Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Системы обработки информации и управления» (ИУ5)

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

Интеграция больших языковых моделей для автономной навигации мобильного робота

Студент ИУ5-34М _____ **А.Н. Кравцов**
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель _____ **Ю.Е. Гапанюк**
(Подпись, дата) (И.О.Фамилия)

2024 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУ5
(Индекс)
_____ В.И. Терехов
(И.О.Фамилия)
« _____» _____2024 г.

З А Д А Н И Е

на выполнение научно-исследовательской работы

по теме Диалоговая генерация архитектурного дизайна зданий
с использованием базы знаний

Студент группы ИУ5-34М
Кравцов Андрей Николаевич
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)
Исследовательская

Источник тематики (кафедра, предприятие, НИР) Кафедра

График выполнения НИР: 25% к ___нед., 50% к ___нед., 75% к __нед., 100% к ___нед.

Техническое задание
Составить описание предметной области
Проанализировать существующие системы-аналоги
Составить обзор интеллектуальных систем и методов представления знаний

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на _____листах формата А4.
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания « 9 » сентября 2024 г.

Руководитель НИР _____ Ю.Е. Гапанюк
(Подпись, дата) (И.О.Фамилия)

Студент _____ **А.Н. Кравцов**
(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре

Оглавление

Введение

Развитие робототехники и искусственного интеллекта открывает новые горизонты в области автономного управления мобильными роботами. В последнее время активно развивается направление интеграции больших языковых моделей (Large Language Models, LLM) в системы управления роботами. Внедрение LLM позволяет добавить больше интеллектуальности роботехническим системам за счет понимания контекста и окружающей среды.

В настоящее время для навигации мобильных роботов обычно используются методы Simultaneous Localization and Mapping (SLAM) [1]. Эти методы позволяют роботу одновременно строить карту местности и определять свое положение ней в реальном времени, однако навигация частично ограничена в те области, которые не картографированы, таким образом, навигация робота в неисследованные области, пока карта местности не построена окончательно, крайне затруднена. Существуют разные методы SLAM в зависимости от входных данных, отличают инерциальный способ, который может использовать данные с датчиков угла поворота колес, визуальный способ, основанный на изображении с камеры и способ, основанный на данных лазерного сканирования пространства. Для локализации и картографирования в SLAM обычно используют графовые методы и фильтр частиц, а для планирования маршрута такие алгоритмы как алгоритм Дейкстры и A* или специально обученные нейросети, например, на основе Long Short-Term Memory (LSTM) [2].

Система управления роботом, основанная на SLAM может одновременно выполнять две задачи: картографирование местности и навигация. Такой подход отлично справляется с навигацией, но требует больших вычислительных мощностей и сильно зависит от качества входных сенсорных данных. Из-за того, что SLAM основан на алгоритмическом подходе, система управления, построенная на его основе, не способна к контекстной интерпретации и обобщению окружающей среды.

Обзор использования LLM при навигации и планировании

Современные задачи робототехники требуют создания таких интеллектуальных систем, которые способны агрегировать поступающую информацию со всех установленных сенсоров и датчиков, для более эффективного принятия решений за счет интерпретации через эти данные окружающего пространства, а также систем способных корректировать решения, адаптируясь к динамической среде. Сам способ взаимодействия с системой управления робота может перейти на более высокий уровень и быть упрощен путем замены сложных структурированных команд на команды на естественном языке, которые будут рассмотрены в контексте всего диалога[3,4]. При таком подходе знания о мире, которые уже изначально заложены в LLM, могут помочь избавиться от излишней дискретизации команд, модель сможет сама разбить задачу по подзадачи и достигнуть цели, как это было продемонстрировано в исследовании “Language Models as Zero-Shot Planners” [5]. В этой статье на вход предобученной модели передается запрос на выполнение задачи, а на выходе получают подробный пошаговый план действий для агента, который уточняется при последующих запросах.

Другой возможный подход - использовать заранее определенные шаблонные инструкции и с помощью модели выделять их из запроса и создавать на их основе план. Такой метод использовали авторы модели FiLM [6].

Еще одна статья описывает метод SayCan (Subtask Evaluation Mode) [7] - итоговое действие принимается на основе взвешенных решений между выполнимым действиями и действиями, которые наиболее вероятно приведут к решению задачи.

