

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (НОВОСИБИРСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет **ФИЗИЧЕСКИЙ**

Кафедра **АВТОМАТИЗАЦИИ ФИЗИКО-ТЕХНИЧЕСКИХ ИССЛЕДОВАНИЙ**

Направление подготовки: **03.03.02 ФИЗИКА**

Образовательная программа: **БАКАЛАВРИАТ**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

ПОПЕЛЕНКОВ ВЛАДЛЕН ВЯЧЕСЛАВОВИЧ

(Фамилия, Имя, Отчество автора)

Тема работы **РАЗРАБОТКА АЛГОРИТМА УПРАВЛЕНИЯ**
ГРУППИРОВКОЙ
МОБИЛЬНЫХ РОБОТОВ

«К защите допущена»

Заведующий кафедрой АФТИ ФФ

К.Т.Н.
ученая степень, звание

Лысаков К. Ф./.....
(фамилия И., О.) / (подпись, МП)

«.....».....20...г

Научный руководитель

К.Т.Н.
ученая степень, звание

ст. науч. сотр., ИАиЭ СО РАН
должность, место работы

Котов К. Ю./.....
(фамилия И., О.) / (подпись, МП)

«.....».....20...г

Дата защиты: «.....».....20...г.

Новосибирск, 2017

Содержание

Введение.....	3
1. Литературный обзор.....	5
2. Постановка задачи и разработка алгоритма управления.....	12
3. Результаты численного моделирования.....	26
Заключение.....	33
Список литературы.....	34

Введение

В настоящее время активно развиваются алгоритмы управления группировкой автономных объектов. Данная задача актуальна в различных областях жизнедеятельности, например в целях транспортировки, поиска, наблюдения, разведки, в труднодоступных местах. Группа может выполнять поставленное задание лучше одного мобильного робота, в случаях разбиения задания на составляющие его компоненты или распределения нагрузки на группу роботов.

Децентрализованная система управления позволяет каждому роботу быть независимой транспортной единицей и потери одного из роботов должны не нарушать выполнения цели поставленного задания или восполняться новым роботом. Дороговизна мобильных роботов и неустойчивость при столкновениях требуют высоких гарантий надежности для алгоритмов.

Автономные единицы собирают данные о внешнем мире средствами восприятия, анализируют собранные данные, планируют стратегию движения и реализуют ее. Для эффективного управления роботами необходимо изучение механической системы, как конфигурации физики роботов, так и механики сред в которых они управляются. Изучаемые механические свойства системы задаются дифференциальными уравнениями, которые являются приближениями к этим свойствам.

На данный момент наибольший интерес представляет изучение роя, стада или толпы среди живых организмов (насекомых, животных, людей и т.д.). Этот самоорганизующийся субъект представляет собой сложное и скоординированное коллективное поведение, достижение результатов которого невозможно добиться одиночными индивидами. Известным примером является кормовая деятельность муравьев, где каждый муравей следует ряду простых правил, но колония в целом представляет сложный процесс, что повышает эффективность выживания муравьев [1].

Групповое поведение замечено в толпе людей не согласованных друг с другом. Например, в случаях очень плотной толпы, в переходах движение людей можно описать в виде потоков [2, 3]. Исследования предполагают, что индивид в толпе стремится к доминирующему потоку. Таким образом, потоки очень плотной толпы напоминают потоки жидкости. На основе данной модели поведение толпы изучается при помощи физического взаимодействия [4].

Собственные поведения индивидов не позволяют охватывать весь спектр состояний толпы в модели потоков, что требует рассматривать их в качестве отдельных единиц. В модели автономных единиц, посредством чувств восприятия, взаимодействие индивида с объектами внешнего мира определяется суперпозицией взаимодействий с каждым объектом. Взаимодействие индивида с конкретным объектом основывается на его когнитивных решениях, которые в некоторых приближениях можно определить физическим взаимодействием [5].

1. Литературный обзор

Основные классификации управления

Существует множество известных алгоритмов управления группировкой мобильных роботов. В зависимости от задачи специфика управления может сильно различаться. Начиная от отличий среде, в которой они движутся и, заканчивая способом транспортировки конкретного груза. Все эти алгоритмы можно классифицировать на три типа: ситуативные методы, методы потенциальных функций и методы, основанные на поведении индивида в коллективе.

Ситуативные методы

Характерной особенностью ситуативных методов является управление роботами при помощи строгих правил. Эти правила работают в рамках конкретных изученных ситуаций, и они базируются на задании дискретных значений, условий, ветвлений. Данные методы зачастую имеют гарантии надежности, получения требуемого результата и оценки производительности. Ситуативные методы хорошо работают как в рамках одной автономной единицы, так и очень большого числа мобильных роботов.

В задачах автономного управления транспортным средством изучаются вопросы избегания столкновений с препятствием на проезжей части, в том числе пешехода [6]. Для решения данной задачи разрабатываются системы распознавания объектов и оценивания их скорости, при помощи при помощи сенсорного восприятия. А так же системы принятия решения связанные с оптимальным совершением маневра и реализация данного маневра. Архитектура данных мероприятий представлена на рисунке 1.

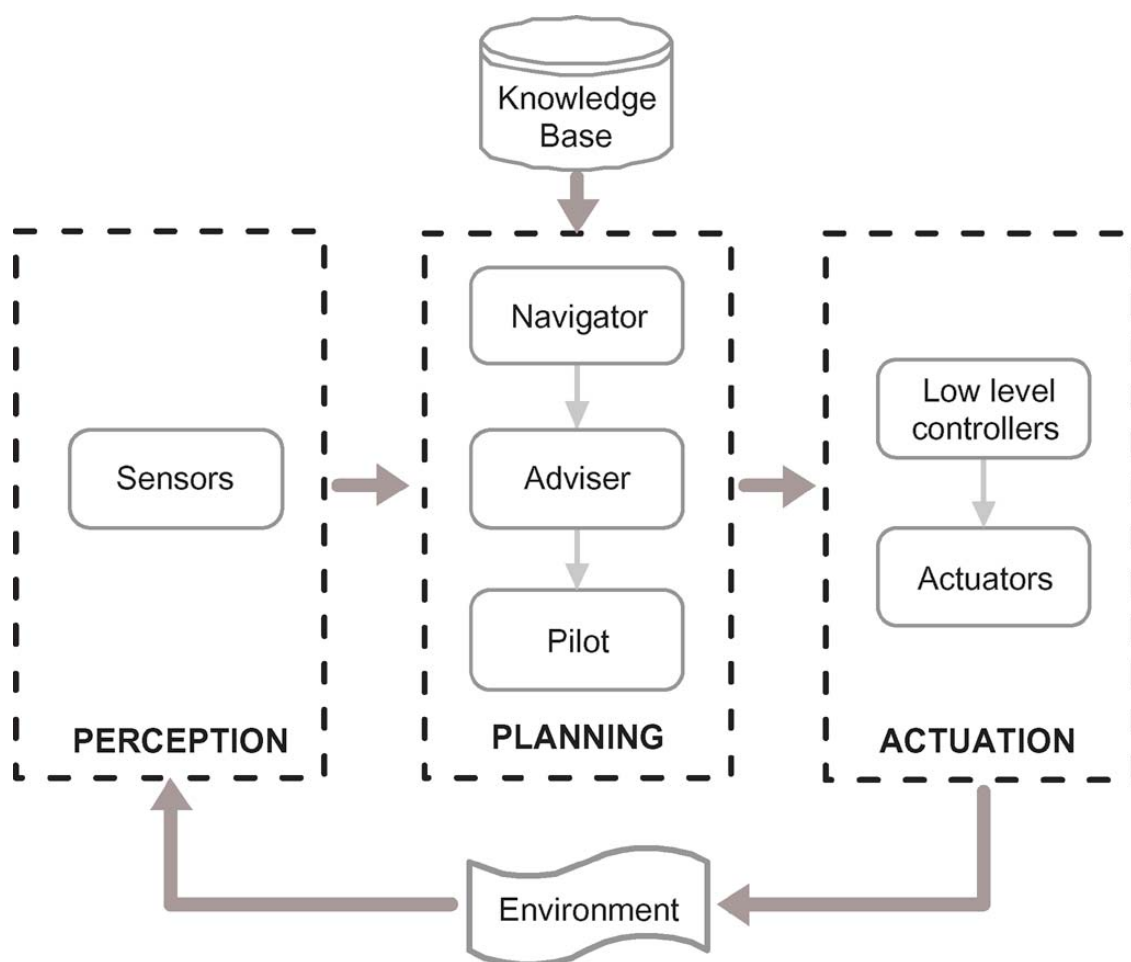


Рисунок 1. Архитектура этапов управления.

