

Министерство образования и науки Российской Федерации
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(государственный университет)
ФАКУЛЬТЕТ УПРАВЛЕНИЯ И ПРИКЛАДНОЙ МАТЕМАТИКИ
КАФЕДРА ИНФОРМАТИКИ
(Специализация 010900 «Прикладные математика и физика»)

Разработка надежной системы навигации и контроля телеуправляемого робота
и ее тестирование в виртуальном окружении

Выпускная квалификационная работа
студента 4 курса 073а группы
Шепелева Дениса Александровича

Научный руководитель
Устюжанин А.Е., к.ф.-м.н.

г. Долгопрудный, 2014 г.

Содержание

1. Задача навигации мобильных роботов.....	4
1.1. Автономные системы управления мобильных роботов.....	4
1.2. Задачи построения навигационной карты и определение координат мобильного робота в пространстве.....	6
1.3. Постановка задач данной работы.....	7
2. Выбор датчиков для решения задачи автономной навигации мобильного робота.....	8
2.1. Основные типы датчиков и принципы их работы.....	8
2.1.1. Датчики перемещения, скорости и ускорения.....	8
2.1.2. Спутниковая навигация.....	11
2.1.3. Стереоскопические видео системы.....	12
2.1.4. Дальнометрические приборы.....	14
2.2. Преимущества и недостатки различных типов датчиков.....	16
2.3. Выбор датчиков.....	18
3. Создание виртуальной модели мобильного робота и рабочего окружения.....	19
3.1. Средства визуализации роботов и рабочей среды.....	19
3.2. Ограничения рабочей среды.....	20
3.3. Виртуальная модель робота.....	20
3.4. Виртуальная среда.....	22

4. Программное обеспечение для разработки робототехнических систем.....	23
4.1. <i>Robotic Operating System</i>	24
4.2. Понимание файловой системы ROS. Концепция графов.....	25
5. Разработка поста управления мобильным роботом.....	26
5.1. Создание Gazebo плагина для управления движением робота.....	26
5.2. Расчет одометрии мобильного робота.....	27
6. Разработка системы автономной навигации.....	29
6.1. <i>move_base</i>	29
6.2. Поведение робота.....	31
6.3. Создание карты.....	32
6.4. Демонстрационный модуль автономной навигации.....	32
6.5. Модуль аварийного управления в случае потери оператором связи с мобильным роботом.....	33
7. Заключение.....	34
Список литературы:.....	36

1. Задача навигации мобильных роботов

Приведена актуальность создания автономных систем управления роботами. Рассмотрены основные положения задачи навигации мобильных роботов. Перечислены основные проблемы, возникающие при решении этой задачи.

1.1. Автономные системы управления мобильных роботов.

В современном мире роботы приобретают огромную роль во многих сферах человеческой деятельности. Они используются в медицине, сфере обслуживания, военном деле, при проведении спасательных операций. Главной задачей роботов является замена человека при выполнении монотонных и опасных операций. На заре робототехники это были не сложные, либо циклически повторяющиеся процедуры, однако развитие робототехники идет по пути увеличения номенклатуры операций, которые должны выполнять роботы без участия человека.

Современные роботы выполняют разнообразные функции. Например, медицинский робот Da Vinci позволяет проводить сложнейшие операции; марсоходы Spirit, Opportunity и Curiosity доказали, что роботы способны хорошо справляться с задачей изучения Марса; различные роботы использует подразделения ФСБ для предотвращения террористических актов, а также роботы применяются в спасательных формированиях Росатома для ликвидации последствий чрезвычайных ситуаций. Однако даже самые современные образцы роботов не способны выполнять весь комплекс работ самостоятельно. Это говорит о сложности решения задач автономного управления роботом. Поэтому большинство мобильных систем работают под управлением человека-оператора, что, очевидно, имеет ряд недостатков: ошибки, присущие человеку, несовершенство органов чувств, недостаточная скорость реакции, неоптимальная оценка ситуации. Все эти недостатки могут быть

скомпенсированы применением систем автоматизации.

По всему миру в передовых научных учреждениях ведутся разработки по проектированию автоматических систем управления силами тысяч высококвалифицированных ученых и инженеров. При этом правительства развитых стран не жалеют денег на их разработку. Например, DARPA Grand Challenge – соревнования мобильных роботов, финансируемые правительством США, имеют призовой фонд несколько миллионов долларов. Целью этих соревнований является создание полностью автономных транспортных средств. Еще одним примером может послужить беспилотный автомобиль Google — проект компании Google по развитию технологии беспилотного автомобиля. Результатом этих работ является закон штата Невада, разрешающий использование беспилотных автомобилей на дорогах общего пользования. А в сентябре 2012 года, власти штата Калифорния легализовали использование автомобилей с функцией автопилота.

Несмотря на огромные усилия разработчиков, хорошее финансирование и некоторые успехи, в данной области существует достаточно много не до конца разрешенных проблем, в частности: адекватное отображение состояния окружающей среды в системе управления робота, реакция робота на изменяющиеся условия среды, взаимодействие в группе автономных роботов. Поэтому в настоящее время наиболее актуальной задачей является создание комбинированных систем, в которых, наряду с телеуправлением человеком-оператором, используются решения, позволяющие системе в автономном режиме выполнять некоторые простые операции или продолжить работу в случае потери связи с оператором.

Особенно важным является применение автономных систем управления роботами при работе в различных недетерминированных условиях, например в зонах с высоким уровнем радиоактивного излучения. При этом возникает актуальная задача автономного перемещения робота при потере связи с постом управления. В этом случае невозможно обеспечить эвакуацию робота с

помощью людского персонала, а использование других роботов для этого может быть невозможным в виду стесненных или неблагоприятных условий, что может приводить к потере робота-эвакуатора. Поэтому подвижный аппарат робота должен быть способен самостоятельно выйти в заданную точку, желательно с выполнением поставленной перед ним первоначальной задачей. Для этого необходимо, чтобы робот мог в автономном режиме решать задачи автономной разведки и навигации.

1.2. Задачи построения навигационной карты и определение координат мобильного робота в пространстве.

Задача автономной навигации мобильного робота состоит из двух основных подзадач – это картографирование и определение собственных координат на карте. Основными методами решения этих задач являются, так называемые, СЛАМ – алгоритмы. СЛАМ (от англ. SLAM – Simultaneous localization and mapping) – метод, используемый роботами и автономными транспортными средствами для построения карты в неизвестном пространстве или для обновления карты в заранее известном пространстве с одновременным контролем текущего местоположения и пройденного пути. Этот метод наиболее распространен для решения подобных задач.

При этом существует несколько принципиальных проблем, связанных с этим методом:

Во-первых, так называемая проблема «яйца и курицы». Очевидно, что для построения карты, нужно знать свое месторасположение, а для определения своих координат – нужна карта. Это порождает множество различных начальных условий для алгоритмов управления.

Во-вторых, невозможно построить абсолютно точную карту и определить на ней абсолютно точное местоположение. Это связано с тем, что каждый датчик имеет ошибку измерений, которая накапливается в процессе работы. В этом случае основной целью алгоритма является минимизация этой ошибки.

