

АВТОМАТНОЕ УПРАВЛЕНИЕ И МЕТОД ПОТЕНЦИАЛЬНЫХ ПОЛЕЙ В ЗАДАЧЕ ДВИЖЕНИЯ МОБИЛЬНОГО РОБОТА В СРЕДЕ С ПРЕПЯТСТВИЯМИ[#]

В. Э. Карпов

НИЦ «Курчатовский институт», г. Москва

✉ karpov-ve@yandex.ru

Аннотация. Рассматривается один из методов решения задачи локальной навигации мобильного робота, работающего в экстремальных условиях, связанных с отсутствием средств глобальной навигации и централизованного управления, а также с ограниченной сенсорикой. В основе локальной навигации лежит метод потенциальных полей, причем построенные на его базе двигательные программы перенесены на уровень элементарных действий. Формирование сложного поведения осуществляется метаавтоматами, реализующими иерархию управления. Основной задачей, решаемой роботом, является движение в заданный район или область пространства. В рамках этой задачи рассматривается базовая процедура – обход препятствий – которая также реализована на базе метода потенциальных полей. Приведены результаты экспериментов трех уровней: вычислительного, имитационного (физического) и на реальных роботах. Во всех них использовались одна и та же вычислительная модель и управляющие автоматы. Показано, что успешность выполнения задачи локальной навигации определяется размером области видимости робота, но при этом унифицированность механизмов управления позволяет расширять эту область естественным образом.

Ключевые слова: метод потенциалов, автоматное управление, метаавтоматы, модели движения, мобильный робот, локальная навигация, обход препятствий, иерархия управления.

ВВЕДЕНИЕ

Одним из прикладных направлений развития систем групповой робототехники является концепция применения массовых группировок наземных мобильных робототехнических систем в экстремальных условиях с потерями. Эта концепция подразумевает использование множества сравнительно простых автономных мобильных платформ, способных функционировать в условиях жестких ограничений. Ограничения подразумевают:

- отсутствие связи с оператором,

– автономность и децентрализованный характер управления,

- отсутствие средств глобальной навигации,
- отсутствие карты местности,
- ограниченность сенсорики (отсутствие средств активной детекции, например мощных лидаров).

При этом определение целей функционирования носит групповой характер.

Среди задач, которые должны решаться группой таких роботов, выделяется базовая – задача доставки некоторой полезной нагрузки в заданный район или область пространства. Роботам задаются целевой пеленг и примерное удаление от точки старта. Сами роботы оснащены камерами, дальномерами или локаторами, способными определить

[#] Работа проведена в рамках выполнения государственного задания НИЦ «Курчатовский институт».



препятствия или непроходимые участки, а также средствами локальной связи для определения состояния соседей – роботов своей группы. В отсутствие глобальной навигации и карты местности роботы могут опираться на анализ локального окружения. Предполагается, что роботы должны двигаться компактной группой, избегая столкновений и обходя препятствия, выдерживая при этом заданное направление. Групповой характер организации подобного рода систем обусловлен, во-первых, требуемой малой чувствительностью к потерям и, во-вторых, ожидаемыми синергетическими эффектами, возникающими при групповом движении (взаимодействии).

В настоящей работе будет рассмотрена основная процедура решения задачи доставки – движение робота по пеленгу с обходом препятствий.

1. ЗАДАЧА ДВИЖЕНИЯ ПО ПЕЛЕНГУ

Среди множества методов планирования пути в средах с препятствиями (см., например, обзор [1]) особое место занимают модели локальной навигации, основанные на использовании потенциальных полей (ПП). Важно, что эти методы позволяют решать не только задачу построения траектории индивидуального движения, но реализовывать также правила согласованного движения группы агентов. Общая идея метода ПП состоит в движении вдоль линий векторного поля, потенциальная функция которого отражает конфигурацию наблюдаемых объектов, препятствий и их форму, а также цель движения. Формально метод ПП относится к более общему семейству – методам так называемого потенциального наведения и, несмотря на более чем полувековую историю, по-прежнему активно исследуется (см., например, работы А. Б. и Н. Б. Филимоновых [2, 3] или обзор Х. Ксинга [4]). В первую очередь это связано с тем, что метод ПП имеет очевидные неустранимые ограничения: тенденцию застревания в локальном минимуме, осцилляции, а также проблемы при решении траекторных задач при обходе препятствий сложной формы (см. работу [5]). К тому же, применение метода ПП становится затруднительным, когда робот отличен от удобной системы с дифференциальным приводом (см. работу [6]). В целом общая тенденция развития методов управления на основе ПП сводится к разработке гибридных систем управления, когда механизм ПП используется для построения траекторий движения на тактическом уровне.

Задача движения в среде с препятствиями обычно относится к категории комплексных, требующих предварительного анализа окружения. Для этого разрабатываются весьма специфические алгоритмы построения пути с огибанием препятствий, использующие как «традиционные» продукционные механизмы (см., например, статью [7]), так и достаточно тяжелые графовые методы поиска (см. публикацию [8] или классическую работу О'Рурка [9]). Несколько особняком стоят так называемые Bug-методы обхода, в которых робот следует контуру препятствия, постоянно оценивая рассогласование между текущей ориентацией и требуемым пеленгом (см. статью [1]). Такой подход обладает существенным ограничением: фактически робот выходит к точке продолжения траектории, находящейся за препятствием, совершая избыточный обход. Но самая большая сложность состоит в том, что процедура следования контуру подразумевает наличие очень развитой сенсорики и способности реконструирования этого контура.

Тем не менее, в настоящей работе в качестве основы локального планирования траектории предлагается именно метод ПП, а вопросы обхода препятствий решаются путем введения дополнительного – автоматного – уровня управления. При этом основными причинами интереса к методу ПП являются его вычислительная простота и общность, что крайне важно для специфики решаемой задачи доставки.

В данной статье не затрагиваются вопросы применения метода ПП для решения групповых задач. Реализация разнообразных законов стайного движения описана в работе [10]. Здесь же сосредоточимся именно на решении частной задачи – обхода препятствий.

Начнем с описания реализации метода ПП для мобильного робота.

1.1. Метод потенциальных полей

Модель агента. Агент представляет собой объект, оснащенный сенсорами и способный к перемещению, для общности, в трехмерном пространстве. Агент A задается пятеркой

$$Agent = \langle name, C, R, S, F \rangle,$$

где $name$ – имя агента; C – множество характеристик агента; R – его пространственное положение; S – состояние сенсорной системы; F – функция управления.

Функция управления F определяет пространственное положение агента:

$$F(C, S, R) \rightarrow R.$$

Базовыми характеристиками агента для рассматриваемой задачи является тройка

$$C = \langle t, r, \omega \rangle.$$

где t – тип агента, $t \in \mathbb{N}$; r – размер агента (его геометрические характеристики, $r \in \mathbb{R}$); ω – вес агента ($\omega \in \mathbb{R}$). Введение типов агентов, весов и размеров необходимо для реализации функций группового управления, когда агенту необходимо ориентироваться на остальных членов группы, причем исходя из наблюдаемых характеристик своих соседей.

Пространственное положение R задается тройкой $R = \langle X, v, A \rangle$. Здесь X – координаты агента, $X \in \mathbb{R}^3$; v – скорость движения ($v \in \mathbb{R}$); $A = \langle \alpha_{yaw}, \alpha_{pitch}, \alpha_{roll} \rangle$ – тройка углов Эйлера, где α_{yaw} , α_{pitch} , α_{roll} – углы рыскания, тангажа и крена соответственно.

