

НАВИГАЦИЯ МОБИЛЬНОГО РОБОТА С ИСПОЛЬЗОВАНИЕМ ЭЛЛИПТИЧЕСКИХ ТРАЕКТОРИЙ И ЭФФЕКТИВНОГО АЛГОРИТМА ОБНАРУЖЕНИЯ ПРЕПЯТСТВИЙ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ

Статья посвящена проблеме навигации мобильного робота в загроможденной среде. Для обхода препятствий при движении по плоскости реализуются адаптивно настраиваемые в режиме реального времени эллиптические траектории, при построении которых используется информация только от нескольких неидеальных измерителей дальности до препятствия. Стратегия безопасного обхода препятствий основана на принципе эллиптического предельного цикла, когда каждое препятствие окружено эллипсом, запретным для движения. Параметры этого эллипса вычисляются в режиме реального времени с использованием последовательности полученных от датчиков данных по дальности до препятствия. Для вычисления параметров эллипса используется эвристический метод в сочетании с расширенным фильтром Калмана. Показано, что этот процесс гарантирует влияние всего набора данных по дальности на вычисленный эллипс. Кроме того, предлагается единый закон управления в мультиконтроллерной системе, где внедрен указанный адаптивный алгоритм обхода препятствий. В основе предлагаемого закона управления лежит закон управления Канаямы. Стабильность этой архитектуры управления доказана на основе метода Ляпунова. Моделирование и эксперименты, проведенные в различных условиях, демонстрируют эффективность и надежность предлагаемого алгоритма навигации в загроможденной среде в режиме реального времени.

1. Введение

Важным вопросом для успешной навигации мобильного робота является обход препятствий. Авторы некоторых работ считают, что управление роботом полностью основано на методах планирования траектории и подразумевает полное априорное знание его окружения. Среди этих методов – диаграммы Вороного и графики видимости [1] или функции искусственных потенциальных полей [2]. Таким образом, на этапе планирования учитываются препятствия всех конфигураций. Эти методы позволяют иметь дело и с изменчивой окружающей обстановкой, регулярно осуществляя перепланировку траектории движения робота [3, 4]. Однако планирование и перепланирование представляют значительную сложность и требуют много времени на вычисление.

Другие авторы сосредоточены на использовании адаптивных методов обхода препятствий, в которых используется информация только от бортовых дат-

Вилка Жозе Мигель. Аспирант Института Паскаля (г. Клермон-Ферран, Франция).

Адуан Луни. Доцент (associate professor) Института Паскаля.

Мезуар Юсеф. Доцент (assistant professor) Института Паскаля.

Научный редактор перевода д.т.н. А.В.Небылов.

чиков, а не предварительные знания об окружающей среде [5–7]. В [8] предлагается подход, основанный на принципе искусственных потенциальных полей, в котором обход препятствий совершается в режиме реального времени. В данной работе предполагается, что в своих действиях робот руководствуется суммой полей сил притяжения и отталкивания. В [9] автор расширяет подход Хатиба, предлагая для навигации мобильных роботов использовать набор простейших базовых форм движения (примитивов).

Другой интересный подход, основанный на рефлекторной реакции, использует концепцию деформируемой виртуальной зоны (ДВЗ), в которой движение робота зависит от зоны риска, окружающей робота [10]. Если новое препятствие обнаружено, оно деформирует ДВЗ. Подход состоит в минимизации этой деформации за счет изменения вектора управления. Этот метод годится для препятствий любой формы, однако его недостаток, как и в случае с использованием примитивов движения, связан с проблемой локального минимума. В общем реактивные методы не требуют высокой вычислительной сложности, поскольку действия робота должны производиться в режиме реального времени в соответствии с темпом поступления информации о препятствиях [11].

В литературе можно найти много других подходов, например обход препятствия с использованием вихревых полей [12] и орбитальных траекторий [13]. Последний подход, представленный в [13–15], построен на дифференциальных уравнениях круговых предельных циклов. Круговые предельные циклы более стабильны, чем вихревые поля, и всегда стремятся к периодической орбите. В данной статье используются эллиптические траектории [16]. Кроме того, предлагается использовать только один закон управления для мультиконтроллерной структуры [17]. Указанный закон управления основан на [18] и адаптирован к этому алгоритму обхода препятствий. Таким образом, обеспечивается более специфический и эффективный обход препятствий, даже если они имеют разные формы, например форму длинной стены. Эллипс лучше подходит для описания такого рода препятствий, чем окружность.

В литературе предложены различные подходы для построения эллипса по данным о препятствии. В [19] автор предлагает методику получения наименьшего огибающего эллипса с помощью набора данных, использующих примитивную операцию при которой линейно растет время обработки с ростом массива данных. В [20] дан краткий обзор методов для работы с набором данных и эллипсом. Описанная методика представляет собой сглаживание методом наименьших квадратов на основе использования алгебры и евклидова расстояния, фильтра Калмана и робастного оценивания. Метод, используемый в данной работе, основан на простом и эффективном эвристическом подходе, базирующемся на оценивании евклидова расстояния [21].

Расширенный фильтр Калмана (РФК) имеет важные применения в различных областях [22–25]. В области управления роботами он используется при позиционировании, интеграции датчиков, оценке положения препятствий. В работе [26] автор представляет метод подбора эллипсов с использованием фильтра Калмана с поправкой на смещение. При этом длина перпендикуляра до эллипса используется для того, чтобы найти оптимальные варианты параметров эллипса. В этой работе РФК применяется для уточнения параметров эллипса, полученных из искаженных погрешностями дальномеров данных о препятствиях любой формы [27].

Далее статья организована следующим образом. В разделе 2 представлена

задача навигации с использованием эллиптических траекторий. В разделе 3 обсуждаются детали структуры системы управления. Приведена модель рассматриваемого робота и реализованные элементарные контроллеры. В разделе 4 подробно излагается алгоритм обхода препятствий. Раздел 5 представляет метод построения ограничивающего эллипса при искаженных погрешностями дальномеров данных по дальности до препятствия. Результаты моделирования рассмотрены в разделе 6.

2. Навигация в загроможденной окружающей среде

Прежде чем перейти к подробному изложению проблем, касающихся навигации робота, допустим, что первое препятствие O в окружающей среде может быть окружено эллипсом (рис. 1). Эллиптическая форма представлена в декартовой системе координат:

$$\frac{(x - h)^2}{a^2} + \frac{(y - k)^2}{b^2} + c(x - h)(y - k) = 1, \quad (1)$$

где $(h, k) \in \mathbb{R}^2$ – координаты центра эллипса, а $(a, b) \in \mathbb{R}^+$ – полуоси ($a \geq b$). $c \in \mathbb{R}$ позволяет дать ориентацию эллипса $\Omega_E = 0.5 \arctan(c/(b^2 - a^2))$.

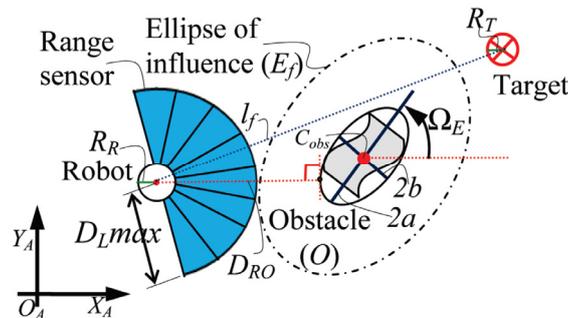


Рис. 1. Препятствие и робот
Надписи слева направо: Робот. Датчик расстояния. Эллипс влияния. Цель. Препятствие

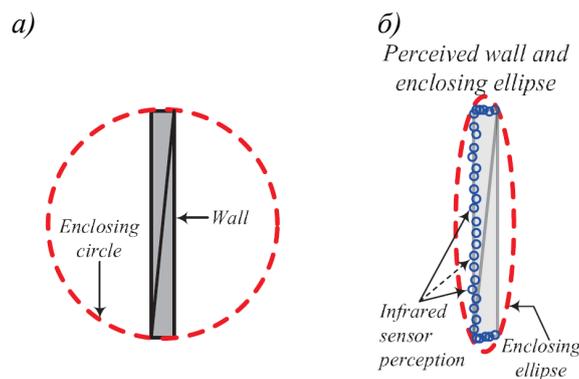


Рис. 2. Стена, интерполированная с использованием окружности и эллипса.
Надписи слева направо: а) Ограждающая окружность. Стена; б) Восприятие ИК-приемника. Воспринимаемая стена и ограждающий эллипс. Ограждающий эллипс

Именно эллипсы, а не окружности используются как типичное и гибкое

средство для окружения и точной подгонки разных видов форм препятствий [16]. Среди примеров форм, которые могут быть правильно подобраны с использованием эллипса, а не окружности, стена (в общем продольные формы). Подобная конфигурация показана на рис. 2. В самом деле, если бы потребовалось описать эту стену окружностью, она бы имела большой радиус, в результате чего путь робота для обхода препятствия соответственно увеличился [13] (см. рис. 2, а). На рис. 2, б видно, что эллипс лучше вписывается в размеры препятствия. На этом же рисунке видна зона восприятия датчика расстояния с левой стороны стены.