На текущий момент в области интеграции LLM в робототехнику ведутся активные исследования [8,9]. Такие компании как OpenAI и Google, разрабатывают модели, которые могут обрабатывать комбинированные данные, такие как: текст, изображения и другие сенсорные данные. Например, в статье [10] описывается применение мультимодальной модели PaLM-E (An Embodied Multimodal Language Model), которая интерпретирует окружающую среду по изображению и планирует стратегию поведения робота.

Важную роль при внедрении мултимодальных моделей играет качество Vision Model. Для задач робототехники в настоящее время активно развиваются специализированные vision models. Например, одной из таких специализированных мултимодальных моделей является DeepSeek-VL[11], целью обучения которой было повысить качество визуального распознавания и embodied intelligence в сложных сценариях.

Однако, несмотря на значительные достижения в области чат-ботов и моделей, анализирующих изображения, остаются области для улучшения. Одной из таких областей является задача улучшения однородности ответов и надежности интерпретации данных в условиях высокой неопределенности. Для преодоления проблемы “галлюцинаций” некоторые исследователи предлагают использовать миварный подход [12, 13].

Таким образом, внедрение LLM в системы управления позволяет расширить область применения мобильных роботов, за счет повышения их интеллектуальности и преодолеть ограничения традиционных методов.

Методология

Архитектура системы***:***

Разработанная архитектура системы управления роботом включает следующие ключевые компоненты:

1) Аппаратная конфигурация:

a) Платформа: двухколесный робот с дифференциальным приводом и пассивной опорой[9]

b) Сенсоры: 2D LiDAR (360° сканирование), Камера RGB, энкодеры угла поворота колес.

2) Программное обеспечение:

a) Операционная система: Ubuntu 20.04 with ROS Noetic

b) Симулятор: Gazebo

c) Язык программирования: Python 3.8

d) Библиотеки: OpenCV, NumPy, rospy

На рисунке 1 представлена диаграмма развертывания системы управления роботом.

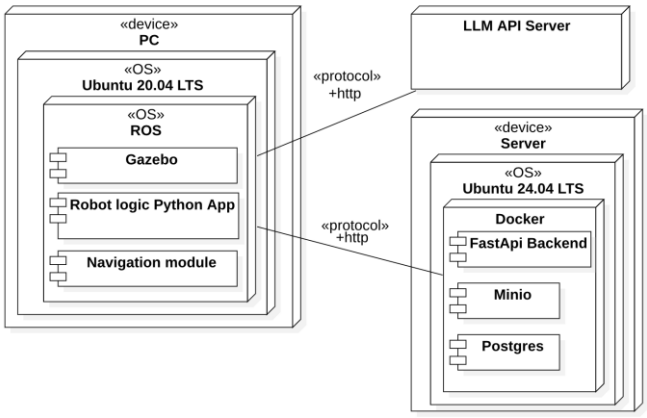

Рисунок 1. Диаграмма развертывания

Для взаимодействия API LLM и ROS с симулятором Gazebo был разработан backend на фреймворке FastApi и программа на Python. Backend отвечает за хранение в БД PostgresSQL прошлых команд LLM и остальной телеметрии робота на маршруте. Для взаимодействия с API LLM необходимо формировать доступную из сети ссылку на изображение, для этого на backend используется объектное хранилище Minio. Программа на Python является центром всей логики работы системы управления. Именно она отвечает за общую коммуникацию всех компонентов системы, а именно: получает из симулятора изображение с камеры и данные LiDAR, обрабатывает их, получает с бэкенда прошлые команды, отправляет запросы в API LLM, получает ответ и выделяет из него команду, добавляет эту команду в контекст, после чего преобразует ее в нужный формат и передает в топики ROS для ее дальнейшего исполнения в симуляторе.

Система получает три типа входных данных:

1) Визуальные данные:
   - *a)* RGB изображение с камеры.
   - *b)* Разрешение: 640x480 пикселей.
   - *c)* Частота кадров: 30 FPS.
   - *d)* Маркировка: наложенная линейка для шкалирования.
   - *e)* Цветовое пространство: RGB.

2) LiDAR сканирование
   - *a)* Угол сканирования: 360°
   - *b)* Точность измерения: ±1 см
   - *c)* Максимальная дальность: 10 м

d) Формат: массив расстояний

1) Контекстная инструкция: В качестве промта в LLM передается сообщение из фото и текста на русском языке, который состоит из нескольких частей. Сначала передается базовая инструкция роботу, в котором описана конфигурация и размеры робота, основная задача, пошаговая инструкция принятия решений о движении, в которой комбинируются такие техники Promt Engineering как:

a) Chain/Tree of Thought,
b) Few-Shot Promting,
c) Self-Consistency,
d) Role/Emotional Promting,
e) Reflexion,
f) Structured Promting.

