Система автономной навигации антропоморфного робота

Подсистема планирования траектории движения антропоморфного робота

Студент: Титов Алексей
Группа: ИВТ - 460

Руководитель: Горобцов А. С.

Цели и задачи

Цель

- Разработать систему автономной навигации для антропоморфного робота AR600E.

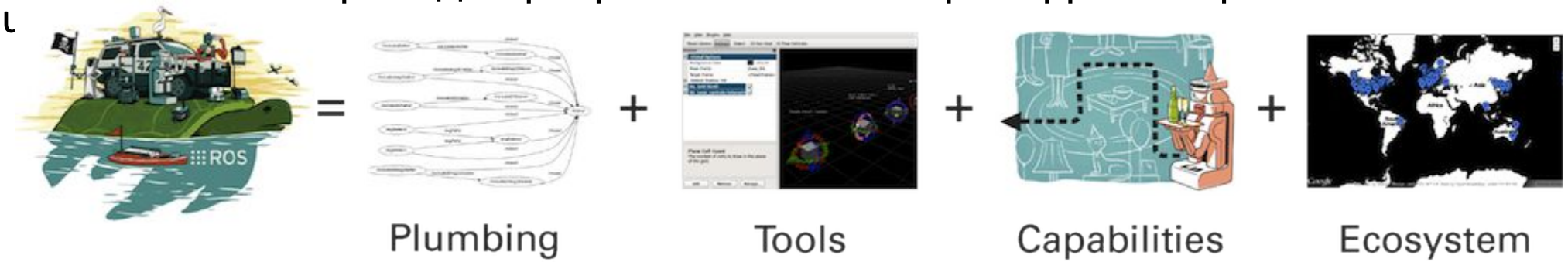
Задачи:

- Поиск и исследование аналогов
- Планирование архитектуры системы
- Разработка:
  - Подсистемы планирования траектории движения робота
  - Подсистемы планирования параметров текущего шага робота (Марков А. Е.)

Средства, методы и подходы

Платформа разработки

В качестве платформы для разработки был выбран фреймворк ROS на ОС Ubu



Причины:

- Предоставляет много нужных пакетов для робототехники
- В частности хорошо развиты стеки:
  - Навигации
  - Локализации
  - Картографии (SLAM алгоритмы)
  - Планирования движений и маршрутов
- Предоставляет визуализаторы, удобную концепцию сообщений и подписчиков и много другого.
- http://www.ros.org/core-components/

SLAM (с англ. одновременная локализация и картография)

Для автономной навигации, да и вообще чего - либо, требующего полной информации об окружении нужна карта самого окружения.

Для этих задач хорошо подходит карта в виде облака точек.

Vision-based SLAM алгоритмы позволяют строить карту окружения и приблизительно оценивать местоположение в ней.

На входе:

- Данные с датчиков, помогающих оценить местоположение
- Данные с Vision датчиков (будет разобрано далее)

На выходе:

- Карта окружения
- Аппроксимация местоположения

В качестве **SLAM** алгоритма была выбрана реализация библиотеки **rtabmap**.

- Качественно задокументирована
- Поддерживает многие датчики в качестве источников данных (см. далее)
- В меру требовательна к ресурсам даже при создании больших карт (хватает ноутбука с Intel Core i3 + 4 Гб RAM)
- Имеет множество настроек и легко расширяема
- Предоставляет много доп. функций (например карту препятствий)
- Интеграция с библиотекой Octomap (спец. структуры для хранения и обработки плотных облаков точек)

Имеется также ряд других реализаций:

- **hector_slam** (строит только 2D карту препятствий)
- **RGBDSLAMv2** (глубокая beta и слабая документация)
- **Kinect Fusion**, **PCL KinFU**, **ElasticFusion** (строят мешь, высокие требования к ресурсам, малый размер карты)