Оптимальным совершением маневра является исполнение наиболее безопасного решения для всех объектов участвующих в дорожно-транспортной системе. В случаях, где торможение транспортного средства не может предотвратить столкновения, предпринимаются попытки разрешить ситуацию перестроением транспортного средства, представленного на рисунке 2, на другую полосу проезжей части и дальнейшее возвращение на исходную полосу.

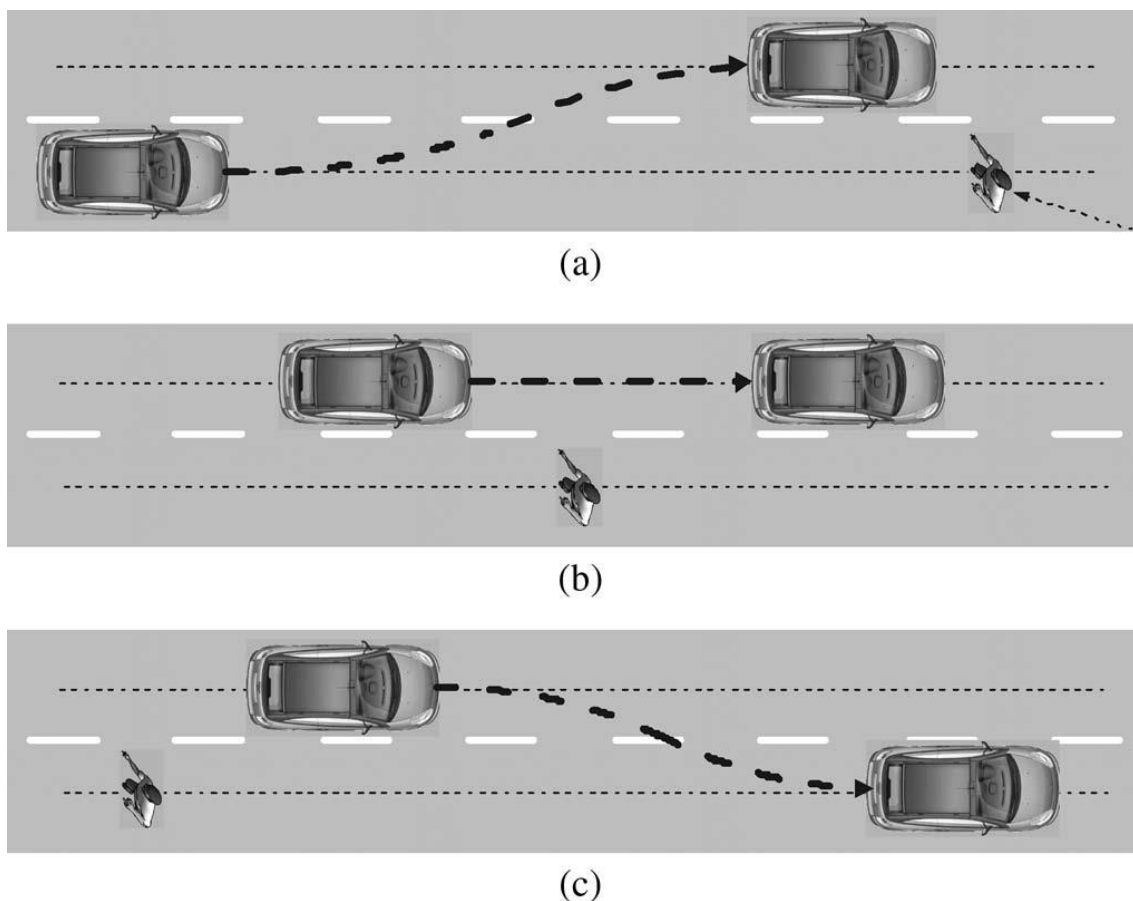


Рисунок 2. Этапы предотвращения столкновения с пешеходом. (а) – перестроение на полосу со свободным путем движения, (б) - отслеживание пути пока не будет преодолено препятствие, (с) - смена полосы движения в исходное положение.

Для роботов, где их число превышает тысячи, изучаются алгоритмы создания двумерных форм из самоорганизующейся группы роботов [7]. Роботы в данных алгоритмах образуют формы, выдерживая дистанции между соседями таким образом, чтобы они представляли модель шестиугольной или четырехугольной сетки. Данную модель они реализуют при помощи дальномерного средства восприятия, вычисляя расстояния до соседних роботов. Начало отсчета в сетчатой модели координат ведется от стационарных роботов, занявших позицию на этой сетке. Последующие роботы двигаются по части поверхности задаваемой формы, образованной стационарными роботами, пока не достигнут ее границы, что показано на рисунке 3.

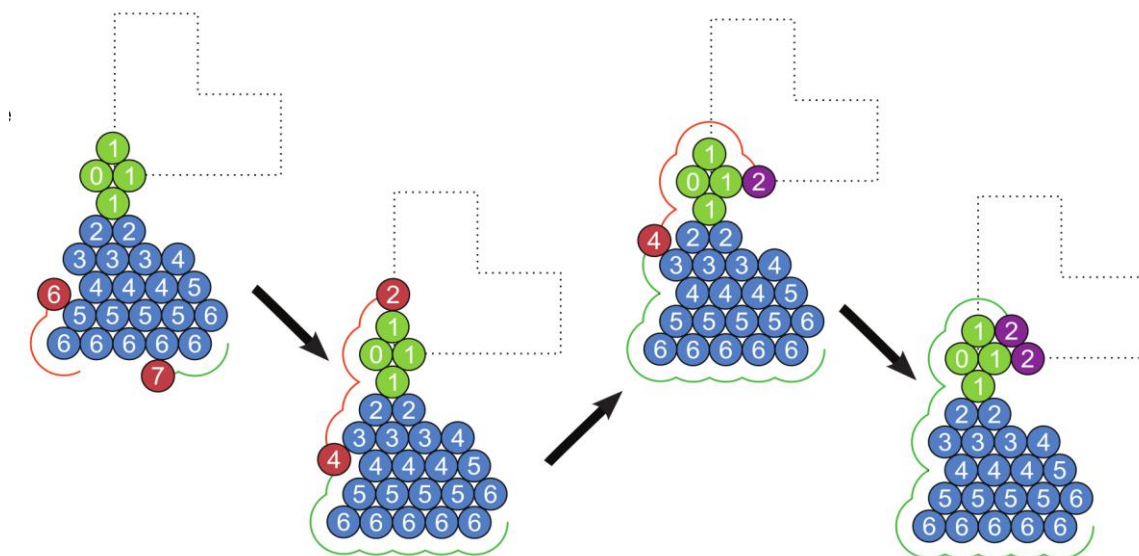


Рисунок 3. Этапы движения роботов вдоль поверхности формы.

Данный алгоритм создания форм имеет ряд проблем и ограничений, таких как: создание только односвязных форм, линейно увеличивающаяся погрешность деформации сетки, квадратично увеличивающееся время исполнения поставленной задачи.

Методы потенциальных функций

Алгоритмы, в которых ключевой особенностью является управление роботами в сплошных средах, называются методами потенциальных функций. Данные методы применимы во многих сферах деятельности, где взаимодействующие силы имеют непрерывный характер, например в космосе или текущих средах таких как, воздух, вода. В силу непрерывности сил в данных системах мы можем описать состояние пространства набором гладких потенциалов. Для задач об оптимальном перемещении в текущих средах возможно составление системы дифференциальных уравнений с искомыми значениями и их вычисление.

На примере автономных подводных планеров для эффективного отбора проб океана, были разработаны алгоритмы с низким потреблением энергии [8]. Данные планеры рассчитаны на работу в течение

нескольких месяцев. Движение производится путем маневрирования между течениями воды в океане. Управление планерами имеет две модели, slocum - путем отклонения внешнего руля (а) или spray - путем смещения внутренней массы (b), что представлено на рисунке 4.



(a) Slocum планер



(b) Spray планер

Рисунок 4. Slocum и spray планеры, используемые в эксперименте ASAP 2006 года.

Методы, основанные на поведении индивида

Данные методы изучают движение индивидов в обществе, базирующихся на живых организмах. Наиболее часто изучается движение группы людей и на основе статистики полученной в заданных ситуациях, прогнозируют стратегии движения каждого индивида в обществе. Избегание столкновений в данных методах является ключевой задачей и существует ряд работ посвященной данной тематике.

Например, изучается движение человека в случаях преграждения движущимся препятствием под разными углами, изображенного на рисунке 5 [9].

