

**Ключевые слова:**

мобильные роботы, алгоритмы навигации, алгоритм ORCA, функционально-воксельный метод (ФВМ), многоагентные системы

# МОДЕЛИРОВАНИЕ АЛГОРИТМОВ УПРАВЛЕНИЯ ГРУППАМИ МОБИЛЬНЫХ РОБОТОВ СРЕДСТВАМИ ФУНКЦИОНАЛЬНО- ВОКСЕЛЬНОГО МЕТОДА

**Алексей ТОЛОК, Михаил ЛОКТЕВ, Анастасия СЫЧЕВА,  
Полина ХАРЛАНОВА, Людмила СИЗОВА**

Рассматривается пример описания одного из алгоритмов децентрализованной навигации больших групп агентов ORCA средствами функционально-воксельного метода, а также случай построения области допустимых скоростей для предотвращения столкновений для двух мобильных роботов. На основе графических образ-моделей разработан подход к построению области допустимых скоростей роботов на плоскости.

## ВВЕДЕНИЕ

Принципы, решающие задачи навигации в многоагентных системах, сопоставимы со взаимодействием социальных общественных групп. Их тоже можно делить на централизованные и децентрализованные [1]. При централизованном подходе действия агентов зачастую рассматриваются с точки зрения выполнения совместной цели общими усилиями. В этом случае агенты собирают в группы, зачастую выбирая лидера, и решают задачу совместным взаимодействием команды. Однако, существует ряд ситуаций, при которых требуется децентрализованная система планирования действий агентов [2], в том числе и система децентрализованной навигации. В таких системах у каждого агента присутствует собственная цель (например, агенты службы доставки). Алгоритм решения такой задачи предполагает учет обстановки по всей сцене и принимает самостоятельное решение относительно поведения остальных агентов. И так происходит с каждым отдельно взятым агентом. Такой алгоритм обязательно применяет оптимизацию решения относительно поставленной цели и множества возникающих ограничений на его пути.

Подобный принцип используется при применении стратегии стайного поведения [3]. В стае достигается более плотное расположение агентов, в результате этого группа занимает меньший объем пространства. Группа по своей форме может быть вытянута вдоль траектории движения или иметь неправильную форму, но члены группы могут вступать в нее или выходить в любой момент, в зависимости от степени совпадения направления движения группы и собственной цели члена группы.

## ДЕЦЕНТРАЛИЗОВАННЫЙ АЛГОРИТМ НАВИГАЦИИ ORCA

Одним из представителей алгоритмов подобного класса является алгоритм ORCA. Он базируется на построении линейных полупространств относительно каждого агента, формирующих на пересечении область возможных решений для применения оптимизационных принципов линейного математического программирования. В работе рассмотрен подход к построению алгоритма ORCA средствами функ-

ционально-воксельного моделирования на основе аналитических принципов  $R$ -функционального построения области сложной геометрии [4]. На этом этапе открываются новые задачи по эффективной реализации такой модели на программно-техническом уровне.

Алгоритм ORCA впервые был представлен в 2011 году [5] как принцип децентрализованной навигации для больших групп агентов, который обеспечивает достаточное условие для избегания столкновений между агентами с индивидуальными целями. Более того, каждый агент принимает на себя минимум половину «ответственности» за избегание столкновения с другим агентом [1]. Данный принцип базируется на более ранней работе [6], посвященной формированию области столкновений – скоростного препятствия VO (Velocity Obstacles). Скоростным препятствием является область, образованная векторами скоростей и приводящая к столкновению агентов в заданном промежутке времени.

Последующие научные исследования других авторских коллективов использовали алгоритм ORCA при решении различных прикладных задач. Так, в работе [7] рассматривается применение различных алгоритмов навигации для взаимного локального уклонения виртуальных агентов в видеоиграх. Симуляция автомобильного движения на перекрестке на основе алгоритма ORCA рассматривается в работе [8]. Помимо этого, разрабатывались собственные модифицированные алгоритмы. В работе [9] представлен новый безаварийный подход к навигации неголономных роботов на основе ORCA, учитывающий особенности кинематики исследуемых роботов.

### ФУНКЦИОНАЛЬНО-ВОКСЕЛЬНОЕ МОДЕЛИРОВАНИЕ В ЗАДАЧАХ НАВИГАЦИИ

Представленные работы основаны на значительном количестве сложных геометрических расчетов на каждом этапе работы алгоритма ORCA. Более того, для большой группы агентов требуется производить взаимный расчет возможных скоростей смещения для каждой пары роботов с последующим вычислением на их основе новой скорости с помощью линейного программирования.

