

## **ЗАДАЧА ПРОКЛАДЫВАНИЯ МАРШРУТА И ВИЗУАЛИЗАЦИИ ПРОСТРАНСТВЕННОГО РАСПОЛОЖЕНИЯ ДЛЯ МОБИЛЬНОГО РОБОТА**

**Воронов С.С.**

*Научный руководитель: Юдин А.В.*

МГТУ им. Н.Э.Баумана, кафедра ИУ4, Москва, Россия

## **TASK OF ROUTE BUILDING AND SPATIAL POSITION VISUALISATION FOR A MOBILE ROBOT**

**Voronov S.S.**

*Supervisor: Yudin A. V.*

Bauman Moscow State Technical University, Moscow, Russia

### **Аннотация**

В данной статье представлены варианты решения проблем навигации мобильных роботов, созданных для участия в конкурсе Eurobot 2014. Были тщательно проанализированы правила конкурса, на их основе поставлены задачи для создания навигационной модели, предложены методы по их решению в виде алгоритмов действий. В завершение статьи автором рассмотрена возможность визуализации результатов работы навигационной системы в удобной для человека форме в виде отдельного программного приложения.

### **Abstract**

This article presents solutions to navigation problems of mobile robots created for the Eurobot 2014 contest. The competition rules were carefully analyzed, tasks for navigation model implementation were set and methods for their optimal solution were proposed. The article finishes with author's presentation of navigation system work results' visualization software tool displaying information in user-friendly form.

### **Введение**

Задача навигации автоматизированных систем с мобильными механическими элементами, будь то робот, средство передвижения человека или фрезерный станок, возникает с момента первых попыток осуществить автоматизированное движение на практике. Степень автоматизации может варьироваться от непосредственного участия оператора в управлении системой до полного автономного поведения. Сложность системы навигации непосредственно связана со степенью автоматизации. Имея в виду, что выше означенные примеры систем подобны в части принципов организации управления движением мобильных элементов будем далее рассматривать только робототехническое приложение для навигационных систем.

Решение навигационных задач включает в себя взаимодействие робота и среды на всевозможных уровнях: следует считывать информацию с датчиков, определяя как положение робота, так и окружающие его объекты; анализировать данную информацию на процессорах, формируя и постоянно обновляя карту окружающего пространства, реагировать на эти данные, подавая соответствующее управление на двигатели и прочие исполнительные механизмы. Компьютерные системы построения маршрута разработаны достаточно хорошо и успешно используются не только в задачах реального мира, но и применяются, например, в компьютерных играх.

Сложной задачей является сопоставление реальных координат объектов с теми, которые на данный момент имеет робот в своей памяти. Датчики окружающей среды имеют те или иные недостатки, все они получают данные с помехами. Например, если взять на рассмотрение ультразвуковые маяки, то небольшое изменение температуры ведёт к изменению скорости распространения звуковых колебаний, основного носителя информации о положении элементов системы, что необходимо обязательно учитывать в подобных системах. Если динамика изменения температуры в окружающем пространстве одинакова

для всех точек системы – достаточно обеспечить замеры температуры и компенсировать изменения программно. Куда более серьезные последствия могут создать различного рода непредсказуемые помехи, такие как точечные турбулентные колебания температуры, которые практически невозможно предсказать.

### 1 Постановка задач навигационной системы

Соревнования мобильных роботов Eurobot 2014 ставит перед участниками ряд задач, каждую из которых необходимо проанализировать и расставить приоритеты выполнения [2]. В зависимости от выполнения задачи разные требования предъявляются и к системе навигации. Кратко рассмотрим задания, опишем принципиальные моменты. В задачах Eurobot помимо разрабатываемого робота принимают так называемые игровые элементы, с которыми робот должен взаимодействовать для успешного решения задачи.