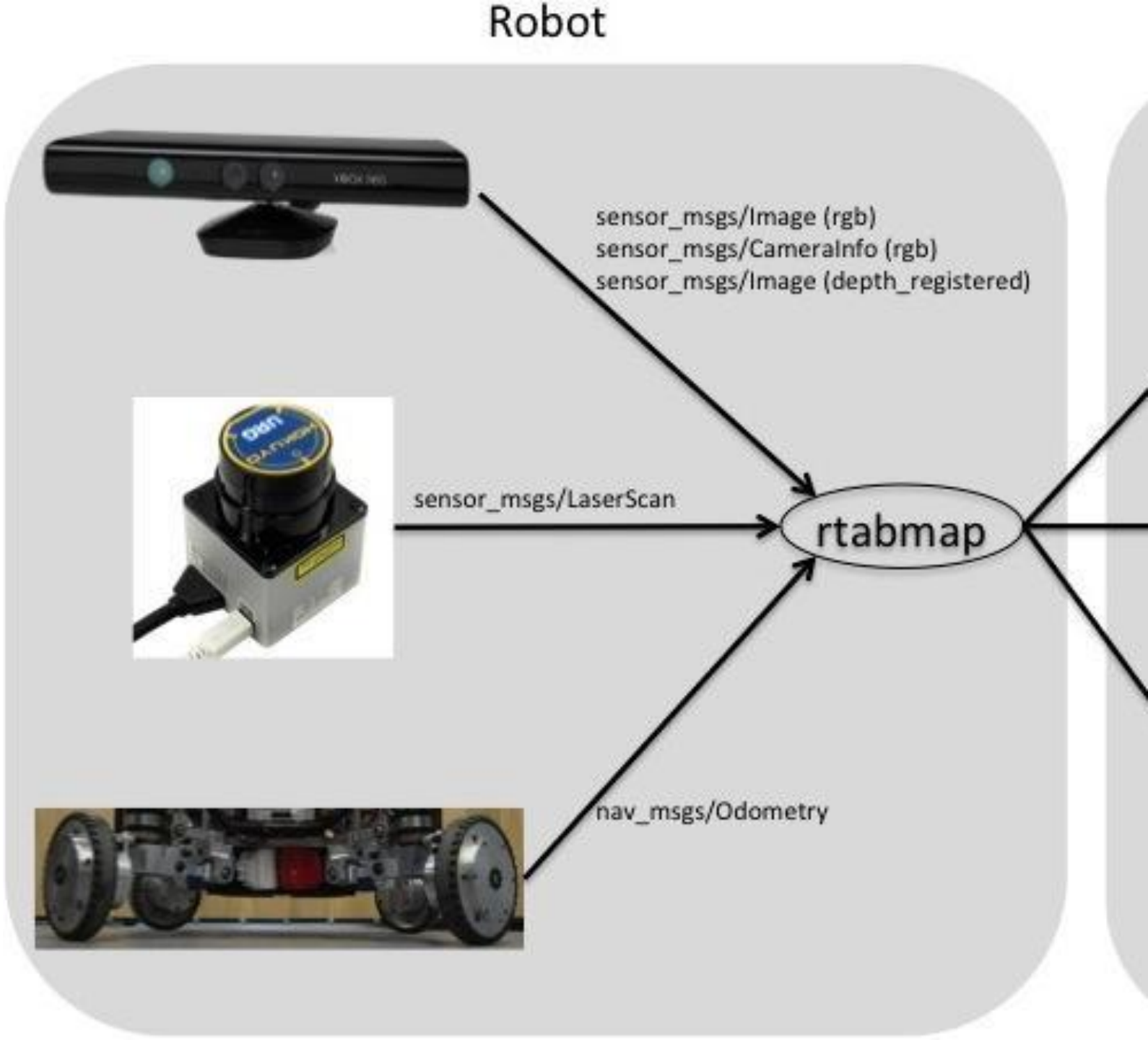
Источники данных для SLAM алгоритма

Источники данных для SLAM алгоритма

В качестве источников данных могут выступать практически любые датчики, помогающие определить местоположение:

- ИНС
- Данные с моторов (для колесных роботов)

Для Visual-based SLAM'а необходимы источники, связанные с «глубиной», которые будут рассмотрены далее



RGB-D камеры

- Предоставляет RGB снимок и карту глубины
- Хорошо поддерживаются много библиотеками
- Дает приемлемую точность (зависит от расстояния. От 1 мм до 5 см)
- Дальность 4 метра
- Слепая зона 0.5 метра

Стерео камеры

- Высокая дальность
- Практически нет слепой зоны
- Точность ниже, чем у RGB-D камер
- **Пара веб камер - не в состоянии дать качественные результаты**

FRONT SIDE

Лидары

- Высокая дальность
- Высокая стоимость
- Возможно использование в будущем
- В данный момент нет необходимости, т.к. работа на открытом пространстве от робота не требуется

Подходы к решению задачи автономной навигации

Планирование по карте препятствий (OccupancyGrid)

Этот подход позволяет решать задачу навигации на плоскости, что довольно просто.