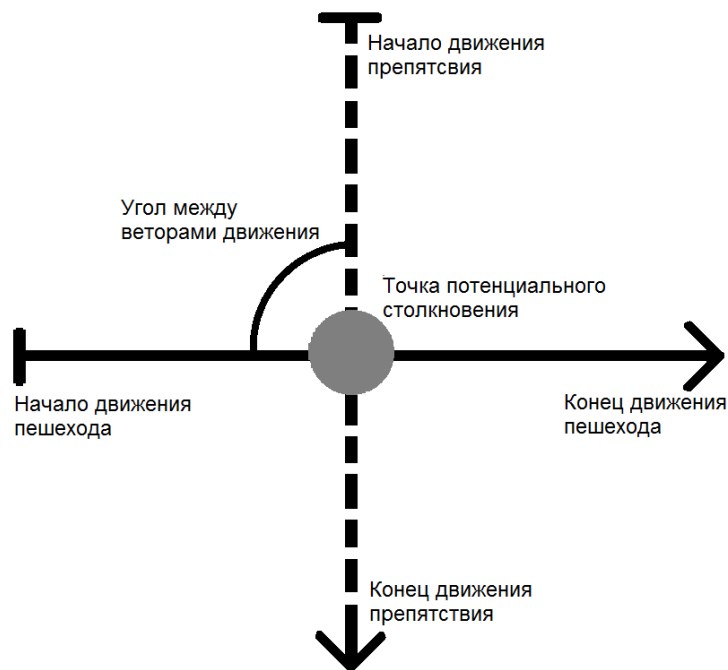


Рисунок 5. Модель проведения эксперимента.

Изучается изменчивость свойств группы путем задания индивидуальных черт поведения каждой единицы, таких как предпочтительная скорость, эффективный радиус, максимальное расстояние видимых соседей, максимальное количество соседей, влияющих на поведение [10].

Наиболее частой практикой управления индивидом является задание когнитивных сил набором дифференциальных уравнений, которые определяют его поведение. Индивид путем эвристических решений вычисляет желаемую стратегию движения, которая является суперпозицией разных стратегий движения. Простая модель поведения – это движение к заданной цели (координата в пространстве), где желаемая стратегия для пешехода является выбором наиболее разреженной траектории движения. Опираясь на скорость и направление движения можно задать управление, тем самым сведя задачу к искомым величинам.

Зададим вспомогательную функцию $D(\alpha)$ и на основе данной функции выразим искомые величины, желаемую скорость v_{des} и желаемый угол движения α_{des} :

$$\alpha_{des} = \underset{\alpha}{\operatorname{argmin}} D(\alpha) \quad (2)$$

При довольно простой математической модели управления, данный алгоритм имеет ряд проблем и ограничений, для оценки которых требуются вычисления и проведение эксперимента.

2. Постановка задачи и разработка алгоритма управления

Основная задача данной работы – разработка алгоритма управления пространственной конфигурацией группы роботов. Роботам индивидуально назначаются целевые координаты, которых они должны достичь. Избегание столкновений роботов является ключевым условием разработки.

В рамках данной работы изучалось управление в двумерном пространстве. Разработка библиотеки алгоритмов проводилась на языке с# (объектно-ориентированный язык программирования с управляемым кодом), а так же использовалась технология WPF (Windows Presentation Foundation – система для построения клиентских приложений Windows). Результаты пользовательского интерфейса представлены на рисунке 7. Все примеры программного кода будут представлены на с#-подобном языке.

Модель робота

Модель робота представлена в виде нескольких компонент: зрительный модуль, когнитивный модуль, модуль управления, физическая модель. Каждая компонента выполняет выделенную роль для моделирования робота. Взаимодействие всех этих компонент между собой является полноценной моделью робота.

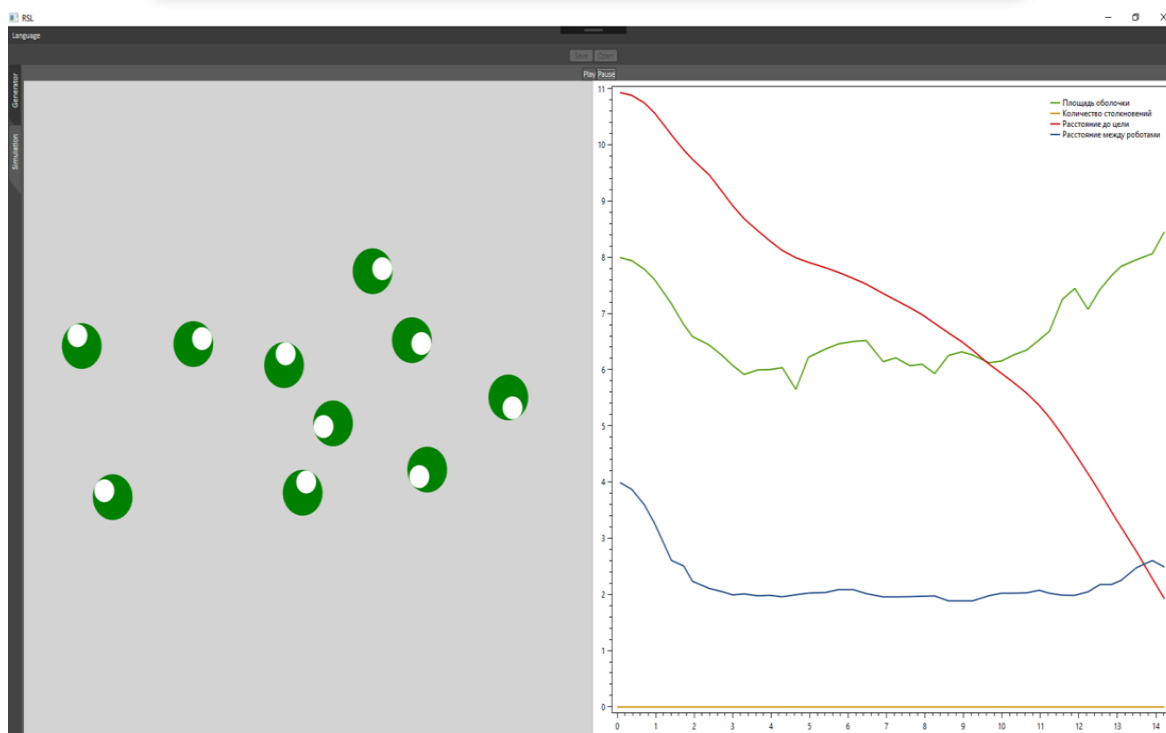
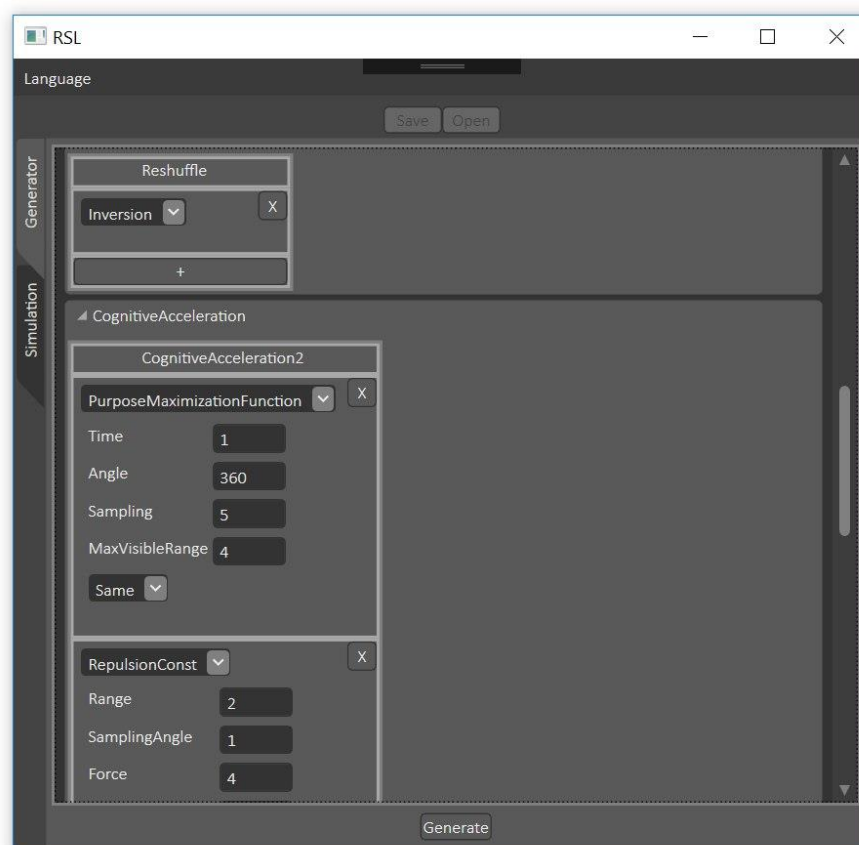


Рисунок 7. Пользовательский интерфейс для численного моделирования.

