Глава 3. Практика

3.1 Мобильный автономный робот

Для решения задачи данной работы необходимо было проводить «живые» тестирования работы алгоритмов. Такая необходимость обусловлена прежде всего тем, что помимо существующей задачи данной ВКР стояла задача в создании робота для распознавания объектов. По этой причине было принято решение делать алгоритмы на реальном роботе[1] с пребыванием данного робота во вполне реальных условиях.

3.1.1 Подбор шасси

Как было сказано в первой главе данной работы, существует большое количество различных шасси, на которых можно располагать различное оборудование. Наш выбор остановился на гусеничном шасси, которое изображено на Рисунке .

Данное шасси за счёт своих размеров является очень мобильным средством передвижения робота и может проникнуть в относительно узкие для роботов пространства и без проблем оттуда выбраться, не повредившись. Для оборудования на данном шасси место тоже нашлось: для этого было принято решение заказать металлическую пластину, которая играла роль второго этажа. поставить рисунок?

[1]Задание данной ВКР можно было бы сделать и в любом симуляторе или игровом движке. Однако решение делать всё в реальной жизни сильно усложнило данную задачу.

3.1.2 Движение шасси

К сожалению, по неизвестной причине к данному шасси не пришёл комплектный контроллер движения, который бы принимал команды от компьютера и заставлял двигаться установленные гусеницы. По этой причине пришлось немного изучить ещё одну предметную область, которая не изучалась в течении университетского курса - электротехнику.

Требования к контроллеру движений

Компьютер, который будет в последствии установлен на робота будет управлять роботом посредством бинарных сигналов с напряжением 3.3В через порт GPIO, где 0 (или по-другому нет напряжения) - это движение не требуется и 1 (когда есть напряжение +3.3В), когда движение требуется.

Контроллер должен, также, уметь по отдельности управлять двумя гусеницами, заставлять их ездить вперёд и назад. Это основные требования. Из дополнительных требований можно выделить умение каким-то образом регулировать скорость движения. Общая структура желаемой модели контроллера изображена на Рисунке .

Схема будущего контроллера

Здесь теоретическое объяснение как работает контроллер

Производство контроллера

В качестве основы для контроллера было принято решение взять текстолитовую пластину с медным покрытием, изображённую на рисунке .

Далее на нём при помощи перманентного маркера были нанесены дорожки, а места вставки деталей были просверлены советской стоматологической бур-машинкой. Далее всё это прошло ванну хлорида железа. Процесс и результаты работы изображены на Рисунке .

Результаты работы

Таким образом был получен полноценный контроллер, который умеет управлять роботом медленной и быстрой скоростями. Однако, не обошлось без недостатков и трудностей. Главной трудностью стал неудачный способ стравливания дорожек на плате, поэтому пришлось дополнительно при помощи олова проводить дорожки и искать потенциально уязвимые места.

После большой ручной работы плату удалось запустить, но появилась другая проблема. И эта проблема заключалась в том, что сопротивление двигателей робота немного отличалось друг от друга, также как и сопротивление резисторов, установленных на плату. Данное отличие было ничтожным, но этого вполне хватало, чтобы робот начинал ездить не очень ровно и его всё время приходилось дополнительно корректировать.

Вторая версия контроллера

Необходимость во второй версии контроллера возникла в первую очередь из-за того, что медленная скорость на роботе работала слишком медленно, а быстрая слишком быстро. Ездить в магазин и вручную подбирать резисторы, каждый раз перепаивая плату совсем не хотелось, а каждый резистор в местных магазинах обходится в неприятную для таких расходных деталей сумму.

Переделка первой версии платы обошлась бы слишком дорого в плане времени, да и к тому же не хотелось портить, то, что и так уже работает. Поэтому вторая версия платы была собрана по той же схеме, что и первая, но с несколькими важными отличиями. Из стабилизаторов остался только ста-

билизатор на 9В, а обычные резисторы были заменены на соответствующие подстроечные резисторы. Таким образом должна была появилась возможность регулировать движение в зависимости от того, в каком положении будет установлен сам резистор. Получившаяся плата изображена на Рисунке .

Однако, что-то пошло не так и настройка резисторов превратилась в процесс настройки «лишь бы работало». Всё дело в том, что на плате образовался какая-то токоутечка (к сожалению, я в плане электротехники и радиофизики являюсь дилетантом, поэтому не могу подобрать настоящего термина к данному процессу) и существовало много вариантов настроить подстроечные резисторы так, что, например, при команде ехать быстро вперёд одна гусеница ехала нормально, а вторая начинала резко дёргаться или вести себя каким-либо случайным и неестественным образом (бывало даже так, что одна гусеница вдруг начинала ехать назад, хотя этому ничего не предвещало).

