ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 123 за 2020 г.

ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

В.Д. Сенотов, А.П. Алисейчик, Е.В. Павловский, А.В. Подопросветов, И.А. Орлов

Алгоритмы стайного децентрализованного управления движением группы роботов с дифференциальным приводом

Рекомендуемая форма библиографической ссылки: Алгоритмы стайного децентрализованного управления движением группы роботов с дифференциальным приводом / В.Д. Сенотов [и др.] // Препринты ИПМ им. М.В.Келдыша. 2020. № 123. 39 с. https://doi.org/10.20948/prepr-2020-123
https://library.keldysh.ru/preprint.asp?id=2020-123

Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук

В.Д. Сенотов, А.П. Алисейчик, Е.В. Павловский,
А.В. Подопросветов, И.А. Орлов

Алгоритмы стайного децентрализованного управления движением группы роботов с дифференциальным приводом

Москва – 2020

УДК 531.38

Сенотов В.Д., Алисейчик А.П., Павловский Е.В., Подопросветов А.В., Орлов И.А. **Алгоритмы стайного децентрализованного управления движением группы роботов с дифференциальным приводом**

Данная работа предлагает метод настройки алгоритма управления группой роботов, движущихся по плоскости, который, с одной стороны, оставляет в основе алгоритма правила локального взаимодействия, с другой стороны, оптимизирует движение роботов относительно критериев, оценивающих поведение стаи в целом.

Ключевые слова: стая роботов, дифференциальный привод, моделирование движения, поведенческое управление, децентрализованные алгоритмы, искусственные силы.

Senotov V.D., Aliseychik A.P., Pavlovsky E.V., Podoprosvetov A.V., Orlov I.A. **Algorithms for swarm decentralized motion control of group of robots with a differential drive**

This work proposes a method for tuning the algorithm for motion control of group of robots moving on a plane, which, on the one hand, leaves the rules of local interaction at the basis of the algorithm, and on the other hand, optimizes the motion of robots with respect to criteria (estimates) evaluating the behavior of the swarm as a whole.

Key words: swarm robotics, differential drive, motion simulation, behavioral control, decentralized algorithms, artificial forces.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (РФФИ: 19-01-00123, 19-08-01159, 18-08-01441).

Введение

Управление движением группы роботов — обширная область робототехники, содержащая множество открытых вопросов. Эта область привлекает много внимания в связи с развитием технологий беспроводной связи, большой доступностью мобильных роботов и преимуществами, которыми обладают группы примитивных роботов при решении сложных задач (это, например, устойчивость к авариям, возможность решать несколько задач, масштабируемость).

Одним из открытых вопросов является настройка алгоритмов управления роботами для оптимального с точки зрения оператора поведения. Алгоритмы управления группами роботов чаще всего содержат локальные правила взаимодействия роботов между собой и внешней средой, а эмерджентное поведение, оказывающееся в результате применения этих правил, не имеет формального описания. Тем самым, у параметров алгоритма отсутствует явная интерпретация относительно поведения всей группы, что затрудняет настройку алгоритма.

Групповая робототехника развивается уже несколько десятилетий, и в течение этих лет были разработаны различные методы управления движением групп роботов. Один из первых подходов описан в работе К. Рейнолдса [1]: автор выделил свойства движения группы, которые должны наблюдаться, чтобы ее движение считалось целостным, и выделил локальные правила, которым должны подчиняться агенты в группе для получения этих свойств. Движение с такими свойствами называют стайным, а этот подход наряду с многими похожими называют биологически инспирированным. В работе [2] также предложен подход, основанный на локальных правилах, однако в нем воспроизводятся поведения, похожие на состояние вещества (отсюда класс физически инспирированных алгоритмов). В работе Матарич [3] строится алгоритм, явно решающий проблемы движения в группе (в первую очередь, уклонение от столкновений).

Важно отметить, что до сих пор есть очень мало реальных проектов с использованием групп роботов. Из известных применений есть 2 проекта для наблюдения за акваториями: в [4] создали группу гомогенных надводных роботов для измерения температуры по всей площади акватории; в проекте CoCoRo [5] группа разных по функциональности подводных роботов проводила мониторинг акватории и получала образцы морского дна. В обеих работах реализованы правила стайного поведения.

Набор задач для групп роботов не ограничивается только задачами исследования водных пространств. В обзорных работах [6], [7] были формали-

зованы задачи, которые могут решать группы роботов: сбор в одной точке, образование рисунка, цепи, самостоятельная сборка, сбор ресурсов, координированное движение, обход препятствий, развертывание, коллективное исследование территории, коллективный транспорт, распределение задач. В более современном обзоре [8] добавляют задачи поиска цели, планирования движения, распределенной оценки измеряемой величины и отмечают, какие из подходов построения управления были использованы в той или иной задаче. Касательно сфер, где встречаются такие задачи, в работе [9] анализируется возможное применение групп роботов в мониторинге состояния конструкций. Использование групп роботов планируется в исследовании зданий при чрезвычайных ситуациях или захватах зданий. В недавнем обзоре [10] индустрии логистики прогнозируется использование модульных роботов.

Стаи роботов являются популярной темой исследований групповой робототехники. Стаи должны совершать безопасное движение группой при выполнении некоторой задачи, при этом поведение не должно зависеть от количества роботов в стае и быть устойчивым к отказам роботов. Предложенный в упомянутой работе К. Рейнолдса [1] алгоритм стайного движения развился в направление алгоритмов, основанных на виртуальных силах (или потенциалах) [7]. Виртуальные силы могут воспроизводить реальные физические силы между молекулами [11] (принимая робота за материальную точку), или отражать правила взаимодействия между роботами и препятствиями [12] и включать в себя случайное передвижение [13]. Подход позволяет изменять в алгоритме отдельные правила взаимодействия в зависимости от задачи и удовлетворять требованиям стайного поведения.

В стандартном подходе к управлению стаями, например, [14], была замечена зависимость поведения стаи от соотношения виртуальных сил. Однако эту зависимость нельзя выразить явно, что стало проблемой в вопросе настройки алгоритма для желаемого поведения. В работах [15], [16] отмечают проблемы в определении состояния и динамики стаи оператором и возникающие в связи с этим трудности управления стаей.

Для решения этих проблем предлагаются алгоритмы автоматической оптимизации движения роботов. Для достижения оптимального движения к целевой точке в [17] предложена оптимизация методом роя частиц. В [18] и [19] используются конечные автоматы для оптимизации стаи по энергии в задаче сбора ресурсов. В [20] используются искусственные нейронные сети для настройки параметров алгоритма. Эволюционные алгоритмы использованы в [21] и [22]. Оптимизация по функционалу над траекторией робота используется в работах [23], [24], [25], [26]. Последний метод отличается в выгодную сторону от предыдущих, т.к. дает решение сразу, без дополнитель-

ной симуляции стаи.

1. Постановка задачи

В данной работе исследуется управление стаей роботов, а именно ставится задача построения децентрализованного алгоритма управления перемещением гомогенной группы колесных роботов при выполнении одного из 3 типов заданий. Помимо этого, необходимо представить методику количественной оценки движения стаи роботов, и алгоритм должен быть настраиваемым для оптимального управления по этим оценкам.

Поясним постановку задачи. Разделим ее на четыре логические части: модель группы роботов, формализация заданий для группы роботов, построение алгоритма управления и оптимальная настройка алгоритма.

В первой части необходимо описать требования к группе роботов. Это должна быть гомогенная (состоящая из одинаковых по физическим параметрам и функциональности объектов) группа колесных роботов. Модель робота должна удовлетворять следующим требованиям:

- *по физическим параметрам*: иметь небольшие размеры (не больше 20 см по каждому из измерений, не тяжелее 1 кг), передвигаться по горизонтальным поверхностям внутри помещений, развивать скорость не менее 3 м/с, иметь малый радиус разворота;

- *по электронным компонентам*: иметь модель датчика расстояния, иметь модель средства связи между роботами, уметь в реальном времени с частотой 20 Гц обрабатывать данные датчиков и сообщения от других роботов с помощью алгоритма, реализующего стайное поведение.

Роботы управляются одинаковыми алгоритмами, однако параметры этих алгоритмов могут различаться между роботами.

Во второй части необходимо формально поставить задачи для групп роботов. Всего этих задач будет 3: (1) достижение заданной точки, (2) следование за лидером, (3) сбор в одной точке.

В задаче (1) рассмотрим группу из N одинаковых роботов, расположенных на расстоянии $r_0 < r_{nbhd}$(максимального расстояния связи между роботами) между собой, покоящиеся в начальный момент на горизонтальной плоскости. Роботам сообщается целевая точка G на этой плоскости. На плоскости могут находиться неподвижные препятствия, представленные в виде ограниченных областей, не содержащих роботов, с произвольной границей. Роботам необходимо переместиться из своего начального положения в окрестность целевой точки в покоящееся положение, сохранив в финальном положении расстояние

между роботами r_0 или заняв устойчивое положение с минимально возможной дистанцией между роботами, соблюдая безопасное расстояние r_{min} от препятствий на пути к целевой точке.

В задаче (2) аналогично имеется группа из N одинаковых роботов, расположенных на расстоянии $r_0 < r_{nbhd}$ между собой, покоящихся в начальный момент на горизонтальной плоскости. На плоскости есть неподвижные препятствия, представленные в виде ограниченных областей, не содержащих роботов, с произвольной границей. Подгруппе из n_L роботов сообщается целевая скорость $\vec{v_g}$, являющаяся непрерывно дифференцируемой вектор-функцией от времени, постоянная по модулю $v_m < v_{max}$(максимальной скорости роботов) и лежащая в горизонтальной плоскости. Группе необходимо следовать с близкой к целевой скоростью, при этом сохраняя расстояние между собой не более r_{nbhd} и между роботами и препятствиями не менее r_{min}.

В задаче (3) N покоящихся одинаковых роботов расположены в горизонтальной плоскости на случайных расстояниях друг от друга и должны за заданное заранее время сформировать минимально возможное количество кластеров (окрестностей диаметром, менее заданного r, содержащих роботов), с минимальным диаметром для этих кластеров. Условие безопасности передвижения относительно неподвижных препятствий произвольной формы также должно выполняться (в любой момент времени каждый робот должен иметь расстояние более r_{min} до любого из препятствий).

Теперь обратимся к третьей части основной задачи: алгоритм управления. Алгоритм управления должен генерировать стайное поведение, быть децентрализованным, использовать локальную информацию на каждом роботе и должен быть относительно прост в вычислении. Стая — группа однородных объектов с одинаковым алгоритмом движения с общей задачей, движение которой удовлетворяет следующим критериям:

- сплоченность: агенты двигаются на относительно близких и схожих дистанциях;
- безопасность: агенты держатся на безопасном расстоянии друг от друга.

И четвертая часть задачи состоит в оптимизации алгоритма. Необходимо составить функции, которые могли бы оценивать выполнение задач группой роботов по следующим критериям: производительность, безопасность и энергоэффективность. Также необходимо разработать метод настройки параметров алгоритма управления для получения оптимального с точки зрения одного из критериев (или их взвешенной совокупности), применяемый до или во время выполнения задачи без предыдущей истории выполнения задач.

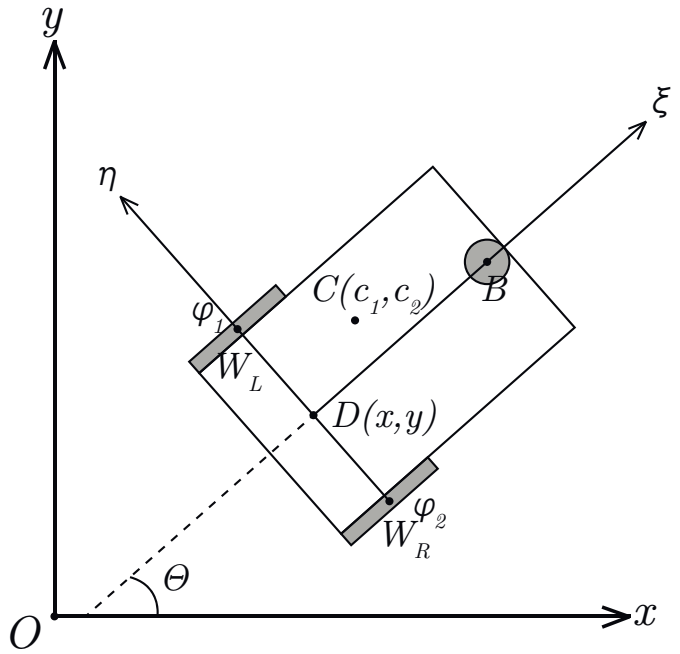

Рис. 2.1. Схема дифференциального привода

2. Математическая модель робота

Для построения модели управления нам необходима связь моментов двигателей со скоростью робота и оценка величины этих моментов. Впоследствии сами уравнения динамики также пригодятся при решении оптимальной задачи.