Зрительный модуль

Зрительный модуль отвечает за восприятие внешнего мира. На основе реальных средств восприятия задаются их модели, имитирующие эксплуатацию. Для описания приборов с различными характеристиками задаются изменяемые параметры модели. Собранные данные со средств восприятия в текущий момент времени, обрабатываются алгоритмами в рамках зрительного модуля для построения картины мира, после обработанные данные передаются когнитивному модулю.

Примером средства восприятия является дальномер, опрашивая который можно узнать расстояние до объекта под наблюдаемым углом. Область обзора простейшего дальномера является сектор, где дуга является границей между свободным пространством и объектом. Дальность обзора есть ключевая характеристика дальномера. На основе простой модели дальномера была разработана модель дальномерного зрения, представленного на рисунке 8, где для каждого угла относительно курса робота задан свой сектор обзора. Курс робота – угол, задающий ориентацию в сферической (полярной) системе координат, направление которого обозначено стрелкой на рисунке 8.

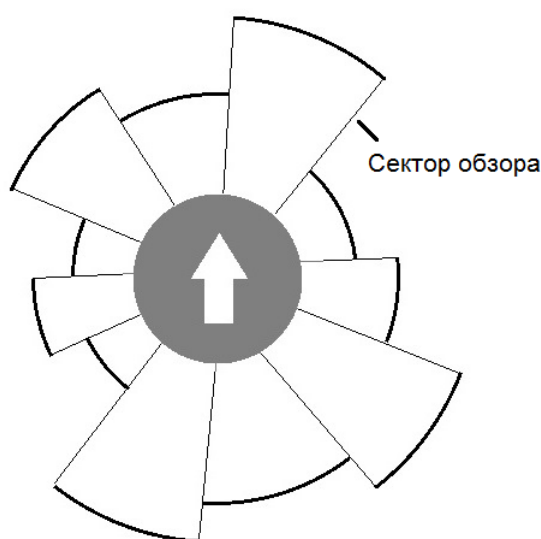


Рисунок 8. Модель дальномерного зрения.

Когнитивный модуль

Когнитивный модуль отвечает за планирование движения робота. Получая данные от зрительного модуля, когнитивные алгоритмы обрабатывают их и задают воздействие на модуль управления роботом. Когнитивный модуль является суперпозицией когнитивных алгоритмов, каждый из которых выполняет заданную роль в активной деятельности робота.

Задание скорости для робота определяет простейшую модель управления, поэтому на основе простого управления конечным результатом работы когнитивного модуля является вектор желаемой скорости, которую необходимо реализовать. Когнитивный модуль представляется в виде нескольких компонентов, таких как целестремительный, отталкивающий алгоритмы. Когнитивные алгоритмы помимо вычисления вектора желаемой скорости для текущего момента времени $\vec{v}_{des}(t)$, так же вычисляют вес $\delta(t)$, декларация которого представлена в программном коде 1.

```
public interface ICognitiveAcceleration
{
    Vector ComputeVelocity(ILogics logics, IPhysics physics);
    double ComputeWeight(ILogics logics, IPhysics physics);
}
```

Программный код 1. Когнитивное ускорение.

Вес алгоритма определяет его вклад в суперпозицию в текущий момент времени, естественная величина которого единица.

Суперпозицией алгоритмов является функционал $P(\vec{v}_i; \delta_i)$, декларация которого в программном коде 2. Данный функционал получает вектора желаемых скоростей и веса вычисленные каждым алгоритмом, и на выходе определяет вектор окончательной желаемой скорости.

```

public interface ICognitiveAccelerationManager
{
    List<ICognitiveAcceleration> ArrayAcceleration;
    Vector ComputeVelocity(ILogics logics, IPhysics physics);
}

```

Программный код 2. Менеджер когнитивного ускорения.

Примером такого функционала является линейная норма n -мерного пространства, которая действует на каждую компоненту вектора скорости таким образом:

$$v_{des} = P(\vec{v}_i; \delta_i) = \left(\sum_i (v_i \delta_i)^n \right)^{\frac{1}{n}} \left(\sum_i (\delta_i)^n \right)^{-\frac{1}{n}} \quad (4)$$

Целестремительный алгоритм

Задачей целестремительного алгоритма является выбор когнитивных сил действующего на робота, которые притягивают к заданной целевой координате в пространстве. Так же в рамках данного алгоритма определяется выбор более разреженного направления движения приближенному к цели. Пример такого алгоритма был описан ранее, поэтому модернизированные вычисления выбора угла движения и скорости описаны в краткой форме.

H – максимальная дальность обзора дальномера.

$f(\alpha)$ – расстояние до объекта под заданным углом определенное дальномером.

v_{max} – максимальная скорость, которую разрешено развивать роботом.

v_{des} – желаемая скорость.

α_{des} – желаемый угол движения.

α_{goal} – угол направления к цели.

r_{goal} – расстояние до целевой координаты.

$D(\alpha)$ – вспомогательная функция.

$$v_{des} = \frac{v_{max}}{2} \sqrt{\frac{\min(r_{goal}; f(\alpha_{des}))}{H}} \quad (5)$$

$$\alpha_{des} = \underset{\alpha}{\operatorname{argmax}} D(\alpha) \quad (6)$$

Вспомогательные функции могут быть разными, они отвечают за выбор более разреженного пути. Мы приведем несколько примеров и в дальнейшем изучим их характер на численном моделировании. В случаях, когда максимум вспомогательной функции достигается при разных углах, выбирается угол более близкий к цели. Примером данной функции является модернизированная функция (1) из статьи, которую условно назовем косинусным затуханием:

$$D(\alpha) = 2H \cos(\alpha - \alpha_{goal}) + f(\alpha) \quad (7)$$

Введем функцию без косинусной зависимости от угла и назовем независимой по углу:

$$D(\alpha) = f(\alpha) \quad (8)$$

Для характерной демонстрации работы данного алгоритма, зададим функцию равную константе. Поведение всего алгоритма в случае с постоянной вспомогательной функцией очевидна, движение всегда будет строго направлено к цели. Сравнение с данной функцией покажет необходимость нахождения более разреженного пути.

$$D(\alpha) = 0 \quad (9)$$

Весовая функция для данного алгоритма равна единице.

Сравнение чисел с плавающей запятой

Поскольку компьютер производит операции с ограниченным дискретным спектром значений, то бесконечные числовые пространства не могут быть полностью представлены в виде компьютерных значений. Общеизвестный стандарт IEEE 754 для представления действительных чисел (чисел с плавающей запятой) в двоичном коде, имеет погрешности при расчете. Поэтому сравнение действительных чисел с учетом погрешности расчетов имеет другой характер, которое используется для максимумов вспомогательной функции.

Простым решением выбора угла доставляющего максимум функции, который более приближен к целевому углу является добавление веса ε в сравнение:

$$\alpha_{des} = \begin{cases} \alpha_i & \text{при } D(\alpha_i) > \varepsilon \cdot D(\alpha_j) \\ \alpha_j & \text{при } D(\alpha_i) \leq \varepsilon \cdot D(\alpha_j) \end{cases} \quad (10)$$

Где $\varepsilon = 0,99$ и $|\alpha_i - \alpha_{goal}| < |\alpha_j - \alpha_{goal}|$.

Особенно, стоит подчеркнуть, что данный метод сравнения желательно использовать для чисел, модуль которых больше единицы, по причине большого разброса значений в области нуля из-за погрешности, которая может повлиять на порядок величины.

Рассуждения о необходимости других алгоритмов

Целестремительный алгоритм решает вопрос движения к цели, он может искать более разреженную траекторию движения и увеличивать скорость работы группы, что будет изучено на моделировании. Однако вопрос избегания столкновения является абсолютно нерешенным. Поскольку информация о других объектах представляется только координатами в пространстве, и нет информации об их скоростях.

Легко привести пример плохого случая с перпендикулярным движением двух роботов, как изображено на рисунке 9.