Упрощение математических операций традиционно считается преимуществом функционально-воксельного моделирования, когда часть вычислений заменяется формированием графического образа с заложенной в нем всей необходимой локальной информацией. Метод функционально-воксельного моделирования (ФВМ), описанный в монографии А.В. Толока [10], позволяет организовать компью-

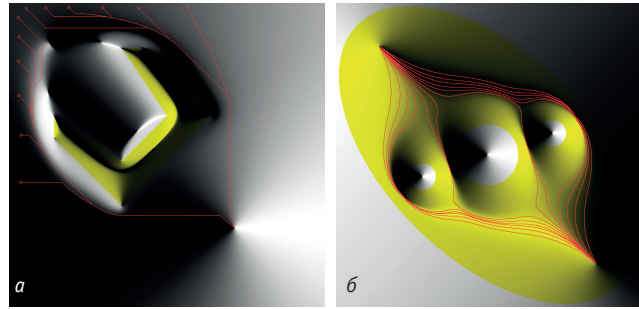


Рис. 1. Решение задачи поиска пути: а – решение задачи градиентного спуска, б – вариационная трассировка

терное представление области функции соразмерными воксельными образами на основе разложения функции на локальные геометрические характеристики в каждой точке воксельного пространства. Данный подход основан на графическом представлении данных об объекте, которое использует аналитический способ описания моделей. При этом функционально-воксельное представление области функции значительно упрощает дальнейшую компьютерную обработку применительно к задачам компьютерного геометрического моделирования с сохранением преимуществ полноты представления геометрии объекта.

Подобные исследования успешно проводились при реализации задач локальной оптимизации пути к цели с обходом препятствий [11]. Также следует отметить успехи в разработке функционально-воксельного моделирования задач математического программирования [12]. На основе предложенного подхода уже были реализованы различные постановки задачи поиска пути [13–14], представленные на рис. 1.

На рис. 1а представлен пример градиентного спуска, как способа определения траектории к цели из любой точки пространства. На рис. 1б продемонстрирован пример так называемой вариативной трассировки, когда на основе анализа функционального пространства определяются характерные элементы рельефа функции (хребты-лощины), по которым и будут проходить возможные маршруты, плавно огибающие препятствия.

Функционально-воксельное представление области столкновения может быть рассчитано предварительно, а в ходе работы алгоритма будут применяться лишь простые вычисления над полученными локальными геометрическими характеристиками модели, что значительно упрощает вычислительный процесс. Такую модель можно построить единожды и далее использовать как графический шаблон при моделировании отношений двух роботов, заданных в пространстве.

**АНАЛИТИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ СКОРОСТНОГО ПРЯПЯТСТВИЯ  $VO_{A|B}^\tau$**

Рассмотрим конкретный пример взаимного расположения роботов **A** и **B** со своими эффективными радиусами  $R_A$  и  $R_B$ , равными единице, где  $P_B(-2, 3)$  и  $P_A(2, -3)$  (рис. 2). Привяжем положение роботов к системе скоростных координат. Для этого перенесем начало координат в точку  $P_A$ , тогда изменит свои координаты  $P_B(-4, 6)$ . Начальные скорости обоих роботов определяются векторами  $v_A(v_{Ax}=1,5, v_{Ay}=1,0)$  и  $v_B(v_{Bx}=3,0, v_{By}=-1,5)$ , а разность этих векторов, образующая вектор сближения роботов, определяется вектором  $v_{(A-B)}(v_{(A-B)x}=-1,5, v_{(A-B)y}=2,5)$ , как показано на рис. 3.

Разность координат центров агентов даст новую точку  $(P_A - P_B)$ , которая в данном случае будет совпадать с положением агента **B**. В этой точке построим новую окружность с общим радиусом двух роботов  $(R_A + R_B)$ . Прямые  $Pr_1(x, y)$  и  $Pr_2(x, y)$  определяют полупространства  $(R_A + R_B)$  для прямых – касательных к окружности с радиусом исходящих из начала координат. Чтобы описать их местоположение, достаточно вычислить углы  $\alpha = \arctg(\frac{y_B}{x_B})$  и

$$\beta = \arccos\left(\frac{\sqrt{(x_B^2 + y_B^2) - (R_A + R_B)^2}}{\sqrt{x_B^2 + y_B^2}}\right).$$