В разделе 5 будет показано, как можно эффективно вычислить параметры эллипса по данным о дальности. Пусть робот и цель представлены окружностями C_R и C_T с радиусами R_R и R_T соответственно (см. рис. 1). Тогда можно определить:

- (1) D_{RO} как минимальное расстояние между роботом и препятствием O ;
- (2) эллипс влияния (E_f) как эллипс, который имеет такой же центр (h, k) и угол наклона Ω_E , что и эллипс, описывающий препятствие (1); его большая и малая полуоси a_{lc} и b_{lc} определяются следующим образом:

$$\begin{cases} a_{lc} = a + R_R + Margin \\ b_{lc} = b + R_R + Margin, \end{cases}$$

где *Margin* представляет собой допуск на безопасность, учитывающий погрешность измерения, точность и надежность управления. Это определение аналогично конфигурации пространства для глобального планирования пути. Тем не менее в данной статье оно используется для того, чтобы сосредоточиться на адаптивном подходе к планированию маршрута робота, т.е. робот получает эллипс влияния в реальном времени по данным об обнаруженном препятствии.

(3) l как линию, проходящую через центр C_R и C_T . Как будет показано ниже, для реализации предлагаемого метода нужно только знать, существуют ли точки пересечения между l и E_f (см. рис. 1).

Цель решения задачи навигации в загроможденной окружающей среде – вести мобильного робота к конкретной цели мимо препятствий. Эта задача должна быть решена путем обхода статических и динамических препятствий O , которые могут иметь различные формы.

3. Структура системы управления

Структура управления основана на [17] (рис. 3).

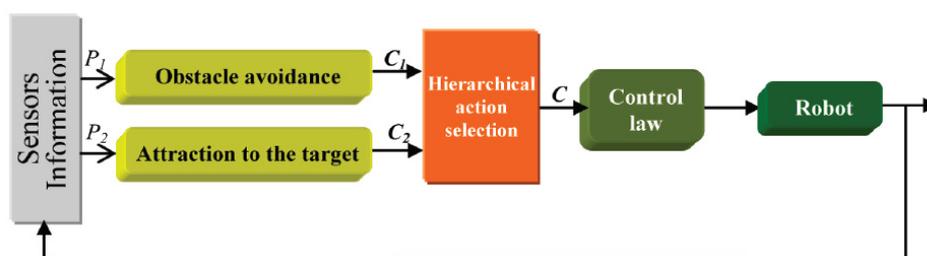


Рис. 3. Система управления движением мобильного робота.

Надписи слева направо: Информация датчиков. Обход препятствий. Притяжение к цели. Иерархический выбор действия. Закон управления. Робот

Она нацелена на управление взаимодействиями между элементарными контроллерами, гарантируя устойчивость управления, предложенного в [7], а также на обеспечении безопасной, быстрой и беспрепятственной навигации робота. Блоки, входящие в систему управления, рассматриваются подробно ниже.

В этой структуре управления используется процедура выбора иерархического действия по управлению переключателями между контроллерами в соответствии с восприятием окружающей среды. Механизм активирует контроллер обхода препятствий, как только возникает по крайней мере одно препятствие, которое может помешать будущим движениям робота к его цели (см. алгоритм 1) [15]. Это позволяет ожидать активизации контроллера, отвечающего за обход препятствий, в отличие от того, что предлагается в [28, 29], когда контроллер ждет, пока робот находится в непосредственной близости от препятствия ($D_{RO} \leq R$, где R – значение определенного радиуса).

Таким образом, алгоритм 1 позволяет уменьшить время, необходимое для достижения цели, особенно в очень загроможденной окружающей среде.

если существует по крайней мере одно препятствие мешающее движению в пространстве {то есть существует хотя бы одна точка пересечения между линией “ I_j ” и эллипсом влияния (см. рис. 1)}, **то**
 Активируй контроллер *Обхода препятствий*;
иначе
 Активируй контроллер *Притяжения к цели*;
конец

Алгоритм 1. Выбор иерархического действия.

3.1. Элементарные контроллеры

Каждый контроллер в структуре управления (см. рис. 3) характеризуется устойчивым номинальным законом. В этой работе только один закон управления, синтезированный согласно теореме Ляпунова, используется для улучшения работы двух различных контроллеров (см. п. 3.2). Прежде чем перейти к описанию элементарного контроллера, кратко приведем кинематическую модель одноколесного робота (моноцикла) (см. рис. 4)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (2)$$

где x, y, θ – пространственное положение одноколесного робота в точке O_m ; v и ω соответственно линейная и угловая скорости робота в точке O_m .

Контроллер притяжения к цели. Этот контроллер направляет робота к цели, которая представлена окружностью C_T с центром (x_T, y_T) и радиусом R_T . Он основан на пространственном положении робота относительно цели, представленной e_x, e_y и e_θ на рис. 4.

Поскольку мы рассматриваем круговую цель с радиусом R_T , чтобы гарантировать, что центр оси робота достигает цели с асимптотической сходимостью, d должно быть меньше R_T , где $d = \sqrt{e_x^2 + e_y^2}$ (см. рис. 4).

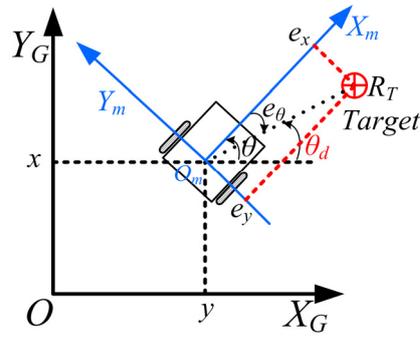


Рис. 4. Конфигурация робота и цели в декартовой системе координат

Ошибки положения и ориентации относительно местной системы координат $OX_m Y_m$ следующие:

$$\begin{cases} e_x = \cos(\theta)(x_T - x) + \sin(\theta)(y_T - y) \\ e_y = -\sin(\theta)(x_T - x) + \cos(\theta)(y_T - y) \\ e_\theta = \theta_d - \theta \end{cases}, \quad (3)$$

где θ_d – ориентация линии, проходящей через робот и цель. Далее считаем, что $\dot{\theta}_d = \omega_r$, имеет вид

$$\omega_r = v \sin(e_\theta) / d, \quad (4)$$

где v – линейная скорость робота [17].

Контроллер обхода препятствий. Чтобы обеспечить обход препятствий, робот должен точно следовать траекториям предельного цикла, как подробно описано в [13, 15].

В этих работах авторы используют круговой предельный цикл, характеризующийся окружностью влияния с радиусом R_l . В работе [16] предлагается распространить эту методику на более гибкую форму предельного цикла (эллипс). Основные идеи этого контроллера приведены далее.

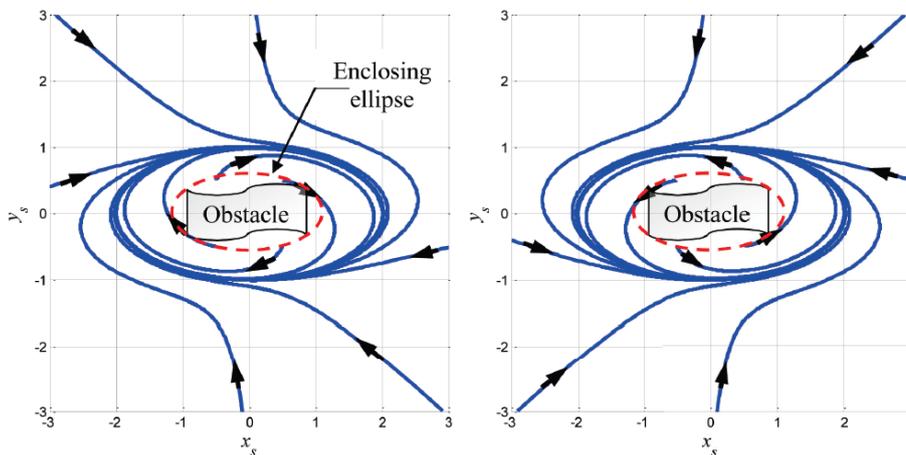


Рис. 5. Формы для используемых эллиптических предельных циклов по часовой ($m = 1$) и против часовой стрелки ($m = -1$).
Надписи: Препятствие. Ограждающий эллипс

Дифференциальные уравнения, дающие эллиптические предельные циклы, следующие:

$$\dot{x}_s = my_s + x_s(1 - x_s^2/a_{lc}^2 - y_s^2/b_{lc}^2 - cx_sy_s), \quad (5)$$

$$\dot{y}_s = -mx_s + y_s(1 - x_s^2/a_{lc}^2 - y_s^2/b_{lc}^2 - cx_sy_s), \quad (6)$$

при этом $m = \pm 1$ соответствует направлению обхода (по часовой или против часовой стрелки (рис. 5); (x_s, y_s) соответствуют положению робота относительно центра эллипса; a_{lc} и b_{lc} характеризуют большую и малую полуоси эллипса соответственно (см. рис. 1); c , если не равно нулю, дает угол эллипса Ω_E .