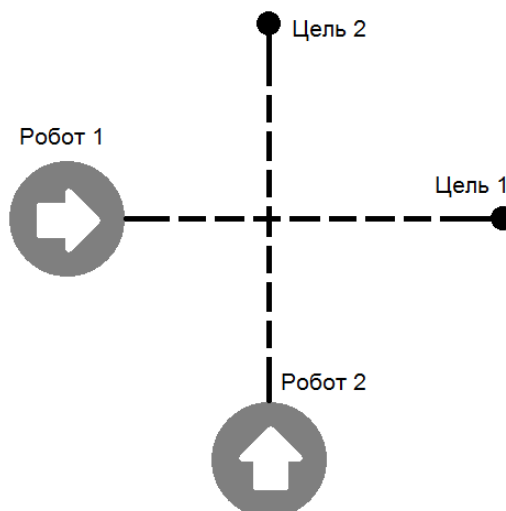


Рисунок 9. Пример перпендикулярного движения роботов.

Данный алгоритм предполагает развивать большие скорости, если преград по направлению движения нет. И действительно, преград в направлении к цели нет, поэтому максимальная скорость зависит только от дальности обзора датномера. Так как оба этих робота находятся в движении, очевидно, что факт неизбежности столкновения может быть обнаружен на расстоянии меньше размеров самого робота. Это условие не позволяет алгоритму избегать столкновений, поэтому было решено разработать алгоритм, предназначенный для избегания столкновений.

Отталкивающий алгоритм

Задача отталкивающего алгоритма держать дистанцию между роботами. Путем задания когнитивных сил взаимодействия между роботами можно получить эффективные потенциалы подобные центральным силам или другие виды потенциальных ям имеющих отличную природу взаимодействия.

Определим желаемую скорость для данного алгоритма с простой потенциальной ямой:

$$\vec{v}_{des} = \frac{\vec{v}_{sum}}{v_{sum}} \min(v_{max}; v_{sum}) \quad (11)$$

$$\vec{v}_{sum} = \sum_i \vec{v}_i \quad (12)$$

$$\vec{v}_i = \frac{\vec{r}_i}{r_i} \left(1 - \frac{R}{r_i}\right) F \quad (13)$$

В данном случае $\vec{r}_i$ является вектором направленном от себя к соседнему роботу и его длина равна расстоянию между ними. В случае дальномерного зрения это перебор всех секторов, где значения r_i равны $f(\alpha)$. Если v_i обнулить для $r_i > R$, то это будет строго отталкивающее взаимодействие.

Весовая функция $\delta(t)$ является не менее важной, которая увеличивается при $v_{sum} > v_{max}$, это означает, что когда запрашиваемая алгоритмом скорость нереализуема, то значимость данного максимального значения скорости возрастает.

$$\delta = \min\left(1; \frac{v_{sum}}{v_{max}}\right) \quad (14)$$

Оценка максимальной скорости

Так как данные алгоритмы реализуют управление конкретными физическими моделями, то оценка максимальной скорости является не менее важным параметром. Запрашиваемые скорости выше физических возможностей модели не только излишни, но и вредны, причем может нарушать условие избегания столкновений.

Рассмотрим пример с группой материальных точек. Каждая точка имеет возможность двигаться с любым ускорением, не превышающим величину a . Дальность обзора точки H , время реакции на изменения внешнего мира T . Чтобы при таких условиях две материальные точки

имели возможность затормозить, требуется максимальная скорость, не превышающая данное значение (17):

$$v = at \quad (15)$$

$$H = 2 \left(v(t + T) - \frac{at^2}{2} \right) \quad (16)$$

$$v^2 + 2vaT - aH = 0 \quad (17)$$

$$v_{max} = \sqrt{(aT)^2 + aH} - aT \quad (18)$$

В случае не согласованности двух автономных единиц двигающихся на встречу друг к другу, совершение маневра одного могут быть скомпенсированы маневром второго, именно поэтому данная задача экстренного торможения является верхней оценкой на максимальную разрешенную скорость передвижения.

Модуль управления и физическая модель

Модуль управления отвечает за реализацию желаемой скорости полученной от когнитивного модуля. Реализацией является управление физической моделью. Так как физическая модель робота может принципиально отличаться от обычного представления движения материальной точки со скоростью, то данный модуль задает управление конкретной физической моделью приближенное к движению с запрашиваемой скоростью.

Физическая модель это приближенное математическое описание физических характеристик реальной модели робота. Физическая модель решает задачу численного моделирования движения робота, что представлено в программном коде 3.

```

public interface IMechanicalAcceleration
{
    Vector ComputePosition(double deltaTime, ILogics logics, IPhysics physics);
    Vector ComputeVelocity(double deltaTime, ILogics logics, IPhysics physics);
    Direction ComputeCourse(double deltaTime, ILogics logics, IPhysics physics);
}

```

Программный код 3. Численное моделирование движения.

Полное состояние в пространстве простой физической модели абсолютно твердого тела можно описать тремя параметрами, такими как координата, векторная скорость, курс робота. В данной работе форма робота представляется в модели окружности. Декларация физических свойств робота представлена в программном коде 4.

```

public interface IPhysics
{
    string Name;
    Direction Course;
    Vector Velocity;
    Vector Position;
    double Proportions;
}

```

Программный код 4. Состояние робота в пространстве.

Самый простой пример физической модели робота, это неограниченная модель, которая всегда реализует желаемую скорость. Пример двухколесного робота с угловой скоростью вращения ω_{max} и поступательным ускорением a_{max} , можно описать математически для численного моделирования. В данной модели направление движения робота строго определяется его курсом.

Δt – время дискретизации численного моделирования.

ω – угловая скорость, заданная управлением.

a – поступательное ускорение, заданное управлением.

$$\angle(\vec{a}; \vec{b}) = \arccos \frac{(\vec{a} \cdot \vec{b})}{ab} \quad (19)$$

$$|\omega| = \min \left(\angle(\vec{v}_{cur}; \vec{v}_{des}) \frac{1}{\Delta t}; \omega_{max} \right) \quad (20)$$

$$|a| = \min \left(\left| \frac{(\vec{v}_{cur} \cdot \vec{v}_{des})}{v_{cur}} - v_{cur} \right| \frac{1}{\Delta t}; a_{max} \right) \quad (21)$$

$$\dot{\varphi} = \omega \quad (22)$$

$$\dot{v} = a \quad (23)$$

$$\dot{x} = v \cdot \cos \varphi \quad (24)$$

$$\dot{y} = v \cdot \sin \varphi \quad (25)$$

Средство анализа

Разработаны средства анализа на основе знаний всех физических состояний роботов. Для удобного визуального представления статистические данные системы представляются в виде графиков разработанных на основе библиотеки OxyPlot на языке с#. Далее перечислены определения статистических данных.

Подсчет количества столкновений – количество пересечений форм всех пар роботов за время все исполнения. Пересечение одной и той же пары роботов засчитывается дважды, если при разъединении они пересеклись вновь.

Размер оболочки группы – квадратный корень площади оболочки группы (25). Оболочка группы это минимальный по периметру многоугольник, внутри которого лежат все роботы, который представлен на рисунке 10.

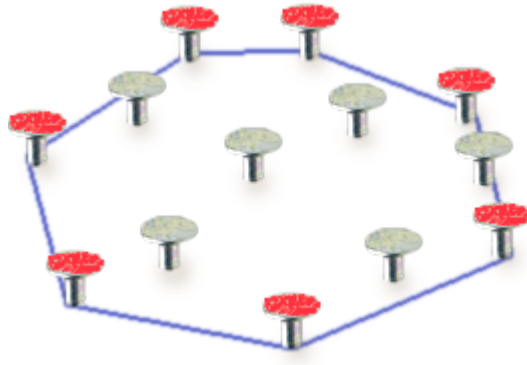


Рисунок 10. Оболочка группы.

$$R(t) = \sqrt{S(t)} \quad (26)$$

$$S(t) = \frac{1}{2} \left| \sum_{i=1}^K (x_i + x_{i+1})(y_i - y_{i+1}) \right| \quad (27)$$

$(x_i; y_i)$ – i -тая вершина многоугольника. Вершина при $K+1$ равна вершине 1.

Среднее расстояние до цели:

$$\bar{G}(t) = \frac{1}{N} \sum_{i=1}^N G_i(t) \quad (28)$$

$G_i(t)$ – расстояние до цели i -того робота в момент времени t .

N – количество роботов.

Среднее отклонение от центра:

$$\Delta r = \frac{1}{N} \sum_{i=1}^N \Delta \rho_i \quad (29)$$

$$\Delta \rho_i = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2} \quad (30)$$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (31)$$

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i \quad (32)$$

Постановка задачи

В ходе разработки алгоритмов и проведения численного моделирования были выявлены экстремальные ситуации перестроения группы роботов числом от 10 до 100 штук. Ситуации являются экстремальными, когда задание условий перестроения вынуждает увеличивать плотность группы во время их движения до величины 1 робот/2 размера робота. Типовой случай экстремального перестроения – вынужденное прохождение каждого робота через центр группы. Примеры представлены на рисунке 11, координаты группы остаются прежними, а задача каждого робота занять противоположную позицию в данной конфигурации.