Карта препятствий - 2D изображение, состоящее из ячеек, в котором каждая ячейка может иметь 1 из 3 значений:

- Препятствие
- Свободно
- Неизвестно

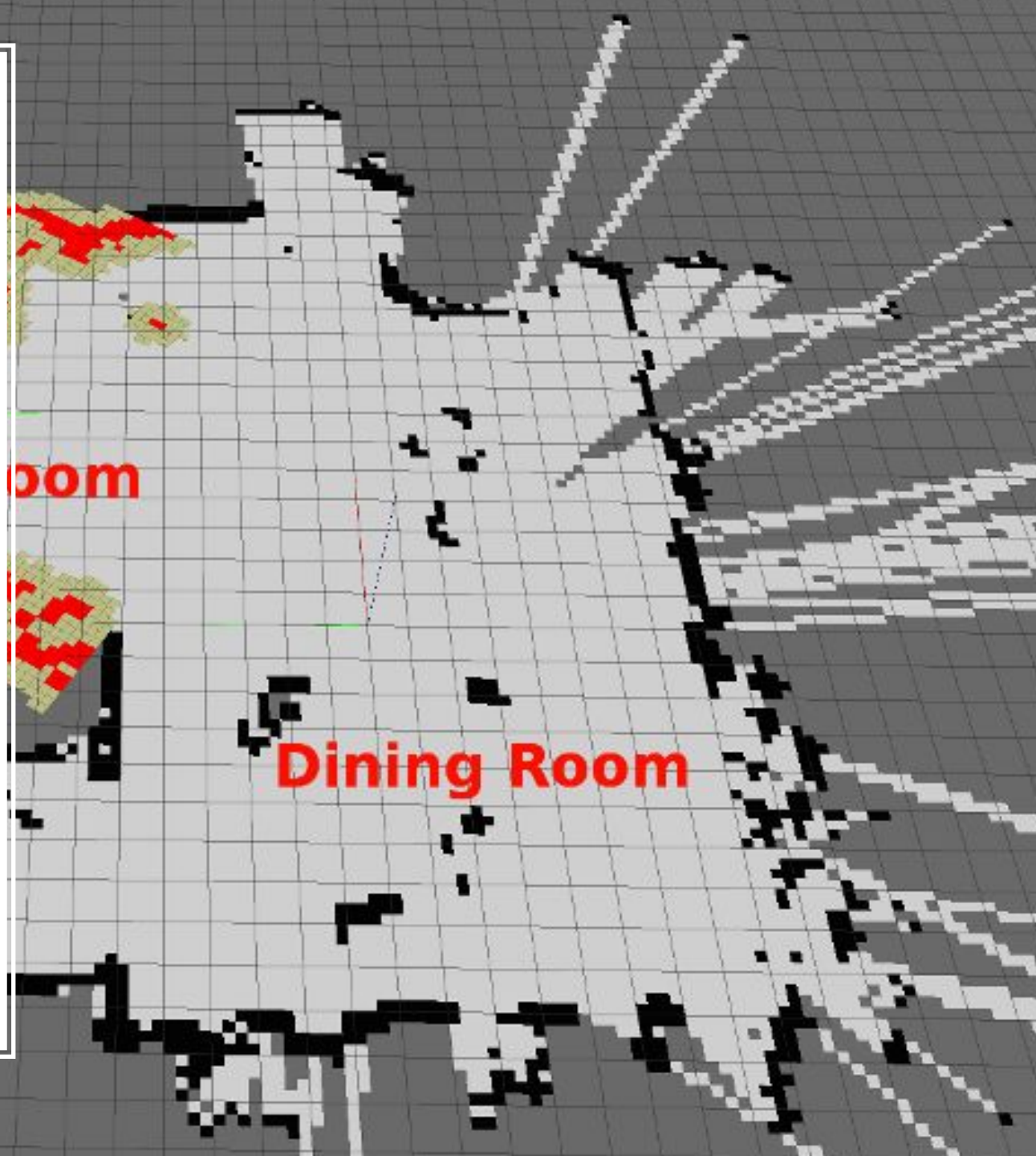
Эту карту можно получить из облака точек несколькими способами:

- Как простой срез на определенной высоте из облака точек
- Как проекцию всех точек на плоскость z = 0 начиная с какой - нибудь заданной высоты z = ?.

Такой вариант предоставляет **rtabmap**.

Для этого имеется много настроек, как то:

- допустимый угол наклона “плоскости”
- количество точек вблизи, которые стоит считать за препятствие
- и т.д.