Для простоты математическая модель робота будет представлять набор твердых тел с неголономными связями. Наличие модели трения в системе моделирования позволяет нам на данном этапе не включать вязкое трение в описание.

Модель робота состоит из корпуса с размерами $2a, 2b$ и $2c$, массой m_1; пары колес радиуса r_1 и ширины w, массы m_2, закрепленных на корпусе параллельно соответствующим граням с помощью вращательных шарниров в точках $(-a + r_1, \pm b, -c + r_1 - 2r_2)$ относительно $D\xi\eta\zeta$; шаровой опоры радиуса r_2 и массы m_3, закрепленной на корпусе в точке $(a - r_2, 0, -c - r_2)$ относительно $D\xi\eta\zeta$.

Шаровой шарнир является свободным. В местах крепления колес к осям находятся точечные цилиндрические шарниры, колеса управляются идеальными электродвигателями, выдающими управляющие моменты M_1 и M_2.

2.1. Кинематика дифференциального привода

Проведем для начала описание кинематики дифференциального привода.

Система дифференциального привода представляет собой два абсолютно твердых диска, находящихся на осях, лежащих на одной прямой, в местах крепления колес к осям находятся точечные цилиндрические шарниры, колеса управляются идеальными электродвигателями. К осям жестко прикреплен корпус, абсолютно твердое тело, которое может двигаться плоскопараллельно. Это устройство движется по абсолютно шероховатой плоскости, колеса в точках касания с плоскостью не проскальзывают. При условии плоскопараллельного движения корпуса положение системы описывается пятью координатами $(x, y, \Theta, \phi_1, \phi_2)$, где x, y — координаты центра корпуса относительно неподвижной системы координат $Oxyz$, угол Θ является углом между горизонтальной осью $D\xi$, перпендикулярной осям колес, и осью Ox, углы ϕ_1, ϕ_2 представляют собой углы поворота ведущих колес робота относительно осей. Центр масс корпуса объекта расположен в точке C. Положение центра масс корпуса в связанной системе координат задается вектором $\vec{c} = (c_1, c_2)$. Влияние пассивных колес на движение системы считаем незначительным.

Из условий непроскальзывания колес возникают дифференциальные связи:

$$(\vec{V}_{W_L}, \vec{e}_\xi) = \dot{\phi}_1 r_1,$$
$$(\vec{V}_{W_R}, \vec{e}_\xi) = \dot{\phi}_2 r_1.$$

Скорости центров колес и центром оси из формулы Эйлера тоже соотносятся следующим образом:

$$\vec{V}_{W_L} = \vec{V}_{W_R} + \vec{\omega} \times \overrightarrow{W_R W_L},$$
$$\vec{V}_D = \vec{V}_{W_R} + \vec{\omega} \times \overrightarrow{W_R D},$$
$$\vec{V}_D = \vec{V}_{W_L} + \vec{\omega} \times \overrightarrow{W_L D}.$$

Приняв за обобщенные координаты координаты точки D и записав $\vec{V}_D = \dot{x}\vec{e}_x + \dot{y}\vec{e}_y$, получим следующие уравнения дифференциальных связей:

$$\begin{cases} \dot{x} = (b\dot{\Theta} + r_1\dot{\phi}_1)\cos\Theta, \\ \dot{y} = (b\dot{\Theta} + r_1\dot{\phi}_1)\sin\Theta, \\ \dot{\phi}_2 = \dot{\phi}_1 + \frac{2b}{r_1}\dot{\Theta}. \end{cases} \tag{2.1}$$

Эта система уже многократно исследовалась, и можно показать, что данная система уравнений не интегрируема и задаваемые ей связи являются неголономными.

В модели, использованной в симуляции, конструкцию дополняет шаровая опора. Значит, к нашей системе добавляется шар, вращающийся без про-

скальзывания, с закрепленным относительно корпуса центром, угловая скорость которого имеет только горизонтальные составляющие. Рассмотрим, какие дополнительные связи возникают в новой системе.

Пусть $\vec{\omega}_{ball} = p\vec{e}_x + q\vec{e}_y$ – угловая скорость шара в неподвижной системе координат. Для скорости центра шара B верны формулы Эйлера для корпуса и шара соответственно.

$$\vec{V}_B = \vec{V}_D + \vec{\omega} \times \overrightarrow{DB},$$
$$\vec{V}_B = \vec{\omega}_{ball} \times \overrightarrow{KB},$$

где K – точка касания шара о плоскость.

После преобразований получается пара дополнительных соотношений, выражающих связи в системе:

$$\begin{cases} \omega_x = \dfrac{\dot{x}}{r_2} + \dfrac{a - r_1}{r_2}\dot{\Theta}\sin\Theta = (\dfrac{b\cos\Theta}{r_2} + \dfrac{(a - r_1)\sin\Theta}{r_2})\dot{\Theta} + \dfrac{r_1\cos\Theta}{r_2}\dot{\phi}_1, \\ \omega_y = -\dfrac{\dot{y}}{r_2} + \dfrac{a - r_1}{r_2}\dot{\Theta}\cos\Theta = (\dfrac{(a - r_1)\cos\Theta}{r_2} - \dfrac{b\sin\Theta}{r_2})\dot{\Theta} - \dfrac{r_1\sin\Theta}{r_2}\dot{\phi}_1. \end{cases} \tag{2.2}$$

2.2. Динамика исследуемой модели

Рассматриваемая система имеет 2 степени свободы, в качестве независимых координат выберем (ϕ_1, Θ). Покажем, что система является системой Чаплыгина, и запишем уравнения движения системы в виде уравнений Чаплыгина [27]. Для этого необходимо выписать кинетические энергии тел системы, активные силы и уравнения связей, и кинетическая энергия системы не должна содержать в своем выражении зависимые координаты.

Будем записывать наш набор координат $(\phi_1, \Theta, x, y, \phi_2, \alpha_x, \alpha_y)$ как $(q_1, q_2, q_3, q_4, q_5, q_6, q_7)$.

Для записи кинетических энергий воспользуемся теоремой Кенига.

$$T_{body} = \frac{1}{2}m_1\vec{V}_C^2 + \frac{1}{2}(\mathcal{J}_C^{body}\vec{\omega}^{body}, \vec{\omega}^{body}),$$

$$\vec{V}_C = \vec{V}_D + \vec{\omega} \times \overrightarrow{DC},$$

$$\mathcal{J}_C^{body} = \begin{pmatrix} I_x & I_{xy} & I_{xz} \\ I_{xy} & I_y & I_{yz} \\ I_{xz} & I_{yz} & I_z \end{pmatrix},$$

$$\vec{\omega}^{body} = \dot{\Theta}\vec{e}_z,$$

$$T_{leftwheel} = \frac{1}{2}m_2\vec{V}_{W_L}^2 + \frac{1}{2}(\mathcal{J}^{wheel}\vec{\omega}_{leftwheel}, \vec{\omega}_{leftwheel}),$$

$$\vec{V}_{W_L} = \vec{V}_D + \vec{\omega} \times \overrightarrow{DW_L},$$

$$\mathcal{J}^{wheel} = \begin{pmatrix} \frac{m_2(3r_1^2+w^2)}{12} & 0 & 0 \\ 0 & \frac{m_2 r_1^2}{2} & 0 \\ 0 & 0 & \frac{m_2(3r_1^2+w^2)}{12} \end{pmatrix},$$

$$\vec{\omega}_{leftwheel} = \dot{\phi}_1 \vec{e}_\eta + \dot{\Theta}\vec{e}_z,$$

$$T_{rightwheel} = \frac{1}{2} m_2 \vec{V}_{W_R}^2 + \frac{1}{2}(\mathcal{J}^{wheel}\vec{\omega}_{rightwheel}, \vec{\omega}_{rightwheel}),$$

$$\vec{V}_{W_R} = \vec{V}_D + \vec{\omega} \times \overrightarrow{DW_R},$$

$$\vec{\omega}_{rightwheel} = \dot{\phi}_2 \vec{e}_\eta + \dot{\Theta}\vec{e}_z,$$

$$T_{ball} = \frac{1}{2} m_2 \vec{V}_B^2 + \frac{1}{2}(\mathcal{J}^{ball}\vec{\omega}_{ball}, \vec{\omega}_{ball}),$$

$$\vec{V}_B = \vec{V}_D + \vec{\omega} \times \overrightarrow{DB},$$

$$\mathcal{J}^{ball} = \begin{pmatrix} \frac{2m_3 r_2^2}{5} & 0 & 0 \\ 0 & \frac{2m_3 r_2^2}{5} & 0 \\ 0 & 0 & \frac{2m_3 r_2^2}{5} \end{pmatrix},$$

$$\vec{\omega}_{ball} = \omega_x \vec{e}_x + \omega_y \vec{e}_y + \dot{\Theta}\vec{e}_z,$$

$$T_{ball} = \frac{1}{2} m_3 (\dot{x}^2 + 2\dot{x}(2a - r_1 - r_2)\sin\Theta\dot{\Theta} + \dot{y}^2 - 2\dot{y}(2a - r_1 - r_2)\cos\Theta\dot{\Theta} + \\ + (2a - r_1 - r_1)^2\dot{\Theta}^2) + \frac{m_3 p^2 r_2^2}{5} + \frac{m_3 q^2 r_2}{5} + \frac{m_3 r_2^2}{5}\dot{\Theta}^2,$$

$$T = \frac{1}{2}(m_1(b^2 - 2bc_2 + c_2^2 + (c_1 + r_1 - a)^2) + I_z + m_2(6b^2 + \frac{r_1^2}{2} + \frac{w^2}{6}) + \\ + \frac{2m_3(b^2 + (a - r_1)^2 + r_2^2)}{5})\dot{\Theta}^2 + \frac{1}{2}(m_1 r_1^2 + 3m_2 r_1^2 + \frac{4m_3 r_1^2}{5})\dot{\phi}_1^2 + \\ + (m_1(br_1 - r_1 c_2) + 3m_2 b r_1 + \frac{4m_3 b r_1}{5})\dot{\Theta}\dot{\phi}_1. \quad (2.3)$$

Среди активных сил, действующих на систему, будем учитывать только вращательные моменты электродвигателей. Таким образом, $Q_2 = Q_3 = Q_4 = Q_6 = Q_7 = 0$, $Q_1 = M_1$, $Q_5 = M_2$, а в самих уравнениях будут записаны Q_1^* и Q_2^*:

$$Q_1^* = Q_1 + \sum_{\mu=1}^{3} Q_{2+\mu} c_{\mu 1} = Q_1 + Q_5, \quad (2.4)$$

$$Q_2^* = Q_2 + \sum_{\mu=1}^{3} Q_{2+\mu} c_{\mu 2} = \frac{2b}{r_1} Q_5, \quad (2.5)$$

где $c_{\mu i}$ — коэффициенты в уравнениях дифференциальных связей. Уравнения дифференциальных связей записаны в 2.1.

Уравнения Чаплыгина записываются в следующем виде:

$$\begin{cases} (m_1 + 3m_2 + \frac{4}{5}m_3)r_1^2\ddot{\phi}_1 + (m_1(br_1 - r_1c_2) + 3m_2br_1 + \frac{4m_3br_1}{5})\ddot{\Theta} = \\ \qquad = Q_1 + Q_5 + m_1(c_1 + r_1 - a)r_1\dot{\Theta}^2 + \\ \qquad + \frac{4m_3r_1}{5}((b\cos 2\Theta + (a - r_1)\sin 2\Theta)\dot{\Theta} + r_1\cos 2\Theta\dot{\phi}_1)\dot{\Theta}, \\ (m_1(b^2 - 2bc_2 + c_2^2 + (c_1 + r_1 - a)^2) + I_z + m_2(6b^2 + \frac{r_1^2}{2} + \frac{w^2}{6}) + \\ \qquad + \frac{2m_3(b^2 + (a - r_1)^2 + r_2^2)}{5})\ddot{\Theta} + (m_1(br_1 - r_1c_2) + \\ \qquad + 3m_2br_1 + \frac{4m_3br_1}{5})\ddot{\phi}_1 = \\ \qquad = \frac{2b}{r_1}Q_5 - m_1(c_1 + r_1 - a)r_1\dot{\Theta}\dot{\phi}_1 - \\ \qquad - \frac{4m_3r_1}{5}((b\cos 2\Theta + (a - r_1)\sin 2\Theta)\dot{\Theta} + r_1\cos 2\Theta\dot{\phi}_1)\dot{\phi}_1. \end{cases} \tag{2.6}$$

3. Компьютерная модель робота

Для симуляции стаи роботов нам необходимо программное обеспечение, имплементирующее нашу модель, с возможностью подключить свой алгоритм управления к каждому роботу. Для симулирования стай роботов существует несколько узкоспециализированных программных комплексов (таких как Swarmrobot, STAGE), однако их возможности по созданию моделей сильно ограничены, и нет описания их алгоритмов управления. Есть более универсальные и популярные симуляторы, такие как Matlab, V-REP, Gazebo, ARGoS, которые подходят для нашего моделирования, однако в них нет модуля для моделирования групп роботов (за исключением групп дронов [28]).