Уравнения прямых  $Pr_1(x, y)$  и  $Pr_2(x, y)$  примут вид:

$$Pr_1(x, y) = y - \operatorname{tg}(\alpha + \beta)x, \tag{1}$$

$$Pr_2(x, y) = \operatorname{tg}(\alpha - \beta)x - y. \tag{2}$$

Образуемая между ними область определяет зону возможного столкновения при попадании

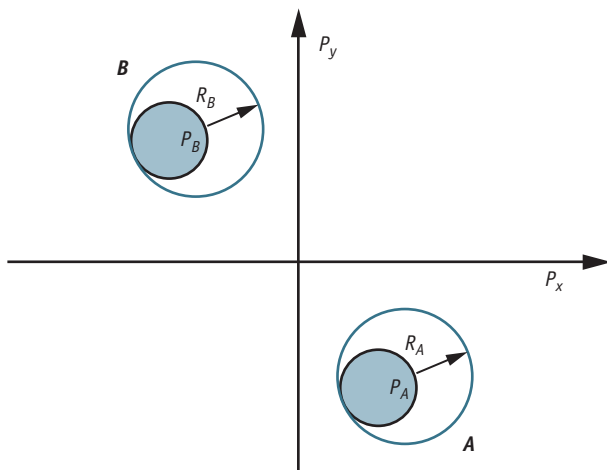


Рис. 2. Начальное положение роботов

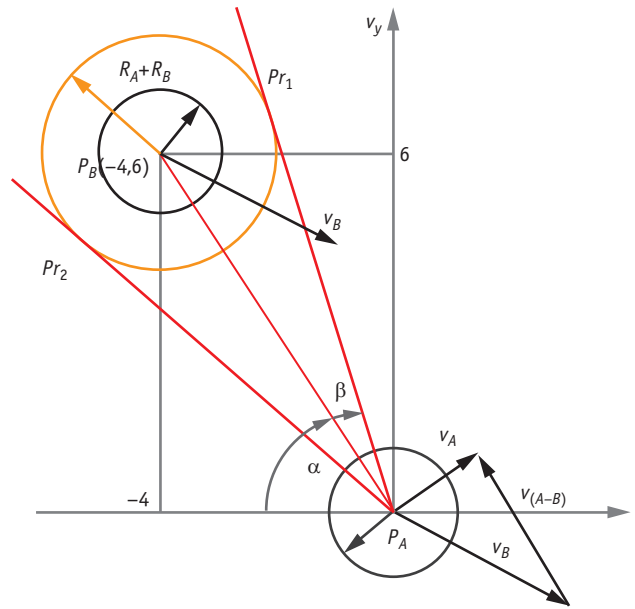


Рис. 3. Описание сцены с двумя агентами **A** и **B** с заданными векторами скоростей

в нее вектора разности скоростей  $v_{(A-B)}$ . Однако следует учитывать динамику происходящего процесса. Для этого стоит ввести параметр временного промежутка взаимодействия роботов  $\tau$ , позволяющий уточнить положение критической зоны с учетом расстояния между двумя роботами. Введение такого параметра позволяет уточнить зону предупреждения столкновения. Введем значение  $\tau = 2,0$ , как промежуток заведомого (запасного) времени, для которого требуется определить область возможных столкновений. Положение новой окружности, определяющей начало опасной зоны, задается точкой центра  $(P_A - P_B)/\tau$  и радиусом  $(R_A + R_B)/\tau$  (рис. 4). Полученная окружность будет определять ближайшую к роботу **A** границу зоны столкновения с роботом **B** на промежутке времени  $\tau$ .

Для построения алгоритма определения принадлежности конечной точки вектора разности скоростей  $v_{(A-B)}$  области полученной зоны столкновения  $VO_{A|B}^\tau$  необходимо эту зону разбить на два участка. Первый участок контролирует попадание вектора  $v_{(A-B)}$  в зону  $VO_{A|B}^\tau$ , ограниченную прямыми:  $Pr_1, Pr_2, Pr_3$  и  $Pr_4$ , где  $Pr_3$  и  $Pr_4$  – перпендикуляры, опущенные на прямые  $Pr_1, Pr_2$  из центра окружности с радиусом  $\frac{(R_A + R_B)}{\tau}$