Сенсорная система агента позволяет получить информацию о состоянии, координатах, скоростях и углах ориентации некоторого множества объектов Ω_s , попавших в его зону видимости – некоторую область пространства $\mathbb{R}_s^3$ вблизи агента, например, его переднюю полусферу. Робот может быть оснащен различными датчиками, показания которых дополняют множество Ω_s , см. рис.1.

Множество определяется как $\Omega_s = \{o_i^k\}$. Здесь o_i^k – распознаваемые сенсорикой робота объекты – соседи, препятствия, целевые ориентиры и т. п. При этом

$$o_i^k = \langle (x_1, x_2, x_3), k \rangle,$$

где (x_1, x_2, x_3) – координаты; k – класс, к которому принадлежит объект (соседи, препятствия, целевые объекты и т. п.). Каждый элемент o_i^k определяет свой вклад (знак и величину) в итоговый вектор движения. Важнейшей характеристикой области Ω_s является ее радиус r_Ω – расстояние, на котором агент может увидеть объект. Радиус r_Ω может определяться как возможностями сенсорики агента, так и полученной извне информацией

(априорными знаниями агента об окружающем пространстве). Если радиус r_Ω соизмерим с размерами области, в которой функционирует агент, то получим ситуацию знания полной информации о среде и локальное планирование движения превратится в глобальную навигацию.

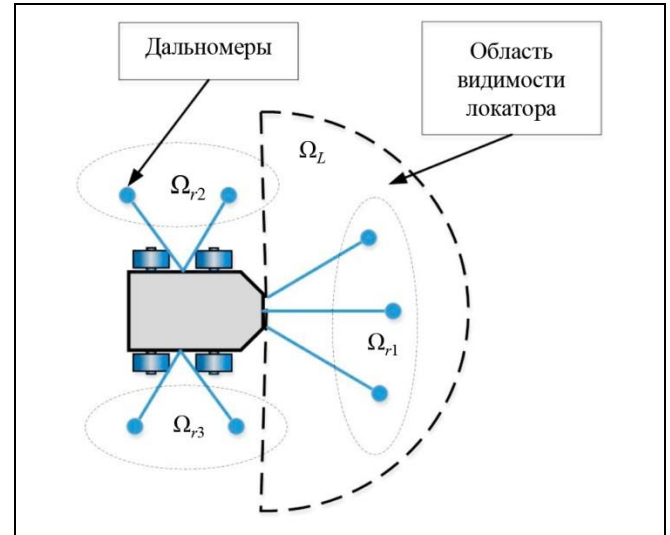


Рис. 1. Область видимости агента Ω_s , формируемая множествами объектов, зарегистрированных локатором Ω_L и дальномерами Ω_{ri}

Движение агента. Направление и скорость движения агента определяются суммой потенциалов полей в его позиции. Это значит, что каждый объект из области видимости Ω_s рассматривается как источник потенциала (положительного или отрицательного), и сумма этих потенциалов определяет значение функции управления как пары

$$F_{ctl} = \langle v_{ctl}, A_{ctl} \rangle, \quad A_{ctl} = \langle \alpha_{yaw}, \alpha_{pitch}, \alpha_{roll} \rangle,$$

где v_{ctl} – скорость движения (модуль); A_{ctl} – направление движения.

Обозначим через $\langle X_s, v_s, A_s \rangle$ средние значения координат, скорости и ориентации наблюдаемых в области Ω_s объектов. Компоненты координат центра масс наблюдаемых объектов вычисляются как

$$X_s^k = \frac{\sum_{o_i \in \Omega_s} x_k^i}{\sum_{o_i \in \Omega_s} \omega_i}, \quad k = 1, 2, 3, \quad (1)$$

где x_k^i – компоненты вектора координат наблюдаемого объекта; ω_i – его вес. Аналогично определяются средние скорости v_s и вектор ориентации A_s .



Эти характеристики области Ω_s позволяют задавать гибкие законы движения: не только двигаться по направлению, определяемому центром масс X_s , но и согласовывать направление движения и скорость с другими агентами, решать задачу преследования, избегания препятствий и т. п.

Итак, на входе имеются состояние агента $\langle X, v, A \rangle$ и наблюдаемые величины $\langle X_s, v_s, A_s \rangle$, задан закон взаимодействия в виде функции $F_A(X, X_s, A_s)$, а искомым выходом является управление $\langle v_{ctl}, A_{ctl} \rangle$. Функция F_A определяет компоненты вектора управления $\langle v_{ctl}, A_{ctl} \rangle$, причем

$$v_{ctl} = |F_A(X, X_s, A_s)|, A_{ctl} = \arg(F_A(X, X_s, A_s)).$$

Например, для закона когезии (стремления держаться поближе к «своим») F_{coh} итоговая скорость в общем случае может определяться собственной скоростью агента и расстоянием до центра масс группы r_s , определяемым по координатам X и X_s :

$$v_{ctl} = |F_{coh}| = \min\left(v_{max}, \frac{k_v}{r_s}\right),$$

где v_{max} – предельная скорость движения; k_v – некоторый коэффициент. А направление движения может быть определено из положения центра масс наблюдаемых «своих».

Далее, чтобы не загромождать изложение дополнительными операторами и преобразованиями, будем полагать, что вектор F естественно представим в виде явной пары <длина, ориентация>, т. е. $F = \langle |F|, \arg(F) \rangle = \langle v, a \rangle$.

Если рассматривается простое движение агента по пеленгу, заданному парой углов α_{yaw} и α_{pitch} , то

$$F_{bear} = \langle v_{ctl}, A_{ctl} \rangle = \langle v_{ctl} = \text{const}, \langle \alpha_{yaw}, \alpha_{pitch}, 0 \rangle \rangle. (2)$$

Общий закон управления движением. Пусть область Ω_s позволяет выделять классы объектов Ω_s^i , по отношению к которым реализуются разные законы движения F_i , как зависящие от области видимости (приближение, отталкивание), так и не связанные с ней (свободное блуждание, движение по пеленгу и т. д.). Управление F_{ctl} в общем случае будет определяться суммарным воздействием всех действующих на агента сил, представленным в виде компонент $\langle v_{ctl}, A_{ctl} \rangle$:

$$F_{ctl} = \langle v_{ctl}, A_{ctl} \rangle = \sum_{i=1}^n k_i F_i(\Omega_s^i), (3)$$

где $F_i(\Omega_s^i)$ – сумма векторов сил, определяемых положением объектов из области Ω_s^i по отношению к агенту; k_i – коэффициенты, значения которых определяют характер поведения агента (как он реагирует на объекты класса i). Обозначим множество этих коэффициентов через K . Подбор коэффициентов позволяет, например, реализовать классическую модель движения группы агентов К. Рейнольдса [11]. Правила движения агентов заключаются в том, что агент не должен приближаться близко к другим агентам, должен двигаться туда же, куда и все, и не отдаляться от других. Кроме того, к ним добавляется стремление придерживаться заданного направления движения (целевой пеленг), а также не приближаться к препятствиям и (или) опасным областям. Формально это означает, что итоговое направление движения агента определяется суммой сил отталкивания-притяжения F_{res} вида

$$F_{res} = k_{goal} F_{goal} + k_{sep} F_{sep} + k_{align} F_{align} + k_{coh} F_{coh} + k_{obst} F_{obst}, (4)$$

где F_{goal} – вектор, определяющий движение по целевому пеленгу; F_{sep} – сила отталкивания от соседей (избегание столкновений); F_{align} – сила выравнивания курса (придерживаться среднего курса соседей); F_{coh} – сила притяжения к «своим» (ориентация на среднее положение соседей); F_{obst} – сила отталкивания от препятствий. Величины k – это коэффициенты, определяющие значимость вклада сил с соответствующими индексами.