Существует множество подходов для компенсации ошибок, наиболее популярными сегодня являются *фильтр Калмана* и *фильтр частиц*. Все эти подходы итеративные, что существенно ускоряет процесс управления и оптимизирует вычислительные затраты.

В-третьих, понятие *карта* не имеет точного определения. Карта может быть 2D, 2.5D, 3D. Она может состоять из множества точек, которые описывают траектории движения или геометрию пространства, множества ячеек, которые обозначают свободные или занятые области. Также возможно использование топологической карты, на которой каждый объект принадлежит к определенной группе других объектов. Все это порождает огромное количество различных подходов к решению задачи, каждый из которых может, как упростить ее, так и усложнить.

В-четвертых, существует множество средств очувствления робота. Выбор датчиков определяет, каким образом будет решаться навигационная задача.

1.3. Постановка задач данной работы.

Основная цель - создание системы управления мобильного робота, которая, в случае потери связи подвижного аппарата робота с постом управления, позволит роботу обеспечить возвращение ПА в заранее заданные координаты в автоматическом режиме.

Для достижения поставленной цели необходимо решить ряд задач:

1. проведение анализа существующих средств организации автономного движения мобильных роботов (датчики) и обоснованный выбор оборудования для реализации автономного перемещения подвижного аппарата мобильного робота при потере связи с постом управления;
2. разработка системы управления автономным перемещением мобильного робота:
 - создание виртуальной модели робота и рабочего окружения;
 - разработка поста управления мобильным роботом;

- разработка системы автономной навигации;
- разработка системы автоматического возвращения ПА в случае отключения от поста управления.

На первом этапе работы был проведен широкий обзор существующих средств оцувствления мобильных роботов. Выбор сенсоров является важным этапом для последующей разработки системы автономного управления мобильным роботом.

2. Выбор датчиков для решения задачи автономной навигации мобильного робота.

Приведены основные типы датчиков, используемые для решения задачи автономной навигации. Проанализированы их достоинства и недостатки. Осуществлен выбор датчиков для реализации системы автономного управления мобильным роботом.

2.1. Основные типы датчиков и принципы их работы.

Приведем основные типов датчиков для системы оцувствления автономного мобильного робота:

- датчики перемещения, скорости и ускорения (акселерометры, гироскопы, энкодеры и др.);
- спутниковая навигация (GPS, ГЛОНАСС);
- стереоскопические видеосистемы;
- дальнометрические приборы;

Отдельно рассмотрим основные принципы работы каждого типа датчиков.

2.1.1. Датчики перемещения, скорости и ускорения.

а) Датчик угла поворота (энкодер).

Энкодер – это электромеханическое устройство, с помощью которого можно определить положение вращающейся оси (вала). В данном устройстве

механическое движение преобразовывается в электрические сигналы, определяющие положение объекта, дают информацию об угле поворота вала, его положении и направлении вращения. С помощью *энкодера* также можно измерить длину и расстояние или установить перемещение.

Выделяют следующие типы энкодеров: *инкрементальные (инкрементные)* и *абсолютные*.

Инкрементальный энкодер - это устройство, которое определяет угол поворота вращающегося объекта, выдавая импульсный цифровой код. Используется для определения скорости вращения вала (оси), когда нет нужды сохранять абсолютное угловое положение при выключении питания. То есть, если вал неподвижен, передача импульсов прекращается. Другими словами, если включить энкодер этого типа, то отсчет поворота угла начнется с нуля, а не с угла на который он был выставлен до момента выключения. Основным преимуществом инкрементальных энкодеров является их простота, надежность и относительно низкая стоимость.

Абсолютный энкодер выдает цифровой код, различный для каждого положения объекта, позволяет определять угол поворота оси даже в случае исчезновения и восстановления питания и не требует возвращения объекта в начальное положение, что является несомненным преимуществом этого типа энкодеров. Так как угол поворота всегда известен, то счетчик импульсов в этом случае не нужен. Сигнал абсолютного энкодера не подвергается помехам и вибрации и тем самым для него не нужна точная установка вала.

По принципу действия различают: энкодеры оптические, магнитные, магниторезисторные. Для примера рассмотрим принцип работы простейшего *оптического энкодера* (Рис 1).

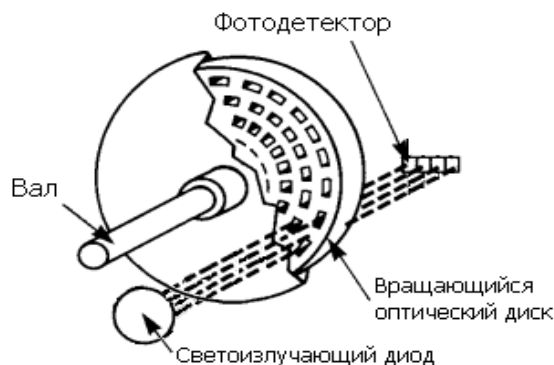


Рис 1. Структурная схема оптического энкодера.

Конструкция оптического энкодера состоит из специального оптического диска, светоизлучающего диода и фотодетектора. Диск с нанесенной оптической шкалой (поверхность диска состоит из прозрачных и непрозрачных участков) жестко закрепляется на валу. При вращении объекта специальный фотодетектор считывает информацию и преобразовывает ее в электрические импульсы.

б) Акселерометр

Акселерометр – датчик, используемый для измерения ускорения объекта. Одним из важных свойств акселерометров - они чувствительны ко всем внешним силам, в том числе и к силе тяжести. По принципу работы акселерометры разделяются на *механические* и *пьезоэлектрические*. *Механический акселерометр* представляет собой чувствительную массу, закрепленную на упругом подвесе. Отклонение массы от ее первоначального положения при наличии кажущегося ускорения несет информацию о величине этого ускорения. *Пьезоэлектрические акселерометры* опираются на свойства некоторых кристаллов, в которых возникает напряжение, когда они находятся под воздействием внешних сил. Это напряжение можно измерить и таким образом узнать ускорение системы.

в) Гироскоп

Гироскоп – устройство, способное реагировать на изменение углов

ориентации тела, на котором оно установлено, относительно инерциальной системы отсчета. Для навигационных задач используются *механические* и *вибрационные* (микро-электромеханические системы - МЭМС) гироскопы.

Среди *механических гироскопов* выделяется *роторный гироскоп* — быстро вращающееся твердое тело, ось вращения которого может свободно изменять ориентацию в пространстве. При этом скорость вращения гироскопа значительно превышает скорость поворота оси его вращения. Основное свойство такого гироскопа — способность сохранять в пространстве неизменное направление оси вращения при отсутствии воздействия на него моментов внешних сил и эффективно сопротивляться действию внешних моментов сил. Это свойство в значительной степени определяется величиной угловой скорости собственного вращения гироскопа.

Принцип работы *вибрационного гироскопа* основан на действии силы Кориолиса на колеблющийся объект. Если объект колеблется с некоторой известной скоростью $v(t)$, находится на вращающейся плоскости и ось вращения перпендикулярна оси колебаний, тогда ускорение Кориолиса равно $a = 2[\Omega \times v]$. Зная закон колебаний и измеряя значение a , можно получить угловую скорость Ω .