(рис. 4). Уравнения этих прямых описываются так:

$$Pr_1(x, y) = \left(y - \frac{y_B}{\tau}\right) - \operatorname{tg}\left(\frac{\pi}{2} + \alpha - \beta\right)\left(x - \frac{x_B}{\tau}\right), \tag{3}$$

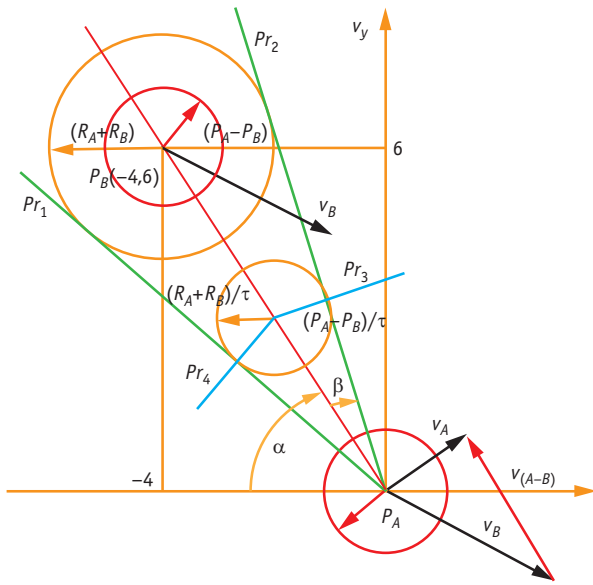


Рис. 4. Формирование области столкновений  $VO_{A|B}^{\tau}$

$$Pr_4(x,y) = (y - \frac{y_B}{\tau}) - tg(\frac{\pi}{2} + \alpha + \beta)(x - \frac{x_B}{\tau}). \quad (4)$$

На начальном этапе алгоритма (условие 1) вычисляем значения  $Pr_1(v_{(A-B)x}, v_{(A-B)y})$  и  $Pr_2(v_{(A-B)x}, v_{(A-B)y})$  с помощью уравнений (1) и (2). Если оба значения положительны, то точка с координатами  $(v_{(A-B)x}, v_{(A-B)y})$  располагается между прямыми  $Pr_1$  и  $Pr_2$ , а значит, подлежит дальнейшему изучению. В противном случае эта точка не попадает в зону  $VO_{A|B}^{\tau}$  и робот **A** может двигаться относительно робота **B** с той же скоростью и в том же направлении.

В случае, если точка  $(v_{(A-B)x}, v_{(A-B)y})$  оказалась между этими прямыми, то следует уточнить ее положение проверкой на положительный знак значения любого из выражений (условие 2):  $Pr_3(v_{(A-B)x}, v_{(A-B)y}) \geq 0$  или  $Pr_4(v_{(A-B)x}, v_{(A-B)y}) \geq 0$ . Выполнения второго условия достаточно, чтобы приступить к определению направления нормали  $\vec{n}$  к ближайшей границе искомой области  $ORCA_{A|B}$  для вычисления вектора корректировки скорости робота **A**, требуемого для обеспечения выхода из зоны  $VO_{A|B}^{\tau}$ . Обозначим такой вектор корректировки  $\vec{w}$ . Направление  $\vec{w}$  совпадает с направлением  $\vec{n}$ , а длина определяется расстоянием до ближайшей границы зоны  $VO_{A|B}^{\tau}$ . Ближайшая граница в этом случае рассматривается как минимальное расстояние до прямых  $Pr_1$  и  $Pr_2$ :

$$dx_1 = \frac{v_{(A-B)x} + tg(\alpha + \beta)v_{(A-B)y}}{(tg(\alpha + \beta))^2 + 1}, \quad (5)$$

$$dy_1 = tg(\alpha + \beta) \frac{v_{(A-B)x} + tg(\alpha + \beta)v_{(A-B)y}}{(tg(\alpha + \beta))^2 + 1} - v_{(A-B)y}, \quad (6)$$

$$d_1 = \sqrt{dx_1^2 + dy_1^2}, \quad (7)$$

$$dx_2 = \frac{v_{(A-B)x} + tg(\alpha - \beta)v_{(A-B)y}}{(tg(\alpha - \beta))^2 + 1}, \quad (8)$$

$$dy_2 = tg(\alpha - \beta) \frac{v_{(A-B)x} + tg(\alpha - \beta)v_{(A-B)y}}{(tg(\alpha - \beta))^2 + 1} - v_{(A-B)y}, \quad (9)$$

$$d_2 = \sqrt{dx_2^2 + dy_2^2}, \quad (10)$$

$$d(dx, dy) = \min(d_1(dx_1, dy_1), d_2(dx_2, dy_2)). \quad (11)$$

Область  $ORCA_{A|B}$  здесь определяется уравнением прямой, параллельной выбранной границе (например,  $Pr_1(x, y)$ ), перенесенной в точку с координатами  $(v_{Ax} + dx, v_{Ay} + dy)$ :

$$ORCA_{A|B}(x, y) = (y - (v_{Ay} + dy)) - tg(\alpha + \beta)(x - (v_{Ax} + dx)). \quad (12)$$