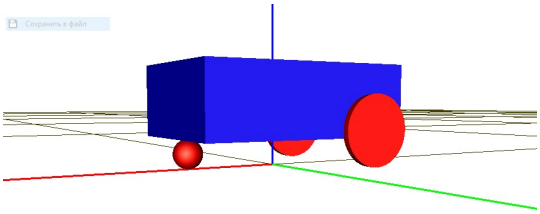

Рис. 3.1. Упрощенная модель робота в ПО «Универсальный механизм»

В этой работе для моделирования движения был выбран программный комплекс «Универсальный механизм» [29]. Этот комплекс позволяет моделировать динамику твердых тел с учетом заданных связей. В нем реализовано несколько методов составления уравнений динамики и несколько численных методов для интегрирования нелинейных уравнений движения. В механические связи между телами можно добавить различные модели трения. С точки зрения функциональности, релевантной нашей задаче, программный комплекс не уступает более популярным симуляторам V-REP, Gazebo, ARGoS [30]: есть возможность графического моделирования, импорта 3d-модели робота, подключения алгоритмов, написанных C++, есть программный интерфейс приложения.

Как и во всех рассмотренных симуляторах, в программном комплексе «Универсальный механизм» нет готового интерфейса для моделирования групп роботов. Для моделирования группы роботов был разработан собственный интерфейс, принимающий физические параметры и 3d-модель деталей робота, параметры сценария моделирования и библиотеку управления роботом на вход и проводящий моделирование на заданном промежутке времени.

В добавление к описанию моделирования в программе «Универсальный механизм» и математической модели было использовано трение между колесами и горизонтальной поверхностью. Трение реализовано в виде модели качения: есть сила трения скольжения и моменты трения верчения и качения, определяющиеся по формуле «модифицированной сигнатуры»:

$$\mathbf{F_f} = \begin{cases} -\frac{fN\mathbf{V}}{\|\mathbf{V}\|}, \|\mathbf{V}\| > V^*, \\ -\frac{fN\mathbf{V}}{V^*}, \|\mathbf{V}\| \leq V^*. \end{cases}$$

$$\mathbf{M}_{spin} = \begin{cases} -\frac{k_{spin}N\omega_{spin}}{|\omega_{spin}|}, |\omega_{spin}| > \omega^*, \\ -\frac{k_{spin}N\omega_{spin}}{\omega^*}, |\omega_{spin}| \leq \omega^*. \end{cases}$$

$$\mathbf{M}_{roll} = \begin{cases} -\frac{k_{roll}N\omega_{roll}}{|\omega_{roll}|}, |\omega_{roll}| > \omega^*, \\ -\frac{k_{roll}N\omega_{roll}}{\omega^*}, |\omega_{roll}| \leq \omega^*. \end{cases}$$

4. Алгоритм управления группой роботов

За основу был взят алгоритм В. Е. Павловского [14], использующий метод искусственных сил, с доработкой для размерных роботов и синтезом управляющих сигналов. Алгоритм обеспечивает согласованное движение к цели и поддержание стаи как структуры над роботами и был проверен на моделировании движения роботов, как точек на плоскости.

В данной модели управления роботы помимо своих механических параметров характеризуются минимальным радиусом сближения r_{min}, радиусом видимости r_{nbhd} и максимальным управляющим моментом в колесах M_{max}. Каждый робот управляется одним и тем же алгоритмом, который на вход получает целевую точку $\vec{r}_g$, положение этого робота $\vec{r}$ и его соседей $\vec{r}_i$, их скорости, положение препятствий, скорости вращения колес этого робота, а также глобальные параметры k_i и γ. На выход каждый шаг времени алгоритм устанавливает управляющие моменты колес.

Алгоритм состоит из четырех последовательных этапов: 1) генерация вектора скорости по правилу, соответствующему поставленной перед роботами задаче; 2) генерация вектора скорости по правилам стайного движения; 3) коррекция суммарного вектора скорости вблизи препятствий; 4) проверка и уход от возможных столкновений.

Напомним, что группа роботов будет решать следующие задачи: (1) достижение заданной точки, (2) следование за лидером, (3) сбор в одной точке. В задаче (1) вектор скорости будет направлен в сторону заданной точки. В задаче (2) вектор скорости будет либо заданным (если робот является лидером), либо определяется средней скоростью ближайших соседей. В задаче (3) вектор скорости будет состоять из случайного вектора и вектора, направленного к ближайшему из соседей.

Для реализации поставленной задачи на втором этапе вектор скорости будет складываться из компонент, отвечающих за сплоченность и безопасность движения стаи. Для плавного обхода препятствий на втором этапе будет учитываться влияние векторных полей скоростей вокруг них. Так как и в симуляции, и на реальных прототипах управляющие сигналы подаются дискретно, один раз в определенный период времени, необходим третий этап, где проводится проверка того, что робот не столкнется с препятствием или другим роботом.

Алгоритм регулируется при помощи изменения следующих параметров:

- k_1 — коэффициент, при увеличении которого увеличивается вклад движения, нацеленного на решение задачи;
- k_2 — коэффициент, при увеличении которого увеличивается вклад движения на центр масс соседей по группе в вектор желаемой скорости;
- k_3 — коэффициент, при увеличении которого увеличивается вклад движения от ближайших роботов в вектор желаемой скорости.

4.1. Построение вектора скорости

Пусть $\vec{r}$ — положение робота, $\vec{v}$ — скорость робота.

Построим вектор скорости в задаче (1). $\vec{r_g}$ — положение цели. Вектор притяжения к цели задается формулой:

$$\vec{V_1}^{goal} = k_1 \frac{\vec{r_g} - \vec{r}}{|\vec{r_g} - \vec{r}|}. \tag{4.1}$$

Построим вектор скорости во задаче (2) для роботов, не являющихся лидерами. n_1 — количество роботов в зоне видимости, $\vec{v_c} = \frac{\sum_{i=1}^{n_1} \vec{v_i}}{n_1}$. Вектор следования за целью задается формулой:

$$\vec{V_1}^{flock} = k_1 \frac{\vec{v_c}}{|\vec{v_c}|}. \tag{4.2}$$

Построим вектор скорости в задаче (3). $\vec{v}_{random}(t)$ — скорость случайного поиска, постоянный по модулю, а его направление является случайным блужданием. Вектор поиска при агрегации задается формулой:

$$\vec{V_1}^{agg} = k_1(1 - \frac{t}{t_f})\vec{v}_{random}(t). \tag{4.3}$$

Для вектора скорости, отражающего стайное поведение, формулы во всех задачах одинаковы. Пусть n_1 — количество роботов в зоне видимости, n_2 — количество роботов в радиусе безопасности, $\vec{r_c} = \frac{\sum_{i=1}^{n_1} \vec{r_i}}{n}$ — геометрический центр соседних роботов, $\vec{r_j}^{dist} = -k_3(2 - \frac{|\vec{r_j} - \vec{r}|}{r_{min}})^2 \frac{\vec{r_j} - \vec{r}}{|\vec{r_j} - \vec{r}|}$ — вектор отталкивания от робота.

Вектор притяжения к центру соседних роботов:

$$\vec{V_2} = k_2 \frac{\vec{r_c} - \vec{r}}{|\vec{r_c} - \vec{r}|}. \tag{4.4}$$

Вектор отталкивания от соседних роботов:

$$\vec{V_3} = \sum_{j=1}^{n_3} \vec{r_j}^{dist}. \tag{4.5}$$

Аналогично подсчитывается отталкивание от препятствий в зоне видимости.

В итоге получаем предварительный вектор скорости, которую должен установить робот:

$$\vec{V} = \vec{V_1} + \vec{V_2} + \vec{V_3}. \tag{4.6}$$

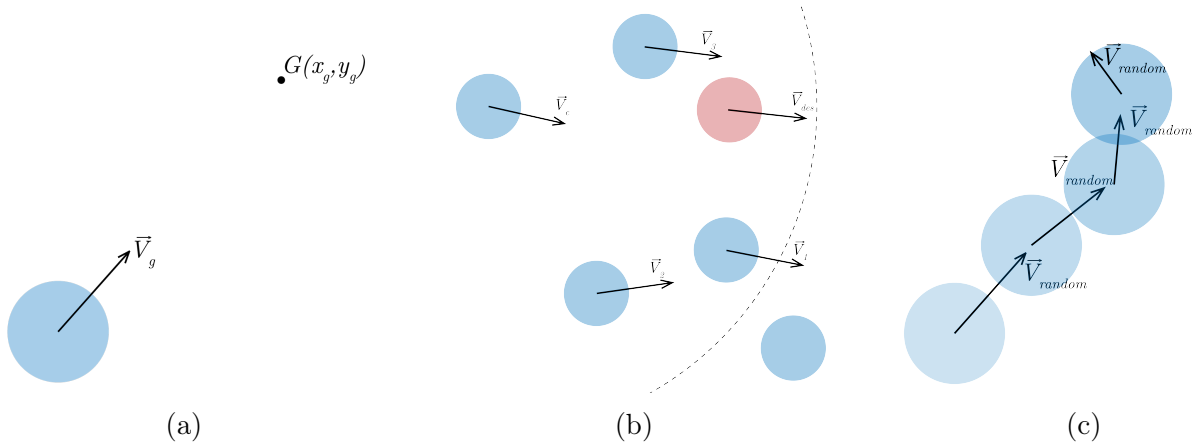

Рис. 4.1. Векторы $\vec{V}_1$ в (a) задаче 1, (b) задаче 2, (c) задаче 3

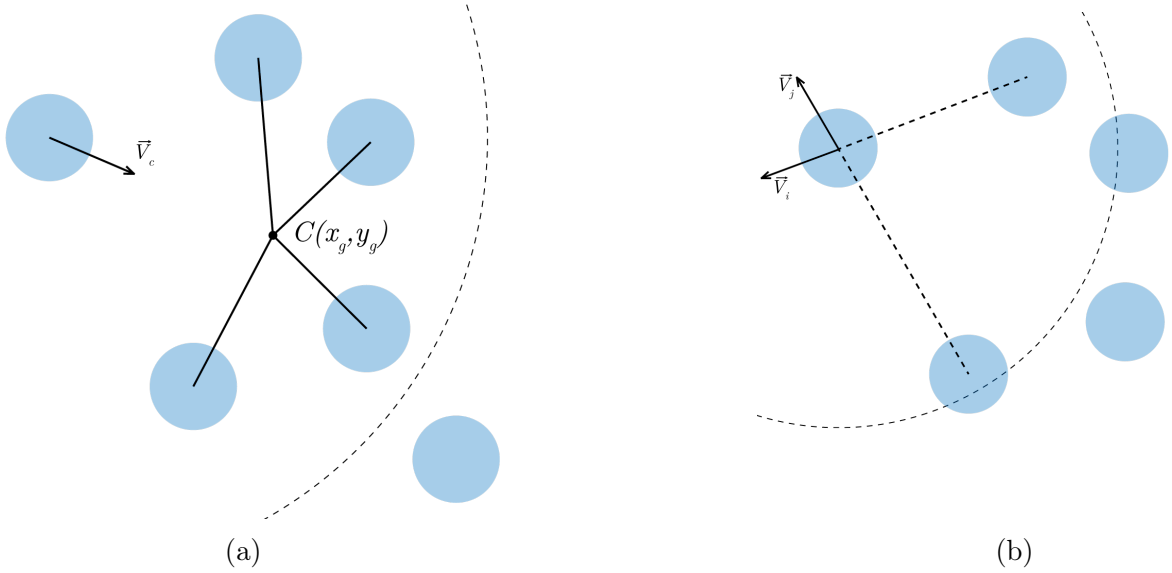

Рис. 4.2. Векторы (a) притягивания к центру масс и (b) отталкивания от соседей

4.2. Уклонение от препятствий

Для плавного объезда препятствий вектор скорости роботов корректируется при обнаружении препятствий в зоне видимости. В основе этой корректировки перенаправление вектора скорости на направление по касательной к границе встречного объекта.