2.1.2. Спутниковая навигация

Спутниковая система навигации - комплексная электронно-техническая система, состоящая из совокупности наземного и космического оборудования, предназначенная для определения местоположения и точного времени, а также параметров движения для наземных, водных и воздушных объектов.

Основные элементы спутниковой системы навигации:

- Орбитальная группировка, состоящая из нескольких (от 2 до 30) спутников, излучающих специальные радиосигналы;
- Наземная система управления и контроля (наземный сегмент), включающая блоки измерения текущего положения спутников и передачи

на них полученной информации для корректировки информации об орбитах;

- Аппаратура потребителя спутниковых навигационных систем («спутниковые навигаторы»), используемое для определения координат;
- Опционально: наземная система радиомаяков, позволяющая значительно повысить точность определения координат.
- Опционально: информационная радиосистема для передачи пользователям поправок, позволяющих значительно повысить точность определения координат.

Принцип работы спутниковых систем основан на получении радиосигналов от спутников расположенных на орбите Земли. Сравнивая время задержки сигналов, приходящих от различных спутников, можно рассчитать координаты. В настоящее время наиболее широко используемыми системами спутниковой навигации являются GPS и ГЛОНАСС.

2.1.3. Стереоскопические видео системы.

Стереоскопические видео системы широко применяются в задачах распознавания образов, слежки и др. Рассмотрим упрощенный принцип работы такой системы [1]. На рисунке 2 изображена идеальная стереоскопическая система.

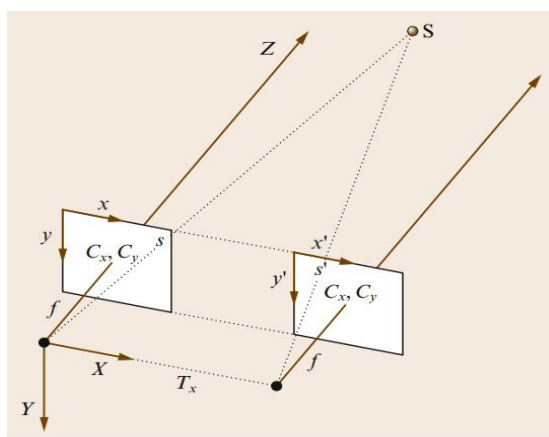


Рис 2. Схема простейшей стереоскопической системы.

Рассмотрим механизм получения координаты точки S в глобальных координатах XYZ с помощью стереоскопической системы. Известно: координаты центров фокальных плоскостей камер C_x, C_y ; фокусные расстояния f ; расстояние между центрами T_x . Точки s и s' — координаты проекции S на кадры xu и $x'u'$. Обычно, точки s и s' имеют равные координаты y и y' . Различие в x координате называется *несоответствием*.

Проекция 3D точки на фокальные плоскости можно представить в виде преобразования, т. е. с помощью *проецирующей матрицы* одной из камер, обозначаемой P .

$$P = \begin{pmatrix} F_x & 0 & C_x & -F_x T_x \\ 0 & F_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

где F_x и F_y фокальные длины изображений, а C_x и C_y — оптические центры. Тогда

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix},$$

где $(x/z, y/z)$ — координаты точек изображения. Тогда с помощью *репроецирующей матрицы* Q можно получить координаты точки S.

$$Q = \begin{pmatrix} 1 & 0 & 0 & -C_x \\ 0 & 1 & 0 & -C_y \\ 0 & 0 & 0 & F_x \\ 0 & 0 & -1/T_x & (C_x - C_{x'})/T_x \end{pmatrix}.$$

Затем, зная что *несоответствие* $d = x - x'$, получим

$$\begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} = Q \begin{pmatrix} x \\ y \\ d \\ 1 \end{pmatrix},$$

где $(X/W, Y/W, Z/W)$ — искомые координаты точки S.

2.1.4. Дальнометрические приборы.

а) Лазерные дальномеры.

Лазерный дальномер (Лидар — от англ. LIDAR) — датчик, применяемый для измерения расстояния и использующий для этого лазерный луч. Существует три основных типа лидаров[1]:

Модуляционный дальномер (modulation range sensor) — лазерный дальномер, который генерирует непрерывный лазерный сигнал с амплитудной или частотной модуляцией. Вычисляя разницу между фазами посылаемого и принятого сигналов, оценивает расстояние до цели.

Триангуляционный дальномер(triangulation range sensors) — основан на тех же принципах, что и стереоскопические системы (Рис 3).

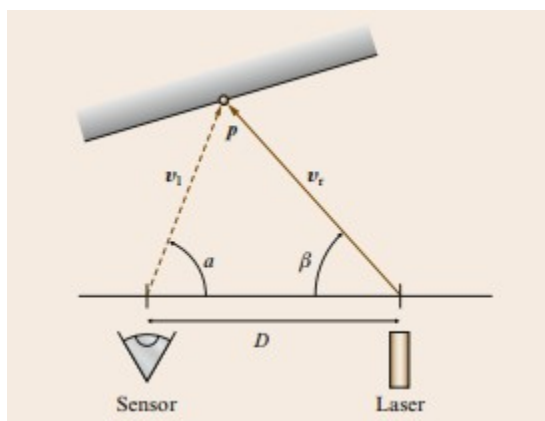


Рис 3. Схема работы триангуляционного лазерного дальномера.

Лазерный луч(Laser) создает световое пятно, датчик (Sensor) фиксирует расположение этого пятна. Зная относительное положение двух датчиков, можно рассчитать координаты точки.

ТОФ дальномер (от англ. TOF - time-of-flight range sensors) — широко используемый в робототехнике тип лазерных дальномеров. Основной принцип работы этого сенсора достаточно прост: излучатель посылает в исследуемое пространство лазерный луч, который сталкивается с препятствием, отражается, и попадает на приемник. Зная время между излучением и приемом сигнала (t)

можно найти расстояние до препятствия: $R = c * t / 2$.

Измерения, получаемые дальномером, называют *сканами*. Сканы могут быть как 2D, так и 3D. Рассмотрим устройство простейшего 2D TOF дальномера. Излучатель испускает лазерный луч, который попадает на вращающееся зеркало, отражающее луч в нужном направлении. Таким образом, возможно получения 2D срезки пространства, скан можно будет изобразить в виде таблицы из двух колонок: первая — направление, например от 0° до 160°; вторая колонка измеренное расстояние в соответствующем направлении. Стандартное разрешение LIDAR — от 0.175° до 0.5°. Скорость получения одного скана, например, SICK LMS 200 получает срежку в 180°, делая 720 измерений, всего за 0.05 секунд.

Лидары используются, например, в работах [2], [3].

б) Ультразвуковые сонары.

Рассмотрим типичный пример эхо-импульсного (от англ. pulse-echo) сонара — Polaroid 6500. Излучатель посылает звуковой импульс продолжительностью в 16 циклов с частотой 49.4 кГц. Затем, после короткой задержки, излучатель переходит в режим приемника (Рис 4).