В этом контроллере заданное положение рассматривается как положение робота $e_x = 0$ и $e_y = 0$ в (3) (рис. 6). Заданная ориентация робота описывается дифференциальным уравнением предельного цикла (5) и (6) (рис. 6):

$$\theta_d = \arctg\left(\frac{\dot{y}_s}{\dot{x}_s}\right). \quad (7)$$

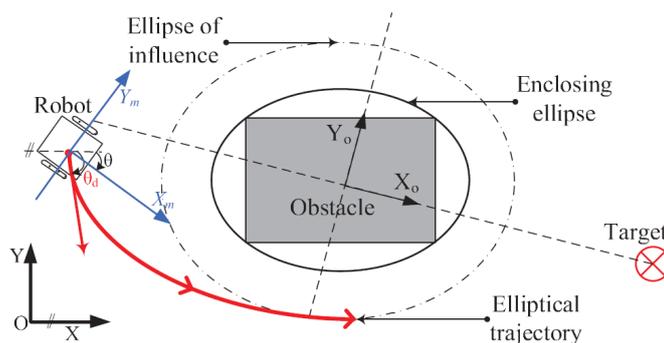


Рис. 6. Контроллер обхода препятствий.

Надписи по часовой стрелке: Робот. Эллипс влияния. Препятствие. Ограждающий эллипс. Цель. Эллиптическая траектория

3.2. Закон управления

Предлагаемый закон управления основан на [18] и адаптирован к мультиконтроллерной структуре. В работе [30] закон управления Канаямы используется для отслеживания заданной траектории. В данной работе предложен закон управления адаптированный для отслеживания заданной траектории (контроллер обхода препятствий) и достижения цели (контроллер притяжения к цели). Кроме того, этот закон позволяет улучшить производительность контроллеров (см. раздел 6).

Рассмотрим случай, когда погрешность между заданным (x_d, y_d, θ_d) и фактическим положениями робота (x, y, θ) в системе координат робота (e_x, e_y, e_θ) имеет следующий вид (см. рис. 4 и 6):

$$\begin{cases} e_x = \cos(\theta)(x_d - x) + \sin(\theta)(y_d - y) \\ e_y = -\sin(\theta)(x_d - x) + \cos(\theta)(y_d - y) \\ e_\theta = \theta_d - \theta. \end{cases} \quad (8)$$

Производные этих погрешностей можно вычислить, используя (2) и (8) [18]:

$$\begin{cases} \dot{e}_x = -v + e_y \omega + v_r \cos(e_\theta) \\ \dot{e}_y = -e_x \omega + v_r \sin(e_\theta) \\ \dot{e}_\theta = -\omega + \omega_r. \end{cases} \quad (9)$$

Следовательно, заданные линейная (v) и угловая скорости (ω) робота, которые заставляют ошибки стремиться к нулю, имеют следующий вид:

$$v = v \cos(e_\theta) + K_x e_x, \quad (10)$$

$$\omega = \omega_r + K_y v_r e_y + K_\theta e^{(e_y/R_R)^2} \sin(e_\theta), \quad (11)$$

где R_R – радиус робота. Следует модифицировать закон управления Канаямы, добавив член уравнения $e^{(e_y/R_R)^2}$ в угловую скорость (11). Цель введения этого члена – улучшить сходимость ошибки для случая, когда $v_r = 0$ (контроллер *притяжения к цели*) (см. раздел 6). Форма члена $e^{(e_y/R_R)^2}$ выбрана таким образом, чтобы улучшить сходимость, когда $e_y \geq R_R$ и когда $e_y \approx 0$, тогда угловая скорость зависит только от e_θ .

Рассмотрим следующую функцию Ляпунова V_0 [18]:

$$V_0 = \frac{1}{2}(e_x^2 + e_y^2) + \frac{1 - \cos(e_\theta)}{K_y}. \quad (12)$$

Следовательно, чтобы гарантировать устойчивость контроллера, \dot{V}_0 должно быть отрицательно определённым. Вычисляя \dot{V}_0 с использованием (9), (10) и (11), получаем:

$$\begin{aligned} \dot{V}_0 &= e_x \dot{e}_x + e_y \dot{e}_y + \frac{\sin(e_\theta) \dot{e}_\theta}{K_y} = \\ &= e_x (-K_x e_x + e_y \omega) + e_y (-e_x \omega + v_r \sin(e_\theta)) + \\ &+ \frac{\sin(e_\theta)}{K_y} (-K_y v_r e_y - K_\theta e^{(e_y/R_R)^2} \sin(e_\theta)) = \\ &= -K_x e_x^2 - \frac{K_\theta e^{(e_y/R_R)^2} \sin(e_\theta)}{K_y} \leq 0, \end{aligned} \quad (13)$$

где K_x , K_y and K_θ – положительные постоянные, определяемые конструктором. Причем $v_r = 0$, когда контроллер *притяжения к цели* активирован, а если активирован *обход препятствий*, то v_r и ω_r вычисляются как

$$v_r = \sqrt{(\dot{x}_d)^2 + (\dot{y}_d)^2} \quad \text{и} \quad \omega_r = \dot{\theta}_d. \quad (14)$$

Интересно отметить, что к роботу применяется только один закон управления, даже если архитектура управления содержит два разных контроллера. В соответствии с используемым контроллером изменяются только задаваемые точки, как в [17].

4. Адаптивный алгоритм обхода препятствий

Далее будет представлена общая методология получения предлагаемого алгоритма обхода [16].

Алгоритм разрабатывается на основе принципа зависимости реакции от воздействия. Для реализации такого поведения, важно:

- обнаружить препятствие, которое нужно обойти (см. раздел 2),
- дать направление обхода препятствия (по часовой стрелке или против часовой стрелки),
- определить критерий обхода, который определяет, можно ли полностью обойти препятствие или пока нет.

Нужно последовательно выполнить все эти разные шаги, гарантируя при этом, что траектория робота является безопасной, гладкой и предотвращает нежелательные ситуации, такие как тупики или локальные минимумы, и что гарантируется устойчивость примененного закона управления (см. п. 3.1). В данной работе наблюдение за препятствием ведется в реальном времени. Робот не имеет глобальной информации ни о препятствии, ни об эллипсе, который его окружает. Таким образом, в каждый момент движения робот самостоятельно обнаруживает форму препятствия и постепенно увеличивает знания об ограждающем эллипсе для получения гладкой эллиптической траектории. Глобальная информация о препятствии не связана с используемой адаптивной навигацией. Ниже приводятся шаги, необходимые для выполнения алгоритма обхода препятствий (2):

(1) Для каждого момента времени получить расстояние D_{ROi} для любого потенциально опасного препятствия I (см. рис. 1).

(2) Среди набора опасных препятствий (которые могут не позволить роботу достичь цели) выбрать одно, наиболее близкое к роботу (наименьшее D_{ROi}). Это конкретное препятствие имеет следующие характеристики: центральную точку (x_{obst}, y_{obst}) , $2a$ - большая ось и $2b$ - малая ось.

(3) После определения ближайшего, стесняющего движение препятствия нужно получить четыре конкретные области (рис. 7), которые определяют поведение робота – обход препятствия по часовой или против часовой стрелки, фаза отталкивания или притяжения (см. алгоритм 2).

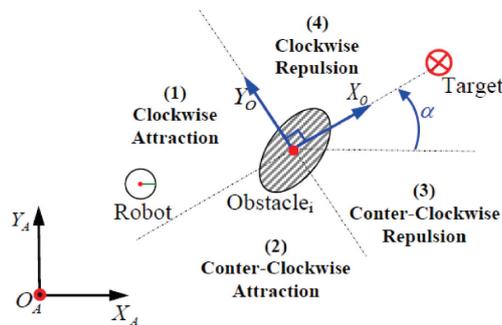


Рис. 7. Четыре области для обхода препятствий [16].