Закон (4) можно естественным образом расширять путем добавления новых факторов, определяющих тенденцию движения агента, получая в итоге общую схему управления (3).

1.2. Управление движением

1.2.1. Метаавтоматы

Процедуру движения агента в потенциальном поле можно рассматривать как некоторое действие. Реализация более сложного поведения, состоящего из последовательности действий, требует

иных механизмов. Одним из таких механизмов является конечно-автоматное управление, например использование автоматов Мили. Сам по себе автомат Мили является весьма ограниченным по своим возможностям, а описание развитого поведения требует построения сложных автоматов, что нивелирует такие важные свойства автоматного управления, как простота и наглядность. Вместе с тем, существует методология метаавтоматного управления, при которой элементарные действия, их последовательности и комплексы последовательностей представлены в виде иерархий соответствующих автоматов.

Метаавтомат – это автомат Мили, у которого пометки дуг определяют выполняемые агентом процедуры. Если действиями автомата являются активизации низкоуровневых двигательных функций, то такой автомат имеет ранг 0 – M^0 . Метаавтомат ранга 1 M^1 – это автомат, действия которого заключаются в управлении автоматами M^0 . Метаавтомат M^n ранга n управляет автоматами M^{n-1} . Таким образом происходит структуризация иерархии управления в автоматном представлении. Более подробно это изложено в работе [10]. Здесь важно отметить, что, во-первых, класс реализуемых метаавтоматами алгоритмов по-прежнему ограничен конечно-автоматными представлениями, а во-вторых, иерархия метаавтоматов – это лишь способ декомпозиции управления, в отличие, допустим, от парадигмы вложенных автоматов (см., например, автоматную школу программирования А. Шалыто [12]).

Итак, имеем двухуровневую систему управления:

1. Базовый уровень. Используются реактивные модели прямого управления, определяющие вектор движения на основе анализа области Ω_s . На этом уровне отрабатываются двигательные программы типа сближения, избегания препятствий, преследования и т. д. Эти действия-программы реализуются методом потенциальных полей. С позиции обобщенной модели управления (3) речь идет о задании коэффициентов из множества K .

2. Уровень поведения. Реализуется на основе дискретно-событийных (автоматных) моделей. На этом уровне метаавтоматы формируют последовательности двигательных программ, реализуя сложные поведенческие процедуры типа движения по пеленгу с обходом препятствий, движение по заданному маршруту (патрулирование) и т. п.

Для решения задачи движения в область пространства по пеленгу требуются две двигательные процедуры: программа свободного движения, реализующая обобщенное управление (3), а также программа движения вдоль препятствия.

1.2.2. Программа свободного движения

Эта программа выполняется тогда, когда робот не видит препятствий на своем пути. Движение робота определяется множеством коэффициентов K , например

$$K = \{k_{\text{bear}}, k_{\text{obj}}, k_{\text{coh}}\},$$

где k_{bear} , k_{obj} , k_{coh} – это величины, определяющие значимость сил движения по заданному пеленгу, движения к объекту интереса и когезии соответственно.

Здесь предполагается, что в отсутствие препятствий робот движется по некоторому пеленгу, в случае обнаружения объекта интереса (точки финиша) робот идет к нему и при этом стремится двигаться в направлении движения группы (правило когезии, «куда все, туда и я»).

Движение осуществляется следующим образом:

- **Движение по пеленгу.** Определяется заданными углами α_{yaw} , α_{pitch} согласно формуле (2)

$$F_{\text{bear}} = \langle v_{\text{ctl}}, A_{\text{ctl}} \rangle = \langle v_{\text{ctl}} = \text{const}, \langle \alpha_{\text{yaw}}, \alpha_{\text{pitch}}, 0 \rangle \rangle.$$

- **Движение к объекту интереса.** Пусть робот регистрирует множество целевых объектов Ω_s^{obj} . Исходя из их координат, направление агента будет определяться как движение к их центру масс, см. формулу (1):

$$F_{\text{obj}} = \langle v_{\text{ctl}}, A_{\text{ctl}} \rangle = \langle v_{\text{ctl}} = \text{const}, A_{\text{obj}} \rangle,$$

где A_{obj} – суммарный вектор направления к центру масс наблюдаемых объектов.

- **Когезия (составление, притяжение).** Пусть робот регистрирует множество своих соседей Ω_s^{coh} . Если известны координаты членов группы – элементы множества Ω_s^{coh} , то скорость и направление агента будет определяться как движение к их центру масс.

Итоговое направление движения определится как

$$F_{\text{ctl}} = \langle v_{\text{ctl}}, A_{\text{ctl}} \rangle = k_{\text{bear}} F_{\text{bear}} + k_{\text{obj}} F_{\text{obj}} + k_{\text{coh}} F_{\text{coh}}.$$



При этом специфика решаемой задачи определяется заданием значимости соответствующих факторов движения. Например, при $k_{coh} \ll k_{bear} \ll k_{obj}$ основным приоритетом будет поиск цели, а при $k_{coh} \ll k_{obj} \ll k_{bear}$ – движение по пеленгу.

1.2.3. Движение вдоль препятствий

Двигательная программа, позволяющая роботу перемещаться вдоль препятствия, выглядит сложнее. Фактически речь идет о процедуре движения вдоль наблюдаемых объектов-ориентиров, в качестве которых могут рассматриваться препятствия или фрагменты таковых. Рассмотрим далее базовую процедуру движения агента вдоль ориентиров с использованием метода потенциальных полей. Данная процедура имеет множество параметров, определяющих характер этого движения, и возвращает закон управления в виде пары $\langle v_{res}, A_{res} \rangle$ – скорости движения и его направления. Процедура реентерабельна и вызывается на каждом шаге моделирования.

Входные параметры:

- координаты агента $c_a = (x_a, y_a, z_a)$ и углы его ориентации $\alpha_a = \alpha_{yaw}^a, \alpha_{pitch}^a, \alpha_{roll}^a$;
- множество наблюдаемых объектов $\Omega = \{s_i\}$, для каждого элемента которого известны его координаты (x_s^i, y_s^i, z_s^i) (все объекты точечные);
- v_0 – линейная крейсерская скорость движения агента;
- α_{add} – угол добавки к рассчитанному курсу, который определяет, под каким углом по отношению к ориентирам движется агент;
- dim_{attr} – степень расстояния между объектами для притяжения;
- dim_{sep} – степень расстояния между объектами для закона отталкивания (чем она больше, тем «точнее» агент идет вдоль пятен (препятствий));
- d_f – коэффициент, определяющий границу сил отталкивания (притяжения) (чем он меньше, тем ближе к препятствиям идет агент).

В итоге параметры dim_{attr} и dim_{sep} определяют точность траектории, а d_f – характер переходного процесса.

Суть процедуры заключается в том, что каж-

дый объект из области видимости Ω обладает как притягивающим, так и отталкивающим потенциалом. При этом отталкивающее поле вблизи объекта должно быть значительно сильнее притягивающего. Это достигается введением некоторого порогового расстояния до препятствия, а также характером отталкивающего поля. Значение потенциала отталкивания, естественно, должно неограниченно возрастать в непосредственной близости от объекта.