Пусть в момент t_0 приемник поймал эхо, длительность которого превышает пороговое значение τ , тогда $r = t_m * c / 2$, где $c = 343$ м/с.

Стоит отметить, что было проведено множество исследований, которые подробно рассматривают физические свойства ультразвукового излучения, поведение звуковой волны при столкновении с различными препятствиями, результирующее эхо, а так же взаимодействие и поведение мультисонарных систем [4-10]. Ультразвуковые датчики на сегодняшний день являются одним из наиболее популярных и дешевых решений для реализации 2D навигации мобильного робота.

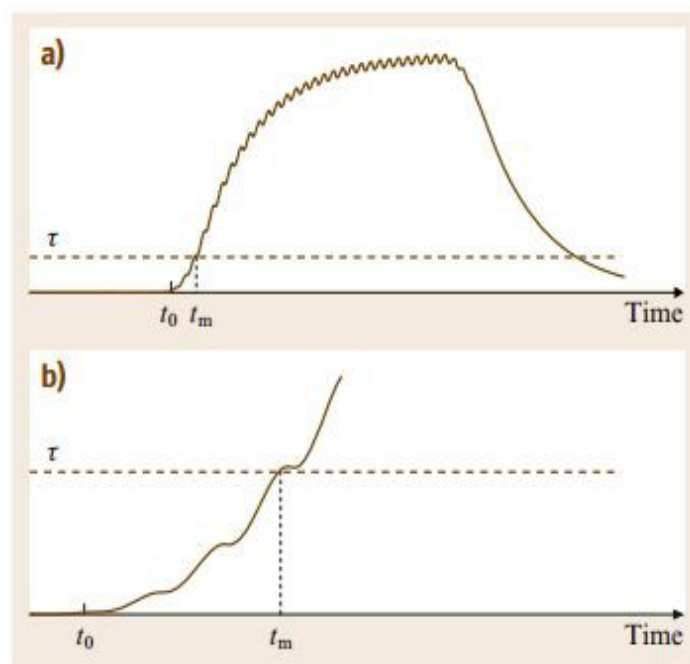


Рис 4. Процесс приема сигнала эхо-импульсного сонара.

2.2. Преимущества и недостатки различных типов датчиков

В таблице 1 приведены основные преимущества и недостатки, рассмотренных типов датчиков

Таблица 1. Сводная таблица преимуществ и недостатков датчиков.

Датчики	Преимущества	Недостатки
Гироскоп	Может измерять как угол поворота и скорость, так и момент. Не взаимодействует с окружающей средой.	Нуждаются в калибровке.
Энкодер	Простой и дешевый датчик. Много готовых вариантов использования. Не взаимодействует с окружающей средой.	Используется только для измерения угла поворота и скорости. Не учитывают эффекта проскальзывания.
Акселерометр	Не взаимодействует с	Используется только для

	окружающей средой.	получения ускорения. Нуждаются в калибровке.
Спутниковая навигация	Глобальная система, которая позволяет определять координаты любой точки планеты.	Малая точность измерений. Осложнение работы в закрытых помещениях
Стереоскопические системы	Возможность построения полноценной 3D карты Большое количество информации получаемое в процессе работы Не взаимодействует с окружающей средой	Сцена должна быть хорошо освещена, от этого зависит точность измерений Точность измерения сильно зависит от расстояния. Сложность алгоритмизации
Лазерные дальномеры	Построение 2D и 3D карт Отличная точность измерений. Хорошее угловое разрешение сканов. Быстрый сбор данных. Большая дальность измерения	Ошибочные вычисления при попадании луча на хорошо отражающую, зеркальную поверхность. Наиболее дорогой из всех видов сенсоров.
Ультразвуковые сонары	Недорогие системы. Приемлемые точность и скорость измерений Не чувствительны к условиям освещенности и световой отражающей способности среды. Интерференция.	Возможно построения только 2D карты. Фантомное эхо.

В таблице приведены численные параметры датчиков для построения карт местности.

Таблица 2. Сводная сравнительная таблица численных параметров датчиков.

Свойства	Тип датчика				
	Системы спутниковой навигации	Стереосистемы видеокамер	Ультразвуковой сонар	Лазерный дальномер	
				2D	3D
Тип	активный	пассивный	активный	активный	активный
Точность измерения расстояния	от 3-5м до 50м	10см	20см	10мм	5см
Диапазон измерения расстояния	—	1-50м	1-5м	0-80м	0-120м
Частота получения измерений	—	10-30Гц	100Гц	75Гц	15Гц
Возможность построения 3D карты	да	да	нет	нет	да

2.3. Выбор датчиков

Поскольку предполагается разработка универсальной системы автономного управления, то в качестве датчиков параметров движения могут быть использованы штатные устройства. При их отсутствии необходима установка комбинации датчиков определяющих все параметры (перемещение, скорость и ускорение), для этого может быть использована комбинация энкодеров и акселерометров, либо универсальные гироскопические датчики.

Система спутниковой навигации может использоваться только в качестве

дополнительного источника информации о положении робота вне помещений, поскольку не обеспечивает необходимую точность.

Использовать стереокамеры для навигации в нашей задаче также нецелесообразно, поскольку не обеспечивается необходимая точность и надежность измерений, так как рабочее окружение может иметь переменное освещение.

Из таблиц 1 и 2 следует, что лазерные дальномеры имеют наибольшие дальность измерений, скорость сбора информации и лучшую точность среди остальных датчиков, поэтому они выбраны для использования в качестве основного инструмента для навигации.

3. Создание виртуальной модели мобильного робота и рабочего окружения.

На сегодняшний день каждый желающий может приобрести себе робота для исследований по вполне приемлемой цене. Но все же даже сегодня, оборудование и комплектующие серьезных промышленных роботов являются достаточно дорогими. А если говорить о цене датчиков, например лазерного дальномера, то эти цифры могут быть очень высокими. Поэтому хорошее программное обеспечение, позволяющее визуализировать реальные робототехнические системы, позволяют исследователям заниматься решением задачи, а не поиском реального робота.

3.1. Средства визуализации роботов и рабочей среды.

В качестве средств визуализации робота и рабочей среды были использованы пакеты Gazebo и Rviz.

Пакеты Gazebo и Rviz обладают богатыми возможностями для создания виртуальных моделей роботов, описания их физических параметров, а также

позволяют симулировать данные различных датчиков робота.

3.2. Ограничения рабочей среды.

В качестве основного датчика был выбран 2D лазерный дальномер, что накладывает некоторые ограничения на рабочую среду робота, а именно:

- поверхность рабочей среды является плоскостью;
- луч лазера параллелен поверхности среды, датчик неподвижно закреплен на роботе;
- высота препятствий превосходят высоту, на которой находится лазерный дальномер;
- поверхности объектов среды не являются зеркальными и прозрачными поверхностями, гарантируется надежность измерения датчика.

Введенные ограничения позволяют свести нашу задачу, к задаче *автономной навигации на плоскости*. Способы решения новой задачи будут рассмотрены в следующих главах.

3.3. Виртуальная модель робота.

Была создана виртуальная модель 4-колесного робота с параметрами указанными в таблице 3.