И всё же существовала такая настройка резисторов, при которой робот начинал нормально слушаться команд, но затем научный консультант предоставил нам ещё одно шасси, на котором затем и продолжилась разработка окончательной версии робота.

3.2 Визуальный анализ пространства

Для анализа окружающего пространства существует довольно большое количество различных датчиков и прочего оборудования. Но закупать сразу всё не выгодно экономически, затратно в плане места размещения на роботе и расточительно в плане потребления электроэнергии этими самыми датчиками. Также для обработки всех этих сигналов нужны соответствующие вычислительные мощности.

Таким образом робот должен иметь совсем небольшое количество сенсоров и при этом не быть «слепым». Исходя из этих соображений, было решено установить на робота два основных сенсора. Лазерный сканер YDLIDAR X4 и CSI камера Sony IMX217. Первый поможет видеть препят-

ствия вокруг робота, второй поможет видеть объекты, размещённые перед роботом.

Схему размещения сенсоров размещена на Рисунке .

3.3 Формирование поведенческой стратегии робота

Основная задача робота - ездить и искать целевые объекты делится на две подзадачи.

3.3.1 Езда

Этот режим поездки можно также назвать режимом исследователя. В идеальном случае, робот в начале работы алгоритма движения должен объездить всё доступное ему пространство, составить карту окружающей местности, параллельно при этом запоминая увиденные им объекты и их примерное местоположение[2], а затем после полного сканирования объехать всю территорию и убедиться в том, что целевые объекты действительно были найдены на этих местах. И это должно стать конечной целью робота. Однако, для упрощения данной задачи будет использоваться другой куда более простой алгоритм.

Режим исследователя будет подразумевать под собой то, что робот будет просто ехать вперёд, параллельно разыскивая целевые объекты и объезжая препятствия.

[2]его нужно вычислять из расчётов того, куда смотрит в данный момент робот, угла прямоугольника на изображении, на котором объект был обнаружен

Объезд препятствий

Робот должен уметь объезжать хотя-бы самые простейшие препятствия, по типу стен, диванов или прочих перегородок. В идеале, он должен уметь справляться и с тонкими препятствиями по типу ножек стула и мягкие поверхности.

Алгоритм объезда препятствий, представленный на данном роботе сводится к схеме, изображённой на Рисунке .

Подсчёт того, где свободнее: слева или справа идёт из соображений того, где находится больше препятствий. Как это считается? Условно робот поделён на несколько направлений. В данном случае он подразделён на «перед», «лево», «право» и «зад». Лидар выдаёт данные в формате обычных чисел в формате float. Всего этих чисел 720, из чего можно сделать вывод, что цена деления лидара это полградуса. По каждой из сторон считается обычное среднее арифметическое число. Как только это число достигает некоторого предела, заданного константой, робот начинает искать путь обхода. Вариантов обойти препятствия всего два: поехать налево или направо. Считается среднее арифметическое число с левой стороны и с правой. Затем выбирается та сторона, у которой это число оказалось меньше. Однако, этого всё ещё недостаточно чтобы объезжать часто встречаемые препятствия.

3.3.2 Поиск целевого объекта

3.4 Программная часть

Здесь пишем об ОС, установленном на робота. О ROS и о собственных программных реализациях.

3.5 Ещё одно шасси

Как и было сказано ранее, от научного консультанта по ВКР было получено шасси другого образца, изображённое на рисунке .

Данное шасси по своей длине гораздо больше, менее мобильно. Это пришлось учитывать при переделке алгоритма движения робота. Однако, пришлось переделать и ещё кое-что.

3.5.1 Делегирование задач

Так как на выданном роботе уже установлен и подключен одноплатный компьютер OrangePI PC было принято решение делегировать на него задачу передвижения робота, а конкретно в обязанности этого компьютера будут входить:

1. Управление лидаром и обработка сигналов от него;
2. Постройка карты окружающей местности и определение внутренних координат на ней;
3. Непосредственное управление электродвигателями робота через разъём GPIO;
4. Управление роботом в режиме исследователя и при нахождении целевого объекта.

3.5.2 Объединение 2 компьютеров

Для выполнения выше поставленных задач компьютер Orange PI PC должен быть напрямую связан со вторым компьютером NVIDIA Jetson NANO.

Локальная сеть

Было принято решение физически связать эти два компьютера при помощи имеющихся на них разъёмах для Ethernet. Программно два компьютера были связаны при помощи встроенной в ОС Ubuntu (установленной на Jetson) функции предоставления сети. Таким образом, подключенный к беспроводной сети Wi-Fi и к проводу Jetson NANO, может не только выходить в интернет, а ещё и предоставлять этот интернет компьютеру Orange PI PC, у которого встроенного Wi-Fi модуля нет.