Кроме того, важен учет влияния объектов, находящихся по курсу движения робота.

Сила притяжения. Обозначим через d_i^a расхождение направлений векторов ориентации агента и вектора направления на объект: $d_i^a = A - A_i$. Тогда кортеж $\langle v_s, A_s \rangle$ определяется как сумма векторов $\langle v_{res}, A_{res} \rangle = \sum_{o_i \in \Omega_B} (F_i^{attr}, A_i)$. Здесь F_i – значение «силы притяжения» объекта o_i из области Ω :

$$F_i^{attr} = \begin{cases} 0, & \text{если } r_i < r_{min}, \\ \frac{k_{cos}}{r_i^{dim_{attr}}}, & \text{иначе,} \end{cases}$$

$$k_{cos} = \cos \max_j (d_i^a)_j,$$

где r_i – расстояние до объекта. Смысл введения коэффициента k_{cos} заключается в том, чтобы для агента были более предпочтительны те ориентиры, которые расположены по ходу его движения. Чтобы не увеличивать влияние более близких ориентиров, целесообразно в знаменателе иметь расстояние r_i в степени $dim_{attr} \geq 1$. Кроме того, совсем близкие ориентиры не должны участвовать в определении направления движения. Для этого вводится некоторое минимальное расстояние r_{min} , кратное, например, размеру агента.

Сила отталкивания F_i^{sep} определяется аналогично, при этом степень расстояния dim_{sep} должна быть больше dim_{attr} .

После определения направления результирующего поля $F_i = F_i^{attr} + F_i^{sep}$ к его значению добавляется дополнительный угол ориентации α_{add} . Этот угол определяет, каким бортом будет происходить обход.

Исходя из этого, ниже представлен алгоритм движения агента вдоль цепочки объектов.

```

1. function SpotMv( $c_a, \alpha_a, \Omega, v_0, \alpha_{add}, dim_{attr}, dim_{sep}, d_f$ )
2.   %-- Этап 1. Если ориентиры не видны, то агент вращается на месте (ищет ори-
   энтиры) .
3.   if  $\Omega = \emptyset$  then
4.      $v_{res} := v_0/2, A_{res} := \langle \alpha_{yaw}^a + d_{yaw}, 0, 0 \rangle$ 
5.     return  $\langle v_{res}, A_{res} \rangle$ 
6.   endif
7.   %-- Этап 2. Инициализация значения результирующего управления,
8.   %-- а также сил притяжения и отталкивания.
9.    $\langle v_{res}, A_{res} \rangle := \langle 0, \langle 0, 0, 0 \rangle \rangle$ 
10.   $\langle v_{attr}, A_{attr} \rangle := \langle 0, \langle 0, 0, 0 \rangle \rangle$ 
11.   $\langle v_{sep}, A_{sep} \rangle := \langle 0, \langle 0, 0, 0 \rangle \rangle$ 
12.  %-- Этап 3. Цикл по всем объектам s из зоны видимости  $\Omega$ 
13.  for s in  $\Omega$  do
14.    %-- 3.1. Определение нормированного расстояния d до объекта и угла  $\alpha_s$ 
15.     $A := (x_s - x_a, y_s - y_a, z_s - z_a)$ 
16.     $d := |A|/d_f$ 
17.    if ( $d_{min} > d$ ) then  $d_{min} := d$ 
18.     $\alpha_s = \langle \alpha_{yaw}^s, \alpha_{pitch}^s, \alpha_{roll}^s \rangle := \text{УглыЭйлера}(A)$ 
19.    %-- 3.2. Учет направления. Косинусный закон: значение имеют те
20.    %-- ориентиры, которые впереди
21.     $k_{cos} := \cos(\alpha_{yaw}^a - \alpha_{yaw}^s)$ 
22.    %-- 3.3. Учет фактора близости (видимость)
23.     $k_{near} := \begin{cases} 1, \text{если } d > 0.5 \\ 0 \text{ иначе} \end{cases}$ 
24.    %-- 3.4. Расчет сил притяжения/отталкивания
25.    if ( $d > 1$ ) then -- Притяжение
26.       $force := k_{cos} k_{near} d^{dim_{attr}}$ 
27.       $\langle v_{attr}, A_{attr} \rangle := \langle v_{attr}, A_{attr} \rangle + \langle force, \alpha_s \rangle$ 
28.    else -- Отталкивание
29.       $\alpha_{sep} := \langle \alpha_{yaw}^s + \frac{\pi}{2}, \alpha_{pitch}^s + \frac{\pi}{2}, \alpha_{roll}^s \rangle$ 
30.       $force := k_{cos} k_{near} d^{dim_{sep}}$ 
31.       $\langle v_{sep}, A_{sep} \rangle := \langle v_{sep}, A_{sep} \rangle + \langle force, \alpha_{sep} \rangle$ 
32.    endif
33.  endfor
34.  %-- Этап 4. Расчет итоговой силы
35.   $\langle v_{res}, A_{res} \rangle := \langle v_{attr}, A_{attr} \rangle + \langle v_{sep}, A_{sep} \rangle$ 
36.  %-- Добавление к углу рыскания дополнительного значения  $\alpha_{rot}$ , чтобы иди
   %-- боком. Значение этой добавки зависит от близости к препятствию r.
37.   $r := \max(d_{force}(d_{min} - 1), 0)$ 
38.   $\alpha_{rot} := \alpha_{add} e^{-r}$ 
39.   $A_{res} := \langle \alpha_{res}^{yaw} + \alpha_{rot}, \alpha_{res}^{pitch}, \alpha_{res}^{roll} \rangle$ 
40.  %-- Скорость постоянная
41.   $v_{res} := v_0$ 
42.  return  $\langle v_{res}, A_{res} \rangle$ 

```

Итак, процедура движения в заданную область пространства требует реализации двух двигательных программ – программы свободного движения и движения вдоль ориентиров. Рассмотрим далее автоматный уровень управления, отвечающий за переключение между этими программами.

1.3. Управляющие автоматы

Для унификации управления будем полагать, что двигательные процедуры запускаются автоматом $dctl$ с одним состоянием. При запуске автомата указывается аргумент arg – имя двигательной программы. Выходом автомата является управление $F_{ctl} = \langle v_{res}, A_{res} \rangle$. Для рассматриваемой задачи аргументами будут имена процедур свободного движения $FreeMv$ и движения вдоль препятствий $SpotMv$.

Метаавтомат движения по пеленгу с обходом препятствия реализует следующий алгоритм поведения: если путь свободен, то робот движется по программе свободного движения (активен автомат $dctl$ с аргументом $FreeMv$). В противном случае управление передается автомату, активизирующему процедуру движения вдоль цепочки объектов ($dctl$ с аргументом $FObstAlong$). Программа $FObstAlong$ – это реализация функции $SpotMv$ с параметрами, определяющими требуемую специфику движения робота (движение поближе (подальше) от препятствия, точность и т. п.).

На рис. 2 изображены автомат $dctl$ и метаавтомат движения по пеленгу. Пометки ребер содержат условие перехода (в числителе) и выполняемую процедуру. Для метаавтомата это запуск автомата $dctl$ с соответствующим аргументом.

Более сложное поведение может потребовать создания соответствующих метаавтоматов того же ранга M^1 и автомата второго ранга для активизации автоматов ранга M^1 . Автомат M^2 может, например, отвечать за определение условий завершения движения, за запуск программы (метаавтомата) преследования и т. п.

