overmars M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Transactions on Robotics and Automation*, 1996, Vol. 12 (4), pp. 566-580.
11. Pshikhopov V., Medvedev M., Kostjukov V., Houssein F., Kadhim A. Trajectory planning algorithms in two-dimensional environment with obstacles, *Informatics and Automation*, 2022, 21:3, pp. 459-492.
12. Tianran Peng, Jun Qian, Bin Zi, Jiakui Liu, Xingwei Wang, Mechanical Design and Control System of an Omni-directional Mobile Robot for Material Conveying, *Procedia CIRP*, 2016, Vol. 56, pp. 412-415. ISSN 2212-8271. Available at: <https://doi.org/10.1016/j.procir.2016.10.068>.
13. Pshikhopov V.Kh. (Ed.), Beloglazov D., Finaev V., Guzik V., Kosenko E., Krukhmaliev V., Medvedev M., Pereverzev V., Pyavchenko A., Saprykin R., Shapovalov I., Soloviev V. Path Planning for Vehicles Operating in Uncertain 2D Environments. Elsevier, Butterworth-Heinemann, 2017, 312 p. ISBN: 9780128123058.
14. Macenski T. Foote, Gerkey B., Lalancette C. Woodall W. Robot Operating System 2: Design, architecture, and uses in the wild, *Science Robotics*, May 2022, Vol. 7.

15. Pshikhopov V.Kh., Medvedev M.Yu., Gaiduk A.R., Fedorenko R.V., Krukhmalev V.A., Gurenko B.V. Position-Trajectory Control System for Unmanned Robotic Airship, *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2014, pp. 8953-8958.
16. Pshikhopov V., Medvedev M. Multi-Loop Adaptive Control of Mobile Objects in Solving Trajectory Tracking Tasks, *Automation and Remote Control*, 2020, Vol. 81 (11), pp. 2078-2093.
17. Bogoslavskiy I., & Stachniss C. Efficient Online Segmentation for Sparse 3D Laser Scans, *ISPRS – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 2017, pp. 1-12. Retrieved from <https://link.springer.com/article/10.1007%2Fs41064-016-0003-y>.
18. Liu C, Xie S, Ma X, Huang Y, Sui X, Guo N, Yang F, Yang X. A Hierarchical Clustering Obstacle Detection Method Applied to RGB-D Cameras, *Electronics*, 2023, 12 (10), 2316. Available at: <https://doi.org/10.3390/electronics12102316>.
19. Kang D, Wong A, Lee B, Kim J. Real-Time Semantic Segmentation of 3D Point Cloud for Autonomous Driving, *Electronics*, 2021, 10 (16), 1960. Available at: <https://doi.org/10.3390/electronics10161960>.
20. Phalgun Chintala, Rolf Dornberger, and Thomas Hanne. Robotic Path Planning by Q Learning and a Performance Comparison with Classical Path Finding Algorithms, *International Journal of Mechanical Engineering and Robotics Research*, June 2022, Vol. 11, No. 6, pp. 373-378. DOI: 10.18178/ijmerr.11.6.373-378.

Статью рекомендовал к опубликованию д.т.н., профессор В.В. Курейчик.

Бросалин Дмитрий Олегович – АО "НКБ РИСУ"; e-mail: ddimonxt@mail.ru; г. Таганрог, Россия; тел.: +79281615949; м.н.с.

Гуренко Борис Викторович – e-mail: boris.gurenko@gmail.com; тел.: +79281687212; к.т.н.; с.н.с.

Медведев Михаил Юрьевич – НИИ робототехники и процессов управления Южного федерального университета; e-mail: medvmihal@sfedu.ru; г. Таганрог, Россия; тел.: 88634371694; д.т.н.; в.н.с.

Brosalin Dmitry Olegovich – Joint stock Company «Scientific-Design bureau of Robotics and Control Systems»; e-mail: ddimonxt@mail.ru; Taganrog, Russia; phone: +79281615949; junior researcher.

Gurenko Boris Viktorovich – e-mail: boris.gurenko@gmail.com; phone: +79281687212; cand. of eng. sc.; senior researcher.

Medvedev Mikhail Yuryevich – Research Institute of Robotics and Control Processes, Southern Federal University; e-mail: medvmihal@sfedu.ru; Taganrog, Russia; phone: +78634371694; dr. of eng. sc.; leading researcher.

УДК 519.712.2

DOI 10.18522/2311-3103-2023-6-24-34

Л.А. Гладков, Н.В. Гладкова

АЛГОРИТМ СИНТЕЗА КОМБИНАЦИОННЫХ ЛОГИЧЕСКИХ СХЕМ НА ОСНОВЕ ЭВОЛЮЦИОННОГО ПОДХОДА

Появление новых технологий изготовления компонентов цифровых электронных устройств привело к необходимости повышения эффективности методов автоматизированного проектирования. Увеличение требований к элементам вызывает рост размерности решаемых задач. Для решения задач, которые ранее невозможно было автоматизировать разрабатываются новые методы и программные приложения. Перед специалистами стоит задача разработки фундаментальных принципов построения систем проектирования нового поколения. Разработка устройств обладающих такими характеристиками как: надежность, живучесть, автоматическое устранение повреждений, является актуальной задачей. В данной работе предлагается подход к решению задачи синтеза комбинационных