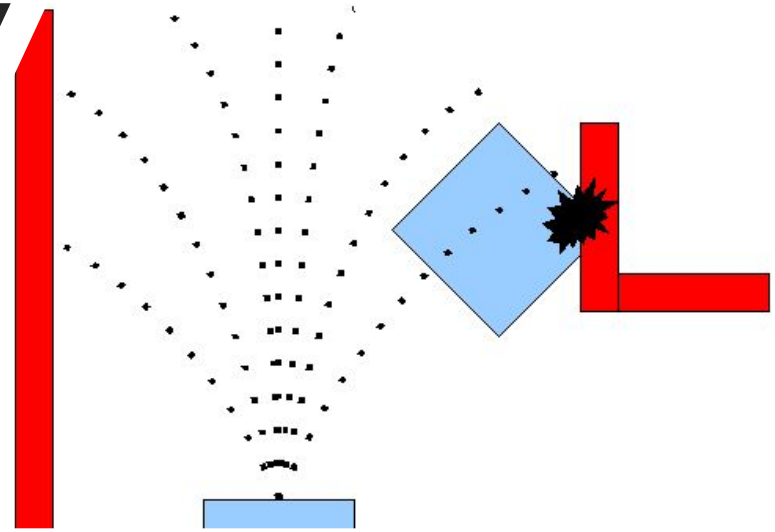
ROS концепция Move Base

Это концепция ROS, которая работает с 2 планерами:

- **Local Planner**

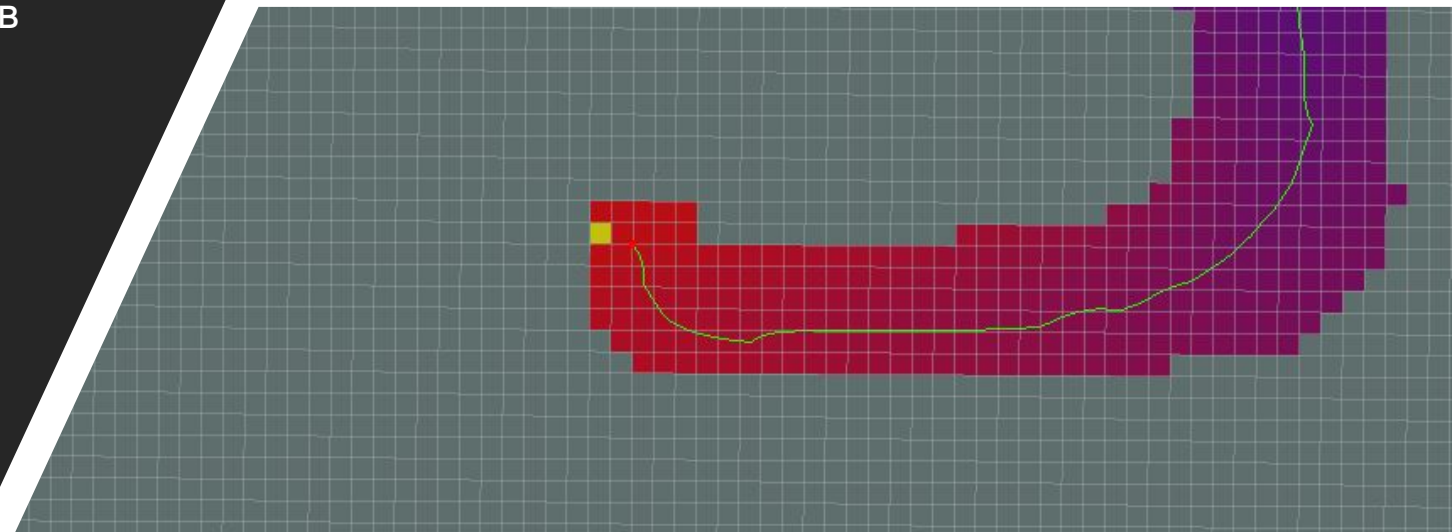
Управляет мобильной платформой (роботом) при прохождении траектории.

На него ложится расчёт скоростей и углов, так чтобы избежать столкновений.



- **Global Planner**

Ищет по Карте Препятствий глобальный маршрут в виде линии



Резюме

В нашем случае роль **Local Planner'a** выполняет ФРУНД.

Имеющиеся же **Global Planner'ы** весьма плохо задокументированы.

Также их использование тянет за собой использование громоздкого move_base.

Footstep planner

Один из сценариев использования этих данных в контексте навигации и движения робота это построение безопасной траектории шагов для робота, по которой он впоследствии сможет пройти.