Далее передаются прошлые команды, вместе с интерпретацией пространства, после чего описан формат команды ответа, который должна выдавать LLM.

Алгоритм управления роботом.

На рисунке 2 представлена блок-схема алгоритма управления роботом.

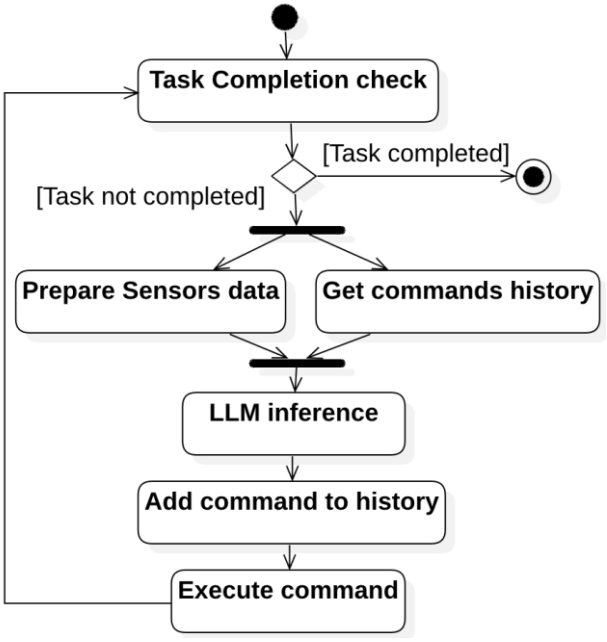

Рисунок 2. Алгоритм управления роботом

Разработанный алгоритм навигации представляет собой итеративный цикл обработки сенсорных данных и принятия решений с использованием больших

языковых моделей (LLM). Процесс навигации включает следующие ключевые этапы:

1) Сбор визуальных данных с камеры: Главным источником информации служит изображение с камеры робота. Для улучшения восприятия пространственных характеристик на исходное изображение накладывается радиальная линейка с тремя зонами безопасности, позволяющая точнее оценивать положение робота.

2)Предобработка данных LiDAR: Данные двухмерного LiDAR сканирования преобразуются в изображение, имитирующее вид сканирования сверху. На этом изображении дополнительно отмечается:

a) Текущая позиция робота
b) Контуры робота и его габариты
c) Область обзора камеры

3)Обработка сенсорных данных: Изображения с камеры и LiDAR объединяются в единое изображения, таким образом предоставляя более полное представление об окружающем пространстве. Такой подход позволяет LLM получать максимально информативный контекст для принятия решений. На рисунке 3 представлен пример объединённого изображения.

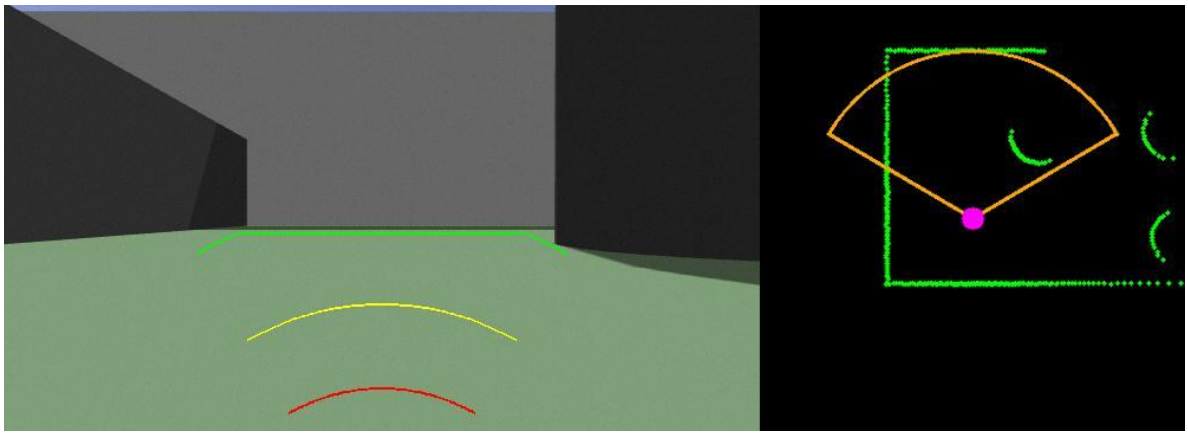

Рисунок 3. Пример объединённого изображения (Изображение с камеры слева, данные LiDAR справа).