2. ЭКСПЕРИМЕНТЫ

Эксперименты проводились на моделях трех видов: в дискретно-событийной системе много-агентного моделирования Kvoqum, в среде физического моделирования Gazebo, а также на реальных роботах – как в лабораторных условиях, так и работающих на открытом пространстве.

Во всех экспериментах робот представлял собой колесную платформу с дифференциальным приводом, оснащенную локатором-дальномером, работающим в передней полуплоскости, а также семью дальномерами, расположенными так, как показано условно на рис. 3.

Кроме того, роботы были оснащены компасом на основе гироскопа. С частотой 10 Гц показания дальномеров формируют сферические координаты регистрируемых препятствий, которые переводятся в координаты объектов-препятствий множества Ω . Локатор, как модельный, так и реальный, тоже обладает дискретностью шага (длина вектора показаний локатора равна 180 из-за минимального шага в 1° , хотя реально шаг составляет 10° в силу того, что используются дальномеры с рабочим углом порядка 15°), поэтому результирующие потенциалы формируются сравнительно небольшим множеством объектов.

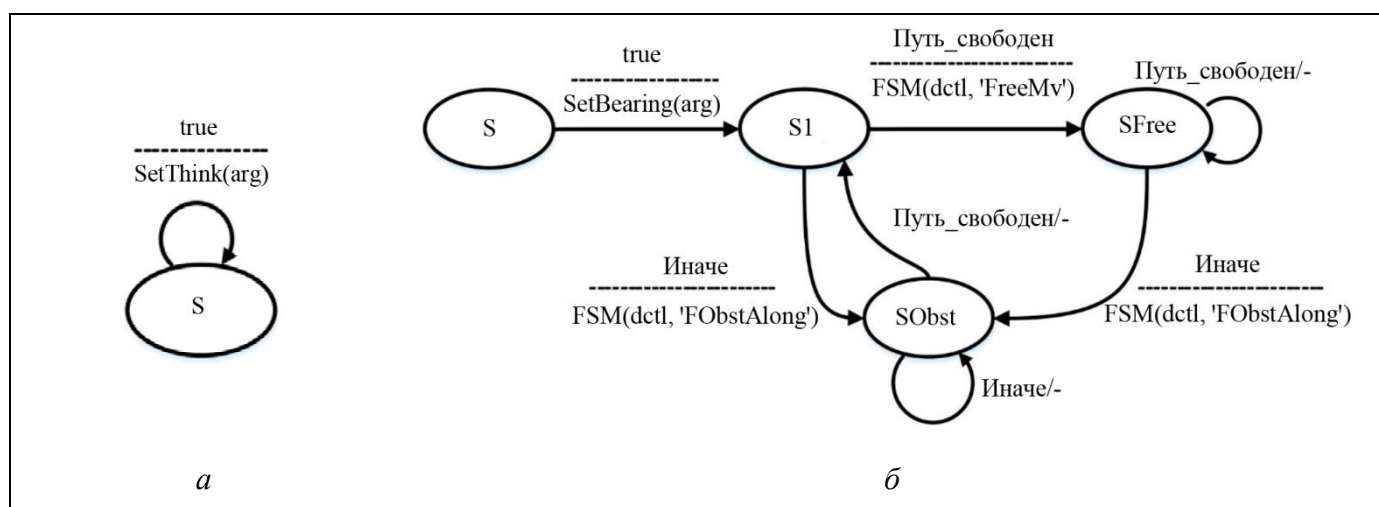


Рис. 2. Автоматы: а – автомат уровня 0 ($dctl$, запуск двигательных программ); б – мета-автомат движения по пеленгу с обходом препятствий

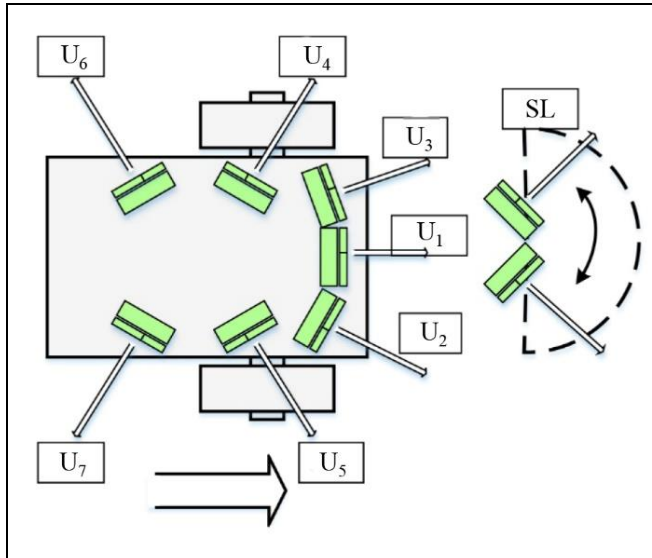


Рис. 3. Расположение датчиков: U_1 – U_7 – дальнометры, SL – локатор с обзором 180°

Важно, что во всех экспериментах – вычислительных, имитационных и реальных – использовались единое вычислительное ядро, рассчитывающее потенциалы полей, а также одни и те же управляющие автоматы.

2.1. Вычислительные эксперименты

Основная серия экспериментов проводилась в среде имитационного моделирования Kvorum. Kvorum – это система дискретно-событийного моделирования, в которой среда агентов представле-

на дискретным (клеточным) полем [13]. Поскольку двигательные процедуры на основе ПП имеют принципиально непрерывный характер, то полученное управление вида $F_{ctl} = \langle v_{res}, A_{res} \rangle$ транслировалось в дискретные команды для модельного объекта: шаг движения с заданной постоянной скоростью и дискретный поворот. Тем не менее, такое загромождение управления не повлияло на итоговый результат. На рис. 4, а приведены пример поля с препятствиями и траектория движения агента. Физическая модель этого поля изображена на рис. 4, б.

Эксперименты заключались в определении зависимости доли успешных миссий (движение в заданную полуплоскость) $E(\rho)$ от степени насыщенности поля препятствиями ρ . Здесь $\rho = N_{obst}/N$, где N_{obst} – количество клеток-препятствий; N – общее количество клеток на модельном поле. Значение ρ выбиралось из диапазона 0,005...0,06 с шагом 0,005. Для каждого значения ρ проводилось десять экспериментов: генерировались поля со случайно расположенными препятствиями (равномерное распределение) и осуществлялся запуск агента, стартующего из одной и той же точки. Миссия считалась успешной, если агент достигал целевой области не более чем за некоторое число тактов $T_{max} = N$ (грубая пессимистическая оценка). Для каждой серии определялось значение $E(\rho)$. На рис. 5 представлены результаты этих экспериментов.