Надписи слева направо: Робот. Притяжение по часовой стрелке. Отталкивание по часовой стрелке. Цель. Отталкивание против часовой стрелки. Притяжение против часовой стрелки. Препятствие

Чтобы разграничить эти четыре области, необходимо:

- определить конкретную систему координат, которая имеет следующие признаки (см. рис. 7):
 - ось X_O соединяет центр препятствия (x_{obst}, y_{obst}) с центром цели. Эта ось ориентирована на цель;
 - ось Y_O перпендикулярна оси X_O , и ее ориентация определяется из законов тригонометрии;
- выполнить изменение в системе координат, чтобы координаты положе-

ния робота $(x, y)_A$, заданные в абсолютной системе координат, преобразовать к системе координат, связанной препятствием $(x, y)_O$. Это достигается за счет использования следующего однородного преобразования:

$$\begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix}_O = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & x_{obst} \\ \sin \alpha & \cos \alpha & 0 & y_{obst} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix}_A. \quad (15)$$

После того как все необходимые данные получены, можно применить стратегию адаптивного обхода препятствий по алгоритму 2.

Вход: Все признаки ближайшего препятствия.

Выход: Признаки траектории предельного цикла, которой надо следовать.

// I) Получение значений a'_{lc} и b'_{lc} предельного цикла, которому надо следовать.

1 если $x_O \leq 0$, то

$$\begin{cases} a'_{lc} = a_{lc} - \xi \\ b'_{lc} = b_{lc} - \xi \end{cases} \quad \text{(Фаза притяжения)}$$

2 {при малом постоянном значении x , удовлетворяющем

3 условию $x \ll \text{Margin}$, гарантирующем, что

робот не подходит близко к препятствию

(см. раздел 2).}

4 или

5 {Критерий уклонения : гладкий выход
из эллипса влияния препятствия}

$$\begin{cases} a'_{lc} = a_{lc} - \xi \\ b'_{lc} = b_{lc} - \xi \end{cases} \quad \text{(Фаза отталкивания)}$$

7 конец

// II) Получение направления предельного цикла

8 если контроллер обхода препятствий был активен в момент $(t - \delta T)$, то

9 Применение уже использованного направления,

т.е. уравнений (5) или (6).

10 {Это позволит избежать колебаний и
некоторых конфликтных ситуаций [15]}

11 или

12 {Задается точка предельного цикла :}

$$\dot{x} = \text{sign}(y_O)y + x(1 - x^2 / a'_{lc}{}^2 - y^2 / b'_{lc}{}^2 - cxy)$$

$$\dot{y} = -\text{sign}(y_O)x + y(1 - x^2 / a'_{lc}{}^2 - y^2 / b'_{lc}{}^2 - cxy)$$

13 Конец

Алгоритм 2: Алгоритм обхода препятствий [16]

Для получения требуемых координат робота необходимо получить значения a_{lc} и b_{lc} (см. раздел 2) орбитального эллипса, а также направление предельного цикла по часовой или против часовой стрелки. Координаты (x_0, y_0) дают конфигурацию (x, y) робота в соответствии с системой координат препятствия. Определение этой конкретной системы отсчета обеспечивает роботу возможность получить знание о том, что он должен делать. Действительно, знак x_0 представляет такой тип поведения, которому должен следовать робот (притяжение или отталкивание).

На фазе отталкивания предельный цикл принимает большие значения для a'_{lc} и b'_{lc} , чтобы гарантировать гладкость траектории. Знак y_0 дает правильное направление для обхода препятствия. Фактически, если $y_0 \geq 0$, то применяется направление обхода по часовой стрелке по принципу предельного цикла или направление против часовой стрелки. Этот выбор позволяет оптимизировать длину траектории робота для обхода препятствий. Тем не менее это направление принудительно склоняется к направлению, принятому до этого, если контроллер обхода препятствий уже был активным в момент $(t - \delta T)$ для обхода локальных минимумов и тупика [15].

В алгоритме 2, для улучшения работы нужно справиться с некоторыми конфликтными ситуациями, которые обусловлены локальными минимумами или тупиками. Эти конкретные локальные правила подробно описаны в [15].

5. Заключение неопределенных данных по дальности в эллипс

Во время движения робота важно обнаружить препятствие в режиме реального времени и обойти его. С этой целью наблюдаемые зашумленные данные по дальности окружаются ближайшим эллипсом, при этом применяется эллиптический подход предельного цикла.

С этой целью рассмотрим набор из n точек в R^2 с координатами $P_i(x_i, y_i)$ (рис. 8). Эти точки вычисляются по данным о дальности до препятствия с датчиков робота, а выбросы удаляются с помощью контроля расстояния Махаланобиса [31]. В этом разделе будет показано, как вычислить эллипс, охватывающий все точки. Для того чтобы метод начал работать, нужно, как минимум, три точки.

В этой статье робот обнаруживает каждое препятствия по отдельности. В будущих работах авторы планируют использовать метод сегментации набора точек для того, чтобы обнаружить более одного препятствия за один раз, а также большие препятствия.

Перед описанием предлагаемого метода для получения ограждающего эллипса в следующем подразделе рассмотрим модель принимаемых от датчика расстояния данных.

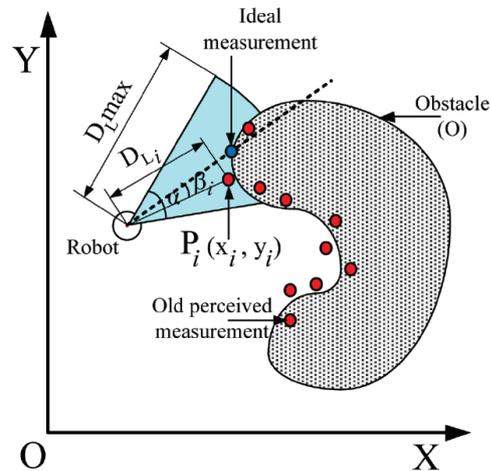


Рис. 8. Модель датчика расстояния и набор данных из n точек.
Надписи слева направо: Робот. Идеальное измерение. Препятствие. Воспринятое старое измерение

5.1. Модель датчика расстояния

Положение препятствия относительно датчика расстояния в R^2 можно обозначить через полярные координаты (D_{Li}, β_i) , где D_{Li} – расстояние между центром робота и ближайшей точкой препятствия, а β_i – ориентация относительно подвижной системы отсчета (см. рис. 8).

Технические условия на датчик и реальное поведение датчика имеют существенные различия [32, 33]. В этой статье внимание акцентировано на точности датчика расстояния, при этом выявлено, что показания на малой дальности более точны, чем при большем радиусе действия. В работе [33] авторы описывают, как среднее и стандартное отклонения ошибок между реальной и измеренной дальностью, как правило, увеличивается с расстоянием.

Данные по дальности, считываемые датчиком расстояния на каждом временном шаге, моделируются нормальным распределением $D_{Li}^i = N(\hat{D}_{Li}^i, P_{Li}^i)$, где $\hat{D}_{Li}^i = [D_{Li}^i, 0]^T$ – вектор математического ожидания, D_{Li} – измерение расстояния датчиком и $P_{Li}^i = \text{diag}([\sigma_{D_{Li}}^2, \sigma_{\beta_i}^2])$ – ковариация, которая определяется как модель погрешностей измерения дальности и угловых ошибок. Последние связаны с погрешностями ультразвуковых датчиков по дальности и по углу β_i (см. рис. 8) [33]. Данные по дальности в декартовой системе координат можно представить следующим образом:

$$z_i = z_i' + v_i, \quad (16)$$

где $z_i = [x_i, y_i]^T$ – точка, вычисленная с использованием зашумленных данных по дальности, z_i' – точка, вычисленная с использованием данных по дальности без шума; гауссов шум v_i имеет $E[v_i] = 0$ и $E[v_i v_i^T] = \mathbf{R}_{v_i}$. Ковариация \mathbf{R}_{v_i} представлена как

$$\mathbf{R}_{v_i} = \begin{bmatrix} \cos(\beta_i) & -D_{Li} \sin(\beta_i) \\ \sin(\beta_i) & D_{Li} \cos(\beta_i) \end{bmatrix} \begin{bmatrix} \sigma_{D_{Li}}^2 & 0 \\ 0 & \sigma_{\beta_i}^2 \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}^T. \quad (17)$$

В следующих разделах будут описаны предлагаемые методы построения эллипсов по данным бортовых ультразвуковых датчиков. В подразделе 5.2 рассматривается проблема построения эллипса, а в подразделе 5.3 предлагаемый метод распространяется на работу с неидеальными датчиками и проблему усовершенствования алгоритма идентификации параметров эллипса для обхода препятствия.