Одной из задач является **бросание копья в мамонта**. Игровой элемент Мамонт – вертикальная плоскость, с прикрепленной липучкой. Игровой элемент Копьё – мяч для пинг-понга, обернутый липучкой. Задача робота – выстрелить шарик так, чтобы он прилип к мамонту. Пусть механизм запуска снаряда требует от робота находиться в конкретно заданной точке игрового поля (это упрощает задачу конструирования механизма метания). Тогда из произвольного/начального места поля необходимо переместиться в точку с фиксированными координатами, что является элементарным действием навигационной модели (рис.1). При этом, как и во всех остальных случаях, необходимо учитывать и тот случай, когда эта точка будет занята другим роботом или игровым элементом.

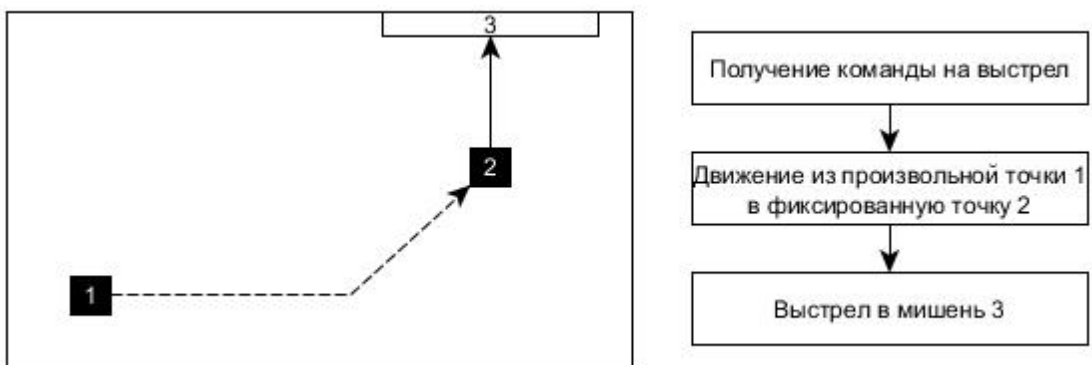


Рисунок 1 — Выстрел в мишень

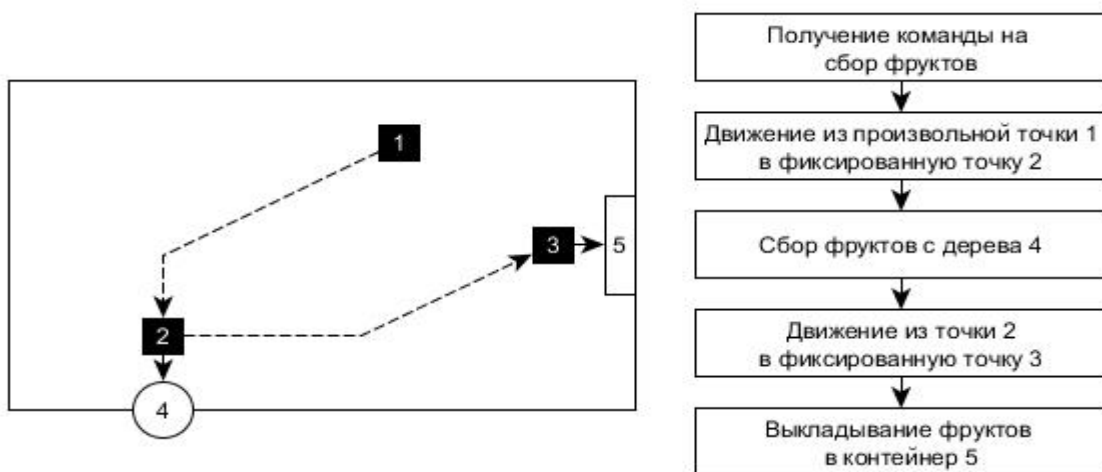


Рисунок 2 — Сбор фруктов

Следующая задача - **сбор фруктов**. Игровой элемент Фрукты - цилиндры, подвешенные на вертикальных столбах по краю поля . Необходимо перенести как можно больше указанных цилиндров в специальный контейнер, расположенный на поле. Задача, которая решается уже в два этапа, которые заключены в циклы. Во-первых, необходимо достигнуть определённой точки рядом со столбом, чтобы специальный механизм мог взять фрукт. Во-вторых, после этого этот фрукт (или несколько сразу) надо положить в определённый контейнер, координаты которого так же известны заранее (рис.2). Необходима постоянная координация с механизмом сбора.