Если после выполнения первого условия не выполняется второе, следует продолжить рассматривать другие условия, позволяющие уточнить отношение точки  $(v_{(A-B)x}, v_{(A-B)y})$  к области  $VO_{A|B}^{\tau}$ .

Уравнение  $Circle_{\tau}(x, y)$  представляет собой уравнение окружности с радиусом  $(R_A + R_B)/\tau$  и центром в точке  $(P_A + P_B)/\tau$ , а значит, описывается выражением:

$$Circle_{\tau}(x, y) = (\frac{R_A}{\tau} + \frac{R_B}{\tau})^2 - (x + \frac{x_B}{\tau})^2 - (y + \frac{y_B}{\tau})^2. \quad (13)$$

Часть этой окружности, не входящая в рассмотренную ранее область  $VO_{A|B}^{\tau}$ , продолжает описывать ее границу. Поэтому выражение  $Circle_{\tau}(v_{(A-B)x}, v_{(A-B)y}) \geq 0$  обеспечивает третье условие принадлежности области  $VO_{A|B}^{\tau}$  как проверку принадлежности точки  $(v_{(A-B)x}, v_{(A-B)y})$  области окружности  $Circle_{\tau}(x, y)$ .

При выполнении третьего условия ближайшей точкой на границе для точки  $(v_{(A-B)x}, v_{(A-B)y})$  является точка на окружности с радиусом  $(R_A + R_B)/\tau$ . Чтобы определить направление к такой точке как направление нормали, достаточно построить прямую, ортогональную прямой, проходящей через точки  $(v_{(A-B)x}, v_{(A-B)y})$  и  $(x_B/\tau, y_B/\tau)$ , и перенести ее в точку вектора  $v_A$ , смещенную на координаты  $(w \cos \gamma, w \sin \gamma)$ , где:

$$\gamma = \arctg \left( \frac{\frac{y_B}{\tau} - v_{(A-B)y}}{\frac{x_B}{\tau} - v_{(A-B)x}} \right),$$

$$w = \frac{(R_A + R_B)}{\tau} - \sqrt{\left(\frac{x_B}{\tau} - v_{(A-B)x}\right)^2 + \left(\frac{y_B}{\tau} - v_{(A-B)y}\right)^2}.$$

В результате имеем:

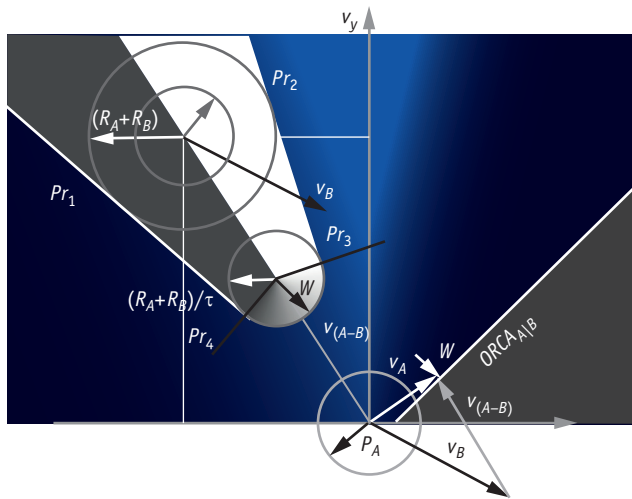


Рис. 5. Построение графического M-образа на основе уравнения (15) и формирования полуплоскости

$$ORCA_{A|B} = \text{tg}\left(\frac{\pi}{2} + \gamma\right) (x - (v_{Ax} + w \cos \gamma)) - (y - (v_{Ay} + w \sin \gamma)). \quad (14)$$

Как видно из рассуждений, представленный алгоритм позволяет описать ситуацию взаимной корректировки скорости для робота по отношению к одному из встречных роботов. Полная картина определяется пересечением всех областей ORCA, построенных для каждого из ближайших роботов сцены, влияющих на корректировку окончательного направления и скорости движения. Полученная область возможных решений требует применения средств оптимизации для получения единственно верного вектора скорости. Любая оптимизационная постановка имеет целевую функцию, которая выражается основным направлением к цели. Такая постановка вполне разрешима средствами линейного математического программирования [12].