Было найдено много публикаций, которые использовали планирование "ступнями" на карте препятствий. Затем траектория в виде набора ступней поступала на исполнение роботу.

Такой вариант возможен, если при планировании траектории также учитываются условия равновесия робота, а также если контроллер робота может "выполнять" задания вида "наступить в точку с заданными координатами".

footstep_planner & humanoid_navigation

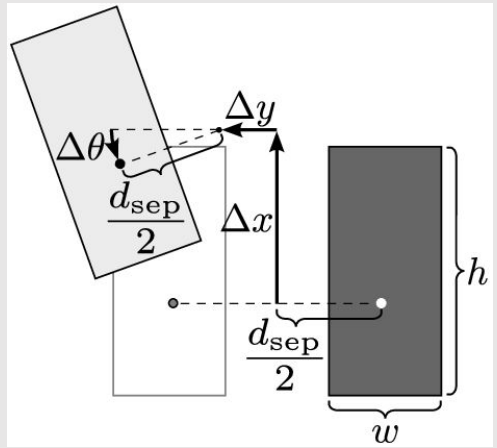
Код данной работы был опубликован в виде ros пакета Armin Hornund'ом.

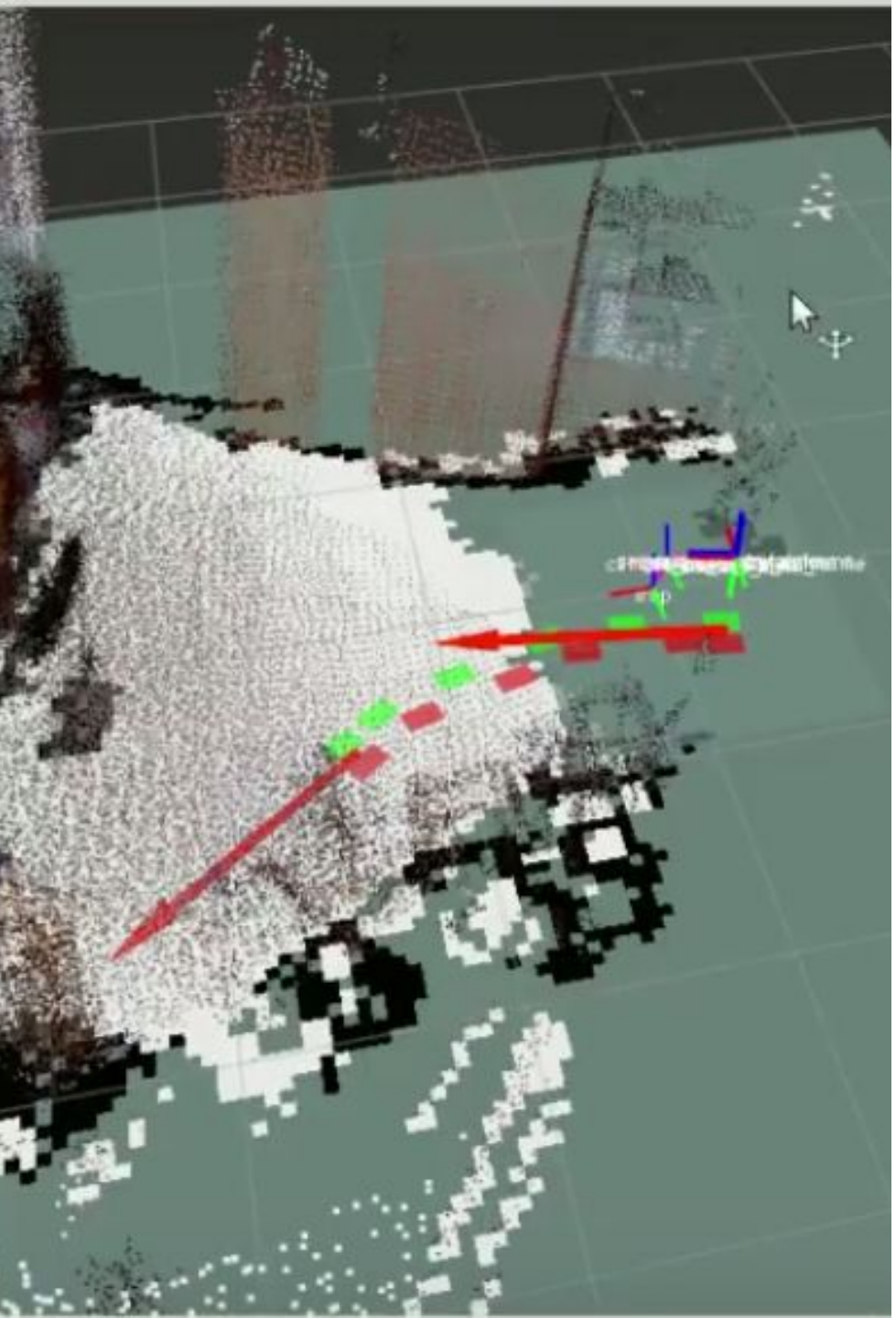
Данный модуль (footstep_planner) предоставляет возможность планировать маршрут в виде набора положений ступней, ведущей из стартовой в конечную позицию.