Ещё одно задание - **работа с «огнём»**. Игровой элемент Огонь - треугольная фигура, заданной толщины, имеет две стороны, каждая покрашена в один из двух цветов (на двух сторонах разные цвета). Необходимо перевернуть огонь цветом своей команды и горизонтально положить. Если положить огонь на специально отведённое место (игровой элемент «сердце огня») - начисляются дополнительные очки (рис.3). Прежде всего, робот должен иметь специальные датчики, определяющие нахождение огня рядом с роботом, потому что он может перемещаться, а значит его положение заранее неизвестно. Именно поэтому представляется наиболее эффективным метод, при котором робот реагирует на огонь при решении остальных задач.

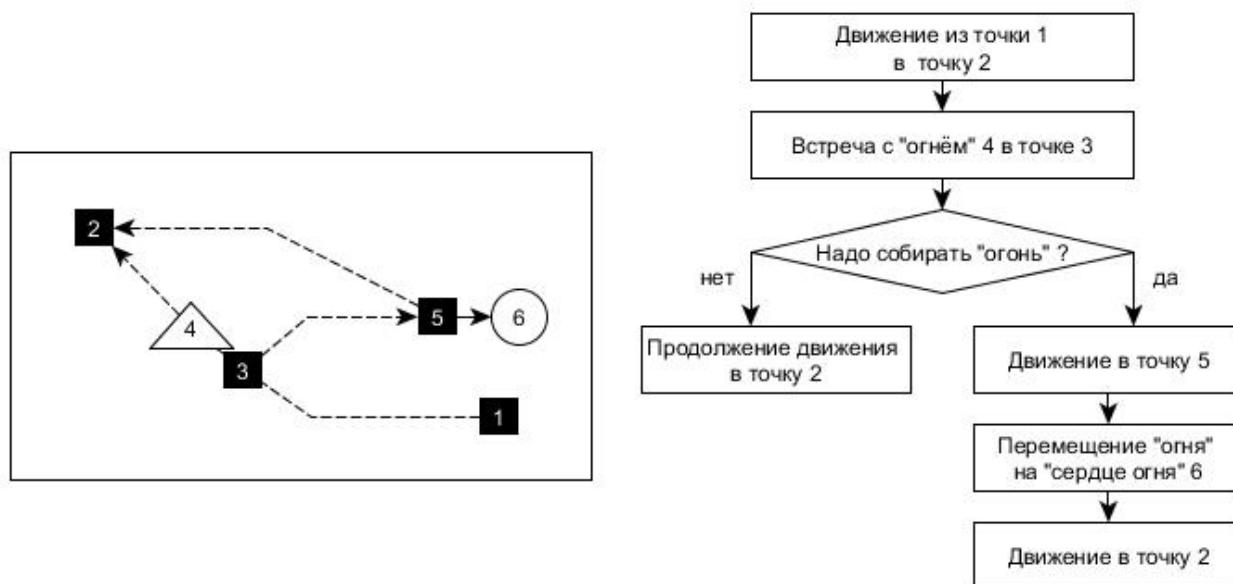


Рисунок 3 — Работа с «огнём»

Если робот может перевозить несколько огней, то он может положить его себе в багаж, чтобы отвезти после несколько сразу. В противном случае робот может или просто перевернуть, или отвезти на сердце огня, а затем возвращаться к отложенной задаче.

## 2 Решение задачи в общем виде

Перед нашей системой ставятся следующие задачи: робот должен уметь двигаться из одной фиксированной точки игрового поля в другую и реагировать на внешнюю среду, получая информацию со специальных датчиков. При получении команды от центрального вычислительного устройства на движение в указанную точку роботу необходимо проложить маршрут последовательных перемещений, чтобы затем по нему двигаться. Надо отметить, что навигация - лишь один из компонентов работающей системы, поэтому необходимо уметь отправлять сигнал в узел, занимающийся вычислениями, о том, что указанная команда выполнена.