1) Подготовка контекста: Перед обращением к языковой модели формируется расширенный контекст:

a) История предыдущих навигационных команд

b) История краткого текстового описание местности
c) Объединенное изображение сенсорных данных
d) Инструкция к действиям (промт)

2) Взаимодействие с LLM: Сформированный контекст передается через API выбранной языковой модели. Модель анализирует изображение, историю команд и описание местности, генерируя наиболее подходящую навигационную инструкцию.

3) Извлечение и выполнение команды: Из ответа LLM извлекается конкретная команда перемещения, которая выполняется роботом в симуляторе.

a) Формат сформированной команды: "linear": <движение в см>, "angular": <поворот в градусах>,"experts": [<безопасность 0-100>,<поиск цели 0-100>,<предыдущие решения 0-100>], "see_goal":<прямая видимость цели true/false>, "summary": <семантическая интерпретация среды> , "task_completed":<true/false>

b) Параллельно команда сохраняется в истории для поддержания контекста в следующих итерациях.

Описанный алгоритм выполняется итеративно до достижения целевой позиции, если в полученной команде получено положительное значение task_completed, то задача считается выполненной и программа завершается. Каждая итерация обновляет контекст и корректирует траекторию движения.

Ограничения системы:

- Максимальная линейная скорость: 0.2 м/с.
- Максимальный угол поворота: ±75°.
- Границы движения: 10 см – 200см
- Запрещено оставаться на месте дольше 5 команд подряд
- Запрещено движение назад, за исключением отъезда от препятствия
- Максимальное количество итераций алгоритма: 50

Алгоритм безопасности:

1. Отдельная проверка столкновений по LiDAR

2. Экстренная остановка при приближении к препятствию ближе 5см от границы робота, прекращение выполнения команды LLM

Экспериментальные исследования

Цель исследования: комплексная оценка влияния характеристик LLM и параметров камеры на эффективность навигации мобильного робота.

A. Факторы исследования

В качестве факторов исследования будут использоваться разные модели LLM и разный горизонтальный угол обзора камеры FOV (Field of View).

Для проведения экспериментов были выбраны следующие модели LLM: GPT-4o, GPT-4o mini, Claude Sonet 3.5, Cohere Command R+. Они обладают разным количеством обученных параметров и разными моделями распознавания изображений. Эти модели являются самыми последними разработками, компаний выпускающих LLM, на момент написания статьи.

Чтобы оценить влияние FOV камеры на выполнение задачи, были выбраны следующие значения:

- Узкий FOV: 45°.
- Средний FOV: 90°.
- Широкий FOV: 120°.

Как экспериментальная среда была смоделирована комната в симуляторе Gazebo размером 10 на 10 метров с статическими препятствиями.

Для каждой комбинации модели и угла обзора камеры проведено по 3 эксперимента. Начальная позиция робота в каждой симуляции оодинаковая, при этом цель на стартовой позиции вне поля зрения камеры. Задача робота переместиться ровно под красную стрелку.

B. Метрики оценки качества

- Время достижения цели
- Количество итераций алгоритма управления

- Точность достижения цели – отклонение от идеального позиционирования. Расчет как расстояние между двумя точками на плоскости.

Таблица 1. Результаты эксперимента

LLM	FOV	Time	Iterations	Task Success	Deviation From goal
Claude Sonet 3.5	45°	363	22	66%	0,629
	90°	196	12,66	100%	2,02
	120°	159	10	100%	4,69
GPT-4o	45°	208,33	20,67	100%	1,17
	90°	119	12	100%	0,85
	120°	205,3	20	100%	2,53
GPT-40 mini	45°	-	-	0%	-
	90°	-	-	0%	-
	120°	147	17	33%	1,5
Command R+	45°	382	38	33%	0,69
	90°	274,33	28	66%	2,24
	120°	190	18,67	100%	1,18

Таким образом, средний процент успешного завершения задачи роботом по всем экспериментам: 66%. Из протестированных моделей GPT-4o mini справилась с задачей только при FOV равным 120 градусов и только в одном случае из трех. Эта модель содержит значительно меньше обучаемых параметров по сравнению с другими протестированными моделями и вполне закономерно, что ее результаты намного хуже. В анализе результатов не будет учитываться данная модель.