Для проверки реализации расчета области на конкретном заданном примере была промоделирована область  $VO_{A|B}^T$  в системе функционально-воксельного моделирования RANOK. Для этого к описанию областей был применен R-функциональный аппарат теоретико-множественных операций над функциями [15], позволивший выделить области с положительным знаком значений:

$$VO_{A|B} = Pr_1 \wedge Pr_2 \wedge (Pr_3 \vee Pr_4) \vee Circle_{\tau}. \quad (15)$$

На рис. 5 демонстрируется объединение в пространстве скоростей области  $VO_{A|B}^T$  и области  $ORCA_{A|B}$ , наглядно демонстрирующий результат вычислений поставленной задачи.

На рис. 6 представлены графические образ-модели (M-образы), описывающие локальное уравнение

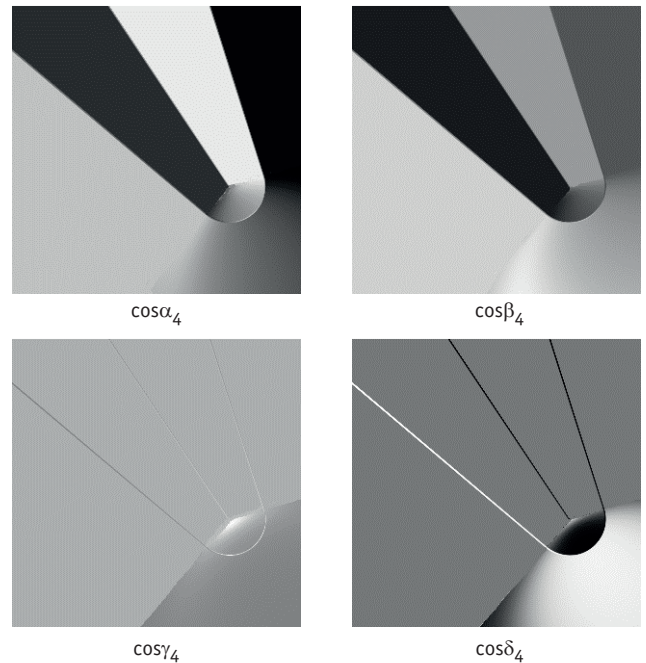


Рис. 6. M-образы, описывающие локальное уравнение в точках области столкновений

в точках области столкновений. M-образы являются одним из ключевых инструментов графического анализа метода функционально-воксельного моделирования [16]. Каждый из этих образов характеризует компоненту нормали, построенную в 4-мерном пространстве. Чтобы описать такое локальное уравнение, необходимо нормальный вектор в каждой точке рассматриваемого пространства представить в четвертом измерении.

Для получения угла наклона области  $ORCA_{A|B}^T$  достаточно в точке приложении вектора разности скоростей рассмотреть компоненту двумерного нормального вектора на оси X (т.е.  $\cos \alpha_2$ ), перенормированную по формуле:

$$\cos \alpha_4 x + \cos \beta_4 y + \cos \gamma_4 z = \cos \delta_4, \quad (16)$$

$$\cos \alpha_2 = \frac{\cos \alpha_4}{\sqrt{\cos \alpha_4^2 + \cos \beta_4^2}} = \cos \alpha_{ORCA}. \quad (17)$$

### ЗАКЛЮЧЕНИЕ

Проведенные исследования доказали возможность рассматривать задачу управления группой автономных мобильных роботов (или других агентов) с помощью функционально-воксельного метода. Полученная функционально-воксельная модель выступает в роли шаблона, которая несет полную информацию о локальных геометрических характеристиках в каждой точке простран-

ства. Эта информация позволяет без труда определять необходимое направление области допустимых скоростей  $ORCA_{A|B}^c$ . Система управления мобильного робота может использовать данный шаблон для определения направления движения, что позволит отказаться от ряда численных вычислений по сравнению с классическим описанием алгоритма. Применение представленных аналитических подходов делает возможным рассмотрение подобных задач без привязки к размерности пространства.