Для одного препятствия в зоне видимости корректировка происходит следующим образом. Для начала считается косинус угла между вектором желаемой скорости $\vec{V}_{des}$ и вектором $\vec{r}_{toobs}$, соединяющим центр робота и центр соответствующего препятствия. Если он оказывается положительным (т.е. проекция вектора скорости на вектор к препятствию сонаправлена с этим вектором), то производится корректировка, иначе — вектор скорости остается исходным. Далее строится вектор $\vec{V}_{tan}$, перпендикулярный вектору от

робота к препятствию, сонаправленный с вектором скорости в проекции на него и равный по модулю вектору скорости. Итоговый вектор скорости равен:

$$\vec{V} = (1 - e)\vec{V}_{des} + e\vec{V}_{tan}, \tag{4.7}$$

где

$$e = e^{-\gamma(||\vec{r}_{toobs}|| - r_{obs} - r_{min})^2}. \tag{4.8}$$

Здесь γ — коэффициент, отвечающий за увеличение корректирующей составляющей при приближении к препятствию;

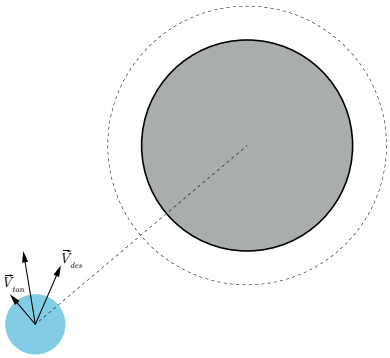

Рис. 4.3. Уклонение от препятствий

При нескольких препятствиях в зоне видимости возможны 2 варианта коррекции. Если граница ни одного из препятствий не находится в зоне безопасности робота, то для каждого препятствия высчитывается вектор $\vec{V}^i_{tan}$ и множитель e_i по формуле (4.8), $i = 1, \ldots, n_1$. В итоге получается вектор

$$\vec{V} = (1 - \frac{\sum_{i=1}^{n_1} e_i}{n_1})\vec{V}_{des} + \frac{1}{n_1}\sum_{i=1}^{n_1} e_i\vec{V}^i_{tan}. \tag{4.9}$$

Если граница препятствий находится в радиусе безопасности робота, то только для таких препятствий высчитывается $\vec{V}^i_{tan}$, считается коэффициент $e_i = e^{-\gamma(||\vec{r}^i_{toobs}|| - r^i_{obs} - r_{min})}$ и затем считается скорректированный вектор скорости

$$\vec{V} = (1 - \frac{\sum_{i=1}^{n_1} e_i}{n_1})\vec{V}_{des} + \frac{1}{n_1}\sum_{i=1}^{n_1} \vec{V}^i_{tan}. \tag{4.10}$$

В итоге корректировки такого вида робот направляется в объезд по касательной препятствия, а также от препятствия, если оно находится в круге безопасности.

4.3. Проверка возможных столкновений

Для проверки допустимости движения в вычисленном направлении проинтегрируем один шаг движения робота:

$$\vec{r}^{new} = \vec{r} + \vec{V}dt.$$

Выберем самый ближайший объект (робот или препятствие) к роботу в следующий момент. Проверим, заедет ли управляемый робот в круг с центром в одном из объектов радиусом r_{min} или r_{obs}. Если этого не происходит, то $\vec{V}$ принимается для дальнейших расчетов моментов. Если для какого-то из объектов проверка дает положительный ответ, то вектор скорости изменяется следующим образом:

Строится радиус из центра объекта, проходящий через концевую точку вектора $\vec{r}^{new}$. Точка окружности, в которую приходит радиус, будет новой концевой точкой $\vec{r}^{new}$, и новый вектор скорости равен $\vec{V}^* = \frac{\vec{r}^{new} - \vec{r}}{dt}$.

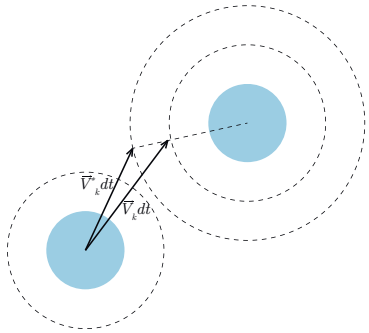

Рис. 4.4. Уход от столкновений

4.4. Расчет управляющих моментов

Для подсчета управляющих моментов использовался PID-регулятор. Воспользовавшись устройством дифференциального привода, можно вывести формулы, переводящие скорость и ускорение робота в модуль скорости по его оси и угловую скорость:

$$v = \sqrt{\dot{x}^2 + \dot{y}^2},$$
$$w = \frac{\ddot{y}\dot{x} - \ddot{x}\dot{x}}{v^2}.$$

Из этих величин можно вывести значения угловых скоростей в колесах, использовав 2.1

$$\begin{cases} \dot{\phi}_1 = \frac{v}{\rho} + \frac{wa}{\rho}, \\ \dot{\phi}_2 = \frac{v}{\rho} - \frac{wa}{\rho}. \end{cases} \tag{4.11}$$

Для создания управляющего момента на каждом колесе используется формула PID-регулятора. Пусть в каждый момент $\dot{\phi}_i$ — угловые скорости колес, $\dot{\phi}_i^*$ — посчитанные алгоритмом угловые скорости.

Введем величину $e_i = \dot{\phi}_i^* - \dot{\phi}_i$. Тогда управляющие моменты будут высчитываться по следующей формуле:

$$M_i = Pe_i + I\int_{t_0}^{t} edt + D\dot{e}_i, \tag{4.12}$$

где $P, I, D > 0$. Коэффициенты были посчитаны методом Циглера-Николса [34].

4.5. Дополнения алгоритмов в задачах (1)-(3)

Задачи (1) и (3) характеризуются условием выполнения задачи, после которого движение прекращается (в задаче (1) это остановка в окрестности целевой точки, в задаче (2) — окончание временного промежутка агрегации). В задаче (2) специфика состоит в том, что у части роботов есть явный управляющий сигнал.

Алгоритм задачи (1) дополнен следующим правилом: когда робот оказывается в окрестности целевой точки, он начинает тормозить, сохраняя проверку столкновений. Остановившись, робот продолжает покоиться.

В алгоритме (2) одному роботу задается желаемая скорость в каждый момент подсчета управляющего сигнала. Желаемая скорость является непрерывно дифференцируемой вектор-функцией с постоянным модулем.

В алгоритме задачи (3) также происходит торможение, после того как закончится заданное время на сбор роботов вместе.

5. Функции оценки эксперимента при различных параметрах алгоритма

Для оценки движения стаи в целом необходимо ввести величины, которые отражают поведение стаи в целом. В существующих работах было введено множество различных метрик, оценивающих выполнение задачи и стайное поведение группы. Например, в [19] вводятся функции, оценивающие только сбор ресурсов роботами, в [31] дается оценка только тому, насколько сплоченно и безопасно движется группа, в [22] используются оценочные функции и для оценки следования, и для оценки «стайности» роботов, в [18] представлена оценка энергоэффективности. В [32] предложены функции для оценки внимания и усилий оператора при управлении группой роботов

В данной работе для оценки движения при разных параметрах алгоритма было введено 14 различных оценочных функций. Эти функции можно разделить на 3 класса: в первом классе оценивается эффективность выполнения задачи, во втором классе (оценки сплоченности и безопасности) оценивается

«стайность» движения, в третьем классе оценивается энергоэффективность стаи (необходимая оценка для разработки стай реальных роботов).

5.1. Оценки в задаче (1):

(a) t_{finish} — время достижения цели всей стаей (чем быстрее стая доберется до цели, тем лучше);

(b) $v_{av} = \frac{1}{T}\frac{1}{N}\int_0^T \sum_{i=1}^N |\vec{v}_i| dt$ — средняя скорость по всей стае (показывает отношение пройденного всеми роботами в стае пути к затраченному на достижение цели времени).

5.2. Оценки в задаче (2):

(a) $\phi_{flock} = \frac{1}{T}\frac{1}{N}\int_0^T \sum_{i=1}^N \frac{1}{1+(|\vec{v}_i|-v_{flock})^2}\frac{\vec{v}_i\vec{v}_{flock}}{|\vec{v}_i||\vec{v}_{flock}|}dt$ — средняя корреляция скорости робота к желаемой скорости по всей стае (при $\phi_{flock} = 1$ стая двигается с желаемой скоростью, чем меньше значение, тем меньше корреляция скоростей с желаемой скоростью);

(b) $\phi_{corr} = \frac{1}{T}\frac{1}{N(N-1)}\int_0^T \sum_{i=1}^N \sum_{j=1, j\neq i}^N \frac{\vec{v}_i\vec{v}_j}{|\vec{v}_i||\vec{v}_j|}dt$ — средняя корреляция вектора скорости робота в стае (показывает, насколько роботы в стае двигаются со схожими скоростями).

5.3. Оценки в задаче (3):

(a) $N_{cluster}$ — количество кластеров в конце выполнения задачи. Кластером считается множество роботов, в котором для каждого робота i найдется хотя бы один робот j, расстояние до которого $|\vec{r}_{ij}|, r_{nbhd}$. Чем меньше кластеров, тем лучше выполнена задача сбора в одной точке;

(b) $n_{cluster} = \frac{1}{N_{cluster}}\sum_{k=1}^{N_{cluster}} n_k$ (n_k — количество роботов в k кластере) — среднее количество роботов в кластере (чем больше эта величина, тем больше кластеры (желаемое выполнение задачи));

(c) $\delta_{cluster} = \frac{1}{N}\sum_{i=1}^N (\vec{r}_i - \vec{r}_{ci})^2$ — среднее расстояние до центра масс соседей на момент остановки.

5.4. Оценки сплоченности:

(a) $\delta_{closeness} = \max_{i,j} |\vec{r}_{ij}|$ — максимальное удаление роботов друг от друга (чем меньше значение, тем больше сплоченность);

(b) $\delta^*_{closeness} = \frac{1}{N}\sum_{i=1}^N \sum_{j=1, j\neq i}^N |\vec{r}_{ij}|$ — расстояние между роботами по всей стае (для оценки сплоченности в средах, где стая не сможет из-за препятствий принимать структуру, близкую к кругу, и предыдущая оценка неизбежно будет принимать высокие значения);

(c) $\delta^{**}_{closeness} = \frac{1}{T}\int_0^T \frac{1}{N}\sum_{i=1}^N \sum_{j=1, j\neq i}^N |\vec{r}_{ij}|dt$ — среднее расстояние между роботами по всей стае (для оценки сплоченности за все время движения до цели).

5.5. Оценка безопасности:

$$\delta_{safety} = \frac{1}{N}\sum_{i=1}^N \int_0^T I(t)dt, \text{ где } I(t) = \begin{cases} 1, \text{если } \exists \text{ object } \mathbb{O} : \vec{r}^{\,i}_{\mathbb{O}}(t) < r_{min} \\ 0, \text{иначе} \end{cases}$$

— штраф за опасное сближение.

5.6. Оценки энергоэффективности:

(a) $c = \sum_{i=1}^N k \int_0^T ((M^i_{left})^2 + (M^i_{right})^2)dt$ — расход топлива по всей стае (оценка оптимальности управления по энергетическим затратам);

(b) $c^* = \max_i k \int_0^T ((M^i_{left})^2 + (M^i_{right})^2)dt$ — максимальный расход (оценка оптимальности управления по энергетическим затратам в случаях, когда для достижения цели может тратиться весь запас топлива).

6. Моделирование движения

Авторами работы была создана в программном комплексе «Универсальный механизм» (версия 8.3.5.3) группа роботов, состоящая из 7 моделей дифференциальных роботов, со следующими параметрами: $a = 0.075$м, $b = 0.05$м, $c = 0.025$м, $r_1 = 0.025$м, $r_2 = 0.01$м $m_1 = 1$кг, $m_2 = m_3 = 0.05$кг, $I_z = 0.0027$кг $\cdot$ м^2 $r_{min} = 0.2$м, $r_{nbhd} = 1$м, $v_{max} = 1$м/с, $M_{max} = 0.02$Нм.

Помимо роботов на плоскости были размещены 2 цилиндрических препятствия.