5.2. Эвристический подход

В [21] дается обзор различных методов построения эллипса; предложен эвристический метод, который авторы сравнивают с существующими подходами (на основе наименьших квадратов или ковариации). Этот метод позволяет **охватить** все данные по дальности и получить гладкие изменения параметров эллипса, что отличается от результатов применения метода наименьших квадратов и ковариационного метода, при использовании которых можно получить точку вне полученного эллипса и резкое изменение параметров (см. рис. 9, а). В этой работе эвристический подход дается более подробно с математическим доказательством того, что полученные параметры эллипса **охватывают** все точки вне зависимости от формы препятствия.

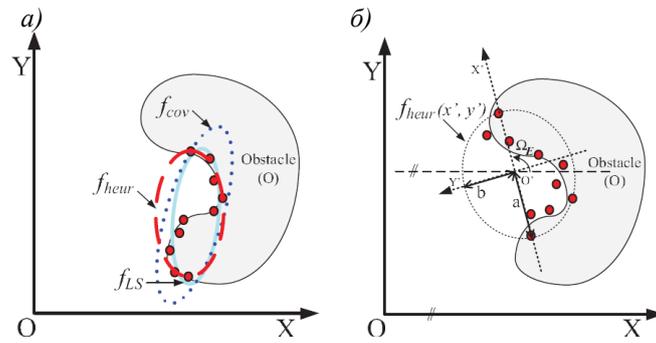


Рис. 9. Эллипсы, полученные с использованием методов наименьших квадратов (f_{LS}), (ФЛС), ковариаций (f_{cov}) и эвристического (f_{heur}) подхода

В данном подходе расстояние между точками используется для получения одной из осей.

Лемма 1. Рассмотрим набор из n точек (\mathbf{P}_N) в R^2 с координатами $\mathbf{P}_i(x_i, y_i)$ при $i = 1, \dots, n$. Параметры эллипса, которые **охватывают** все точки, вычисляются следующим образом:

(1) Вычисление расстояния между всеми точками $d_{ij} = \|p_i - p_j\|$ при $i, j = 1, \dots, n$ и выберем максимальное расстояние d_{max} . Это d_{max} не уменьшается, если еще добавить точки данных.

(2) Центр эллипса C_O – средняя точка между точками с максимальными расстояниями, а первая полуось – $a_1 = d_{max} / 2$ (рис. 9, б).

(3) Преобразуем точки n в новую систему координат $X' - Y'$ с использованием (18) для того, чтобы получить вторую полуось a_2 эллипса

$$\mathbf{P}'_i = \begin{bmatrix} \cos(\Omega) & \sin(\Omega) \\ -\sin(\Omega) & \cos(\Omega) \end{bmatrix} (\mathbf{P}_i - C_O), \quad (18)$$

где Ω – ориентация линии между двумя точками, которые имеют максимальное расстояние. $\mathbf{P}'_i(x'_i, y'_i)$ – координаты в новой системе, $\mathbf{P}_i(x_i, y_i)$ – координаты в исходной системе и C_O – координаты центра эллипса в исходной системе.

(4) Вычисление расстояния \mathbf{P}'_i до начала координат O' , если значение $|y'_i|$ точек больше, чем пороговая величина $\varepsilon > 0$. Эта пороговая величина используется для устранения точек, которые лежат на одной прямой с двумя точками, имеющими максимальное расстояние между собой (первая ось), и точками на линии, перпендикулярной к первой оси (которая могла бы дать большую ось).

(5) выберем $a_2 = \max \{b_i\}$, где b_i полуось, полученная с использованием \mathbf{P}'_i в (1).

(6) В завершение алгоритма получаем полуоси эллипса (1):

$$\begin{aligned} a &= \max \{a_1, a_2\}, \\ b &= \min \{a_1, a_2\} \end{aligned} \quad (19)$$

и его ориентацию $\Omega_E = \Omega + \Pi/2$, если a_2 – большая ось, иначе $\Omega_E = \Omega$.

Доказательство 1. Чтобы доказать, что этот эвристический метод охватывает все точки, будем считать, что точка данных $p_i \in \mathbf{P}_N$; \mathbf{P}_N представляет собой набор из n точек, который используется для вычисления параметров эллипса. Сначала, если $p_i \notin \text{Ellipse}$, то $d_{ij} = \|p_i - p_j\|$ при $p_i \in \mathbf{P}_N - \{p_i\}$ могло бы быть

$d_{ij} \geq d_{max}$. Однако первая ось эллипса – максимальное расстояние между всеми точками, тогда d_{ij} – новая первая ось, а $p_i \in Ellipse$. Следовательно, если $p_i \notin Ellipse$, то $d_{ij} \leq d_{max}$. Теперь мы знаем, что вторая ось удовлетворяет условию $b_i^2 \leq b^2$. Используя (18) и (1), получаем:

$$\begin{aligned} b_i^2 &\leq b^2, \\ \frac{y_i'^2}{1 - x_i'^2 / a^2} &\leq b^2, \\ x_i'^2 / a^2 + y_i'^2 / b^2 &\leq 1. \end{aligned} \quad (20)$$

Следовательно, p_i удовлетворяют уравнению эллипса ($p_i \in Ellipse$) при $i, j = 1, \dots, n$.

Эвристический подход является эффективным методом для охвата данных эллипсом. Он обеспечивает плавное изменение параметров эллипса; среднее время вычисления $O(n \log(n))$, где n – размер данных. Однако этот метод не учитывает ни неопределенные данные, ни последовательность полученных данных, которые характеризуют реальные эксперименты. Следующий подраздел посвящен этому вопросу.

5.3. Идентификация оптимальных параметров с использованием РФК

Фильтр Калмана используется во многих областях как общий метод интегрирования зашумленных измерений [24, 25]. В данной статье фильтр Калмана используется для улучшения параметров полученного эллипса. Общее коническое уравнение имеет вид

$$f(x, y) = Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0. \quad (21)$$

В соответствии с действительными константами A, B, C, D, E и F получаем аналитическое уравнение различного вида – парабола, эллипс и гипербола. Эллипс получается, если конические параметры (21) удовлетворяют условию $B^2 - AC < 0$.

Проблема заключается в подборе конического сечения (21), соответствующего набору из n точек $\{p_i\} = \{(x_i, y_i)\} |, i = 1, \dots, n$. Выбор осуществляется с помощью выражения (21) с параметрами эллипса, полученными эвристическим методом. Эти точки удовлетворяют условию $f(x_i, y_i) < \delta$, а $\delta \in \mathbb{R}^+$ близко к нулю, т. е. эти точки близки к границе полученного эллипса. Другие точки для этого метода не рассматриваются. Поскольку данные зашумленные, маловероятно, что можно найти набор параметров (A, B, C, D, E, F) (за исключением тривиального решения $A = B = C = D = E = F = 0$), так что $f(x_i, y_i) = 0$. Этот метод применяется для подбора конического сечения.

Вектор состояния определяется коническими параметрами следующим образом $x = [A, B, C, D, E, F]^T$, а вектор измерения – точкой, как $z_i = [x_i, y_i]^T$; линейная динамическая система (в форме для дискретного времени) может быть описана следующим образом:

$$x_{i+1} = F_i x_i + w_i, \quad (22)$$

$$z_i = H_i x_i + v_i, \quad (23)$$

где $i = 0, 1, \dots, n$. Матрица состояния F_i представляет собой единичную матрицу 6-го порядка (I_6), w_i – вектор случайных возмущений, который обычно моде-

лируется, как белый шум:

$$E[\mathbf{w}_i] = 0, \quad E[\mathbf{w}_i \mathbf{w}_i^T] = \mathbf{Q}_i.$$

Уравнение измерений (23) является нелинейным в функции идеального измерения \mathbf{z}'_i (т.е. полиномиального уравнения, которое не удовлетворяет принципу суперпозиции [34]), и оно описывается функцией наблюдения:

$$f_i(\mathbf{z}'_i, \mathbf{x}_i) = x_i^2 A + 2x'_i y'_i B + y_i^2 C + 2x'_i D + 2y'_i E + F. \quad (24)$$

Реальное измерение \mathbf{z}_i считается поврежденным аддитивным шумом \mathbf{v}_i . Модель шума \mathbf{v}_i описывается в подразделе 5.1. Далее $f_i(\mathbf{z}'_i, \mathbf{x}_i)$ следует разложить в ряд Тейлора вблизи $(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})$

$$\begin{aligned} f_i(\mathbf{z}'_i, \mathbf{x}_i) &= f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1}) + \frac{\partial f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})}{\partial \mathbf{z}'_i} (\mathbf{z}'_i - \mathbf{z}_i) + \\ &+ \frac{\partial f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})}{\partial \mathbf{x}_i} (\mathbf{x}_i - \hat{\mathbf{x}}_{i|i-1}) - O((\mathbf{z}'_i - \mathbf{z}_i)^2) + \\ &+ O((\mathbf{x}_i - \hat{\mathbf{x}}_{i|i-1})^2). \end{aligned} \quad (25)$$