Планирование осуществляется на 2D карте препятствий.

Следует заметить, что данный пакет предоставляет много нужных настроек.

Начиная с физических параметров ступней и их положений друг относительно друга при шаге, заканчива выбором алгоритма поиска, используемого при планировании.





Применение пакета footstep_planner

Мне удалось исправить в нем ошибку, которая не давала применить его на динамически меняющейся карте.

Также я написал программу, которая передает этому пакету данные о местоположении камеры и карте препятствий, которую предоставляет SLAM алгоритм библиотеки rtabmap.

Благодаря этому стало возможным использование алгоритма планирования с данными от выбранного нами SLAM алгоритма в реальном времени с некоторой задержкой на расчёты.

Подходы к решению задачи автономной навигации

Навигация в [плотных] облаках точек

Vigir footstep planner

Move It

Пакет планирования сложных движений для роботов любой конструкции.

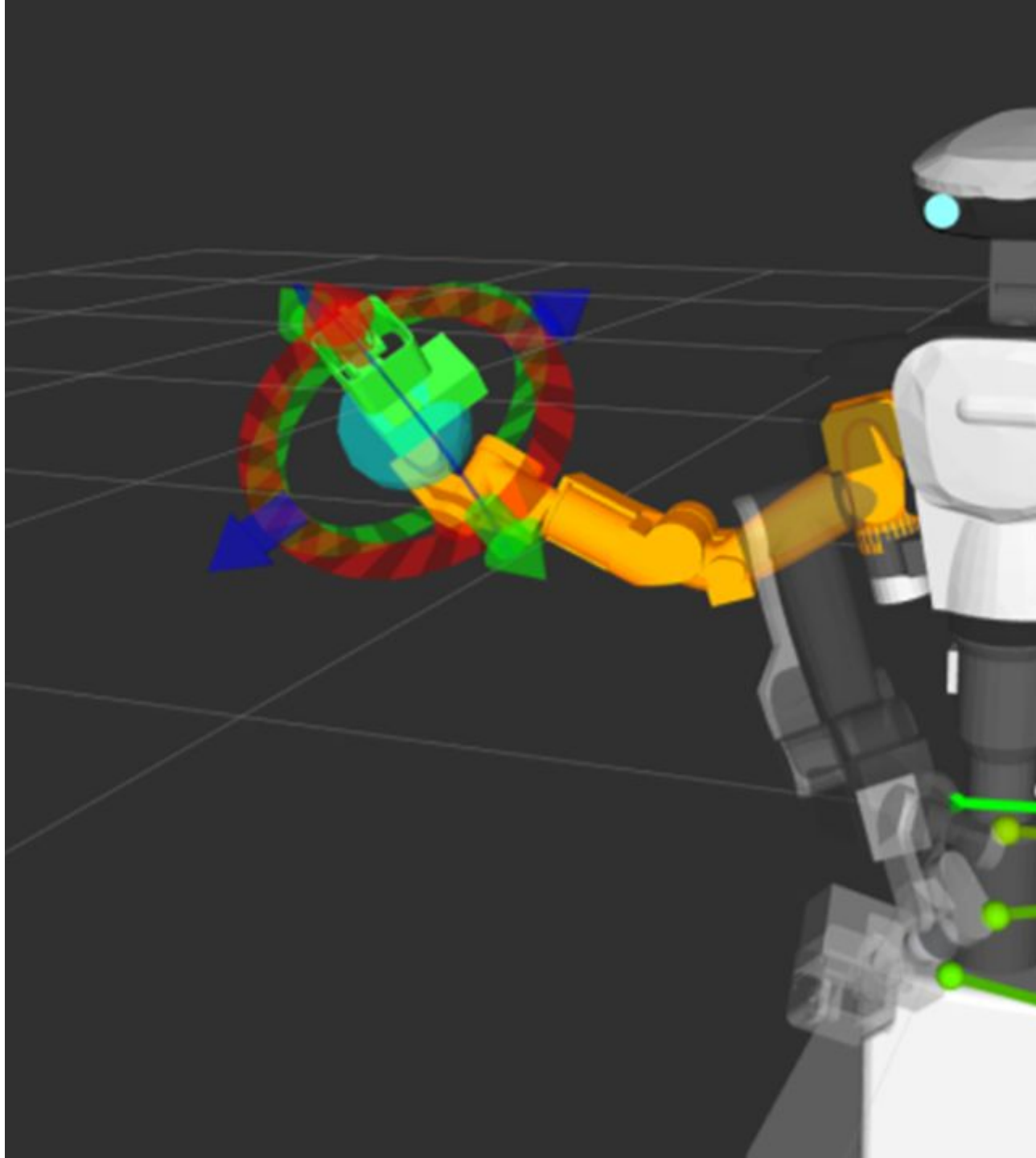
Робот представляется в виде модели с учетом всех подвижных частей и ограничений на их передвижения.