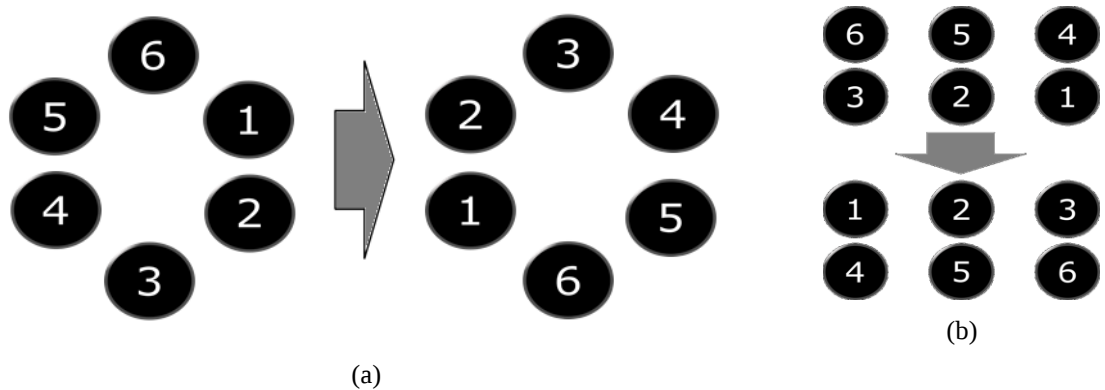


Рисунок 11. Экстремальное перестроение группы (a) из окружности в окружность, (b) из двух линий в две линии.

3. Результаты численного моделирования

Была проведена серия схожих моделирований в случае экстремальных перестроений с небольшими отличиями в заданных системах. При помощи средств статистические состояния систем были сопоставлены анализа для выявления характера изменчивости. В каждом рисунке представлены по 3 графика: среднее расстояние до цели, размер оболочки группы и количество столкновений.

Параметры заданной системы:

Частота моделирования: 20 перерасчетов/сек.

Диаметры роботов: 1 метр.

Угловые размеры секторов дальномерного зрения: 15 градусов.

Дальность обзора дальномерного зрения: 6 метров.

Параметры двухколесной модели: $\omega_{max} = 70^\circ/\text{с}$; $a_{max} = 6\text{м}/\text{с}^2$.

*Количество роботов: 50 штук.

*Вспомогательная функция целестремительного алгоритма: $D(\alpha) = f(\alpha)$.

*Параметры отталкивающего алгоритма: $R = 3\text{ м}$; $F = 3\text{м}/\text{с}$.

*Конфигурация группы: окружность, радиус 25 м.

**-параметры, которые менялись для серии моделирований.*

Список отличий систем для каждой серии:

- I. Рисунок 12, разные целестремительные алгоритмы.
- II. Рисунок 13, разные отталкивающие алгоритмы.
- III. Рисунок 14, разное количество роботов.
- IV. Рисунок 15, разные конфигурации групп.

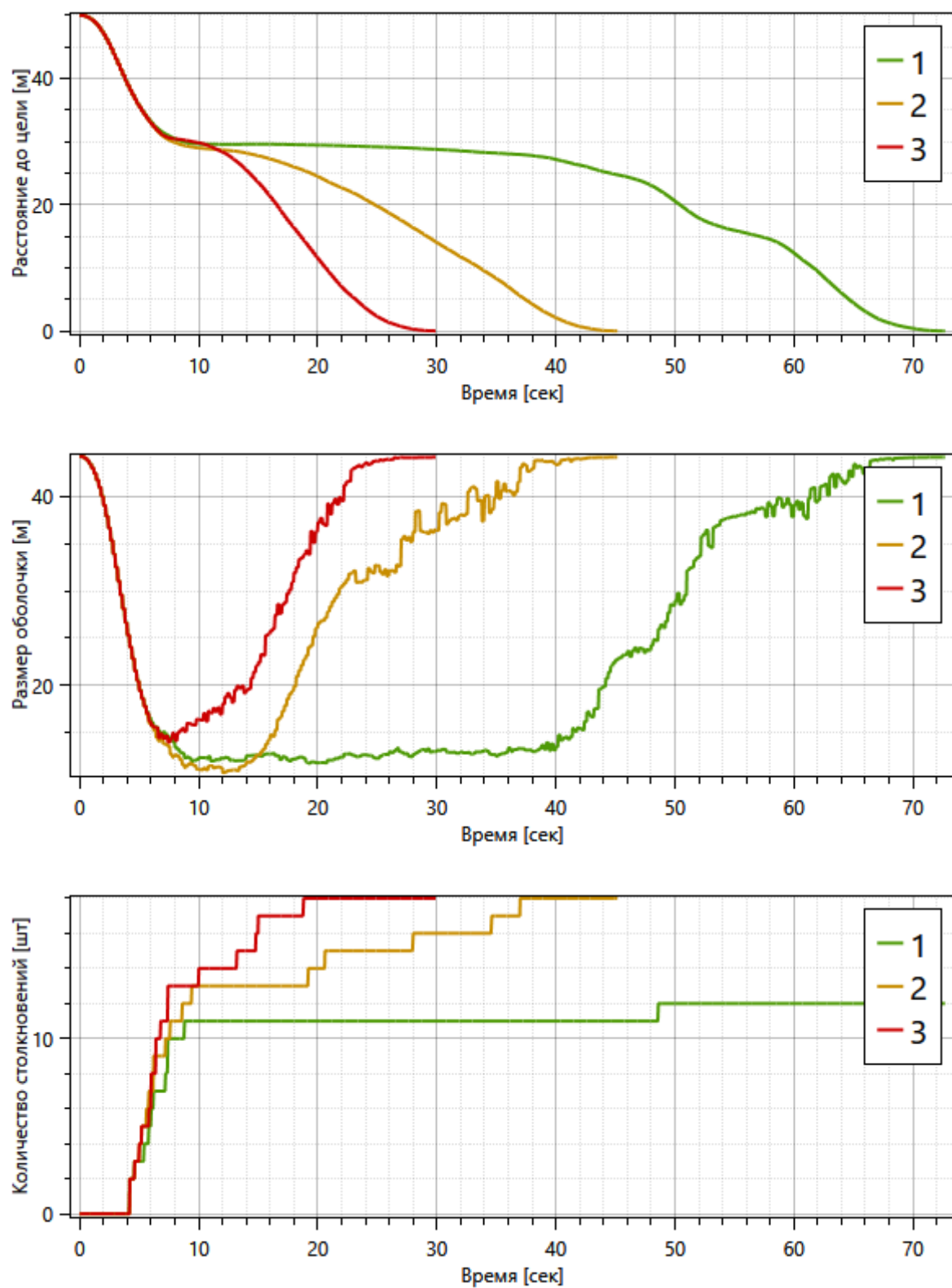


Рисунок 12. Целестремительный алгоритм.

(1) - $D(\alpha) = f(\alpha)$;

(2) - $D(\alpha) = 2H \cos(\alpha - \alpha_{goal}) + f(\alpha)$;

(3) - $D(\alpha) = 0$.