Управляющие сигналы были представлены в виде моментов, приложенных в шарнирах крепления колес к корпусу. Моменты обновляются с частотой 50 Гц.

В начальный момент сигналы нулевые, роботы покоятся. Затем в течение 40 секунд роботы выполняют задание.

Уравнения динамики системы синтезированы методом составных тел. Интегрирование динамических уравнений в программе «Универсальный механизм» выполнено прямым методом Парка с погрешностью 1E-6.

Моделирование экспериментов со следующими наборами параметров:
$k_1 = [0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 4]$,
$k_2 = [0.01, 0.02, 0.05, 0.1, 0.2, 0.4, 1]$,
$k_3 = [1, 2, 4, 10, 20, 50, 100]$.

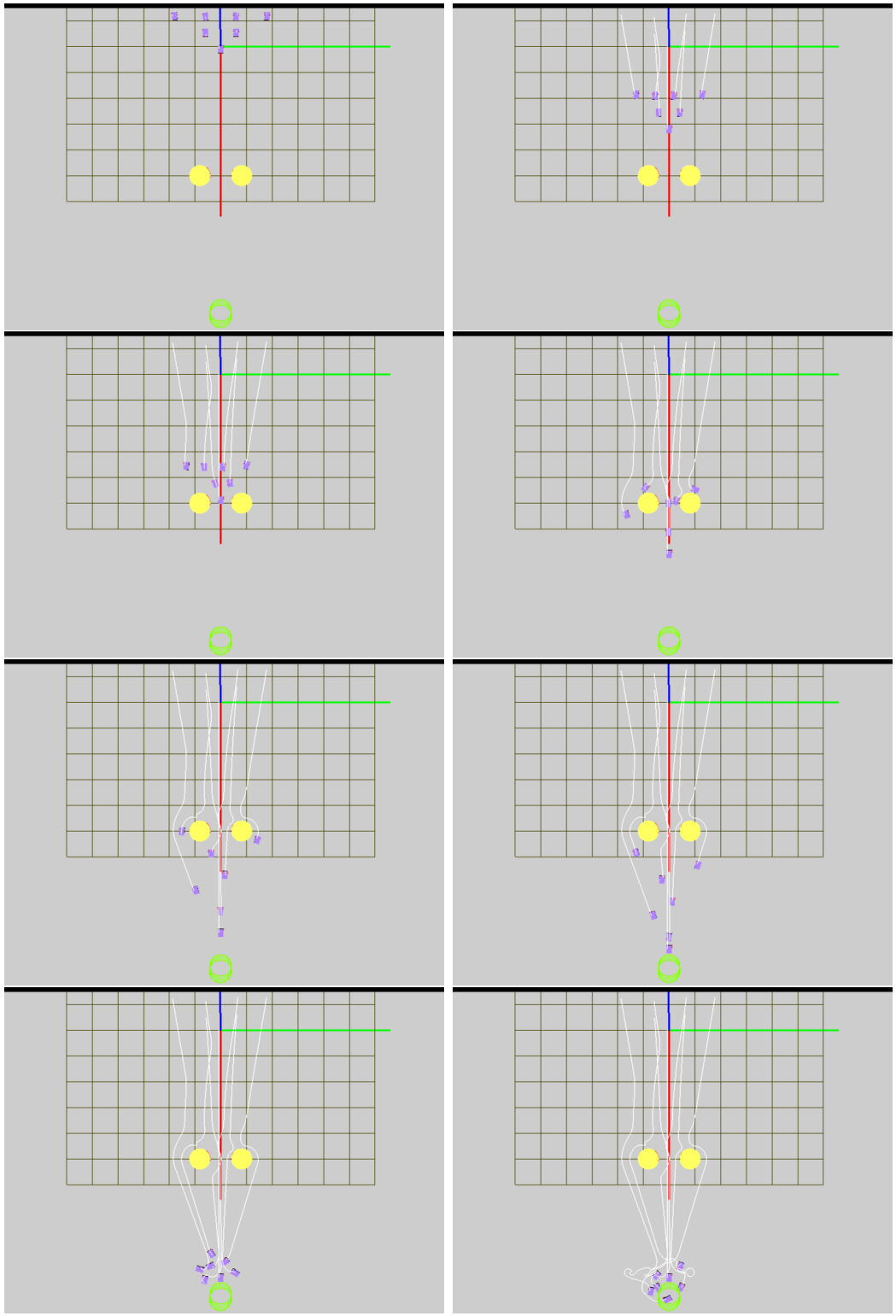

Рис. 6.1. Моделирование движения группы при $k_1 = 4$, $k_2 = 0.05$, $k_3 = 10$ (кадры сняты каждые 5 секунд)

7. Зависимость оценок алгоритма от его параметров

По результатам моделирования были выявлены некоторые характерные зависимости оценок от параметров алгоритма. Зависимости определялись

среди линейных зависимостей от одного параметра, квадратичных и гиперболических от пар параметров.

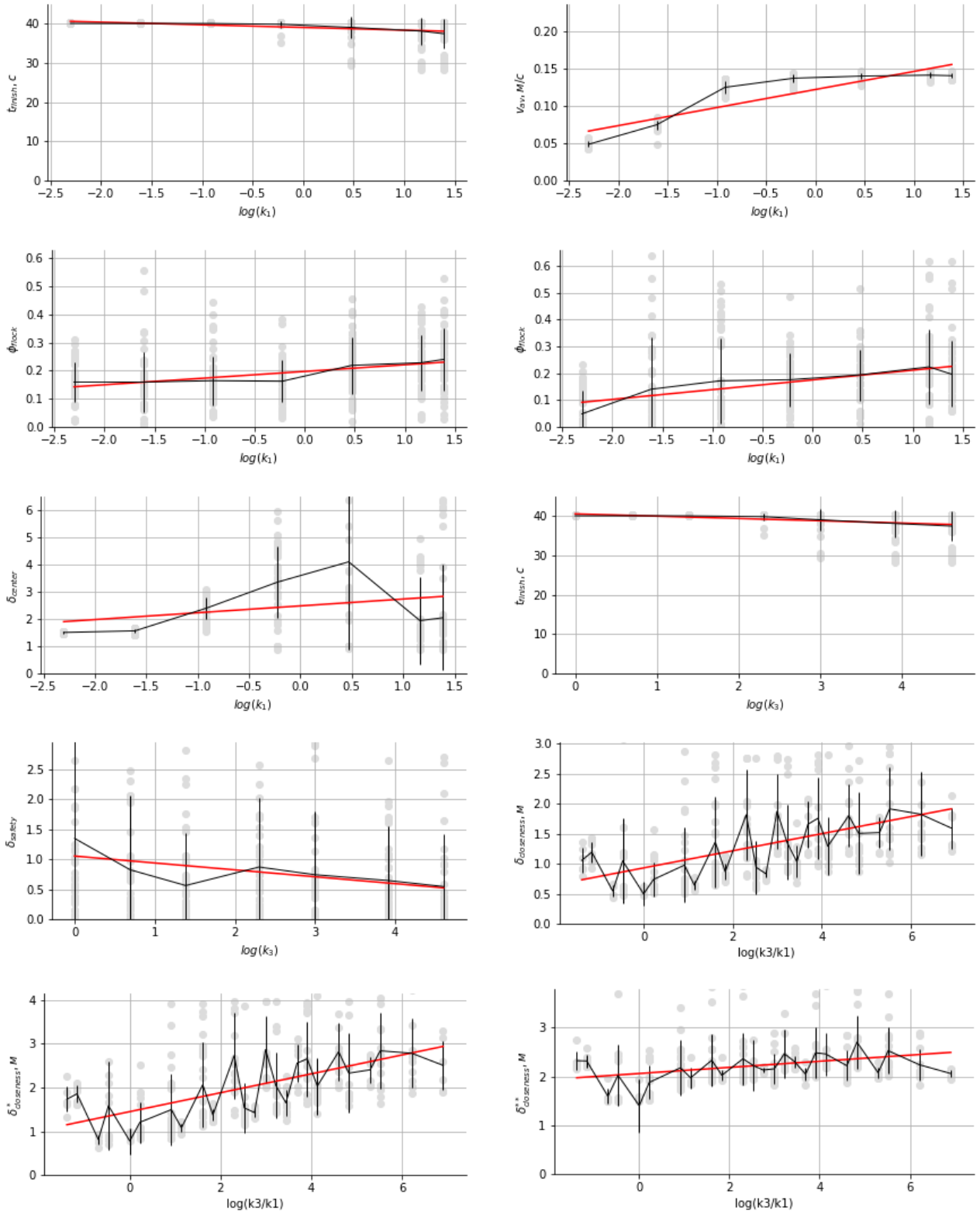

Рис. 7.1. Значения различных оценки ($t_{finish}, v_{av}, \phi_{flock}, \phi_{corr}, \delta_{center}, \delta_{closeness}, \delta^*_{closeness}, \delta^{**}_{closeness}, \delta_{safety}$) при различных параметрах k_1, k_2, k_3, по горизонтальной оси значения параметров по логарифмической шкале, **черным** соединены средние значения для фиксированного параметра (с отрезками среднеквадратического отклонения), **красным** отмечена линейная аппроксимация.

Как и ожидалось, время достижения целевой точки в задаче (1) и среднее количество роботов в кластере в задаче (3) уменьшались с увеличением k_1, а средняя скорость, среднее расстояние до соседей и корреляции следования и скоростей в стае при этом увеличивались. Изменение k_3 приводило к обратному изменению времени достижения целевой точки и среднего расстояния до соседей. Максимум расстояния между роботами увеличивался с увеличением соотношения $\frac{k_2}{k_3}$. Аналогичные зависимости наблюдались для остальных оценок, связанных с расстояниями между роботами. Функция штрафа за столкновения уменьшалась с увеличением параметра отталкивания k_3 между роботами. Для функций расхода топлива для данного набора экспериментов не было обнаружено зависимости от параметров. Зависимости проиллюстрированы на 7.1.

8. Оптимизация стайного движения

Для оптимизации движения с учетом энергопотребления используем так называемый определяемый ограничениями метод, развитый в [23], [24], [25], [26]. Метод заключается в минимизации некоторого функционала, характеризующего поведение стаи, от управления с учетом ограничений, заданных неравенствами. В решении используются принцип максимума Понтрягина и метод множителей Лагранжа.

В указанных работах метод используется для оптимизации в задаче следования. Однако метод может применяться и для других задач. Опишем, как можно осуществить оптимизацию в алгоритмах управления стай роботов, описанных в тех же терминах, что и алгоритмы к задачам (1)-(3).

Пусть рассматривается стая из N роботов. Для каждого робота i выполнены уравнения динамики:

$$\dot{\vec{x}}_i(t) = \begin{pmatrix} \dot{\vec{r}}_i(x) \\ \dot{\vec{v}}_i(x) \end{pmatrix} = \begin{pmatrix} \vec{v}_i(t) \\ \vec{u}_i(t) \end{pmatrix}, \tag{8.1}$$

где $\vec{r}_i(t), \vec{v}_i(t), \vec{u}_i(t) \in \mathbb{R}^2$. Скорость и компоненты управления ограничены по модулю:

$$\|v_i(t)\| \leq v_{max}, \tag{8.2}$$

$$|u_{ij}(t)| \leq u_{max} \text{ для каждой компоненты вектора } \vec{u}_i. \tag{8.3}$$

В дальнейшем нам необходимо формализовать поведение группы роботов. Свойства движения группы можно разделить на 2 класса: необходимые и желаемые. Необходимые свойства должны быть записаны, как неравенства

$f_k(\vec{x}(t)) \leq 0$, где f_k — гладкая функция. Выполнение неравенства означает наличие свойства у движения и наоборот. Желаемые свойства движения должны быть записаны в виде функции $\Phi_k(\vec{x}(t), t)$, непрерывно дифференцируемой по своим аргументам. Функции f_k и Φ_k принимают в качестве аргументов положение и скорость робота и его соседей, таким образом, в методе используется только локальная информация, что обусловливает его децентрализованность.

В терминах стаи роботов необходимым свойством будет безопасность. Роботы не должны приближаться друг к другу ближе, чем на $2r_{min}$. Свойство сплоченности и другие свойства, относящиеся к конкретным задачам, мы запишем в виде функций желаемых свойств движения. Свойство сплоченности имеет параметр желаемого радиуса стаи, который можно варьировать при решении задачи. Функции должны иметь экстремумы в желаемых траекториях роботов.