Из всего этого следует необходимость в следующей архитектуре общей системы (рис.4): вычислительное устройство обменивается данными с исполнительным, которое непосредственно взаимодействует с внешней средой.

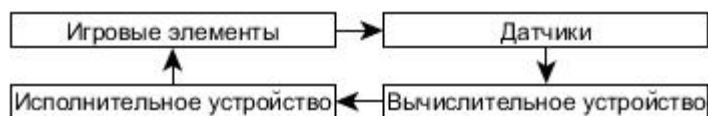


Рисунок 4 - Архитектура системы

При этом у нас есть датчики, которые способны получать информацию из внешней среды и отсылать её на обработку вычислительному устройству.

### 3 Основы алгоритма перемещения из текущей точки в заданную

Примем, что мы имеем следующий функционал: мы можем посылать команду исполнительному устройству вращать каждое из колёс независимо друг от друга на фиксированный оборот. Выведем простейшие манипуляции. При равном обороте на обоих колёсах робот движется вперёд-назад. При вращении колёс в разные стороны мы имеем поворот робота на произвольное количество градусов.

Разобьём наше игровое поле на достаточно малые квадраты, так чтобы они образовывали сетку размера  $n^2$ , каждой клетке дадим соответствующие ей координаты. Наш робот, имея указанное функционирование, может поворачиваться и двигаться так, чтобы оказаться в любой из восьми соседних клеток (рис.5). Чтобы оказаться в клетках 2, 4, 6, 8 следует при необходимости повернуть робота на 90 градусов в одну из сторон и двигаться вперёд/назад. Для движения по диагонали — повернуть на количество градусов, кратное 45 и двигаться. Тем самым мы получаем прямолинейное движение по ломаной линии: перемещение-поворот-перемещение-... и так далее. Остаётся сформулировать алгоритм последовательного передвижения из квадрата в квадрат до достижения нужной цели.

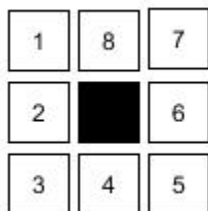


Рисунок 5 — Поле вариантов для выбора близлежащей цели движения робота

Воспользуемся наиболее эффективным для этих целей алгоритмом, известным как  $A^*$ , который позволяет сделать последовательный список из координат, по которым нужно совершить движение. Мы учитываем возможные препятствия (эти координаты помечаем как заблокированные), и алгоритм строит маршрут так, чтобы объезжать их (рис.6).

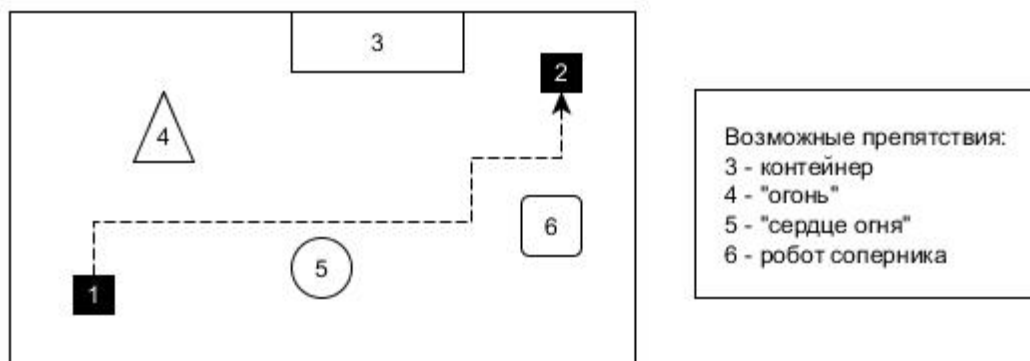


Рисунок 6 — Объезд препятствий

Подробнее с этим алгоритмом на псевдокоде или реализованном на одном из реальных языков программирования можно ознакомиться в соответствующем источнике [3].