Далее можно рассчитать траекторию всех конечностей робота при его переходе из одного состояния в другое.

Возможен учет коллизий с окружением в виде плотных облаков точек (OcTree).

Также пока остается загадкой возможность реализации ходьбы в этом пакете.

Данный пакет не применим в нашем проекте, т.к. система компьютерного зрения не может влиять на генерацию движений робота. Иными словами роль этого пакета выполняет ФРУНД.



MoveIt!

moveit.ros.org

Проектирование архитектуры

Обоснование

- В нашем случае система расчёта движений (ФРУНД) не может реализовывать шаги, поступающие извне. Наоборот, ФРУНД генерирует шаги так, чтобы держать равновесие.
- Также система компьютерного зрения не может воздействовать на движения, генерируемые ФРУНД’ом.
- Это приводит к реализованной нами архитектуре.

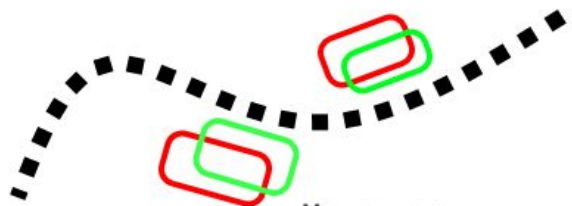
Глобальное планирование траектории
Предоставляет ФРУНД'у траекторию в виде линии (набора точек)

Планирование на 2D Карте Препятствий может упускать некоторые особенности поверзнности

ФРУНД

1) Запрашивает траекторию в виде набора точек
2) Генерирует шаги так, чтобы не сильно откланяться от траектории
3) В процессе планирования шагов ФРУНДОм каждый шаг "проверяется" путем запросов к модулю планирования параметров шага.
// Эти запросы позволяют скорректировать шаг, если в требуемой точке
// обнаружены препятствия. Или остановить робота, если шаг невозможен.

Корректировка параметров шага
Корректирует шаг, если в требуемой точке обнаружены препятствия. Сообщает, если шаг невозможен



Например,
*** слишком большой наклон для робота**
*** мелкие посторонние предметы**

act Автономная навигация

[Параметры шага, координаты и углы с.к. карты в с.к. робота]

Ubuntu + ROS

Windows + ФРУНД

Рассчет параметров шага (локальное планирование)

[Возможность совершения шага и скорректированные координаты шага]

[Координаты и углы с.к. карты в с. к. робота]

Рассчет траектории (глобальное планирование)

[Траектория]

+Прием запросов и данных от ФРУНДа и их передача модулям глобального и локального планирования
+ Передача ответов модулей планирования ФРУНДУ

[Параметры шага, и робота, смещение с.к. карты в с.к. робота от предыдущего запроса]

[Траектория или информация о возможности совершения шага]

Управление роботом (ФРУНД)
+ запросить информацию о шаге
+ запросить информацию о траектории

[Карта, с.к. карты, местоположение камеры в с.к. карты]

SLAM

RGB - D камера

Направления стрелок в стиле:
Кто передает данные -> Кому

Дальнейшие работы

- Возможно разработка упрощенного варианта 2D планера. Т. к. многие возможности footstep_planner’a в данном варианте архитектуры не задействуются
- Доработка взаимодействия с ФРУНД'ом
- Исследования пакета Move It