C. Графические результаты

На рисунке 4 представлена зависимость времени от FOV камеры. Видно, что увеличение FOV ведет к уменьшению времени выполнения задачи у GPT-4o и Claude Sonet 3.5.

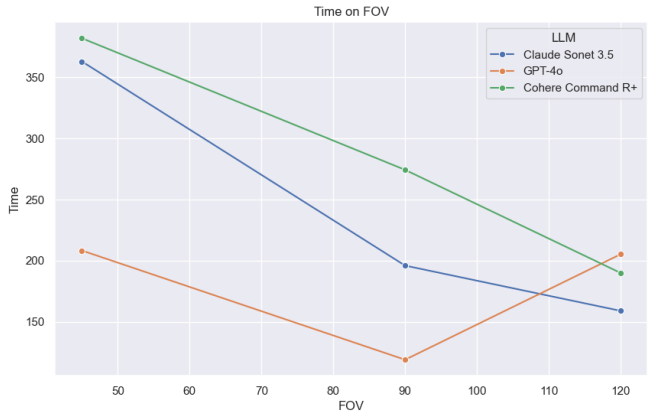

Рисунок 4. . Зависимость времени от FOV.

На рисунке 5 представлен график итогового отклонения от цели. У Claude Sonet 3.5 разброс отклонений намного больше, чем у других двух моделей.

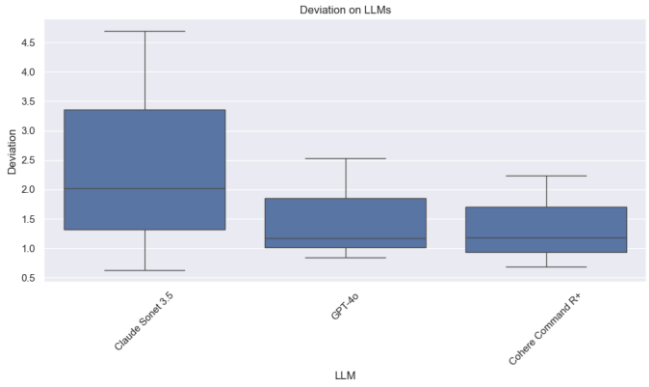

Рисунок 5. Отклонения от цели по моделям.

На рисунке 6 представлены графики влияния FOV на успешное завершение задачи, количество итераций и отклонения от цели. Из графиков видно, что наименьшее отклонение от цели при меньшем значении FOV, но при этом меньший FOV сказывается на меньшую вероятность успеха выполнения задачи у всех моделей.

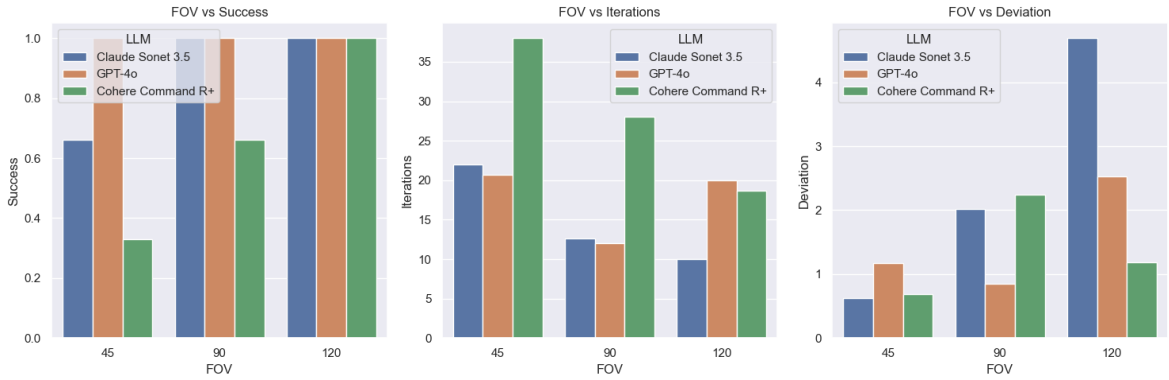

Рисунок 6. графики влияния FOV на Success, Itterations, Deviation.

Проведен анализ эффективности всех моделей на основе трех критериев с разными весами.