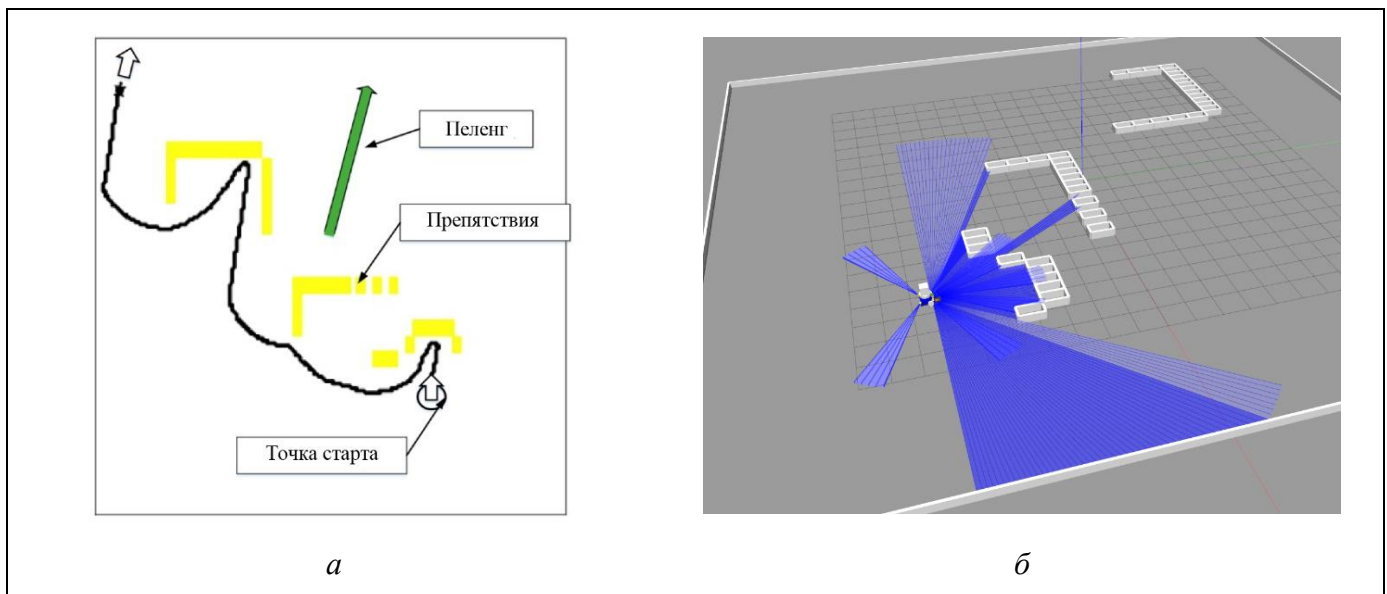


Рис. 4. Примеры: а – иллюстративного эксперимента в системе имитационного моделирования Kvorum; б – модели робота в среде Gazebo. Топология препятствий одна и та же

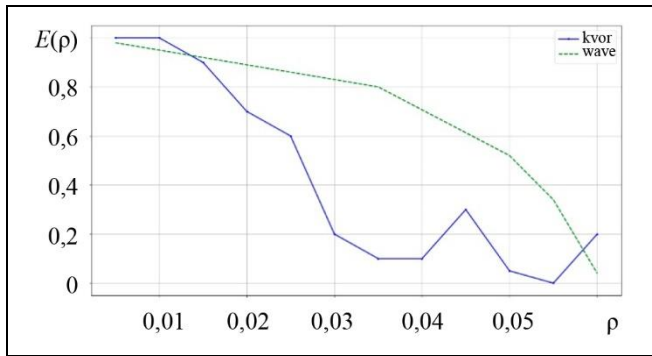


Рис. 5. Результаты вычислительных экспериментов

На рис. 5 изображены зависимости $E(\rho)$, полученные в ходе имитационного моделирования в системе Kvorum (график kvor), а также результаты работы волнового алгоритма (график wave) в тех же условиях. Волновой алгоритм здесь выбран как эталонный. Для обоих алгоритмов стартовая точка (начальное положение робота) находилась на нижней грани прямоугольного поля, а целевая – на верхней. Кроме того, волновой алгоритм работал с полем, на котором препятствия масштабировались. Это связано с тем, что должна существовать зона безопасности при движении в среде с препятствиями: робот должен находиться на некотором расстоянии от них.

Разумеется, волновой алгоритм имеет лучшие показатели в силу гарантированного нахождения решения для сред, в которых путь существует. Тем не менее видно, что как для разреженных сред, так и для сред с большой плотностью препятствий ρ показатели эффективности близки. Для метода потенциальных полей более раннее ухудшение показателей с ростом значения ρ связано с тем, что агент может попасть в локальный экстремум. Для волнового алгоритма уменьшение значения $E(\rho)$ означает появление теоретически непроходимых зон.

Очевидно, что введение дополнительных уровней управления все равно не может гарантировать успешного решения задачи локальной навигации методом ПП. Гарантировать корректное движение с обходом препятствий можно лишь в области, размер которой определяется величиной радиуса области видимости r_Ω .

Фактически это означает, что данные эксперименты можно интерпретировать как определение зависимости эффективности навигации от радиуса r_Ω и среднего расстояния между препятствиями (показателя насыщенности).

Действительно, пусть S – это общая площадь поля. Тогда каждое препятствие будет центром области с площадью $S_0 = S / N_{\text{obst}}$ и радиусом

$$r_0 = \sqrt{S_0}. \text{ Поскольку } S = N, \text{ то } S_0 = \frac{N}{N_{\text{obst}}} = \frac{N}{\rho N} = \frac{1}{\rho}, \text{ откуда можно оценить среднее расстояние}$$

между препятствиями r_0 :

$$r_0 = \sqrt{S_0} = \frac{1}{\sqrt{\rho}}.$$

В экспериментах значение r_Ω было постоянным и равным 10 % от линейного размера всей области, т. е. $r_\Omega = 0,1\sqrt{N}$.

В итоге получен следующий результат: если размер радиуса области видимости агента r_Ω был больше, чем минимальное расстояние между препятствиями r_0 , то агент успешно проходил маршрут.

О сравнении с другими алгоритмами и методами планирования. Формально речь шла об экспериментах на картах класса $\text{random}(m \times n, R)$, что означает прямоугольное поле из m строк и n столбцов, при этом R процентов случайно выбранных клеток из них являются заблокированными (обозначают препятствия). Среди множества метрик, описывающих как свойства среды, так и эффективность решения задачи, были выбраны две – ρ (плотность препятствий, англ. *Obstacle Density*) и E (доля успешно завершенных миссий, англ. *Success Rate*). Чаще всего в исследованиях оценка свойств рассматриваемых алгоритмов и методов проводится на примерах фиксированных карт препятствий. При этом номенклатура метрик может быть весьма разнообразной (см. работу [14]). Так, для оценки сложности статических сред широко используется метрика Tr (проходимость, англ. *Traversability*). Она определяется как среднее проходимое расстояние для всех равномерно выбранных положений и направлений:

$$Tr = \frac{1}{N_s} \sum_{i=1}^{N_s} d_i,$$

где d_i – проходимое расстояние в i -й выбранной позиции и направлении; N_s – общее количество выбранных позиций и направлений. Однако очевидно, что в силу вычислительных затрат использование данной метрики затруднительно для экспериментов с генерацией множества карт класса random .

Пожалуй, наиболее близка к $E(\rho)$ оценка, данная в работе [15], где приводится зависимость отношения E от параметра, характеризующего плотность препятствий, однако приведенное там срав-

нение алгоритмов имеет лишь качественный характер, показывающий монотонное убывание значения $E(p)$. Именно поэтому в настоящей работе использовалось сравнение результатов метода потенциальных полей с эталонным – волновым алгоритмом.

2.2. Эксперименты в Gazebo и на реальных роботах

Эти эксперименты носили прежде всего иллюстративный характер, демонстрируя работоспособность предложенных моделей в реальных условиях. Кроме того, в этих экспериментах управление $F_{ctl} = \langle v_{res}, A_{res} \rangle$ более естественным образом переводилось в команды движения, задаваемые парой $\langle v_{lin}, v_{ang} \rangle$, где v_{lin}, v_{ang} – линейная и угловая скорости движения робота.