## ЛИТЕРАТУРА

- Дергачев С.А.** Экспериментальное исследование реактивного алгоритма навигации для групп агентов — ORCA // Сборник научных трудов 17-й национальной конференции по искусственному интеллекту с международным участием (КИИ-2019), г. Ульяновск. Т. 1. 21–25 октября 2019. С. 102.
- Яковлев К.С., Баскин Е.С., Андрейчук А.А.** Метод автоматического планирования совокупности траекторий для навигации беспилотных транспортных средств // Управление большими системами. Выпуск. — М.: ИПУ РАН, 2015.
- Харланова П.М.** Исследование применения стратегии стайного поведения в ограниченных пространствах // Труды 16-й Всероссийской школы-конференция молодых ученых «Управление большими системами» (УБС'2019, Тамбов). Тамбов: Издательский центр ФГБОУ ВО «ТГТУ», 2019. С. 175–178.
- Тимофеев А.В., Юсупов Р.М.** Принципы построения интегрированных систем мультиагентной навигации и интеллектуального управления мехатронными роботами // Information Technologies & Knowledge. 2011. Т. 5. № 3.
- Jur van den Berg, Stephen J. Guy, Ming Lin, Dinesh Manocha.** Reciprocal n-body Collision Avoidance // Robotics Research: The 14th International Symposium ISRR, Springer Tracts in Advanced Robotics, v. 70, Springer-Verlag, May 2011, pp. 3–19.
- Fiorini P., Shiller Z.** Motion planning in dynamic environments using Velocity Obstacles. Int. Journal of Robotics Research 17(7), pp. 760–772, 1998.
- Lake A.T. et al.** Reciprocal collision avoidance and navigation for video games. 2012.
- Schaefer M.** Planning and coordination in driving simulation // Acta Polytechnica CTU Proceedings. 2015. Т. 2. № 2. С. 40–44.
- Mao R., Gao H., Guo L.** A Novel Collision-Free Navigation Approach for Multiple Nonholonomic Robots Based on ORCA and Linear MPC // Mathematical Problems in Engineering. 2020. Т. 2020.
- Толок А.В.** Функционально-воксельный метод в компьютерном моделировании / Под ред. акад. РАН С.Н. Васильева. — М.: ФИЗМАТЛИТ, 2016. 112 с. ISBN: 978-5-9221-1680-0.
- Васильев С.Н., Локтев М.А., Толок А.В., Толок Н.Б., Ульянов С.А.** К планированию маршрутов в 3D-среде с многовариантной моделью // Труды СПИИРАН, Вып. 2 (45). ISSN 2078-9181. — СПб, 2016. С. 5–25.
- Толок А.В., Толок Н.Б.** Mathematical Programming Problems Solving by Functional Voxel Method // Automation and Remote Control. 2018. V. 79. № 9. PP. 1703–1712.
- Локтев М.А.** Особенности применения функционально-воксельного моделирования в задачах поиска пути с препятствиями // Информационные технологии в проектировании и производстве. 2016. № 1. С. 45–49.
- Григорьев С.Н., Толок А.В., Толок Н.Б.** Построение градиентного алгоритма локального перебора точек на основе метода функционально-воксельного моделирования // Программирование. 2017. № 5. С. 32–38.
- Рвачев В.Л.** Теория R-функций и некоторые ее приложения. — Киев: Наукова думка, 1982. 552 с.
- Толок А.В.** Графические образы-модели в информационных технологиях // Прикладная информатика. 2009. № 4. С. 31–40.

---

### ТОЛОК Алексей Вячеславович –

*доктор технических наук, главный научный сотрудник лаборатории компьютерной графики Института проблем управления им. В.А. Трапезникова РАН*

### ЛОКТЕВ Михаил Александрович –

*кандидат технических наук, старший научный сотрудник лаборатории компьютерной графики Института проблем управления им. В.А. Трапезникова РАН*

### СЫЧЕВА Анастасия Антоновна –

*младший научный сотрудник лаборатории компьютерной графики Института проблем управления им. В.А. Трапезникова РАН*

### ХАРЛАНОВА Полина Михайловна –

*младший научный сотрудник лаборатории Систем логического управления Института проблем управления им. В.А. Трапезникова РАН*

### СИЗОВА Людмила Николаевна –

*ведущий инженер-программист лаборатории компьютерной графики Института проблем управления им. В.А. Трапезникова РАН*