Игнорируя члены второго порядка, получаем линеаризованное уравнение измерения:

$$\mathbf{y}_i = \mathbf{M}_i \mathbf{x}_i - \xi_i, \quad (26)$$

где \mathbf{y}_i – вектор нового измерения, ξ_i – вектор шума нового измерения и \mathbf{M}_i – линеаризованная матрица преобразования:

$$\begin{aligned} \mathbf{y}_i &= -f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1}) + \frac{\partial f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})}{\partial \mathbf{x}_i} \hat{\mathbf{x}}_{i|i-1}; \\ \xi_i &= \frac{\partial f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})}{\partial \mathbf{z}'_i} (\mathbf{z}'_i - \mathbf{z}_i); \\ \mathbf{M}_i &= \frac{\partial f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})}{\partial \mathbf{x}_i}. \end{aligned}$$

Ясно, что $E[\xi_i] = 0$ и $E[\xi_i \xi_i^T] = R_{\xi_i}$ и между процессом шума системы и наблюдением корреляция отсутствует. Производные от $f_i(\mathbf{z}_i, \mathbf{x}_i)$ относительно \mathbf{x} и \mathbf{z}_i имеют вид:

$$\frac{\partial f_i(\mathbf{z}_i, \mathbf{x})}{\partial \mathbf{x}} = [x_i^2, 2x_i y_i, y_i^2, 2x_i, 2y_i, 1], \quad (27)$$

$$\frac{\partial f_i(\mathbf{z}_i, \mathbf{x})}{\partial \mathbf{z}_i} = 2[x_i A + y_i B + D, \quad y_i C + x_i B + E]. \quad (28)$$

Затем используется РФК с учетом того, что уравнение для выхода фильтра (23) получают из нелинейного уравнения измерения (24). РФК описывается хорошо известными шагами:

- инициализация: $\Pi_{0|0} = \Pi_0, \hat{\mathbf{x}}_{0|0} = E[\mathbf{x}_0]$;
- предсказание состояний: $\hat{\mathbf{x}}_{i|i+1} = \mathbf{F}_{i-1} \hat{\mathbf{x}}_{i-1}$;
- предсказание ковариационной матрицы состояния: $\Pi_{i|i-1} = \mathbf{F}_{i-1} \Pi_0 \mathbf{F}_{i-1}^T + \mathbf{Q}_{i-1}$;
- матрица коэффициентов усиления Калмана: $K_i = \Pi_{i|i-1} \mathbf{H}_{i-1}^T (\mathbf{H}_{i-1} \Pi_{i|i-1} \mathbf{H}_{i-1}^T + \mathbf{R}_v)^{-1}$;

- расчет оценки состояния: $\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_{i|i-1} + \mathbf{K}_i(\mathbf{z}_i - \mathbf{H}_i\hat{\mathbf{x}}_{i|i-1})$;
- расчет ковариационной матрицы состояний: $\mathbf{\Pi}_i = (\mathbf{I} - \mathbf{K}_i\mathbf{H}_i)\mathbf{\Pi}_{i|i-1}$.

Обратим внимание на то, что метод фильтра Калмана применяется обычно к временной последовательности. Здесь же он применяется к пространственно-временной последовательности, включающей данные от каждого датчика в каждый момент времени. Благодаря своей рекурсивной природе она больше подходит для задач, в которых измерения поступают в последовательном порядке.

В противном случае, если имеются все измерения или их можно получить одновременно, целесообразно применить фильтр Калмана для совместного оценивания (т.е. обработки всей последовательности одновременно). Таким образом, метод фильтра Калмана эквивалентен методу наименьших квадратов, только если система линейна.

Для нелинейных задач РФК будет приводить к различным результатам в зависимости от порядка обработки измерений одного за другим, и, возможно, возникнет риск оказаться в ловушке локальных минимумов [20].

6. Результаты моделирования

Чтобы продемонстрировать эффективность предлагаемого закона управления в мультиконтроллерной структуре, а также метод обнаружения препятствий в реальном времени при окружении препятствия эллипсом в целях его обхода, было проведено два статистических обследования. В этой статье рассматривается мобильный робот с радиусом $R_R = 0,065$ м и шесть инфракрасных датчиков расстояний с максимальной дальностью обнаружения до $D_{Lmax} = 0,30$ м (см. рис. 8). Эти датчики размещены в передней части робота под углом 30° между каждыми парами датчиков (рис. 10). Точность датчиков согласно техническому описанию составляет около 10% от D_{Lmax} . При моделировании мы рассматриваем область неопределенности с максимальным значением 20% от D_{Lmax} , обеспечивающей область худших значений.

Для выражений (10) и (11), являющихся законами управления, коэффициенты усиления определяются как $K_x = 0,8$, $K_y = 5$ и $K_\theta = 3$. Значения этих коэффициентов были выбраны эвристически для того, чтобы получить гладкую траекторию, быстрые реакцию и скорость, но в рамках следующих предельных скоростей мобильного робота: $v_{max} = 0,4$ м/с и $w_{max} = 3$ рад/с. Время выборки составляет 0,01 с. В каждом сеансе моделирования робот начинает двигаться из одного и того же положения и достигает одного и того же положения в конце пути. Эвристический метод не работает, пока не получит достаточного количества данных о дальности ($n_{data} \geq 3$).

Первое исследование проводилось для сравнения выполнения адаптивного обхода препятствий между предлагаемым законом управления и законом управления Канаямы в представленной выше мультиконтроллерной архитектуре для обеспечения безопасной и надежной навигации (см. подраздел 3.1).

На рис. 10 показана траектория движения робота для двух различных законов управления в специфической среде, где параметры эллипса, который ограждает препятствие, известны. Отмечено, что траектория при использовании закона управления Канаямы (пунктирная линия) находится ближе к препятствию, чем траектория при использовании предлагаемого закона управления (сплошная линия) в контроллере обхода препятствий. Кроме того, с использованием контроллера притяжения к цели траектория при предложенном законе управ-

ления получается короче. Этот рисунок показывает, что предложенный закон управления повышает безопасность траектории для достижения цели.

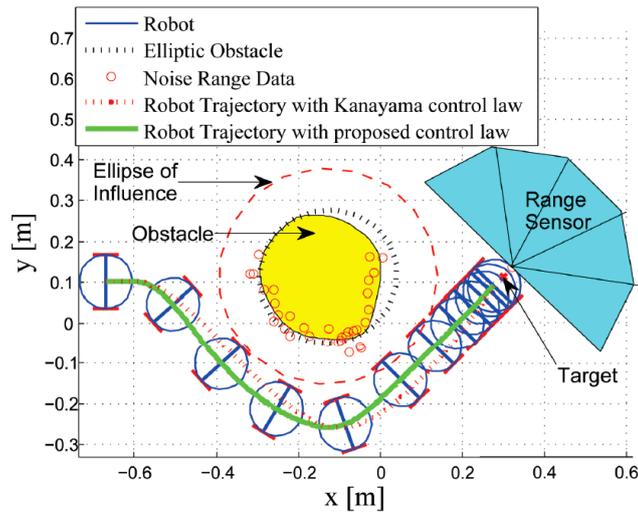


Рис. 10. Траектория робота при использовании двух разных законов управления.
 Надписи сверху вниз и слева направо: Робот. Препятствие в форме эллипса. Зашумленные данные по дальности. Траектория робота при использовании закона управления Канаемы. Траектория робота при использовании предложенного закона управления.
 Эллипс влияния. Препятствие. Датчик расстояния. Цель

На рис. 11 представлены ошибки положения и ориентации для двух разных законов управления. Видно, что предложенный закон управления улучшает сходимость в контроллере притяжения к цели. Заметим также, что e_x демонстрирует близкую сходимость между двумя разными законами управления, потому что изменения для получения предлагаемого закона управления касаются только угловой скорости (больше связанной с e_y и e_θ).