Поскольку в управлении A_{res} определяет итоговое направление движения, то значение v_{ang} определялось как поворот, устраняющий рассогласование между требуемой и текущей компасной ориентацией робота.

2.2.1. Физическое моделирование в Gazebo

Одним из распространенных средств для реализации физического моделирования в робототехнике является система Gazebo [16]. Симулятор Gazebo позволяет добавлять реалистичные датчики и использовать объемные модели агентов, облегчая переход от моделей управления к реальным робототехническим устройствам.

На рис. 6 представлены изображения модели робота и одного из тестовых полей.

2.2.2. Эксперименты с реальными роботами

Эксперименты проводились как в лабораторных условиях, так и на открытом пространстве на соответствующих роботах. Имея единую архитектуру, роботы отличались габаритами и типами используемых дальномеров на локаторах: на лабораторном роботе использовались лазерные дальнометры VL53L0X (угол 15° , диапазон – до 2 м с точностью 3 %), а на «уличном» – дальнометры на основе УЗД HC-SR04 (угол 15° , диапазон – до 4 м).

Локаторы роботов представляют собой спаренные дальнометры, расположенные на сервоприводе. Это обеспечивает угол обзора в 180° при повороте сервопривода на 90° (шесть шагов при эффективном угле 15°). Компас построен на основе IMU-сенсора MPU6050 (акселерометр и гироскоп со стандартным отклонением $0,3^\circ$). Дрейф показаний компаса игнорировался.

На рис. 7 изображены «уличный» (предназначенный для работы на открытых пространствах) и лабораторный роботы.

Для удобства отладки и управления вычисления осуществлялись на удаленном компьютере, а задача бортового вычислителя на основе Raspberry Pi 4 сводилась к передаче сенсорных данных и отработке команд управления скоростью. Задача роботов заключалась в движении в направлении, определенном их начальной ориентацией. Аппаратные рефлексy – реакция на препятствия в непосредственной близости – были отключены. В целом результаты огибания препятствий, размеры которых соизмеримы с габаритами робота, были удовлетворительными в смысле отсутствия столкновений. Это связано с широким углом охвата дальномеров робота.

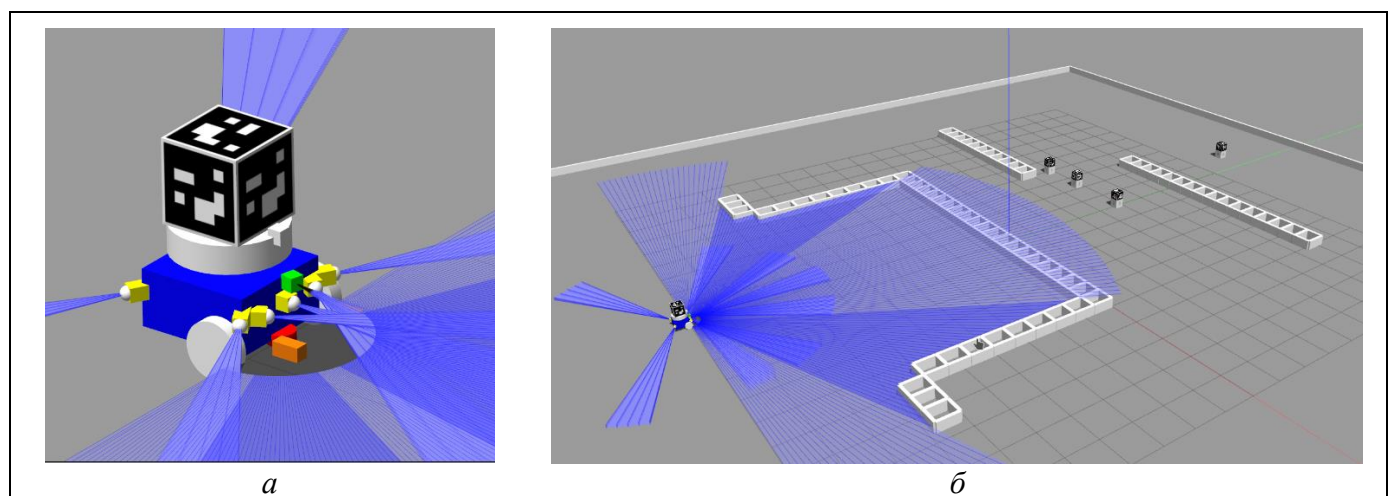


Рис. 6. Имитационное моделирование в среде Gazebo: а – модель робота; б – робот на полигоне

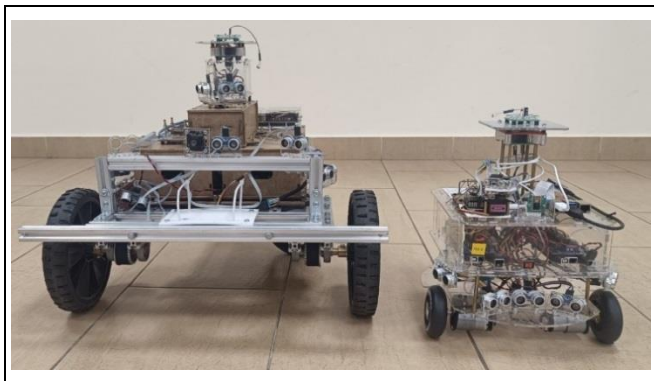


Рис. 7. Робот для работы на открытых пространствах (слева) и лабораторный робот (справа)

О погрешностях сенсорной системы. Как в вычислительных экспериментах, так и при работе с реальными роботами возникает вопрос влияния погрешностей измерений на работоспособность системы. Погрешности относятся к трем категориям: пропуск препятствий, точность и инерционность. Несмотря на дискретность работы описанного выше локатора, использование дальномеров с широким углом в 15° позволяет нивелировать пропуск – локатор работает с заведомым перекрытием. Точности измерения в 3 % более чем достаточно для безопасного функционирования системы: если минимально безопасное расстояние до препятствий составляет один линейный размер робота (0,5 м), то при скорости 1 м/с и времени обновления всех показаний датчиков даже в 0,5 с допустимая погрешность может составлять до 20 %. То же самое касается и инерционности сенсорной системы (поскольку робот движется, локатор выдает искаженную картину препятствий). Вопрос в том, сколько робот успеет пройти за время обновления показателей дальномеров, разрешается так же, как и вопрос точности. Таким образом, при указанных параметрах инерционность тоже несущественна.

ЗАКЛЮЧЕНИЕ

Навигация в экстремальных условиях, когда робот может ориентироваться исключительно на показания достаточно простых датчиков и у него нет средств глобальной навигации и априорных сведений об окружении, требует специфических методов управления. Специфика определяется, во-первых, ограниченностью вычислительных ресурсов, во-вторых, гибкостью и универсальностью моделей управления и, в-третьих, нетребовательностью к качеству (точности) решений. В этом плане метод потенциальных полей выгодно выделяется в широком ряду механизмов навигации.

В силу принципиальных ограничений метода ПП продолжающийся интерес к нему определяется разработкой различного рода модификаций, построением новых гибридных архитектур, применением методов машинного обучения и т. п. В настоящей работе был предложен именно архитектурный подход к локальной навигации, когда управление на основе метода ПП переносится на уровень элементарного движения, а анализ ситуации и переключение между двигательными программами переносится на уровень метаавтоматного управления.