Таблица 3. Физические параметры четырех колесного робота.

Габариты основания робота	(0.125 м; 0.3 м; 0.4 м)
Масса основания	7 кг
Радиус колеса	0.1 м
Ширина колеса	0.05 м
Масса колеса	0.5 кг
Координаты правого переднего колеса	(0.15; -0.175; -0.03125)

Координаты правого заднего колеса	(-0.15; -0.175; -0.03125)
Координаты левого переднего колеса	(0.15; 0.175; -0.03125)
Координаты левого заднего колеса	(-0.15; 0.175; -0.03125)
Координаты лазерного дальномера	(0.1; 0; 0.1125)

Таблица 4. Параметры лазерного дальномера.

Габариты датчика	(0.1 м; 0.1 м; 0.1 м)
Масса датчика	1e-5 кг
Минимальная дистанция измерения	0.10 м
Максимальная дистанция измерения	30.0 м
Разрешение	0.01 м
Число измерений за одно сканирование	720
Минимальный угол обзора	-1.570796
Максимальный угол обзора	1.570796
Угловое разрешение	1

Также в роботе установлен IMU датчик, для более точного просчета одометрии. На рисунке ниже представлена сама виртуальная модель робота.

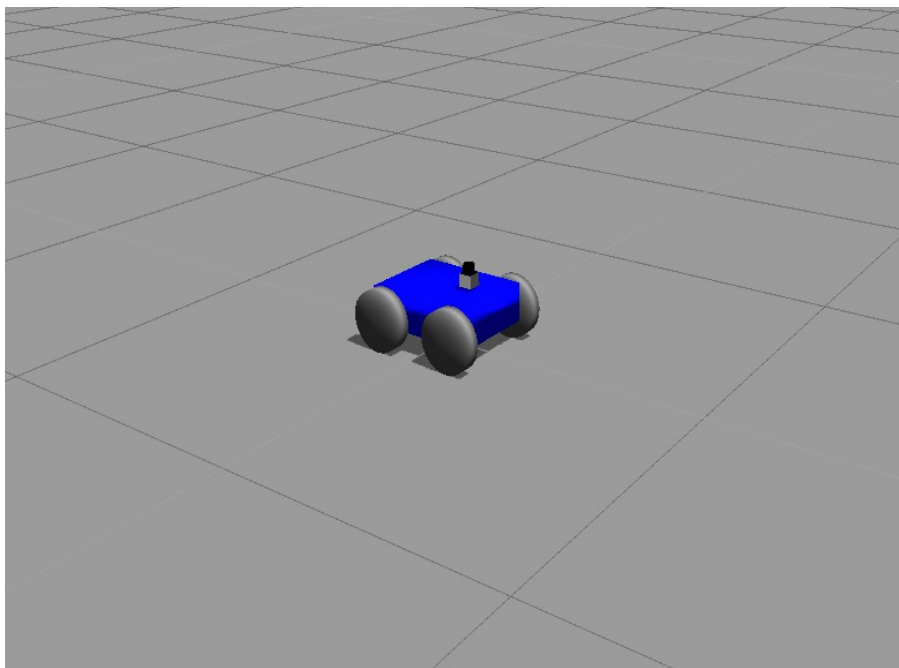


Рис 5. Виртуальная модель робота.

3.4. Виртуальная среда.

Ниже представлено изображение виртуальной среды и начальное положение робота на ней.

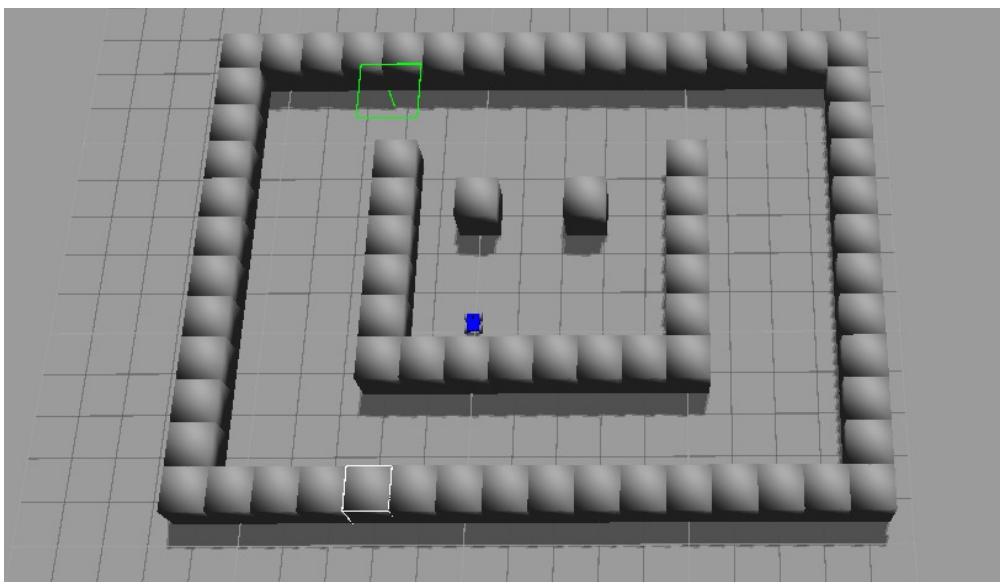


Рис 6. Виртуальная среда, вид сверху.

4. Программное обеспечение для разработки робототехнических систем.

Существуют библиотеки и программное обеспечение, помогающие разработчику сэкономить силы и время при разработке своей собственной робототехнической системы. При разработке программного обеспечения робототехнических систем возникают следующие трудности:

- Модульность системы. Приходится разбивать задачу на несколько подзадач, разрабатывать для каждой из них отдельный модуль, причем эти модули могут быть тесно связаны.
- Взаимодействие процессов. Часто множество модулей работают в отдельном процессе, и они могут активно взаимодействовать между собой. Поэтому приходится заботиться о синхронизации между отдельными процессами.
- Реализации алгоритмов. Сложные системы могут выполнять множество функций, например: определение координат ПА, создание карты помещения на основе данных датчиков, автономная навигация, планирование и т. д. Если реализация алгоритмов не является основной целью, то лучше воспользоваться готовыми, максимально совместимыми решениями.

В силу перечисленных выше причин, было решено использовать программное обеспечение ROS (Robot Operating System). В этой части работы рассмотрены основные принципы работы и устройство библиотеки.

4.1. Robotic Operating System.

ROS – это фреймворк для программирования роботов, предоставляющий функциональность для распределённой работы. ROS был первоначально разработан в 2007 году под названием *switchyard* в Лаборатории Искусственного Интеллекта Стэнфордского Университета. В 2008 году развитие продолжается в Willow Garage, научно-исследовательском институте/инкубаторе робототехники,

совместно с более чем двадцатью сотрудничающими институтами.

ROS обеспечивает стандартные службы операционной системы, такие как: аппаратную абстракцию, низкоуровневый контроль устройств, реализацию часто используемых функций, передачу сообщений между процессами, и управление пакетами. ROS основан на архитектуре графов, где обработка данных происходит в узлах, которые могут получать и передавать сообщения между собой. Библиотека ориентирована на Unix-подобные (Ubuntu) системы, однако сейчас активно разрабатывается поддержка платформ Windows и MacOS.