Рассчитаем параметр эффективности каждой модели на основе таких параметров как: процент успешного завершения задачи - S, количества итераций - I и отклонения от цели - D. Каждому критерию присвоим свой вес.

Формула расчета эффективности:

$$Efficiency = 0.5S + 0.3D + 0.2I$$

На рисунке 7 представлен график результатов многофакторного анализа.

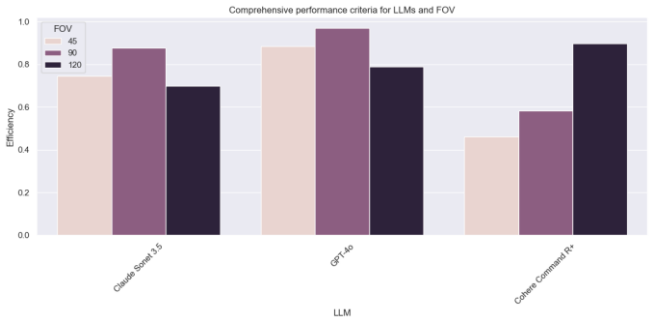

Рисунок 7. . Комплексный многофакторный анализ

На основе проведенного анализа можно сделать вывод что лучше всего с задачей справляется модель GPT-4o, а лучший FOV камеры – 90°. Немного ниже эффективность у Claude Sonet 3.5. Хуже всего справилась Cohere Command R+ с FOV камеры – 45, однако у данной модели можно увидеть тенденцию к увеличению эффективности с увеличением FOV.

Таблица 2. Корреляционный анализ

Variables	**FOV**	**Time (sec)**	**Iterations**	**Task Success %**	**Target Deviation**
FOV	1.00	-0.71	-0.56	0.27	0.59
Time (sec)	-0.71	1.00	0.83	-0.52	-0.38
Iterations	-0.56	0.83	1.00	-0.65	-0.44
Task Success %	0.27	-0.52	-0.65	1.00	0.35
Target Deviation	0.59	-0.38	-0.44	0.35	1.00

1)Сильные отрицательные корреляции:

a) “FOV” и “Время” (-0.71): чем больше поле зрения, тем меньше времени требуется.

b) “FOV” и “Итерации” (-0.56): чем больше поле зрения, тем меньше итераций нужно.

2)Сильные положительные корреляции:

a) “Время” и “Итерации” (0.83): чем больше времени, тем больше итераций, что логично.

b) “FOV” и “Отклонение от цели” (0.59): увеличение поля зрения ведет к большему отклонению от цели.

3)Средние корреляции:

a) “Задача выполнена” и “Итерации” (-0.65): больше итераций связано с меньшим процентом выполнения.

b) “Задача выполнена” и “Время” (-0.52): больше времени также связано с меньшим процентом выполнения.

4)Слабые корреляции:

a) “Отклонение от цели” и “Задача выполнена” (0.35)

b) “FOV” и “Задача выполнена” (0.27)

Корреляционный анализ показывает, что параметры взаимосвязаны. Особенно важны взаимосвязи между временем, итерациями и полем зрения (FOV).

Таблица 3. Статистический анализ

Параметр	**Влияние**	**F-статистика**	**p-значение**	**Статистически значимое различие**
Time	LLM	1.0519	0.4059	Нет
Iterations	LLM	2,86	0.1338	Нет
Success	LLM	1,75	0.2517	Нет
Deviation	LLM	0.5396	0.6088	Нет
Time	FOV	3.1173	0.1179	Нет
Iterations	FOV	1,48	0.2988	Нет
Success	FOV	1,75	0.2517	Нет
Deviation	FOV	2,32	0.1793	Нет

Влияние модели LLM и FOV:

- Для всех измеряемых параметров по влиянию LLM (Time, Iterations, Success, Deviation) p-значения > 0.05.
- Аналогично, для всех параметров по влиянию FOV p-значения> 0.05.

Таким образом, нет статистически значимых различий между разными LLM и разными FOV.