#### 4 Реализация алгоритма перемещения из текущей точки в заданную

Опишем функцию, которую мы назовём элементарной: движение из точки с одними координатами в точку с другими (рис 7.).



Рисунок 7 - Основной алгоритм навигации мобильного робота

Прежде всего мы передаём функции начальные и конечные координаты и массив занятых препятствиями клеток. Алгоритм A\* вычисляет маршрут, учитывая препятствия и выдаёт массив последовательных соседних координат, по которым должен двигаться робот. Далее мы прогоняем массив с начала, выбирая оттуда пары координат, по которым специальная функция формирует команду вращения колёс. Если нам надо двигаться прямо, то это, как было сказано ранее, пара одинаковых чисел вида (S1, S1), где S1 – некоторое расстояние. Если по диагонали, то две пары чисел: поворот, затем движение. Эта пара

добавляется в специальную очередь. После обработки всего массива мы указанную очередь передаём на контроллер управления колёсами, который последовательно обрабатывает их и вращает колёса.

### 5 Прерывания движения

Во всей построенной системе не обойтись без качественного механизма прерываний движений робота. Они нужны по нескольким причинам:

1. Встреча робота соперника. Чтобы избежать столкновения надо остановить робота и спроектировать новый маршрут.
2. Встреча перемещаемого объекта. В этом случае надо понять, стоит ли делать что-то с ним или вернуться к предыдущим действиям.

Одно из решений выглядит следующим образом. Пусть датчик способен выдавать по запросу массив координат, которые чем-то заняты и присылать ответ на запрос, был ли объект тем, который надо переместить. Тогда опишем две функции: первая — сравнение информации о текущих занятых клетках с полученной. Если в определённом радиусе есть несовпадения, причём занята хотя бы одна клетка нашего маршрута, то маршрут надо переформировать, пошлав текущие и конечные координаты функции, описанной выше.

Вторая — оповещение центрального алгоритма о том, что встречен объект, который можно переместить. В зависимости от принятого решения может потребоваться выполнить некоторые действия и сформировать новый маршрут.

Каждая из функций может вызываться или по таймеру, или по завершении каждой итерации перемещения на соседнюю клетку.

### 6 Визуальное средство отслеживания робота на поле.

Для наглядного представления ситуации на игровом поле с точки зрения робота необходима система визуализации. Она представляет собой графическое представление объектов, которое можно отобразить, например, на экране монитора компьютера.

В простейшем случае мы имеем некоторую заготовку, на которой изображены все элементы поля. Управляя движением робота, мы знаем текущие координаты робота и текущий маршрут. Всё это можно схематично визуализировать. Если усложнить подход, то имея датчики, передающие информацию об игровом поле, мы в определенном радиусе можем заметить, например, робота соперника и следить за его движением (рис.8).

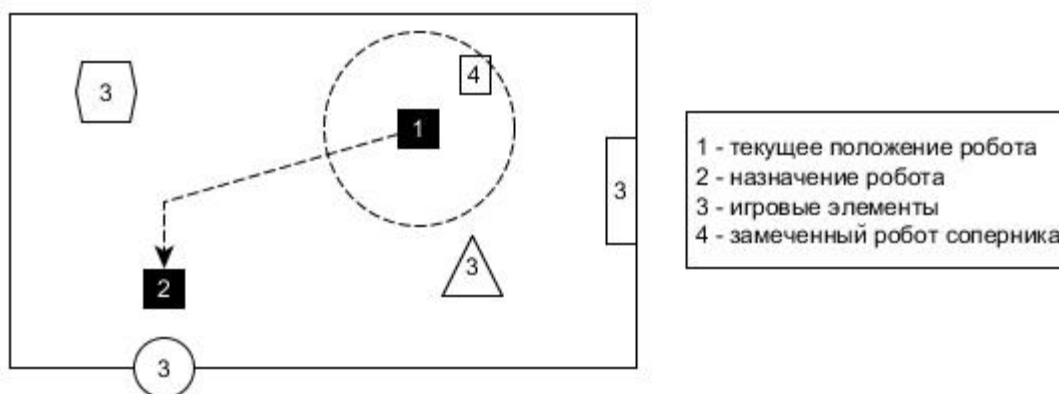


Рисунок 8 — Визуализация (робот 4 не виден без датчиков)