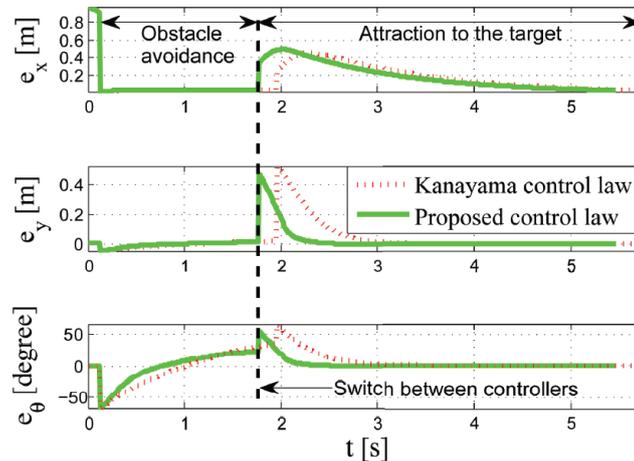


Рис. 11. Ошибки положения и ориентации
 Надписи слева направо и сверху вниз
 Обход препятствия. Притяжение к цели. Закон управления Канаемы. Предложенный закон управления. Переключение между контроллерами

Рис. 12 показывает изменение функции Ляпунова V_0 (12), связанное с законом управления (см. рис. 3), в процессе навигации. Эта функция убывает асимптотически к точке равновесия. Заметим, что функция Ляпунова предложенного закона управления имеет более низкое значение при переключении между контроллерами и более быструю сходимость, чем при законе управления Канаямы.

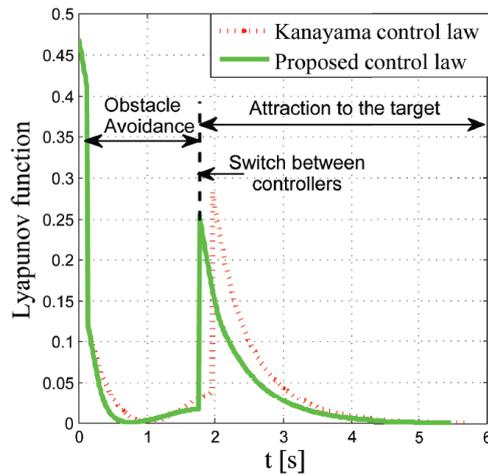


Рис. 12. Эволюция функций Ляпунова для двух разных законов управления во время навигации робота.

Надписи сверху вниз: Закон управления Канаямы. Предложенный закон управления. Притяжение к цели. Обход препятствия. Переключение между контроллерами

Второе исследование сосредоточено на предложенном эвристическом методе и фильтре Калмана, которые дают удовлетворительные результаты, когда они используются для навигации в режиме реального времени в загроможденной окружающей среде (рис. 13). На этом рисунке показана траектория движения робота в среде с тремя препятствиями. Буфер данных по дальности, используемый для вычисления параметров эллипса, очищается для каждого нового обнаруженного препятствия. Показано, что эллипс, полученный с использованием эвристического метода, охватывает все точки, как это было показано выше. Кроме того, отмечается, что робот обходит препятствия по гладкой траектории.

Эта траектория была получена с помощью алгоритма обхода препятствий в режиме реального времени [16] с использованием предложенного закона управления. Этот алгоритм получает параметры с использованием сочетания предлагаемого эвристического подхода и РФК.

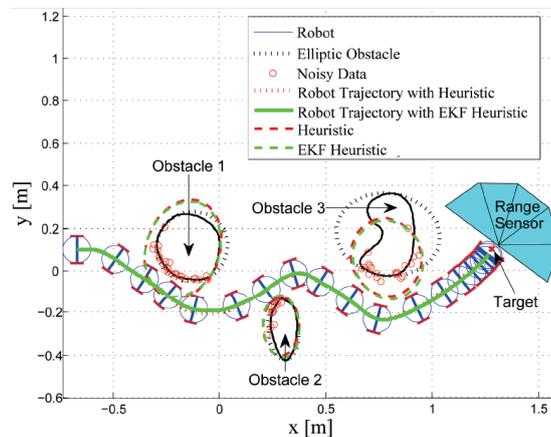


Рис. 13. Траектория робота при использовании подхода, сочетающего эвристический метод и РФК для описания препятствий.

Надписи в рамке сверху вниз: Робот. Препятствие в форме эллипса. Зашумленные данные по дальности. Траектория робота при использовании закона эвристического метода. Траектория робота при сочетании эвристического метода и РФК.

Эвристический метод. Эвристический метод + РФК.

Obstacle – Препятствие, Range sensor – Датчик расстояния, Target – Цель

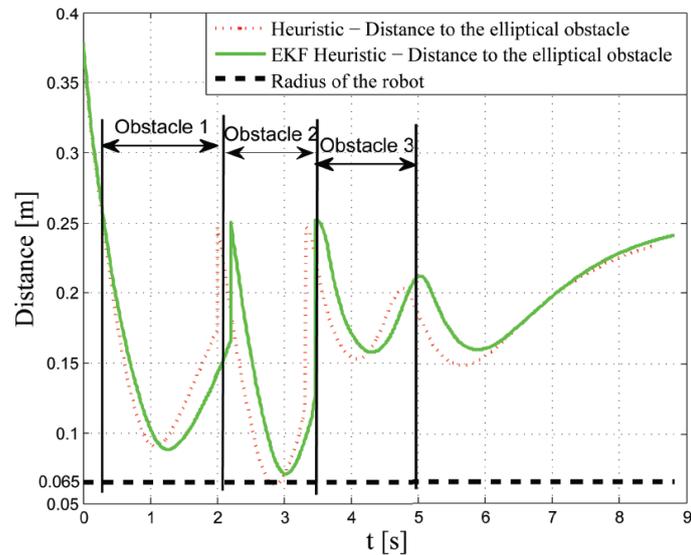


Рис. 14. Расстояние от робота до препятствия в форме эллипса.
 Сверху вниз: Эвристический метод. Расстояние до препятствия в форме эллипса
 Эвристический метод + РФК. Расстояние до препятствия в форме эллипса

На рис. 14 показано минимальное расстояние между эллиптическими препятствиями, полученное из знания всех данных по дальности (без шума), и положением робота по эллиптической траектории при использовании только эвристического метода (точечная линия) и сочетании эвристического метода с РФК (сплошная линия). Это показывает, что робот никогда не сталкивается с препятствиями в случае использования предлагаемого закона управления, поэтому предлагаемый подход эффективен при использовании в загроможденной и неструктурированной среде.

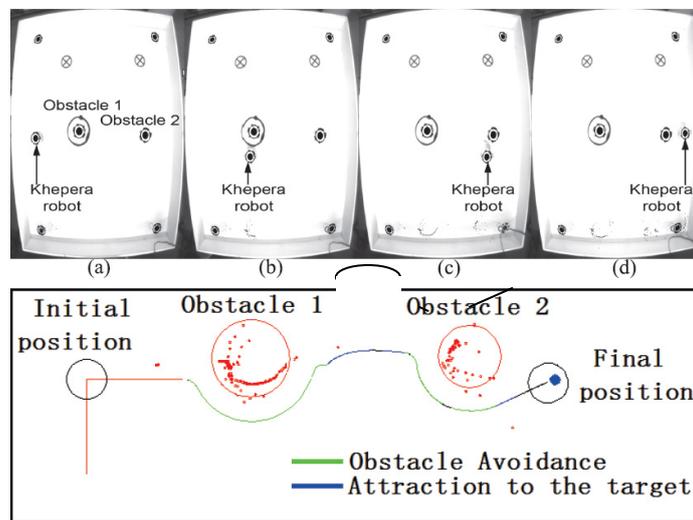


Рис. 15. Вид сверху траектории робота на платформе и не определенные данные по расстоянию, наблюдаемые с местоположения робота.
 Нижний рисунок. Слева направо: Исходное положение. Конечное положение.
 Сверху вниз: Траектория обхода препятствия. Траектория стремления к цели

Эксперименты проводились с использованием робота Kheperar® III (рис. 15). Его кинематическая модель имеет вид (2). В самых первых испытаниях навигация осуществлялась на платформе, оснащенной камерой наверху, которая дает положения и ориентации роботов и препятствий, которые следует обойти.

Навигация осуществляется на платформе с использованием бортовых инфракрасных датчиков робота. Данный тест показывает эффективность предложенного робастного подхода с использованием ограждающих эллипсов.

На этом же рисунке показана реальная траектория робота с обходом двух препятствий. Видно, что робот успешно стремится к своей цели после обхода двух препятствий (окруженных двумя эллипсами влияния).