Искомым оптимальным управлением является минимум функционала: $\Phi_k(\vec{x}_i(t_f), t) + \frac{1}{2}\int_{t_0}^{t_f} ||\vec{u}_i||^2 dt$ — при условии выполнения введенных выше уравнений и неравенств.

Итак, для обеспечения безопасности введем ограничение, выраженное в следующем неравенстве:

$$||r_{ij}(t)|| \geq 4r_{min}^2, \forall j, \tag{8.4}$$

$r_{ij}(t)$ по-прежнему означает вектор от i робота к j роботу.

Определение 8.1. *Соседней окрестностью робота i определим множество*

$$N_i(t) = \{j : |\vec{r}_{ij}(t)| < r_{nbhd}\}. \tag{8.5}$$

Для полноценного решения должно соблюдаться условие $r_{nbhd} \geq 2r_{min}$. Робот может общаться с любым роботом из $N_i(t)$ и знать его положение. Для каждого робота траектория будет генерироваться на промежутке $t \in [t_0, t_f]$.

Чтобы построить оптимальное управление, нам нужно записать функционал над траекторией и управлением, который бы характеризовал движение системы относительно выполнения задачи.

Для задачи (1) введем следующую функцию:

Определение 8.2. *Для каждого робота i определена ошибка достижения целевой точки в виде следующей функции:*

$$\Phi_{pos}(\vec{x}_i, t_f) = \omega_1 \phi_g(\vec{x}_i, t_f) + \omega_2 \phi_a(\vec{x}_i, t_f), \tag{8.6}$$

$$\phi_g(\vec{x}_i, t_f) = ||\vec{r}_i(t_f) - \vec{r}_g||^2, \tag{8.7}$$

$$\phi_a(\vec{x}_i, t_f) = \begin{cases} (||\vec{r}_c(t_f) - \vec{r}_i(t_f)|| - D)^2, |N_i(t_0)| = 1, \\ \sum_{j \in N_i(t_0)} (\frac{\vec{r}_{ij}(t_f)\vec{r}_{ij}(t_f)}{||\vec{r}_{ij}(t_f)||} - D)^2, |N_i(t_0)| > 1. \end{cases} \tag{8.8}$$

ω_1 и ω_2 являются нормализующими параметрами, которые взвешивают влияние перемещения к целевой точке ϕ_g и сплоченности ϕ_a в поведении системы. D — желаемое расстояние между роботами. Предполагается, что роботы могут посчитать положение центра всей стаи $\vec{r}_c(t) = \frac{1}{N}\sum_{i=1}^{N} \vec{r}_i(t)$ (хотя для этого нужны более сложные датчики, чем моделировавшиеся ранее).

Для задачи (2) введем следующую функцию:

Определение 8.3. *Для каждого робота i определена ошибка следования в виде следующей функции:*

$$\Phi_{flock}(\vec{x}_i, t_f) = \omega_1\phi_d(\vec{x}_i, t_f) + \omega_2\phi_v(\vec{x}_i, t_f) + \omega_3\phi_a(\vec{x}_i, t_f), \tag{8.9}$$

$$\phi_d(\vec{x}_i, t_f) = ||\vec{v}_i(t_f) - \vec{v}_d(t_f)||^2, \tag{8.10}$$

$$\phi_v(\vec{x}_i, t_f) = ||\vec{v}_i(t_f) - \vec{v}_{avg}(t_f)||^2, \tag{8.11}$$

$$\phi_a(\vec{x}_i, t_f) = \begin{cases} (||\vec{r}_c(t_f) - \vec{r}_i(t_f)|| - D)^2, & |N_i(t_0)| = 1, \\ \sum_{j \in N_i(t_0)} (\frac{\vec{r}_{ij}(t_f)\vec{r}_{ij}(t_f)}{||\vec{r}_{ij}(t_f)||} - D)^2, & |N_i(t_0)| > 1. \end{cases} \tag{8.12}$$

ω_1, ω_2 и ω_3 являются нормализующими параметрами, которые взвешивают влияние скорости следования ϕ_d, соответствия скорости в стае ϕ_v и сплоченности ϕ_a в поведении системы. $\vec{v}_d$ — желаемая скорость, D — желаемое расстояние между роботами. Средняя скорость считается по следующей формуле:

$$\vec{v}_{avg}(t_f) = \begin{cases} \dot{\vec{r}}_c(t_f), & |N_i(t_0)| = 1, \\ \frac{1}{N_i(t_0)|}\sum_{j \in N_i(t_0)} \vec{v}_j(t_f), & |N_i(t_0)| > 1. \end{cases}$$

Для задачи (3) введем следующую функцию:

Определение 8.4. *Для каждого робота i определена ошибка агрегации в виде следующей функции:*

$$\Phi_{agg}(\vec{x}_i, t_f) = \omega_1\phi_a(\vec{x}_i, t_f), \tag{8.13}$$

$$\phi_a(\vec{x}_i, t_f) = \begin{cases} (||\vec{r}_c(t_f) - \vec{r}_i(t_f)|| - D)^2, & |N_i(t_0)| = 1, \\ \sum_{j \in N_i(t_0)} (\frac{\vec{r}_{ij}(t_f)\vec{r}_{ij}(t_f)}{||\vec{r}_{ij}(t_f)||} - D)^2, & |N_i(t_0)| > 1. \end{cases} \tag{8.14}$$

ω_1 является нормализующим параметром, D — желаемое расстояние между роботами.

Отдельные формулы на случай $|N_i(t_0)|$ обеспечивают нам то, что отделившийся от стаи робот будет возвращаться к ней. Однако это не предостерегает от случаев, когда отделяется подгруппа роботов.

Таким образом, в каждой задаче мы имеем функционал над траекторией робота, поиск минимума которого на фиксированном промежутке времени дает оптимальное управление. А именно, каждый робот строит оптимальную траекторию, решая следующую децентрализованную проблему оптимального управления.

Минимизация энергии с учетом функционала задачи.

Для каждого робота i на заданном промежутке времени $[t_i^0, t_i^f]$ ищем минимум функционала по управлению $\vec{u}_i(t)$

$$\min_{\vec{u}_i(t)} (\Phi(\vec{x}_i, t_f) + \frac{1}{2} \int_{t_i^0}^{t_i^f} ||u_i(t)||^2 dt), \tag{8.15}$$

с учетом выполнения начальных условий $\vec{x}_i(t_0) = \vec{x}_i^0$ и уравнений 8.1 и неравенств 8.2, 8.3, 8.4.

Для решения задачи допускаем следующие предположения.

Предположение 8.1. *Нет внешних возмущений и препятствий. Это предположение может быть ослаблено с введением функции оценки устойчивости.*

Предположение 8.2. *Нет задержек и ошибок в обмене информацией между роботами. В общем, было показано, что периодические обновления траектории могут быть достаточны для построения оптимальной траектории, так что предположение не сильно влияет на точность модели.*

Предположение 8.3. *Плотность стаи низкая (только одна пара роботов может приблизиться на расстояние $2r_{min}$ друг к другу в определенный момент). Это предположение может быть опущено, если роботы упорядочены между собой.*

Для решения задачи приведем ее в гамильтонову форму. На первом шаге приведем ограничение безопасности и ограничение модуля скорости к виду,

содержащему управление, продифференцировав их необходимое количество раз.

$$C_i^{safety}(t) = \begin{bmatrix} 4r_{min}^2 - \vec{r}_{ij}(t)\vec{r}_{ij}(t) \\ -\vec{r}_{ij}(t)\dot{\vec{r}}_{ij}(t) \\ -\vec{r}_{ij}(t)\ddot{\vec{r}}_{ij}(t) - \dot{\vec{r}}_{ij}(t)\dot{\vec{r}}_{ij}(t) \end{bmatrix} \leq 0, \tag{8.16}$$

$$C_i^{v}(t) = \begin{bmatrix} v_{max}^2 - \dot{\vec{r}}_{ij}(t)\dot{\vec{r}}_{ij}(t) \\ -\dot{\vec{r}}_{ij}(t)\ddot{\vec{r}}_{ij}(t) \end{bmatrix} \leq 0. \tag{8.17}$$

Пусть $f_j(t)$ — ограничения в задаче.

$$\mathcal{L}(\vec{x}_i(t), t) = l(\vec{x}_i(t_f)) + \int_{t_0}^{t_f} L(\vec{x}_i(t))dt,$$

$$L(\vec{x}_i(t))dt = \frac{1}{2}||u(\vec{x}_i(t))||^2.$$

Уравнения записываются в виде $\dot{\vec{x}}(t) = g(\vec{x}(t)) = \begin{pmatrix} \vec{v}(t) \\ \vec{u}(t) \end{pmatrix}$. Функция Понтрягина записывается в виде $H = L(\vec{x}_i(t)) + \lambda^T g(\vec{x}(t))$. К ней добавляются множители Лагранжа $\sum_j \mu_j(t) f_j(t)$.

В нашем случае функция Понтрягина будет выглядеть следующим образом:

$$\begin{aligned} H_i = &\frac{1}{2}||\vec{u}_i(t)||^2 + \vec{\lambda}_i^r(t)\vec{v}_i(t) + \vec{\lambda}_i^v(t)\vec{u}_i(t) - \\ &- \sum_{j \in N_i} \mu_{ij}(t)(\vec{r}_{ij}(t)\ddot{r}_{ij}(t) + \dot{\vec{r}}_{ij}(t)\dot{\vec{r}}_{ij}(t)) + \vec{\mu}_i^u(\vec{u}_i - \vec{u}_{max}) + \mu_i^v(\|v_i\| - v_{max}). \end{aligned} \tag{8.18}$$

Множители Лагранжа на оптимальном управлении удовлетворяют следующим условиям:

$$\mu_{ij}(t) = \begin{cases} \geq 0, \text{ если } \vec{r}_{ij}(t)\ddot{r}_{ij}(t) + \dot{\vec{r}}_{ij}(t)\dot{\vec{r}}_{ij}(t) = 0, \\ 0, \text{ если } \vec{r}_{ij}(t)\ddot{r}_{ij}(t) + \dot{\vec{r}}_{ij}(t)\dot{\vec{r}}_{ij}(t) > 0. \end{cases} \tag{8.19}$$

$$\mu_i^u(t) = \begin{cases} \geq 0, \text{ если } u_i(t) - u_{max} = 0, \\ 0, \text{ если } u_i(t) - u_{max} < 0. \end{cases} \tag{8.20}$$

$$\mu_i^v(t) = \begin{cases} \geq 0, \text{ если } \|v_i(t)\| - v_{max} = 0, \\ 0, \text{ если } \|v_i(t)\| - v_{max} < 0. \end{cases} \tag{8.21}$$

Для получения траектории рассмотрим случаи, где выполняются ограничения и где они нарушаются, и составим ее из участков, относящихся к соответствующим случаям.

9. Свободное движение

Свободным движением мы называем траекторию, на которой ни одно из неравенств не обращается в равенство. В этом случае $\mu_{ij} = \mu_i^u = \mu_i^v = 0$. Необходимое условие оптимального управления, условие стационарности, выглядит следующим образом:

$$\frac{\partial H_i}{\partial \vec{u}_i} = \vec{u}_i + \vec{\lambda}_i^v = 0. \tag{9.1}$$

Уравнения сопряженной системы выглядят так:

$$\begin{cases} \dot{\vec{\lambda}}_i^r = -\frac{\partial H_i}{\partial \vec{r}_i} = 0, \\ \dot{\vec{\lambda}}_i^v = -\frac{\partial H_i}{\partial \vec{v}_i} = -\lambda_i^r. \end{cases} \tag{9.2}$$

$$\begin{cases} \vec{\lambda}_i^r = \vec{a}_i, \\ \vec{\lambda}_i^v = -(\vec{a}_i t + \vec{b}_i). \end{cases} \tag{9.3}$$

$$\begin{cases} \vec{u}_i(t) = \vec{a}_i t + \vec{b}_i, \\ \vec{v}_i(t) = \frac{1}{2}\vec{a}_i t + \vec{b}_i t + \vec{c}_i, \\ \vec{r}_i(t) = \frac{1}{6}\vec{a}_i t^3 + \frac{1}{2}\vec{b}_i t^2 + \vec{c}_i t + \vec{d}_i. \end{cases} \tag{9.4}$$

Константы $\vec{a}_i, \vec{b}_i, \vec{c}_i, \vec{d}_i$ находятся из начальных условий и условий трансверсальности:

$$\begin{cases} \vec{x}_i(t_0) = \vec{x}_i^0, \\ \vec{\lambda} = \frac{\partial \Phi(\vec{x}_i, t)}{\partial \vec{x}_i}|_{t_f}. \end{cases} \tag{9.5}$$

Для задачи (1) условия трансверсальности записываются так:
$\vec{\lambda}_i^r(t_f) = 2(\omega_1(\vec{r}_i(t_f) - \vec{r}_g) - \omega_2 \sum_{j \in N_i(t_0)}((||\vec{r}_{ij}|| - D)\vec{r}_{ij}(t_f)))$,
$\vec{\lambda}_i^v(t_f) = 0$.
После замены левой части на равные величины получим

$$\begin{cases} \vec{r}_i(t_f) = \frac{\vec{a}_i}{2(\omega_1 - \omega_2 \sum_j (||\vec{r}_{ij}(t_f)|| - D))} + \omega_1 \vec{r}_g + \omega_2 \sum_{j \in N_i} \vec{r}_j(t_f), \\ \vec{u}_i(t_f) = 0. \end{cases} \tag{9.6}$$

Из этих уравнений находится решение в задаче (1).