Основные достоинства ROS :

- архитектура операционной системы;
- набор поддерживаемых сообществом пакетов (ros-pkg);
- бесплатен для использования в коммерческих и исследовательских проектах.

4.2. Понимание файловой системы ROS. Концепция графов.

Основной единицей файловой системы является *пакет* (package). Каждый *пакет* содержит библиотеки, исполняемые файлы, скрипты, и т. д. Для каждого пакета существует свой *манифест* (manifest), в котором есть краткое описание пакета, зависимости между пакетами, и различная мета информация, такая как версия пакета, лицензия и т.д.

Введем несколько понятий:

- *Узел* (Node) — некоторый исполняемый процесс, который использует ROS для связи с другими *узлами*.
- *Сообщения* (Messages) – специальный тип данных ROS используемый *узлом* для публикации или подписки на *топики*.

- *Топики* (Topics) — узлы могут подписываться на *топик*, чтобы получать *сообщения*, или публиковать в *топик*, чтобы отправлять *сообщения*.
- *Мастер* (Master) — имя сервиса для ROS.
- `rosout` — эквивалент `stdout/stderr`.

В ROS существует множество инструментов, помогающих программисту в процессе разработки и обработки ошибок, можно легко проследить как взаимодействуют между собой различные процессы внутри ROS. Пакет Rviz является частью ROS, а Gazebo — ROS совместим.

Основная цель ROS – выделить из задачи несколько подзадач, максимально упростить код, сделать его понятным даже для человека, который близко не знаком с темой.

Рассмотрим в качестве примера пакет `tf`, который помогает избавиться от кода связанным с преобразованием координат. Типичный робот имеет множество 3D координатных систем, которые меняются с течением времени. `tf` продолжает наблюдать эти системы в течении времени, и позволяет определить в любой момент времени (от некоторого $t=0$ и до текущего момента времени), как расположена одна система координат относительно другой системы. Для этого узлу достаточно подписаться/публиковать в топик `tf` необходимые запросы. Можно например вывести все преобразования между системами координат в качестве графа.

5. Разработка поста управления мобильным роботом.

Для 4-колесного робота необходимо разработать модуль, позволяющий с помощью определенного оборудования (джойстик, клавиатура) передавать ПА скорость движения. В ограничениях нашей задачи достаточно транслировать мобильному роботу линейную и угловую скорости.

5.1. Создание Gazebo плагина для управления движением робота.

Был создан плагин позволяющий во время симуляции, читать из топика gazebo/cmd_vel скорость робота, назовем её *желаемой скоростью* и обозначим за w , и передавать эту скорость на колеса робота, таким образом достигая некоторой скорости v , которая должна быть максимально близка к w . Изначально было решено воспользоваться моделью:

r – радиус колес;

c – расстояние между центрами колес (между центрами правого переднего и левого переднего колес);

Тогда

$$V_{lin} = w_{lin}/r$$

$$V_{rot} = w_{ang} \cdot c/2r$$

и скорости вращения левого и правых колес можно записать как

$$V_{lin} - V_{rot}$$

$$V_{lin} + V_{rot}$$

соответственно. Эти скорости правых и левых колес должны приближать скорость v к w . Однако этого не удалось достичь. Пришлось изменить модель:

$$V_{lin} = v/r;$$

$$V_{rot} = (\pm b + w_{ang} \cdot k) \cdot c/2r$$

где знак «+» если

$$w_{ang} > 0$$

и «—» иначе. Формулы скоростей вращения колес остаются неизменными. Коэффициенты $b = 0.09433$ и $k = 1.106553$ вычислялись с помощью метода линейной регрессии, стандартные ошибки 0.007446 и 0.014231 соответственно, $p\text{-value} < 2.2e-16$, данные на которых проверялась гипотеза и скрипт на R, приложены к работе.

Затем был создан модуль, обрабатывающий нажатия кнопок джойстика, преобразующий их в линейную и угловую скорости и публикующий в /gazebo/cmd_vel.

В заключение стоит отметить, что разработанный плагин не только работает как контроллер колес, но и публикует такую информацию как

положение оси колеса(угол поворота, скорость вращения) и положение робота в виртуальной среде(ground truth pose).

5.2. Расчет одометрии мобильного робота.

Одометрия — использование данных о движении приводов, для оценки перемещения. Одометрия помогает получить некоторую оценку положения робота. Эта оценка затем обычно используется в алгоритмах СЛАМ, в качестве начальной оценки перемещения робота. Стандартной схемой одометрии робота, является использование энкодеров, считывающих угол поворота колес.

Пусть v_x v_y v_θ — скорость робота по x , y и θ соответственно, и

$$w_{left} \quad w_{right}$$

угловые скорости правого и левого колес соответственно. Тогда обозначим

$$v_{ang} = (w_{right} - w_{left}) \cdot r / c$$

$$v_{ang} = (w_{right} + w_{left}) \cdot r / 2$$

и получим, что скорости робота равны

$$v_x = v_{lin} \cdot \cos(\theta)$$

$$v_y = v_{lin} \cdot \sin(\theta)$$

$$v_\theta = v_{ang}$$

Таким образом, одометрия — недорогое средство, которое даёт хорошую кратковременную точность и позволяет реализовать большую частоту дискретизации.

Однако, необходимо учитывать, что при рассмотрении более длительных промежутков времени, накопление погрешностей приводит к увеличению ошибки ориентации, которая растёт пропорционально пути, пройденному роботом.

Источники погрешности:

- погрешность измерения радиуса колёс;
- различные размеры колёс (для роботов у которых больше одного колеса);
- ошибки подсчёта импульсов от энкодеров;
- низкая частота обработки одометрии.

В ходе экспериментов выяснилось, что данные одометрии недостаточно точны при высоких линейных скоростях робота и при повороте вокруг своей оси. Поэтому было решено ограничить скорость робота до 2 м/с, поставить на него IMU датчик для того, чтобы получать более точные данные о повороте робота и его ускорении. Данные одометрии и IMU датчика затем обрабатывались с помощью `robot_pose_ekf` (пакет ROS), использующий расширенный фильтр Калмана (`extended Kalman filter`) для получения окончательных данных одометрии, которые в дальнейшем будут использованы для определения положения ПА в пространстве.

6. Разработка системы автономной навигации.

После того, как был создан модули управления роботом и одометрии, можно приступить к созданию модуля навигации. В этой работе использованы возможности ROS, которая предлагает решение этой задачи с помощью пакета `move_base` ([19]).