В полевых условиях (в которых в последствии робот должен будет оказаться), естественно никакой беспроводной сети с интернетом не будет. Однако, это удобно для разработки и отладки. А в случае, если понадобится подключить мобильный интернет, это можно будет также сделать через Jetson NANO.

Объединение ROS

После того, как два компьютера соединились друг с другом, можно начать пересылку информации. Для этих целей можно написать ряд отдельных клиент-серверных приложений, которые при помощи сокетов будут передавать различные данные. Например, Jetson NANO будет передавать на Orange PI PC координаты объекта, которые он видит в кадре.

Оказалось, что писать для этого отдельные приложения не требуется. Объединение нескольких компьютеров возможно и при помощи встроенных в Robot Operating System штатных средств.

При объединении компьютеров в начале нужно выяснить кто из компьютеров будет главным ядром системы (в последствии на этом компьютере будет запущен мастер-узел roscore). Было решено, что ядром станет Jetson NANO, так как он имеет более высокие вычислительные мощности.

Зависимый от главного компьютер (в нашем случае Orange PI PC) при этом должен хранить в своём рабочем окружении две ключевых переменных ROS_MASTER_URI и ROS_IP. Первая задаётся в

виде: http://ip-адрес-компьютера-с-roscore:11311/. Во второй переменной просто должен содержаться собственный IP адрес компьютера. На Orange PI PC эти переменные установлены в этих значениях: ROS_MASTER_URI=http://10.42.0.1:11311/ ROS_IP=10.42.0.240

После объявления этих переменных, между компьютерами создаётся двухсторонняя связь, в которой оба компьютера видят друг у друга все публикующиеся сообщения и могут присоединяться к различным сервисам. Схема соединения компьютеров представлена на Рисунке .

3.5.3 Лазерное сканирование

Следующая проблема возникает в связи с конструкцией, которая установлена на робота. В середине корпуса имеется непрозрачное препятствие, которое мешает лазерному сканированию при помощи LIDAR. Поэтому было принято решение установить второй LIDAR с другой стороны, а затем в алгоритме движения робота учесть этот момент.

Здесь расписать как решена задача объединения лидаров...

3.6 Таблица обыкновенная

Так размещается таблица:

Таблица 7 — Название таблицы

Месяц	T_{min}, К	T_{max}, К	$(T_{max} - T_{min})$, К
Декабрь	253.575	257.778	4.203
Январь	262.431	263.214	0.783
Февраль	261.184	260.381	−0.803

Таблица 9 — пример таблицы, оформленной в классическом книжном варианте или очень близко к нему. ГОСТу по сути не противоречит. Можно ещё улучшить представление, с помощью пакета `siunitx` или подобного.

Таблица 8

Оконная функция	$2N$	$4N$	$8N$
Прямоугольное	8.72	8.77	8.77
Ханна	7.96	7.93	7.93
Хэмминга	8.72	8.77	8.77
Блэкмана	8.72	8.77	8.77

Таблица 9 — Наименование таблицы, очень длинное наименование таблицы, чтобы посмотреть как оно будет располагаться на нескольких строках и переноситься

Оконная функция	$2N$	$4N$	$8N$
Прямоугольное	8.72	8.77	8.77
Ханна	7.96	7.93	7.93
Хэмминга	8.72	8.77	8.77
Блэкмана	8.72	8.77	8.77

3.7 Таблица с многострочными ячейками и примечанием

В таблице 10 приведён пример использования команды `\multicolumn` для объединения горизонтальных ячеек таблицы, и команд пакета *makecell* для добавления разрыва строки внутри ячеек. При форматировании таблицы 10 использован стиль подписей `split`. Глобально этот стиль может быть включён в файле `Dissertation/setup.tex` для диссертации и в файле `Synopsis/setup.tex` для автореферата. Однако такое оформление не соответствует ГОСТ.

Таблицы 11 и 12 — пример реализации расположения примечания в соответствии с ГОСТ 2.105. Каждый вариант со своими достоинствами и недостатками. Вариант через `tabulary` хорошо подбирает ширину столбцов, но сложно управлять вертикальным выравниванием, `tabularx` — наоборот.

Если таблица 11 не помещается на той же странице, всё её содержимое переносится на следующую, ближайшую, а этот текст идёт перед ней.