Для задачи (2) условия трансверсальности записываются так:
$\vec{\lambda}_i^r(t_f) = -2\omega_3 \sum_{j \in N_i(t_0)}((||\vec{r}_{ij}|| - D)\vec{r}_{ij}(t_f))$,
$\vec{\lambda}_i^v(t_f) = 2\omega_2(\vec{v}_i(t_f) - \vec{v}_{avg}) + 2\omega_1(\vec{v}_i(t_f) - \vec{v}_d)$.
После замены левой части на равные величины получим

$$\begin{cases} \vec{r}_i(t_f) = \frac{\vec{a}_i}{2\omega_3 \sum_j (||\vec{r}_{ij}(t_f)|| - D)} + \sum_{j \in N_i} \vec{r}_j(t_f), \\ \vec{u}_i(t_f) = -2\omega_2(\vec{v}_i(t_f) - \vec{v}_{avg}) - 2\omega_1(\vec{v}_i(t_f) - \vec{v}_d). \end{cases} \tag{9.7}$$

Из этих уравнений находится решение в задаче (2).

Для задачи (3) условия трансверсальности записываются так:
$\vec{\lambda}_i^r(t_f) = -2\omega_1 \sum_{j\in N_i(t_0)}((||\vec{r}_{ij}|| - D)\vec{r}_{ij}(t_f)))$,
$\vec{\lambda}_i^v(t_f) = 0$.

После замены левой части на равные величины получим

$$\begin{cases} \vec{r}_i(t_f) = \frac{\vec{a}_i}{2\omega_1 \sum_j (||\vec{r}_{ij}(t_f)||-D)} + \sum_{j\in N_i} \vec{r}_j(t_f), \\ \vec{u}_i(t_f) = 0. \end{cases} \tag{9.8}$$

Из этих уравнений находится решение в задаче (3).

Значения $\vec{b}_i, \vec{c}_i, \vec{d}_i$ можно выразить явно, значения $\vec{a}_i$ считаются численно.

9.1. Движение под действием ограничения безопасности

Уравнение стационарности в этом случае будет выглядеть следующим образом:

$$\vec{u}_i(t) = -\vec{\lambda}_i^v(t) - \sum \mu_{ij}(t)\vec{r}_{ij}(t). \tag{9.9}$$

Уравнения сопряженной системы представляются в виде:

$$\begin{cases} -\dot{\vec{\lambda}}_i^v(t) = \vec{\lambda}_i^r(t) + \sum \mu_{ij}(t)\dot{\vec{r}}_{ij}(t), \\ -\dot{\vec{\lambda}}_i^r(t) = \sum \mu_{ij}(t)\ddot{\vec{r}}_{ij}(t). \end{cases} \tag{9.10}$$

Объединив эти уравнения с 8.1 и подставив 9.9 и 8.16, получаем линейную неоднородную систему, которую необходимо решать численно. Граничными условиями являются начальные условия и условия трансверсальности.

В [26] выводятся дифференциальные уравнения на скорости и множители Лагранжа, позволяющие применить гибридный оптимизационный метод и показать, что движение при нулевой относительной скорости между роботами i и j является оптимальным для участка с ограничением безопасности.

9.2. Движение под действием максимального управления

$|\vec{u}_i(t)| = u_{max}, t \geq t_1, ||\vec{v}_i(t)|| < v_{max}$

Функция Понтрягина в таком случае непрерывна, и можно подставить значение управления в 8.1.

$$\begin{cases} \vec{v}_i(t) = \vec{u}_{max}t + \vec{v}_{0i}, \\ \vec{r}_i(t) = \frac{1}{2}\vec{u}_{max}t^2 + \vec{v}_{0i}t + \vec{r}_{0i}. \end{cases} \tag{9.11}$$

Значение констант находится в момент t_1.

9.3. Движение при максимальной скорости

Для рассмотрения случая с максимальной скоростью представим скорость как функцию модуля скорости и угла Θ направления скорости:

$$\vec{v_i} = \begin{pmatrix} \|v_i\| cos\Theta \\ \|v_i\| sin\Theta \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}.$$

Из этого можно вывести, что угол Θ зависит от управления по следующему дифференциальному закону:

$$\Theta = -\frac{u_1 sin\Theta + u_2 cos\Theta}{v_{max}}.$$

Ограничение по скорости дает следующее равенство: $u_1 v_1 + u_2 v_2 = 0$.

Уравнение стационарности дает 2 уравнения, с помощью которых выразим значение множителя Лагранжа:

$$u_1 - \lambda_1^v + \mu^v v_1 = 0,$$

$$u_2 - \lambda_2^v + \mu^v v_2 = 0,$$

$$\mu^v = -\frac{\lambda_1^v v_1 + \lambda_2^v v_2}{v_{max}^2}.$$

Из этих уравнений можно выразить сами компоненты управления:

$$\begin{cases} u_1 = -\lambda_1^v sin^2\Theta + \lambda_2^v sin\Theta cos\Theta, \\ u_2 = \lambda_1^v cos\Theta sin\Theta - \lambda_2^v cos^2\Theta. \end{cases} \tag{9.12}$$

При подстановке управлений дифференциальное уравнение на Θ станет следующим:

$$\dot{\Theta} = \frac{\lambda_1^v sin\Theta - \lambda_2^v cos\Theta}{v_{max}}. \tag{9.13}$$

Уравнения на сопряженные переменные в этом случае выглядят следующим образом:

$$\begin{cases} \dot{\vec{\lambda}}^r = 0, \\ \dot{\vec{\lambda}}^v = -\vec{\lambda}^r - \mu^v \vec{u}. \end{cases} \tag{9.14}$$

Таким образом, уравнения 9.13 и 9.14 составляют уравнения движения в случае ограничения по скорости, которые можно посчитать численно. Граничные значения $\vec{\lambda}^r, \vec{\lambda}^v$ находятся из уравнений трансверсальности в момент, когда модуль скорости робота перестанет быть максимальным. Для угла Θ и положения (x, y) граничными условиями являются значения на момент достижения максимальной скорости.

9.4. Движение при нескольких ограничениях

На траектории возможны моменты (или отрезки) времени, когда выполняется несколько равенств, выражающих ограничение при движении, и в эти моменты управление будет определяться особенным образом. Для получения управления в этих случаях нам необходимо попарно рассмотреть выполнение условий, а затем случай, когда все условия выполняются.

1. $\|v_i\| = v_{max}, |u_{i1}| = u_{max}$

Заметим, что принимается, что u_{i1} принимает максимальное по модулю значение, однако случай максимального значения у u_{i2} рассматривается аналогично. Ограничивающие равенства в этом случае можно записать следующим образом:

$$\begin{pmatrix} v_{i1} & v_{i2} \\ \pm 1 & 0 \end{pmatrix} \begin{pmatrix} u_{i1} \\ u_{i2} \end{pmatrix} = \begin{pmatrix} 0 \\ u_{max} \end{pmatrix}.$$

Если $v_{i2} \neq 0$, то матрица $\begin{pmatrix} v_{i1} & v_{i2} \\ \pm 1 & 0 \end{pmatrix}$ обратима, управления выражаются однозначно с ее помощью, и дальнейшее решение проводится численно аналогично случаю с движением при максимальной скоростью.

Если $v_{i2} = 0$, то принимаем $u_{i1} = 0$. Теперь выполняется неравенство $|u_i| < u_{max}$, и решение также сводится к решению для движения при максимальной скорости.

2. $\|v_i\| = v_{max}, |u_{i1}| = |u_{i2}| = u_{max}$

В этом случае количество равенств связей превышает размерность вектора управления, поэтому невозможно построить непрерывное управление. Управление с разрывами будет оптимально, если в точках разрыва гамильтониан H, сопряженные переменные $\vec{\lambda}$ и $\frac{\partial H}{\partial u}$ остаются непрерывными, т.е. должны выполняться следующие уравнения:

$$\begin{cases} \vec{\lambda}^r(t^-) = \vec{\lambda}^r(t^+) + \pi \frac{\partial N(t)}{\partial \vec{r}}, \\ \vec{\lambda}^v(t^-) = \vec{\lambda}^v(t^+) + \pi \frac{\partial N(t)}{\partial \vec{v}}, \\ H(t^-) = H(t^+) + \pi \frac{\partial N}{\partial t}, \\ \frac{\partial H(t^-)}{\partial u} = \frac{\partial H(t^+)}{\partial u}. \end{cases}$$

Система разрешима относительно управления, управление при этом становится меньше максимума по модулю, и далее вычисляется траектория движения при максимальной скорости.

3. $\|r_{ij}\| = 2r_{min}, |u_{i1}| = u_{max}$

Так как уравнения связей представимы в виде

$$\begin{pmatrix} r_{i1}-r_{j1} & r_{i2}-r_{j2} \\ \pm 1 & 0 \end{pmatrix} \begin{pmatrix} u_{i1} \\ u_{i2} \end{pmatrix} = \\ = \begin{pmatrix} (r_{i1}-r_{j1})u_{j1} + (r_{i2}-r_{j2})u_{j2} + (v_{j1}-v_{i1})^2 + (v_{j2}-v_{i2})^2 \\ u_{max} \end{pmatrix}.$$

Матрица обратима и существует единственное решение для управления, если $r_{i2} - r_{j2} \neq 0$, иначе разрешим первое уравнение относительно u_{i1}:

$$u_{i1} - u_{j1} = -\frac{(v_{j1}-v_{i1})^2 + (v_{j2}-v_{i2})^2}{r_{i1}-r_{j1}}.$$

Правая часть равенства должна быть ограничена сверху по модулю, чтобы управление не превысило максимальное, а именно

$$|\frac{(v_{j1}-v_{i1})^2 + (v_{j2}-v_{i2})^2}{r_{j1}-r_{i1}}| \leq \frac{4v_{max}^2}{r_{min}} \leq max|u_{i1}-u_{j1}| = 2u_max,$$

$$u_{max} \geq \frac{2v_{max}^2}{r_{min}}. \tag{9.15}$$

Если такое условие выполняется, то мы всегда можем разрешить систему и перейти к движению при ограничении безопасности.

4. $\|r_{ij}\| = 2r_{min}, |u_{i1}| = |u_{i2}| = u_{max}$

Аналогично случаю с максимальными скоростью и управлением строится разрывное управление из уравнений непрерывности для H, $\vec{\lambda}$ и $\frac{\partial H}{\partial u}$ при $N = \begin{pmatrix} 4r_{min}^2 - (\vec{r}_{ij}, \vec{r}_{ij}) \\ -2(\vec{r}_{ij}, \dot{\vec{r}}_{ij} \end{pmatrix}$. В этот раз решение системы по управлению неоднозначно и всегда возможно только в задачах (1) и (3).

5. $\|r_{ij}\| = 2r_{min}, \|v_i\| = v_{max}$

Уравнения ограничений записываются в следующем виде:

$$\begin{pmatrix} r_{i1}-r_{j1} & r_{i2}-r_{j2} \\ v_{i1} & v_{i2} \end{pmatrix} \begin{pmatrix} u_{i1} \\ u_{i2} \end{pmatrix} = \\ = \begin{pmatrix} (r_{i1}-r_{j1})u_{j1} + (r_{i2}-r_{j2})u_{j2} + (v_{j1}-v_{i1})^2 + (v_{j2}-v_{i2})^2 \\ 0 \end{pmatrix}.$$

Матрица обратима тогда и только тогда, когда векторы $\vec{v}_i$ и $\vec{r}_{ij}$ не параллельны. Если эти вектора параллельны и сонаправлены, то управление находится из первого уравнения, если $(r_{i1}-r_{j1})u_{j1}+(r_{i2}-r_{j2})u_{j2}+(v_{j1}-v_{i1})^2+(v_{j2}-v_{i2})^2 < 0$, и из второго в ином случае. Если вектора параллельны и противоположно

направлены, то неравенства могут не иметь решения, и необходимо применять алгоритмы с упорядоченным построением управления.