6.1. move_base

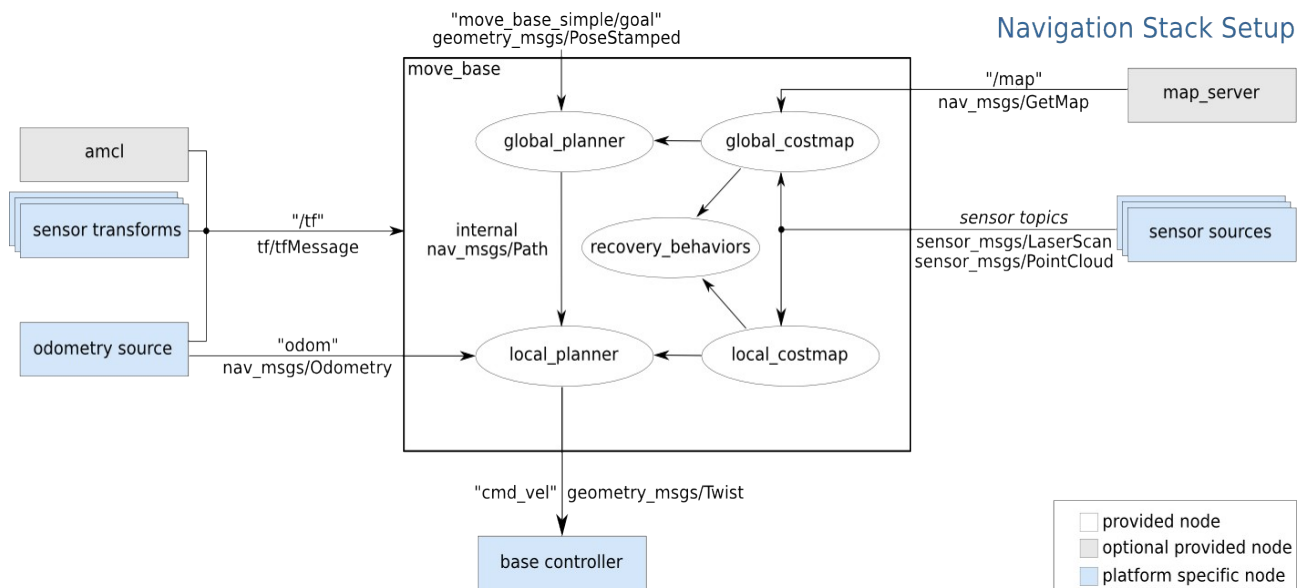


Рис 7. Схема конфигурации пакета `move_base`.

Рассмотрим компоненты этой схемы:

- Карта (`map_server`). Узел `map_server` подает на вход некоторую карту рабочей среды, эта карта будет использована для построения `global_costmap` – глобальной карты препятствий.
- Одометрия (`odometry source`). Необходимо, чтобы через `tf` модуль мог достать преобразование `odom` → `base_link`, где `base_link` — система координат связанная с роботом, а `odom` — система координат, в которой происходит движение ПА.
- Координаты мобильного робота (`amcl`, `sensor transforms`). Необходимо, чтобы с помощью некоторого алгоритма (например, адаптивного метода Монте-Карло — `amcl`, [29],[30]), который в через `tf` предоставляет группу последовательных преобразований координат: `map` → `odom` → `base_link`. В этой работе, чтобы уменьшить вычислительные затраты, был использован пакет `fake_localization`, который предоставляет те же преобразования в `tf`, что и `amcl`, но в отличие от последнего, он использует данные о положении робота на карте из симуляции.

- Данные датчиков (sensor sources), которые будут использованы для построения local_costmap.
- global_planner. Существует интерфейс nav_core::BaseGlobalPlanner, который используется планировщиками пути. Один из таких планировщиков из navfn. Он использует алгоритм Дейкстры для расчета минимального пути из начальной точки в конечную.
- local_planner. nav_core::BaseLocalPlanner предоставляет интерфейс используемый локальными планировщиками. base_local_planner — один из плагинов использующих этот интерфейс. Этот пакет предоставляет реализации методов TRA (Trajectory Rollout approach) и DWA(Dynamic Window approach) для локальной навигации робота на плоскости. Принимая на вход путь и карту препятствий (costmap), на выходе выдает скорость, которая затем передается контроллеру робота.

6.2. Поведение робота.

move_base Default Recovery Behaviors

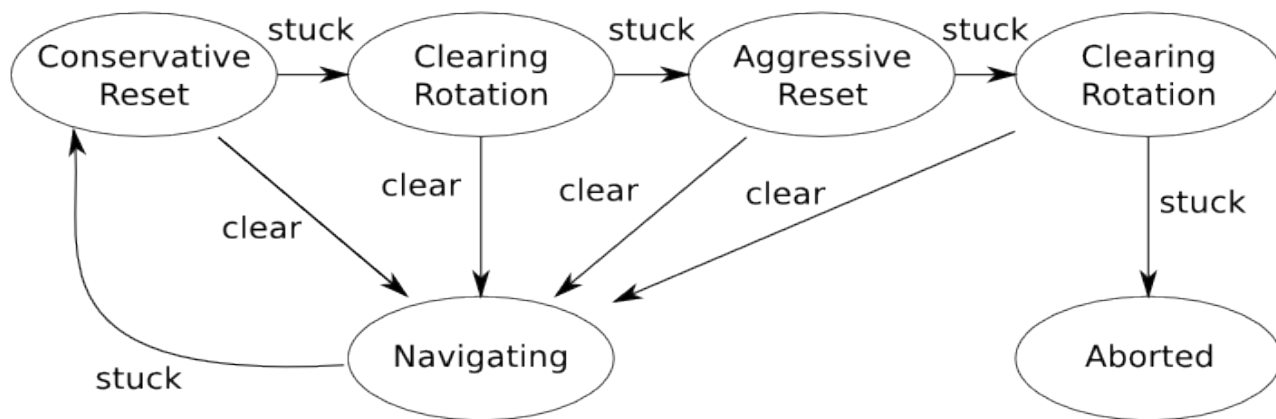


Рис 8. Схема поведения робота при восстановлении.

При вызове move_base, если робот правильно настроен, будет пытаться достичь цели, которую поставил ему пользователь в пределах некоторой погрешности, так же установленной пользователем. При отсутствии динамических препятствий, move_base в конечном итоге должен сообщить пользователю о том, что цель достигнута или же поставленная цель

недостижима. Если робот застревает, то модуль переходит в режим восстановления. По умолчанию, `move_base` выполнит следующие действия, чтобы восстановить движение робота:

1. препятствия за пределами заданной пользователем области будут удалены с карты робота; ,
2. если возможно, робот будет выполнять вращение на месте, чтобы очистить пространство перед ним;
3. если это тоже не удастся, робот полностью очистит область карты, в которой робот может вращаться;
4. после этого он продолжит свое вращение.

Если все это не поможет, цель будет считаться недостижимой и будет прервана. Работу режима восстановления можно настроить с помощью параметра `recovery_behaviors`, или отключить с помощью параметра `recovery_behavior_enabled`.

6.3. Создание карты.

С помощью пакета `gmapping` была создана карта рабочей среды. Этот пакет использует алгоритм Rao-Blackwellized Particle Filters([27], [28]). Затем эта карта загружается на `map_server` и используется для навигации.

Рис 9. Карта.

6.4. Демонстрационный модуль автономной навигации.