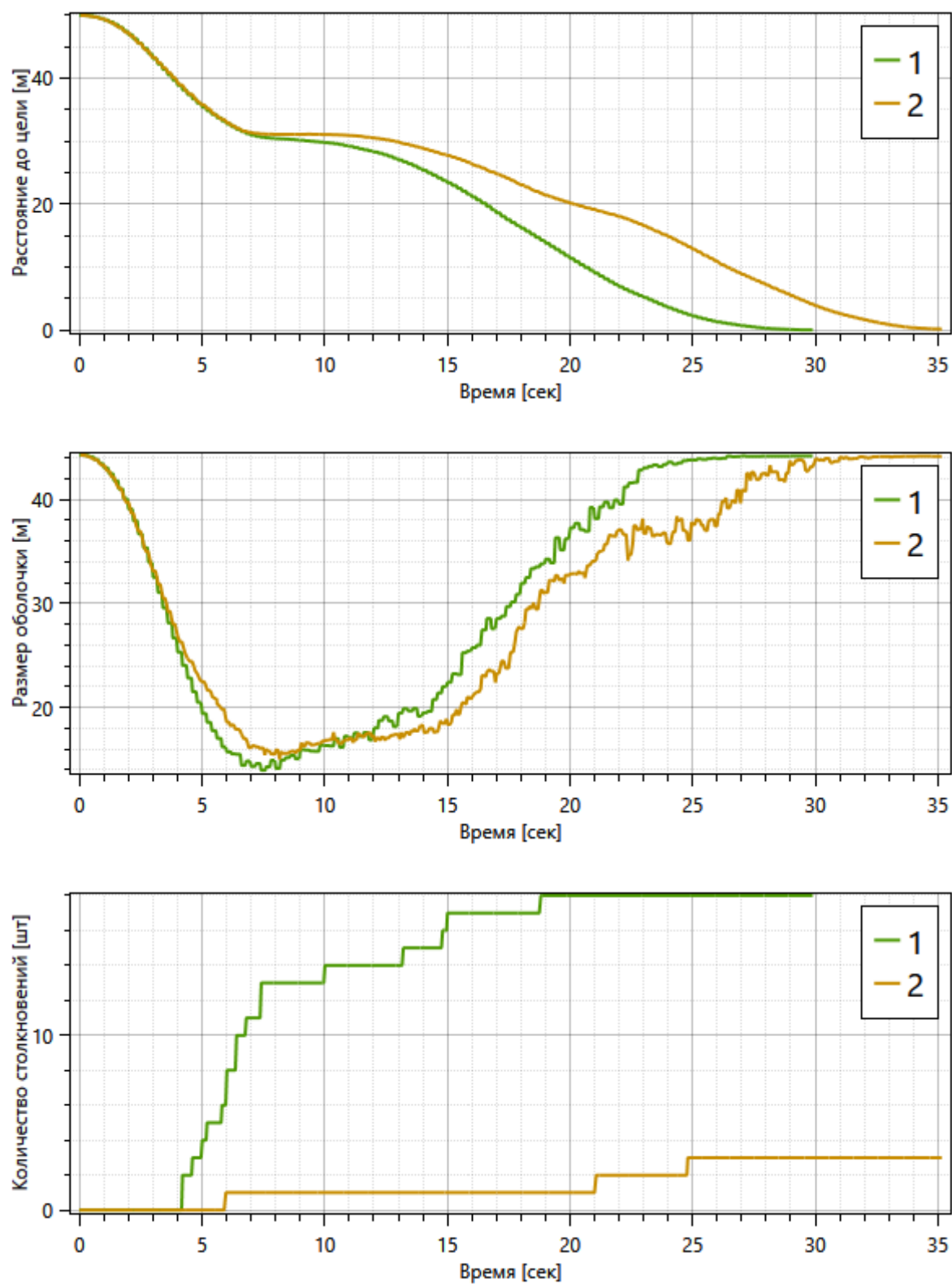


Рисунок 13. Отталкивающий алгоритм.

(1) - $R = 3$ м; $F = 3$ м/с;

(2) - $R = 6$ м; $F = 6$ м/с.

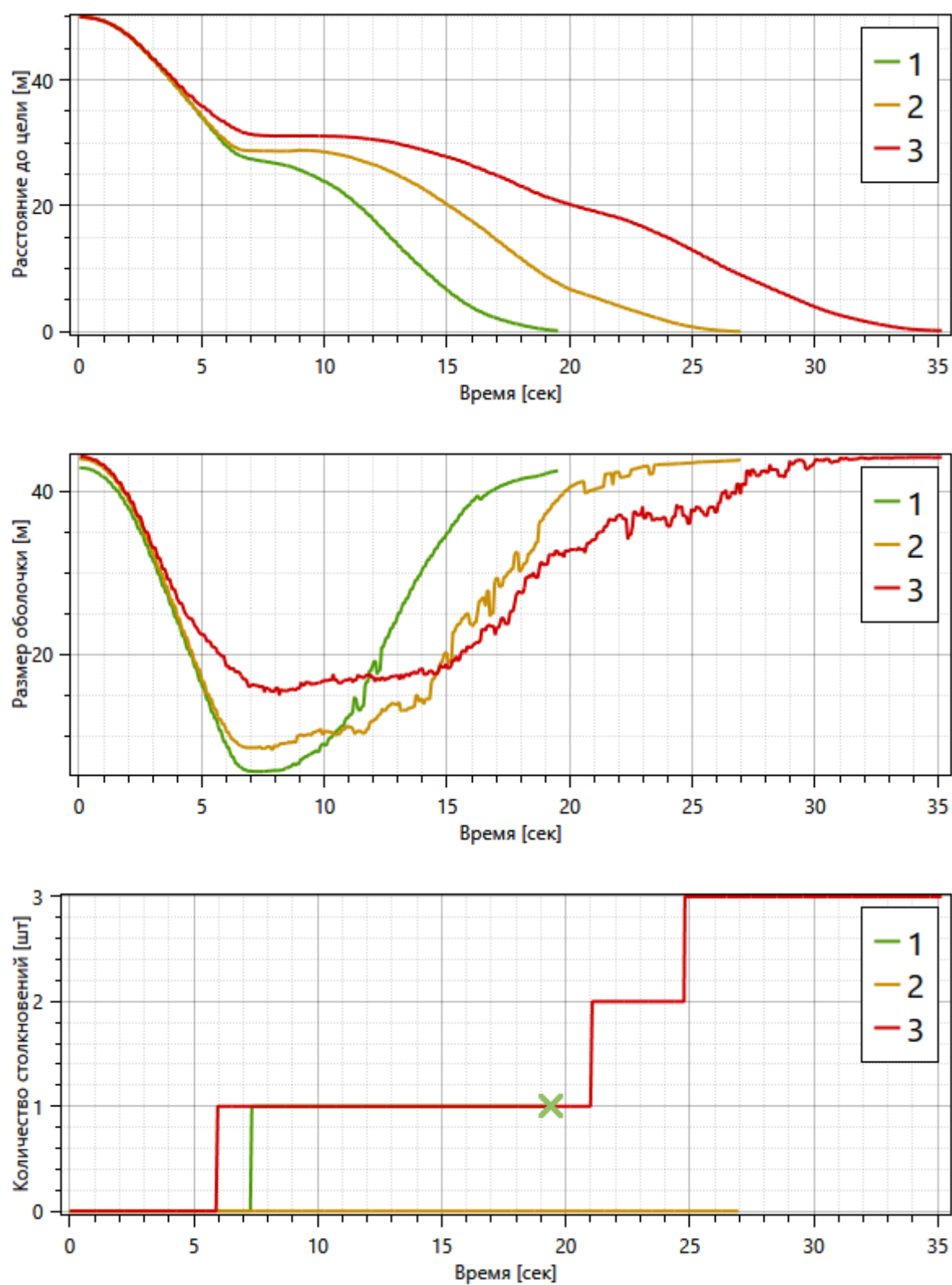


Рисунок 14. Количество роботов.

- (1) - $N = 10$ шт; $R = 6$ м; $F = 6$ м/с;
- (2) - $N = 20$ шт; $R = 6$ м; $F = 6$ м/с;
- (3) - $N = 50$ шт; $R = 6$ м; $F = 6$ м/с.