7. Выводы

В данной статье предлагается адаптивная эллиптическая траектория, вычисляемая в режиме реального времени для осуществления гладкой и безопасной автономной навигации мобильного робота. Траектории создаются по принципу предельного цикла в целях получения универсального и гибкого алгоритма навигации в неструктурированных средах. Траекторию, построенную по принципу эллиптического предельного цикла, получают при использовании предложенного эвристического метода в сочетании с РФК, который для получения параметров эллипса использует не достоверные данные о расстоянии. Работа этого метода был продемонстрирован в режиме реального времени, с использованием всего спектра данных. Предлагаемый адаптивный алгоритм навигации был запрограммирован в системе с мультиконтроллерной структурой. Приведено доказательство устойчивости структуры управления в целом с использованием только одного закона управления для двух различных контроллеров. Представлены результаты моделирования и экспериментов в различных условиях. Доказана эффективность и гибкость предлагаемой структуры управления и обнаружения препятствий.

В будущем предполагается тщательное рассмотрение проблемы обнаружения выбросов данных, и предлагаемая структура управления будет развита для системы с несколькими роботами. Кроме того, будут проведены исследования алгоритма при наличии динамических препятствий.

ЛИТЕРАТУРА

1. **Latombe, J.C.**, Robot Motion Planning, Kluwer Academic Publishers, Boston, MA, 1991.
2. **Rimon, E. and Koditschek, D.**, Exact Robot Navigation Using Artificial Potential Fields, IEEE Transactions on Robotics and Automation, 1992, vol. 8, no. 5, pp. 501–518.
3. **Fraichard, T.**, Trajectory Planning in a Dynamic Workspace: a ‘State-Time Space’ Approach, Advanced Robotics, 1999, vol. 13, no. 1, pp. 75–94.
4. **Jur-Van-Den, B., and Overmars, M.**, Roadmap-Based Motion Planning in Dynamic Environments, IEEE Transactions on Robotics, 2005, vol. 21(5), pp. 885–897.
5. **Egerstedt, M. and Hu, X.**, A Hybrid Control Approach to Action Coordination for Mobile Robots, Automatica, 2002, vol. 38(1), pp. 125–130.
6. **Toibero, J., Carelli, R., and Kuchen, B.**, Switching Control of Mobile Robots for Autonomous Navigation in Unknown Environments, IEEE International Conference on Robotics and Automation, 2007, pp. 1974–1979.
7. **Adouane, L.**, Hybrid and Safe Control Architecture for Mobile Robot Navigation, 9th Conference on Autonomous Robot Systems and Competitions, Portugal, May 2009.
8. **Khatib, O.**, Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, International Journal of Robotics Research, 1986, vol. 5, pp. 90–99.

9. **Arkin, R. C.**, Motor Schema-based Mobile Robot Navigation, *International Journal of Robotics Research*, 1989, vol. 8, no. 4, pp. 92–112.
10. **Zapata, R., Cacitti, A., and Lepinay, P.**, DVZ-Based Collision Avoidance Control of Non-holonomic Mobile Manipulators, *JESA, European Journal of Automated Systems*, 2004, vol. 38(5), pp. 559–588.
11. **Arkin, R.C.**, *Behavior-Based Robotics*, MIT Press, 1998.
12. **De Luca, A. and Oriolo, G.**, Local Incremental Planning for Nonholonomic Mobile Robots, *IEEE International Conference on Robotics and Automation*, May 1994, vol. 1, pp. 104–110.
13. **Kim D.-H. and Kim, J.-H.**, A Real-Time Limit-Cycle Navigation Method for Fast Mobile Robots and its Application to Robot Soccer, *Robotics and Autonomous Systems*, 2003, vol. 42(1), pp. 17–30.
14. **Jie, M.S., Baek, J.H., Hong, Y.S., and Lee, K.W.**, Real Time Obstacle Avoidance for Mobile Robot Using Limit-Cycle and Vector Field Method, *Knowledge-Based Intelligent Information and Engineering Systems*, October 2006.
15. **Adouane, L.**, Orbital Obstacle Avoidance Algorithm for Reliable and On-Line Mobile Robot Navigation, *9th Conference on Autonomous Robot Systems and Competitions*, May 2009, Portugal.
16. **Adouane, L., Benzerrouk, A., and Martinet, P.**, Mobile Robot Navigation in Cluttered Environment Using Reactive Elliptic Trajectories, *18th IFAC World Congress*, August 2011.
17. **Benzerrouk, A., Adouane, L., and Martinet, P.**, Lyapunov Global Stability for a Reactive Mobile Robot Navigation in Presence of Obstacles,” *ICRA’10 International Workshop on Robotics and Intelligent Transportation System*, 2010.
18. **Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T.**, A Stable Tracking Control Method for an autonomous mobile robot, *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1990, pp. 384 – 389.
19. **Welzl, E.**, *Smallest Enclosing Disks (Balls and Ellipsoids), Results and New Trends in Computer Science*, Springer-Verlag, 1991, pp. 359–370.
20. **Zhang, Z.**, *Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting, Image and Vision Computing*, 1997, vol. 15, pp. 59–76.
21. **Vilca, J., Adouane, L., and Mezouar, Y.**, On-Line Obstacle Detection Using Data Range for Reactive Obstacle Avoidance, *12th International Conference on Intelligent Autonomous Systems*, Korea, June 2012.
22. **Xiong, K., Wei, C., and Liu, L.**, *Robust Kalman Filtering for Discrete-Time Nonlinear Systems with Parameter Uncertainties*, *Aerospace Science and Technology*, 2011.
23. **Fouque, C., Bonnifait, P., and Betaille, D.**, Enhancement of Global Vehicle Localization Using Navigable Road Maps and Dead-Reckoning, *IEEE Position Location and Navigation Symposium*, 2008.
24. **Rigatos, G.G.**, Extended Kalman and Particle Filtering for Sensor Fusion in Motion Control of Mobile Robots, *Mathematics and Computers in Simulation*, November 2010, vol. 81, no. 3, pp. 590–607.
25. **Levinson, J. and Thrun, S.**, Robust Vehicle Localization in Urban Environments Using Probabilistic Maps, *IEEE International Conference on Robotics and Automation*, Alaska, USA, May 2010.
26. **Porrill, J.**, Fitting Ellipses and Predicting Confidence Envelopes Using a Bias Corrected Kalman Filter, *Image and Vision Computing*, February 1990, vol. 8, no. 1, pp. 37–41.
27. **Vilca, J., Adouane, L., and Mezouar, Y.**, Robust Online Obstacle Detection Using Range Data for Reactive Navigation, *10th International IFAC Symposium on Robot Control*, Croatia, September 2012.
28. **Brooks, R.A.**, A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, vol. RA-2, March 1986, pp. 14–23.
29. **Adouane, L. and Le Fort-Piat, N.**, Behavioral and Distributed Control Architecture of Control for Minimalist Mobile Robots, *Journal Européen des Systèmes Automatisés*, 2006, vol. 40, no. 2, pp. 177–196.
30. **Maalouf, E., Saad, M., and Saliyah, H.**, A Higher Level Path Tracking Controller for a Four-Wheel Differentially Steered Mobile Robot, *Robotics and Autonomous Systems*, 2006, vol. 54, pp. 23–33.
31. **De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D.**, The Mahalanobis Distance, *Chemometrics and Intelligent Laboratory Systems*, 2000, vol. 50, no. 1, pp. 1–18.
32. **Barshan, B. and Kuc, R.**, Active Sonar for Obstacle Localization Using Envelope Shape Information, *International Conference on Acoustics, Speech, and Signal Processing*, April 1991, vol. 2, pp. 1273–1276.

33. **Burguera, A., Gonzlez, Y., and Oliver, G.,** Sonar Sensor Models and Their Application to Mobile Robot Localization, *Sensors*, December 2009, vol. 9, no. 12, pp. 10217–10243.
34. **Khalil, H.K.,** *Nonlinear Systems*, 3rd ed., P. Hall, Ed., 2002.

Abstract. This paper deals with the problem of mobile robot navigation in cluttered environment. Adaptive elliptic trajectories are exploited for reactive obstacle avoidance using only position information and uncertain range data. The obstacle avoidance strategy used is based on the elliptic limit-cycle principle where each obstacle is surrounded by an ellipse. The ellipse parameters are computed online using a sequence of uncertain range data. An online heuristic method combined with the extended Kalman filter (EKF) is used to compute the ellipse parameters. It is demonstrated that this process ensures that all range data are surrounded by a computed ellipse. Moreover, this paper proposes a single control law to the multicontroller architecture where a reactive obstacle avoidance algorithm is embedded. The proposed control law is based on the Kanayama control law; it is designed to improve the performance of the controllers. The stability of this control architecture is proved according to the Lyapunov synthesis. Simulations and experiments in different environments have been performed to demonstrate the efficiency and reliability of the proposed online navigation in cluttered environment.

Key words: Mobile robots navigation; Multi-controller architecture; Reactive control; Obstacle detection and avoidance; Telemetry; Parameter identification; Extend Kalman Filter.

Материал поступил 19.07.12