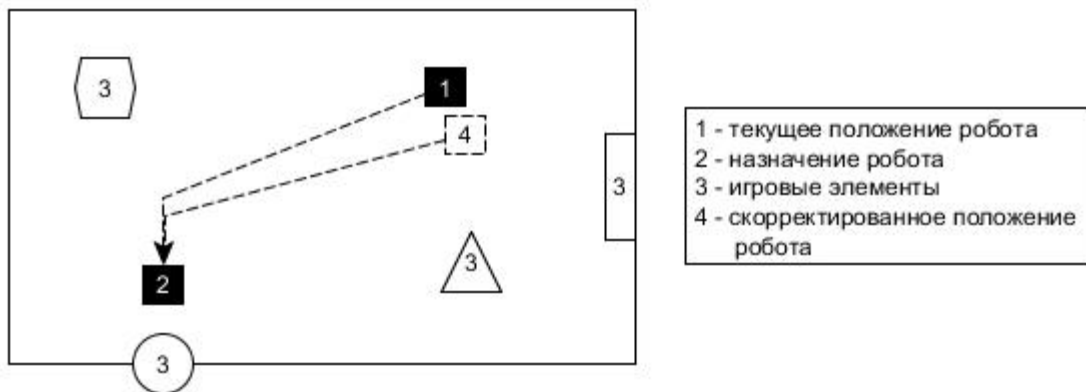


Рисунок 9 — Визуализация с корректировкой

Наконец, мы можем отслеживать реальное положение робота, используя соответствующие датчики. Как известно, ожидаемое и реальное положение робота могут не совпадать из-за неточности работы, например, механизма колёс робота.

### Заключение

Разработка навигационных моделей является одной из приоритетных задач в автономной мобильной робототехнике. Усовершенствование систем на любом из этапов даёт огромный рывок в плане эффективности роботов. При этом важны как решения локальных задач, так и глобальные теоретические разработки, например совершенствование алгоритмов прокладки маршрутов.

В статье дан один из возможных подходов к поставленной задаче. Автор планирует дальнейшую работу над реализацией описанной модели на различных устройствах и визуализацией действий робота.

### Литература

1. Молодежные соревнования мобильных автономных роботов Евробот. Электронный ресурс. Режим доступа: <http://www.eurobot.org>, <http://www.eurobot-russia.ru> — Проверено 31.01.2014.
2. Правила молодежных соревнований роботов Евробот 2014: «Первоботная ЭРА»: Пер. с англ. С дополнениями НОК Евробот России./ Общ. Ред. А.В. Юдина. Изд. 1-е.— М.: Национальный организационный комитет Евробот России, 2013. — 35с., ил. Электронный ресурс. Режим доступа: [http://eurobot-russia.org/main/media/EA2014\\_rules\\_RU\\_alpha.pdf](http://eurobot-russia.org/main/media/EA2014_rules_RU_alpha.pdf) — Проверено 31.01.2014.
3. Информация по алгоритму A\*. Электронный ресурс. Режим доступа: [http://neerc.ifmo.ru/wiki/index.php?title=Алгоритм\\_A\\*](http://neerc.ifmo.ru/wiki/index.php?title=Алгоритм_A*) — Проверено 31.01.2014.
4. Расчет траектории движения мобильного робота в частной задаче перемещения объектов — Электронный ресурс. Режим доступа: [http://class.skycluster.net/\\_media/documentation/archive/2011/files/2011\\_iu4\\_conference\\_chistyako v.pdf](http://class.skycluster.net/_media/documentation/archive/2011/files/2011_iu4_conference_chistyako v.pdf) — Проверено 31.01.2014.
5. Афонин В.Л., Макушкин В.А. Интеллектуальные робототехнические системы. — М.: Интернет-университет информационных технологий - ИНТУИТ.ру, 2005 — 208 с.