На основе всего выше сказанного был создан демонстрационный модуль автономного движения мобильного робота (`move_base_demo`). Программа работает следующим образом:

1. Запуск:

- запускается симуляция робота и виртуальной среды командой

```
$ roslaunch robot_gazebo demo.launch
```

- запускается Rviz

```
$ roslaunch robot_description 2dnav.launch
```

- запускается `move_base_demo`

```
$ roslaunch robot_2dnav move_base_demo.launch
```

2. Работа:

- через интерфейс Rviz можно задать цель робота;
- будет построен маршрут и робот начнет по нему движение;
- движение будет закончено когда робот достигнет цели, или решит, что цель недостижима, или робот застрянет.

6.5. Модуль аварийного управления в случае потери оператором связи с мобильным роботом.

Идея работы модуля очень проста. Пока оператор управляет роботом, через каждые l метров положение робота запоминается в специальной очереди. Если оператор теряет управление, в этой работе — это простое отключение

джойстика/клавиатуры, робот возьмет управление на себя, если не дожидется оператора в течении некоторого времени t , и в автономном режиме пройдет по тем точкам, которые на тот момент находились в очереди целей. Если одна из целей окажется не достижимой, робот попытается приблизиться к ней как можно ближе, а затем перейдет к следующей цели из очереди, если она окажется не пустой.

Были созданы две программы:

- `teleop_tracker` — следит за передвижением робота, и заполняет очередь целей, если необходимо, прослушивает состояние связи с оператором, и если не дожидается ответа в течении некоторого времени (в этой работе 4 секунды), то отправляет цель из очереди в `goal_controller`, если она не пуста. Если оператор вернул управление, то `teleop_tracker` посылает в `goal_controller` запрос на отмену выполнения текущей цели.
- `goal_controller` — получает цель от `teleop_tracker`, отправляет эту цель в `move_base`, и проверяет достигнута ли она, или нет, и сообщает о результате. Если контроллер получит запрос о отмене выполнения текущей цели, `goal_controller` отправит аналогичный запрос в `move_base`, и движение автономное движение робота прекратиться, управление вернется оператору.

Для запуска программы необходимо выполнить следующие команды:

```
$ roslaunch robot_gazebo willow_garage_teleop.launch
```

```
$ roslaunch robot_description 2dnav.launch
```

```
$ roslaunch robot_2dnav emergency_move_base.launch
```

7. Заключение.

В данной работе была показана актуальность проблемы автономной навигации мобильных роботов. Были продемонстрированы и изучены

современные типы датчиков используемые в робототехнических системах, проведен обзор современных решений и программного обеспечения.

Были созданы:

- виртуальные модели 4-колесного робота и рабочей среды;
- контроллер 4-колесного робота;
- модули управления роботом по средством джойстика/клавиатуры;
- демонстрационный модуль автономной навигации;
- модуль аварийного управления в случае потери оператором связи с мобильным роботом.

Созданный проект и его исходный код доступен на репозитории <https://github.com/denzist/robot-navigation>.

В дальнейших планах работы по проекту — усовершенствование модуля `move_base`, создание собственного, плагина `local_controller`, тестирование системы на реальном прототипе робота.

Список литературы:

- 1) Bruno Siciliano, Oussama Khatib. Springer Handbook of Robotics. Springer-Verlag Berlin Heidelberg. 2008. 1611 с.
- 2) Све Лин Хту Аунг. Навигация и управление движением мобильного робота в городских условиях: диссертация кандидата технических наук: 50.02.05, Москва, 2011.
- 3) Минин А. А. Навигация и управление мобильным роботом, оснащенным лазерным дальномером: диссертация кандидата технических наук: 50.02.05, Москва, 2008.
- 4) Antoni Burguera, Yolanda Gonzalez and Gabriel Oliver. Sonar Sensor Models and Their Application to Mobile Robot Localization. Sensors (Basel). 2009; 9(12): 10217–10243.
- 5) Teruko YATA, Akihisa OHYA, Shinichi YUTA. A Fast and Accurate Sonar-Ring Sensor for a Mobile Robot. Proceedings of the 1999 IEEE International Conference on Robotics and Automation, May, 1999, pp. 630-636.
- 6) Elfes A. A sonar-based mapping and navigation system. Robotics and Automation. Proceedings. 1986 IEEE International Conference on (Volume:3). Apr 1986. pp. 1151 — 1156.
- 7) Heale A., Kleeman L. Fast target classification using sonar. Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on (Volume:3). 2001. 1446 - 1451 vol.3.
- 8) Lindsay Kleeman, Roman Kuc. Mobile Robot Sonar for Target Localization and Classification. The International Journal of Robotics Research (Impact Factor: 2.86). 01/1995; 14:295-318.
- 9) Kiyoshi Okuda, Masamichi Miyake, Hiroyuki Takai, Keihachiro Tachibana. Obstacle arrangement detection using multichannel ultrasonic sonar for indoor mobile robots. Artificial Life and Robotics. September 2010, Volume 15, Issue 2, pp 229-233.
- 10) M. A. Martinez and J. L. Martinez. The dual-frequency sonar system of the

mobile robot RAM-2. Journal Robotica Volume 22 Issue 3, June 2004. pp. 263 - 270

11) Maryum F. Ahmed. Development of a stereo vision system for outdoor mobile Robots. Abstract of Thesis Presented to the Graduate School of the University of Florida in Partial Fulfillment of the Requirements for the Degree of Master of Science. 2006. pp 78.

12) Kok Seng Chong , Lindsay Kleeman, Mobile Robot Map Building from an Advanced Sonar Array and Accurate Odometry. 1996.

13) http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/jmd/

14) Sebastian Thrun and others. Stanley: The Robot that Won the DARPA Grand Challenge. Journal of Field Robotics 23(9), 661–692 (2006)

15) http://www.maxbotix.com/Ultrasonic_Sensors/MB1403.htm

16) <http://www.sick.com>

17) Scanning Laser Range Finder UTM-30LX/LN Specification

18) <http://www.fastwel.ru/products/356046/epic/432846.html>

19) http://wiki.ros.org/move_base

20) http://wiki.ros.org/costmap_2d

21) http://wiki.ros.org/nav_core

22) http://wiki.ros.org/base_local_planner

23) <http://wiki.ros.org/navfn>

24) Brian P. Gerkey and Kurt Konolige. "Planning and Control in Unstructured Terrain ". Discussion of the Trajectory Rollout algorithm in use on the LAGR robot. (2008)

25) D. Fox, W. Burgard, and S. Thrun. "The dynamic window approach to collision avoidance". The Dynamic Window Approach to local control. Robotics & Automation Magazine, IEEE, 1997, pp 23 – 33

26) Alonzo Kelly. "An Intelligent Predictive Controller for Autonomous Vehicles". An previous system that takes a similar approach to control, tech. report CMU-RI-TR-94-20, Robotics Institute, Carnegie Mellon University, May, 1994

27) Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard: Improved

Techniques for Grid Mapping with Rao-Blackwellized Particle Filters, IEEE Transactions on Robotics, Volume 23, pages 34-46, 2007

28)Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard: Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling, In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2005

29)<http://wiki.ros.org/amcl>

30)Probabilistic Robotics, by Thrun, Burgard, and Fox.

31)http://wiki.ros.org/robot_pose_ekf