Разумеется, успешность выполнения задачи локальной навигации определяется размером области видимости робота, однако отметим еще раз, что унифицированность механизмов управления позволяет расширять эту область совершенно естественным образом.

ЛИТЕРАТУРА

1. Katona, K., Neamah, H.A., Korondi, P. Obstacle Avoidance and Path Planning Methods for Autonomous Navigation of Mobile Robot // *Sensors*. – 2024. – Vol. 24, no. 11. – Art. no. 3573.
2. Филимонов А.Б., Филимонов Н.Б. Вопросы управления движением мобильных роботов методом потенциального наведения // *Мехатроника, автоматизация, управление*. – 2019. – Т. 20, № 11. – С. 677–685. [Filimonov, A.B., Filimonov, N.B. Issues of Motion Control of Mobile Robots Based on the Potential Guidance Method // *Mekhatronika, Avtomatizatsiya, Upravlenie*. – 2019. – Vol. 20, no. 11. – P. 677–685. (In Russian)]
3. Филимонов А.Б., Филимонов Н.Б. Новый подход к использованию метода потенциальных полей в мобильной робототехнике // *IV Международная научная конференция по проблемам управления в технических системах*. – Санкт-Петербург, 2021. – С. 314–317. [Filimonov, A.B., Filimonov, N.B. Novyi podkhod k ispol'zovaniyu metoda potentsial'nykh polei v mobil'noi robototekhnike // *Proceedings of the 4th International Conference on Control in Technical Systems (CTS'2021)*. – Saint-Petersburg, 2021. – P. 314–317. (In Russian)]
4. Xing, H. Review of unmanned aerial vehicle obstacle avoidance planning based on artificial potential field // *Appl. Comput. Eng.* – 2023. – Vol. 10, no. 1. – P. 55–63.
5. Koren, Y., Borenstein, J. Potential field methods and their inherent limitations for mobile robot navigation // *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*. – Sacramento, CA, USA, 1991. – Vol. 2. – P. 1398–1404.
6. Xu, Z., Hess, R., Schilling, K. Constraints of potential field for obstacle avoidance on car-like mobile robots // *IFAC Proceedings Volumes*. – 2012. – Vol. 45, no. 4. – P. 169–175.
7. Сидоренко А.В., Солодухо Н.А. Алгоритм обигания препятствий при перемещении мобильного робота // *Системный анализ и прикладная информатика*. – 2024. – Т. 4. – С. 29–33. [Sidorenko, A.V., Saladukha, N.A. Obstacle avoidance algorithm in mobile robot motion // *System analysis and applied information science*. – 2024. – Vol. 4. – P. 29–33. (In Russian)]

8. *Алдошкин Д.Н., Царев Р.Ю.* Поиск пути мобильного робота в условиях наличия препятствий и неполноты информации о среде // Мехатроника, автоматизация, управление. – 2016. – Т. 17, № 7. – С. 465–470. [Aldoshkin, D.N., Tsarev, R.Yu. Mobile Robot Path Planning in the Presence of Obstacles and Lack of Information about the Environment // Mekhatronika, Avtomatizatsiya, Upravlenie. – 2016. – Vol., 17, no. 7. – P. 465–470. (In Russian)]
9. *O'Rourke, J.* Computational Geometry in C. – Cambridge: Cambridge University Press, 1998.
10. *Карпов В.Э., Ровбо М.А., Воробьев В.В.* О некоторых аспектах применения автоматных моделей в групповом управлении // Мехатроника. Автоматизация. Управление. – 2023. – Т. 24, № 4. – С. 171–180. [Karpov, V.E., Vorobiev, V.V., Rovbo, M.A. About Some Aspects of Finite State Machine Models Application to Group Control // Mekhatronika, Avtomatizatsiya, Upravlenie. – 2023. – Vol. 24, no. 4. – P. 171–180. (In Russian)]
11. *Reynolds, C.W.* Flocks, herds, and schools: A distributed behavioral model // Proc. 14th Annu. Conf. Comput. Graph. Interact. Tech. (SIGGRAPH'1987). – Anaheim, 1987. – Vol. 21, no. 4. – P. 25–34.
12. *Поликарпова Н.И., Шалыто А.А.* Автоматное программирование. – СПб, 2008. – 167 с. [Polikarpova, N.I., Shalyto, A.A. Avtomatnoe programmirovaniye. – SPb, 2008. – 167 s (In Russian)]
13. *Карпов В.Э., Ровбо М.А., Овсянникова Е.Е.* Система моделирования поведения групп робототехнических агентов с элементами социальной организации Кворум // Программные продукты и системы. – 2018. – Т. 31, № 3. – С. 581–590. [Karpov, V.E., Rovbo, M.A., and Ovsyannikova, E.E. Kvorum – the system for modeling the behavior of robotic agent groups with elements of social organization // Software & Systems. – 2018. – Vol. 31, no. 3. – P. 581–590. (In Russian)]
14. *Shi, M., Chen, G., Serra Gómez, A., et al.* Evaluating Dynamic Environment Difficulty for Obstacle Avoidance Benchmarking // arXiv:2404.14848. – 2024. – DOI: <https://doi.org/10.48550/arXiv.2404.14848>
15. *Diab, M., Mohammadkarimi, M., Rajan, R.T.* Artificial Potential Field-Based Path Planning for Cluttered Environments // Proceedings of 2023 IEEE Aerospace Conference. – Big Sky, MT, USA, 2023. – P. 1–8. – DOI: 10.1109/AERO55745.2023.10115857
16. *Gazebo.* – URL: <https://gazebo.org/> (дата обращения: 03.09.2025). [Accessed September 3, 2025].
- Статья представлена к публикации членом редколлегии Е. Я. Рубиновичем.*
- Поступила в редакцию 04.09.2025,
после доработки 27.01.2026.
Принята к публикации 16.03.2026.*
- Карпов Валерий Эдуардович** – д-р техн. наук, НИЦ «Курчатовский институт», г. Москва
ORCID iD: <https://orcid.org/0000-0002-9364-1223>
- © 2026 г. Карпов В. Э.



Эта статья доступна по лицензии Creative Commons «Attribution» («Атрибуция») 4.0 Всемирная.



FINITE-STATE MACHINE CONTROL AND POTENTIAL FIELD METHOD FOR MOBILE ROBOT NAVIGATION IN AN ENVIRONMENT WITH OBSTACLES

V. E. Karpov

National Research Center Kurchatov Institute, Moscow, Russia

✉ karpov-ve@yandex.ru

Abstract. This paper considers a local navigation method for a mobile robot operating in extreme conditions, i.e., the absence of global navigation and centralized control means, as well as limited-capability sensors. Local navigation is based on the potential field method, and the motion programs developed with this method are transferred to the level of elementary actions. Complex behavior is formed by meta finite-state machines (FSMs) implementing a control hierarchy. The robot's basic task is to move to a given area or region of space. Within this task, an elementary procedure—obstacle avoidance—is considered and implemented based on the potential field method as well. The results of experiments are provided for three different levels, namely, computations, simulations (the physical level), and real robots, with the same computational model and FSMs used in all experiments. As shown, the successful solution of the local navigation task is determined by the size of the robot's field of view, but the latter can be naturally expanded due to the universality of control mechanisms.

Keywords: potential field method, finite-state machine control, meta finite-state machine, movement models, mobile robot, local navigation, obstacle avoidance, control system hierarchy.

Acknowledgments. This work was carried out within the state assignment of NRC Kurchatov Institute.