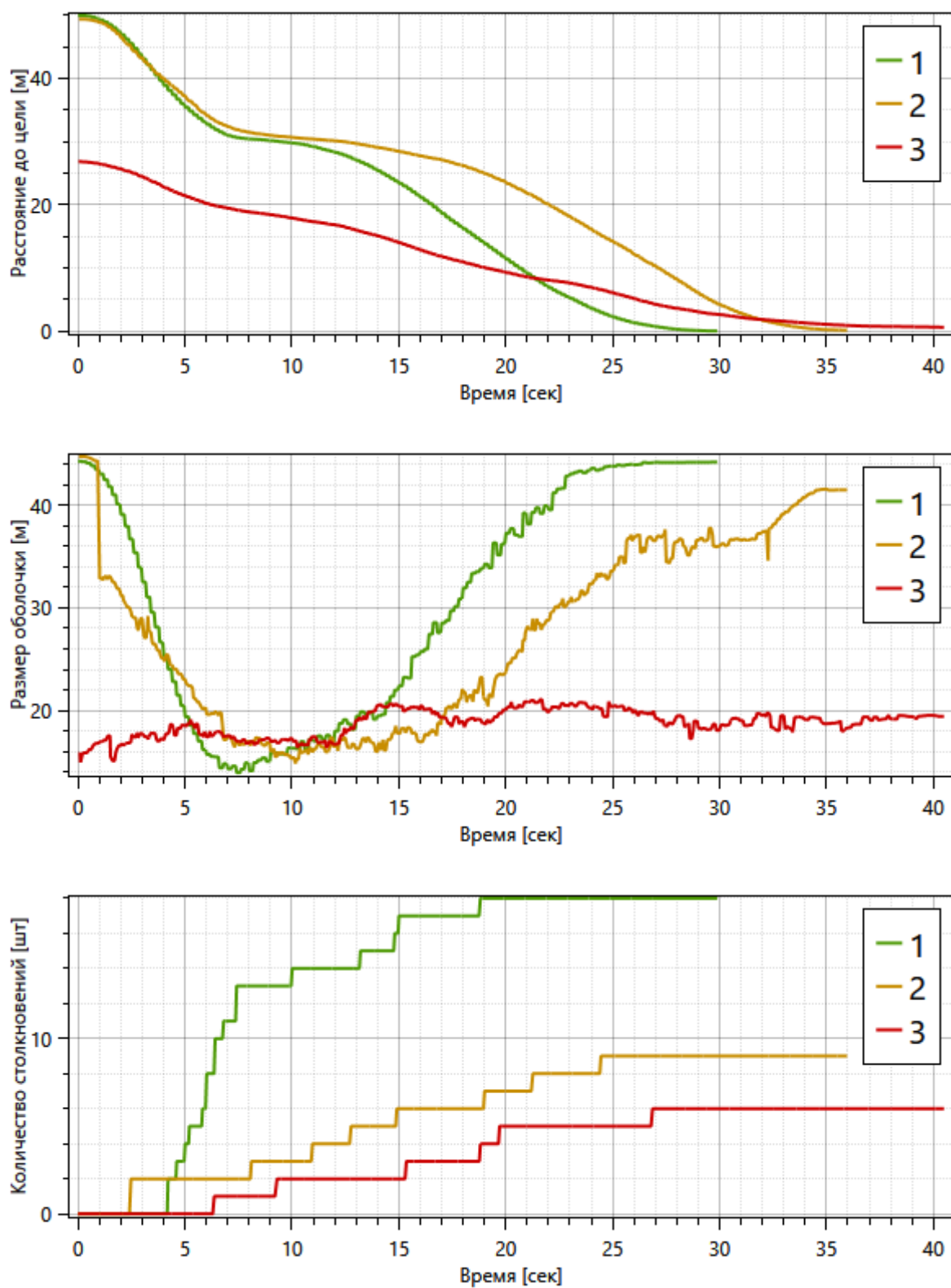


Рисунок 15. Конфигурация группы.

- (1) – окружность, радиус 25 м;
- (2) – две линии, длина линий 50 м, расстояние между линиями 40 м;
- (3) – две линии, длина линий 50 м, расстояние между линиями 10 м.

Анализ графиков

I. Рисунок 12. Целестремительный алгоритм.

$$(1) - D(\alpha) = f(\alpha);$$

$$(2) - D(\alpha) = 2H \cos(\alpha - \alpha_{goal}) + f(\alpha);$$

$$(3) - D(\alpha) = 0.$$

Количество столкновений меняется не значительно при изменении вспомогательной функции, но разрежённость группы, а так же скорость перестроения больше у алгоритма (1), при независимом выборе направления движения, чем у алгоритма косинусного затухания по углу (2). Движение, где направление движения всегда выбирается строго равное направлению к цели, является худшим алгоритмом из трех, что говорит о перспективности преждевременного выбора более свободной траектории движения, для ситуаций экстремального перестроения.

II. Рисунок 13. Отталкивающий алгоритм.

$$(1) - R = 3 \text{ м}; F = 3 \text{ м/с};$$

$$(2) - R = 6 \text{ м}; F = 6 \text{ м/с}.$$

Увеличение силы отталкивающего взаимодействия между роботами (2), может незначительно влиять на скорость перестроения, однако играет ключевую роль в пиковой плотности группы, как следствие риск столкновений может резко снизиться.

III. Рисунок 14. Количество роботов.

$$(1) - N = 10 \text{ шт}; R = 6 \text{ м}; F = 6 \text{ м/с};$$

$$(2) - N = 20 \text{ шт}; R = 6 \text{ м}; F = 6 \text{ м/с};$$

$$(3) - N = 50 \text{ шт}; R = 6 \text{ м}; F = 6 \text{ м/с}.$$

Для группы роботов, чье количество превышает 10 автономных единиц, увеличение количества в несколько раз в экстремальных случаях в приближении может увеличивать продолжительность перестроения по логарифмированному закону.

IV. Рисунок 15. Конфигурация группы.

- (1) – окружность, радиус 25 м;
- (2) – две линии, длина линий 50 м, расстояние между линиями 40 м;
- (3) – две линии, длина линий 50 м, расстояние между линиями 10 м.

Разные конфигурации групп роботов, могут иметь совершенно другой характер и ряд проблем в реализации их перестроения, что было продемонстрировано при похожих симметриях перестроения (1) и (2) постановок задач. Для постановки задачи (3) с близким прилеганием двух рядов роботов в течении характерного времени ~ 1 мин, не была достигнута цель перестроения. 4 из 50 роботов, были заблокированы выстроенным рядом достигнувших точки назначения роботов, что не позволяло оставшимся роботам перейти на другую сторону ряда. Данный факт говорит о сложном характере перестроения из плотной группы в плотную.

На протяжении всех серий экспериментов, наблюдались расстояния между любыми соседними роботами порядка 1-4 метра, при размерах самих роботов в 1 метр. Возникало довольно большое количество столкновений роботов, что говорит о не надежности избегания столкновений у текущих алгоритмов. Очевидны попадания в локальные минимумы, из которых группа роботов выбирается либо очень долго, либо не выбирается совсем.

Заключение

Был разработан алгоритм управления группой мобильных роботов избегающий столкновения на основе эвристики моделирующей поведение индивида в коллективе. Было разработано программное обеспечение для проведения моделирования приближенное к модели реального робота для удобного анализа состояния моделирования. Проведена серия численных моделирований для задач перестроения группы роботов в особых условиях. Изучалась изменчивость свойств группы роботов для перестроений в экстремальных условиях на основе статистических данных. Были выявлены преимущества и недостатки разработанных целестремительного и отталкивающего алгоритмов поведения автономных единиц в коллективах, а так же были приведены оценки ограничений для данных алгоритмов. Для уменьшения риска столкновений и избегания попаданий в локальные минимумы требуется разработка алгоритмов являющихся композицией ситуативных методов и методов, основанных на поведении индивида, над которыми на данный момент ведется работа.

Список литературы

- [1] Ant Colony Optimization and Swarm Intelligence / M. Dorigo, M. Birattari, C. Blum, et al. // 6th International Conference, Brussels, Belgium, 2008. - Vol. 3172, Springer.
- [2] Crowd Behavior Analysis: A Review where Physics meets Biology / V. J. Kok, M. K. Lim, C. S. Chan // Neuro-computing 2015;177:342-62.
- [3] Crowd turbulence: the physics of crowd disasters / D. Helbing, A. Johansson, H. Z. Al-Abideen // International Conference on Nonlinear Mechanics (ICNM-V), 2007, pp. 967–969.
- [4] How simple rules determine pedestrian behavior and crowd disasters / M. Mehdi, H. Dirk, T. Guy // Proceedings of the national academy of sciences of the united states of america, 108(17):6884-8, 2011.
- [5] Gigerenzer G. Simple Heuristics That Make Us Smart / P. Todd // behavioral and brain sciences, 23, pp. 727–780, 2000.
- [6] Autonomous Pedestrian Collision Avoidance Using a Fuzzy Steering / D. F. Llorca, V. Milanes, I. P. Alonso, et al. // Controller IEEE Transactions on Intelligent Transportation Systems, Vol. 12, Issue 2, pp. 390 – 401, 2011.
- [7] Programmable self-assembly in a thousand-robot swarm / M. Rubenstein, A. Cornejo, R. Nagpal// Science, Vol. 345, Issue 6198, pp. 795-799, 2014.
- [8] Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay / E. L. Naomi, A. P. Derek, E. D. Russ, et al. // Journal of field robotics, 27(6), pp. 718-740, 2010.
- [9] Adjustments of Speed and Path when Avoiding Collisions with Another Pedestrian/ H. Markus, S. Yi-Huang, K. Melanie, et al. // Plos one [Электронный ресурс]. – 2014. – URL:

<https://doi.org/10.1371/journal.pone.0089589>. – (дата обращения: 31.05.2017).

[10] Simulation heterogeneous crowd behaviors using personality trait theory / J. G. Stephen, K. Sujeong, C. L. Ming, M. Dinesh // In ACM SIGGRAPH, Eurographics symposium on computer animation, pp. 43-52, 2011.

[11] Bioinspired Obstacle Avoidance Algorithms for Robot Swarms / J. Guzzi, A. Giusti, L. M. Gambardella, et al. // International Conference on Bio-Inspired Models of Network, Information, and Computing Systems, Bio-Inspired Models of Network, Information, and Computing Systems, pp. 120-134, 2012.