6. $\|r_{ij}\| = 2r_{min}, \|v_i\| = v_{max}, |u_{i1}| = u_{max}$

Управление будет разрывным, как в предыдущем случае, поэтому решение будет аналогичным при условии соблюдения непрерывности гамильтониана, его производной по управлению и сопряженных переменных.

9.5. Построение управляющего сигнала

Проинтегрировав уравнения в предыдущих пунктах, мы получим значения ускорения робота, которое должно действовать на него для реализации траектории. Воспользовавшись 2.1, можно привести эти ускорения к координатам ϕ_1, Θ.

$$\begin{cases} \ddot{x} = (b\ddot{\Theta} + r_1\ddot{\phi}_1)cos\Theta - \dot{\Theta}(b\dot{\Theta} + r_1\dot{\phi}_1)sin\Theta, \\ \ddot{y} = (b\ddot{\Theta} + r_1\ddot{\phi}_1)sin\Theta + \dot{\Theta}(b\dot{\Theta} + r_1\dot{\phi}_1)cos\Theta. \end{cases}$$

Из этих уравнений можно выразить $\ddot{\phi}_1, \ddot{\Theta}$ через сами углы, их скорости и управления. Затем, подставив углы, их скорости и ускорения в уравнения Чаплыгина 2.6, можно получить выражение для моментов.

9.6. Алгоритм построения траектории

Для генерации траектории используем следующие определения.

Определение 9.1. *Множеством контакта называется* $\nu_i(t) = \{j|\mu_{ij}(t) > 0, j \neq i\}$.

Определение 9.2. *Контактным интервалом для робота* i *называется такой интервал* $\tau_i^k \subset [t_0, t_f]$, *что* $\forall t_a, t_b \in \tau_i^k, \nu_i(t_a) = \nu_i(t_b) = \varnothing$, *для которых верно, что* $\forall p, q \in \mathbb{N}, p \neq q : t_p \in \tau_i^p, t_q \in \tau_i^q, t_p < t_q$.

Контактный интервал обозначает временной интервал для робота i, в который нарушается ограничение безопасности при свободном движении. Для предотвращения столкновений на этих интервалах принимается централизованное решение случая с ограничением безопасности. Окончательный алгоритм, выполняющийся параллельно на каждом роботе стаи, устроен так:

- роботы строят траекторию при свободном движении на всем временном интервале;
- роботы обмениваются траекториями с роботами из соседней окрестности;

- роботы вычисляют множества контакта $\nu_i(t)$;
- роботы строят движение под действием ограничения безопасности на каждом контактном интервале;
- на интервалах между контактными интервалами строятся траектории при свободном движении с сохранением непрерывности траектории;
- строятся траектории при свободном движении для начального и конечного интервала.

Заключение

В данной работе построены математическая и компьютерная модели дифференциального привода, которые вместе позволяют построить схему управления группой роботов, отражающих стайное поведение. Построен децентрализованный алгоритм перемещения группы роботов к целевой точке, следования за лидером и сбора в одной точке. Реализация алгоритма позволила наблюдать особенности стайного поведения, такие как, например, образование очереди при прохождении узостей, эффект обтекания препятствий группой роботов и др. Введены классы оценок и функции оценки движения стаи при выполнении задач. Проведено моделирование движения при различных параметрах алгоритма. Произведен анализ зависимостей оценок от параметров алгоритма. Обобщен алгоритм построения децентрализованного энергетически оптимального управления для группы роботов в различных задачах. Приведено решение задачи оптимального управления движением стаи из задач (1)-(3).

Набор исследованных методов позволяет легко в алгоритмическом и вычислительном смыслах прототипировать поведение стаи алгоритмами управления на основе локальных взаимодействий и затем преобразовать этот алгоритм в оптимальный по энергии и, возможно, дополнительным параметрам децентрализованный алгоритм. Однако в данной работе недостает дополнительных исследований алгоритма. В первую очередь, это его применение на реальных роботах, так как в алгоритмах есть несколько предположений, сильно упрощающих взаимодействие роботов. Необходимо оценить вычислительную сложность алгоритма оптимизации. Также моделирование при большем количестве сценариев поможет сделать более утвердительные выводы о поведении стаи при изменении параметров алгоритма. Экспериментальное сравнение с текущими алгоритмами оптимизации поможет определить границы применимости алгоритма.

Список литературы

1. *Reynolds C.* Flocks, Herds, and Schools: A Distributed Behavioral Model // ACM SIGGRAPH Computer Graphics. 1987. 21(4). Pp. 25–34.

2. *Spears W.M., Spears D.F., Heil R., Kerr W., Hettiarachchi S.* An Overview of Physicomimetics // Lecture Notes in Computer Science. 2005. Vol 3342. Pp. 84–97.

3. *Matarić, M. J.* Designing and Understanding Adaptive Group Behavior // Adaptive Behavior. 1995. 4(1). Pp. 51—80. 1995

4. *Duarte M., Costa V., Gomes J., Rodrigues T., Silva F., Oliveira S., Christensen A.* Evolution of Collective Behaviors for a Real Swarm of Aquatic Surface Robots // PLoS ONE. 2016. 11(3): e0151834.

5. *Schmickl T., Thenius R., Moslinger C., Timmis J., Tyrrell A., Read M., Hilder J., Halloy J., Campo A., Stefanini C., Manfredi L., Orofino S., Kernbach S., Dipper T., Sutantyo D.* CoCoRo - The self-aware underwater swarm // SASO 2011 - Fifth IEEE International Conference on Self-Adaptive and Self-Organizing Systems. 2011. Pp. 120–126.

6. *Bayindir L., Sahin E.* A review of studies in swarm robotics // Turkish Journal of Electrical Engineering and Computer Sciences. 2007. 15. Pp. 115-147.

7. *Brambilla, M., Ferrante, E., Birattari, M. et al.* Swarm robotics: a review from the swarm engineering perspective // Swarm Intelligence. 2013. 7. Pp. 1–41.

8. *Rossi F., Bandyopadhyay S., Wolf M., Pavone M.* Review of Multi-Agent Algorithms for Collective Behavior: a Structural Taxonomy // arXiv preprint. 2015.

9. *Jahanshahi, M.R., Shen, W., Mondal, T.G. et al.* Reconfigurable swarm robots for structural health monitoring: a brief review // International Journal of Intelligent Robotics and Applications. 2017. Vol. 1. Pp. 287—305.

10. *Gutelius B., Theodore N.* The Future of Warehouse Work: Technological Change in the U.S. Logistics Industry // UC Berkeley Labor Center. 2019.

11. *Spears, W.M., Spears, D.F., Hamann, J.C. et al.* Distributed, Physics-Based Control of Swarms of Vehicles // Autonomous Robots. 2004. 17. Pp. 137-–162.

12. *Nagy G., Vaughan R.* Self-Organization of a Robot Swarm into Concentric Shapes // 2017 14th Conference on Computer and Robot Vision (CRV). 2017. Pp. 155–160.

13. *Chen J., Gauci M., Price M., Groß R.* Segregation in Swarms of e-puck Robots Based on the Brazil Nut Effect. // Proceedings of the 11th International Conference on Autonomous Agents and Multi-Agent Systems. 2012. Vol. 1. Pp. 163–170.

14. *Павловский В.Е., Павловский В.В.* Математическая модель двумерной гомогенной стаи роботов // Искусственный интеллект и принятие решений. 2015. 4. С. 62–71.

15. *Kolling A., Walker P., Chakraborty N., Sycara K., Lewis M.* Human Interaction With Robot Swarms: A Survey // IEEE Transactions on Human-Machine Systems. 2016. Vol. 46(1). pp. 9–26.

16. *Hussein A., Ghignone L., Nguyen T., Salimi N., Nguyen H., Wang M., Abbass H.*Towards Bi-Directional Communication in Human-Swarm Teaming: A Survey // arXiv preprint. 2018.

17. *Parker L., Butterworth J. and Luo S.* Fly Safe: Aerial Swarm Robotics using Force Field Particle Swarm Optimisation // arXiv preprint. 2019.

18. *Chen A., Harwell J., Gini M.* Maximizing Energy Battery Efficiency in Swarm Robotics // arXiv preprint. 2019.

19. *Castello, E., Yamamoto, T., Libera, F.D. et al.* Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach // Swarm Intell. 2016. 10. Pp. 1—31.

20. *Furferi, R., Conti, R., Meli, E., Ridolfi, A.* Optimization of potential field method parameters through networks for swarm cooperative manipulation tasks // International Journal of Advanced Robotic Systems. 2016. 13(5). Pp. 1–13.

21. *Gomes, J., Urbano, P., Christensen, A.L.* Evolution of swarm robotics systems with novelty search // Swarm Intell. 2013. 7. Pp. 115-–144.

22. *Vásárhelyi G., Virágh Cs., Somorjai G., Nepusz T., Eiben A. E., Vicsek T.* Optimized flocking of autonomous drones in confined environments // Science Robotics. 2018. 3(20). eaat3536.

23. *Notomista G., Egerstedt M.* Constraint-driven coordinated control of multi-robot systems // arXiv preprint. 2018.

24. *Beaver L.E., Malikopoulos A.A.* An energy-optimal framework for assignment and trajectory generation in teams of autonomous agents // Systems & Control Letters. 2020. 138:104670.

25. *Beaver L.E., Malikopoulos A.A.* Beyond Reynolds: A Constraint-Driven Approach to Cluster Flocking // arXiv preprint. 2020.

26. *Beaver L.E., Kroninger C., Malikopoulos A. A.* An Optimal Control Approach to Flocking // arXiv preprint. 2019.

27. *Голубев Ю. Ф.* Основы теоретической механики: Учебник. 3-е издание, переработанное и дополненное // Издательство Московского университета. 2019.

28. *Sardinha H., Dragone M., Vargas P.*Closing the Gap in Swarm Robotics Simulations: An Extended Ardupilot/Gazebo plugin // arXiv preprint. 2018.

29. *Погорелов Д.Ю* Компьютерное моделирование динамики технических систем с использованием программного комплекса "Универсальный механизм"// Вестник компьютерных и информационных технологий. 2005.

30. *Pitonakova L., Giuliani M., Pipe A., Winfield A.* Feature and Performance Comparison of the V-REP, Gazebo and ARGoS Robot Simulators // Towards Autonomous Robotic Systems. TAROS 2018. Lecture Notes in Computer Science. 2018. Vol. 10965. Pp. 357–368.

31. *Soria E., Schiano F., Floreano D.* The Influence of Limited Visual Sensing on the Reynolds Flocking Algorithm // 2019 Third IEEE International Conference on Robotic Computing (IRC). 2019. pp. 138–145.

32. *Croix J-P.,Egerstedt M.* Analyzing human-swarm interactions using control Lyapunov functions and optimal control // Networks & Heterogeneous Media. 2015. 10(3): 609–630.

33. *Павловский В.Е., Евграфов В.В., Павловский В.В., Петровская Н.В.* Динамика, моделирование, управление мобильными роботами // Искусственный интеллект и принятие решений. 2009. 2.

34. *Ziegler J. B., Nichols N. B.* Optimum settings for automatic controllers // Transactions of the ASME. 1942. 64. 759–768.

Содержание

Введение 3
1 Постановка задачи 5
2 Математическая модель робота 7
2.1 Кинематика дифференциального привода 7
2.2 Динамика исследуемой модели 9
3 Компьютерная модель робота 11
4 Алгоритм управления группой роботов 12
4.1 Построение вектора скорости 13
4.2 Уклонение от препятствий 15
4.3 Проверка возможных столкновений 16
4.4 Расчет управляющих моментов 17
4.5 Дополнения алгоритмов в задачах (1)-(3) 18
5 Функции оценки эксперимента при различных параметрах алгоритма 18
6 Моделирование движения 20
7 Зависимость оценок алгоритма от его параметров 21
8 Оптимизация стайного движения 23
9 Свободное движение 28
9.1 Движение под действием ограничения безопасности . . . 29
9.2 Движение под действием максимального управления . . 29
9.3 Движение при максимальной скорости 30
9.4 Движение при нескольких ограничениях 31
9.5 Построение управляющего сигнала 33
9.6 Алгоритм построения траектории 33
Список литературы 35