Стоит отметить, что наиболее близкое к значимому результату оказалось влияние FOV на время выполнения ($p = 0.1179$), а также влияние LLM на количество итераций ($p = 0.1338$)

Отсутствие статистической значимости может быть связано с:

- Маленьким размером выборки
- Высокой вариативностью результатов

- Малым числом проведенных экспериментов

D.Результаты

- Не выявлено статистически значимого влияния FOV на результат, однако FOV 90° на всех моделях демонстрирует более стабильные результаты.
- Не выявлено статистически значимого влияния модели LLM на результат, однако GPT-4o показала в среднем лучшие результаты.
- Увеличение FOV повышает процент успешного выполнения задачи и уменьшает количество итераций и затраченного времени, но при этом увеличивается отклонение от цели, напротив уменьшение FOV показывает более точное достижение цели, но при этом общий процент успеха выполнения задачи сильно падает.

Заключение

В рамках проведенного исследования получены следующие ключевые результаты:

- Разработана архитектура системы управления мобильным роботом с использованием больших языковых моделей (LLM).
- Создан алгоритм семантической интерпретации сенсорных данных с преобразованием в целенаправленные навигационные команды.
- Экспериментально подтверждена общая применимость предложенного подхода: Общий средний успех завершения задачи роботом по всем экспериментам : 66%, а если не учитывать результаты модели GPT-4o mini, то процент составляет: 85%. Лучше всего с задачей справилась модель GPT-4o, с FOV равным 90°.

Список используемых источников

1) Wang G.; Fomichev A.V.. "Simultaneous localization and mapping method for a planet rover based on a Gaussian filter." *AIP Conference Proceedings*, vol. 2171, no. -, 2019, pp. -. doi: 10.1063/1.5133307
2) Yin S.; Yuschenko A.. "Planning of service mobile robot based on convolutional LSTM network." *Journal of Physics: Conference Series*, vol. 1828, no. 1, 2021, pp. -. doi: 10.1088/1742-6596/1828/1/012002
3) Yuschenko A.S.; Yin S.. "Dialogue Control of Collaborative Robots Based on Artificial Neural Networks; [Диалоговое управление коллаборативными роботами с помощью искусственных нейронных сетей]." *Mekhatronika, Avtomatizatsiya, Upravlenie*, vol. 22, no. 11, 2021, pp. 567-576. doi: 10.17587/mau.22.567-576
4) Kotov A.; Zaidelman L.; Zinina A. et al.. "Conceptual Operations with Semantics for a Companion Robot." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12335, no. -, 2020, pp. 232-243. doi: 10.1007/978-3-030-60276-5_24
5) Huang, Wenlong, et al. "Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents." 2022. arXiv, https://arxiv.org/abs/2201.07207.
6) Min, So Yeon, et al. "FILM: Following Instructions in Language with Modular Methods." 2022. arXiv, https://arxiv.org/abs/2110.07342.
7) Ahn, Michael, et al. "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances." 2022. arXiv, https://arxiv.org/abs/2204.01691.
8) W. Zu *et al.*, "Language and Sketching: An LLM-driven Interactive Multimodal Multitask Robot Navigation Framework," *2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, 2024, pp. 1019-1025, doi: 10.1109/ICRA57147.2024.10611462.
9) V. S. Dorbala, J. F. Mullen and D. Manocha, "Can an Embodied Agent Find Your “Cat-shaped Mug”? LLM-Based Zero-Shot Object Navigation," in *IEEE Robotics*

and Automation Letters, vol. 9, no. 5, pp. 4083-4090, May 2024, doi: 10.1109/LRA.2023.3346800. keywords: {Navigation;Task analysis;Robots;Grounding;Natural languages;Decision

10) Driess Danny, et al. "PaLM-E: An Embodied Multimodal Language Model." 2023. arXiv, https://arxiv.org/abs/2303.03378.

11) Aixin Liu, DeepSeek-AI, et al. "DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model." 2024. arXiv, https://arxiv.org/abs/2405.04434.

12) Kim R.; Kotsenko A.; Andreev A. et al.. "Comparison of ChatGPT and Bard for using in hybrid intelligent information systems." *E3S Web of Conferences*, vol. 549, no. -, 2024, pp. -. doi: 10.1051/e3sconf/202454908009

13) Varlamov O.. "“Brains” for Robots: Application of the Mivar Expert Systems for Implementation of Autonomous Intelligent Robots." *Big Data Research*, vol. 25, no. -, 2021, pp. -. doi: 10.1016/j.bdr.2021.100241

14) Volgina A. D., Kirillov D. S., Kravtsov A. N., Dyakonova S. S. and Kanev A. I., "The Robot-Guide for Indoor Navigation," 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Moscow, Russian Federation, 2023, pp. 1-6, doi: 10.1109/REEPE57272.2023.10086707.