Таблица 10

Пример использования функций пакета *makecell*

<table>
<tr><th>Колонка 1</th><th>Колонка 2</th><th>Название колонки 3,
не помещающееся в одну строку</th><th>Колонка 4</th></tr>
<tr><td colspan="4">Выравнивание по центру</td></tr>
<tr><td colspan="2">Выравнивание
к правому краю</td><td colspan="2">Выравнивание к левому краю</td></tr>
<tr><td rowspan="2">В этой ячейке
много информации
А в этой мало</td><td rowspan="2">8.72

8.22</td><td>8.55</td><td>8.44</td></tr>
<tr><td colspan="2">5</td></tr>
</table>

3.8 Таблицы с форматированными числами

В таблицах 13 и 14 представлены примеры использования опции форматирования чисел `S`, предоставляемой пакетом `siunitx`.

3.9 Параграф — два

Некоторый текст.

3.10 Параграф с подпараграфами

3.10.1 Подпараграф — один

Некоторый текст.

3.10.2 Подпараграф – два

Некоторый текст.

Таблица 11 — Нэ про натюм фюйзчыт квюальизквюэ

доминг лаборамюз эи ыам (Общий съём цен шляп (юфть))	Шеф взъярён	адвыр-жаряюм	тебик-вюэ эльъэеф-энд мэдио-крета-тым	Чэнзэ-рет мны-жарк-хюм
Эй, жлоб! Где туз? Прячь юных съёмщиц в шкаф Плюш изъят. Бьём чуждый цен хвощ!	≈	≈	≈	+
Эх, чужак! Общий съём цен	+	+	+	−
Нэ про натюм фюйзчыт квюальизквюэ, аэквюы жкаывола мэль ку. Ад граэкйж плььатонэм адвыржаряюм квуй, вим емпыдит коммюны ат, ат шэа одео	≈	−	−	−
Любя, съешь щипцы, — вздохнёт мэр, — кайф жгуч.	−	+	+	≈
Нэ про натюм фюйзчыт квюальизквюэ, аэквюы жкаывола мэль ку. Ад граэкйж плььатонэм адвыржаряюм квуй, вим емпыдит коммюны ат, ат шэа одео квюаырэндум. Вёртюты ажжынтиор эффикеэнди эож нэ.	+	−	≈	−

Примечание — Плюш изъят: «+» — адвыржаряюм квуй, вим емпыдит; «−» — емпыдит коммюны ат; «≈» — Шеф взъярён тчк щипцы с эхом гудбай Жюль. Эй, жлоб! Где туз? Прячь юных съёмщиц в шкаф. Экс-граф?

Таблица 12 — Любя, съешь щипцы, — вздохнёт мэр, — кайф жгуч

доминг лаборамюз эи ыам (Общий съём цен шляп (юфть))	Шеф взъярён	адвыр-жаряюм	тебиквюэ эльъэеф-энд мэдио-крета-тым	Чэнзэрет мны-жарк-хюм
Эй, жлоб! Где туз? Прячь юных съёмщиц в шкаф Плюш изъят. Бьём чуждый цен хвощ!	≈	≈	≈	+
Эх, чужак! Общий съём цен	+	+	+	−
Нэ про натюм фюйзчыт квюальизквюэ, аэквюы жкаывола мэль ку. Ад граэкйж плльатонэм адвыржаряюм квуй, вим емпыдит коммюны ат, ат шэа одео	≈	−	−	−
Любя, съешь щипцы, — вздохнёт мэр, — кайф жгуч.	−	+	+	≈
Нэ про натюм фюйзчыт квюальизквюэ, аэквюы жкаывола мэль ку. Ад граэкйж плльатонэм адвыржаряюм квуй, вим емпыдит коммюны ат, ат шэа одео квюаырэндум. Вёртюты ажжынтиор эффикеэнди эож нэ.	+	−	≈	−

Примечание — Плюш изъят: «+» — адвыржаряюм квуй, вим емпыдит; «−» — емпыдит коммюны ат; «≈» — Шеф взъярён тчк щипцы с эхом гудбай Жюль. Эй, жлоб! Где туз? Прячь юных съёмщиц в шкаф. Экс-граф?

Таблица 13 — Выравнивание столбцов

Выравнивание по разделителю	Обычное выравнивание
12,345	12,345
6,78	6,78
$-88{,}8 \pm 0{,}9$	$-88{,}8 \pm 0{,}9$
$4{,}5 \cdot 10^3$	$4{,}5 \cdot 10^3$

Таблица 14 — Выравнивание с использованием опции `S`

Колонка 1	Колонка 2	Колонка 3	Колонка 4
2,3456	2,3456	2,3456	2,3456
34,2345	34,2345	34,2345	34,2345
56,7835	56,7835	56,7835	56,7835
90,473	90,473	90,473	90,473