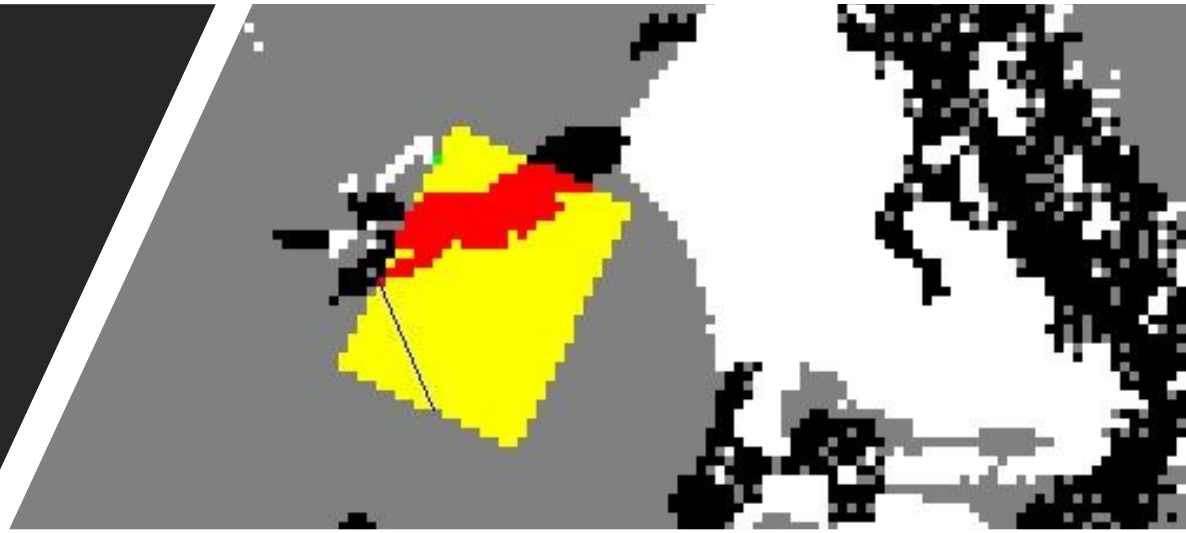
Дополнительные работы

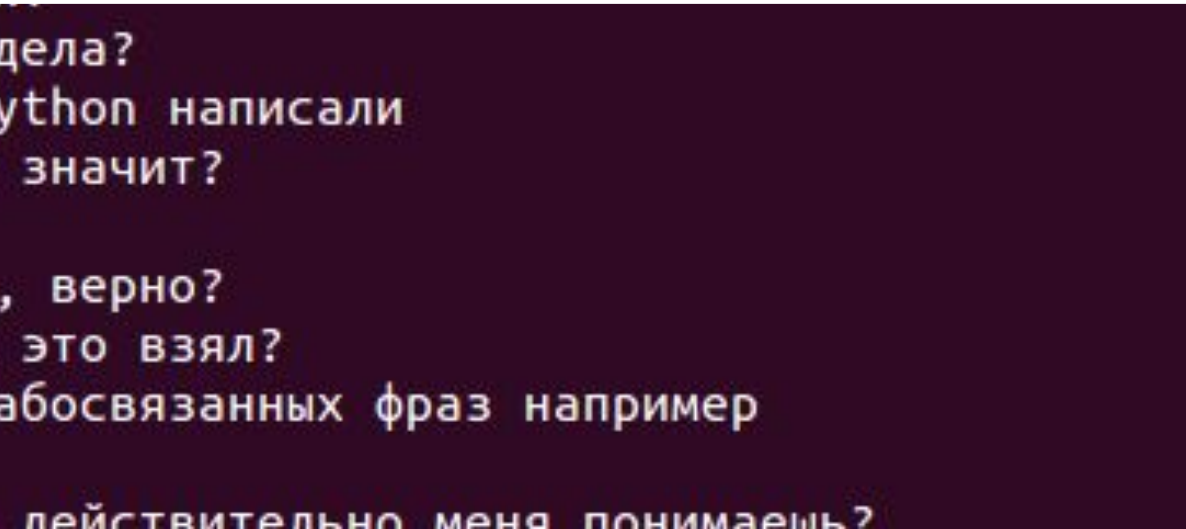
- Обнаружение препятствий в облаке точек по направлению движения робота

- **SpeechAI**

Разработанный мной лингвистический ИИ с распознаванием и синтезом речи.

Одно из реализованных применений - голосовое управление роботом и озвучивание информации о препятствиях.



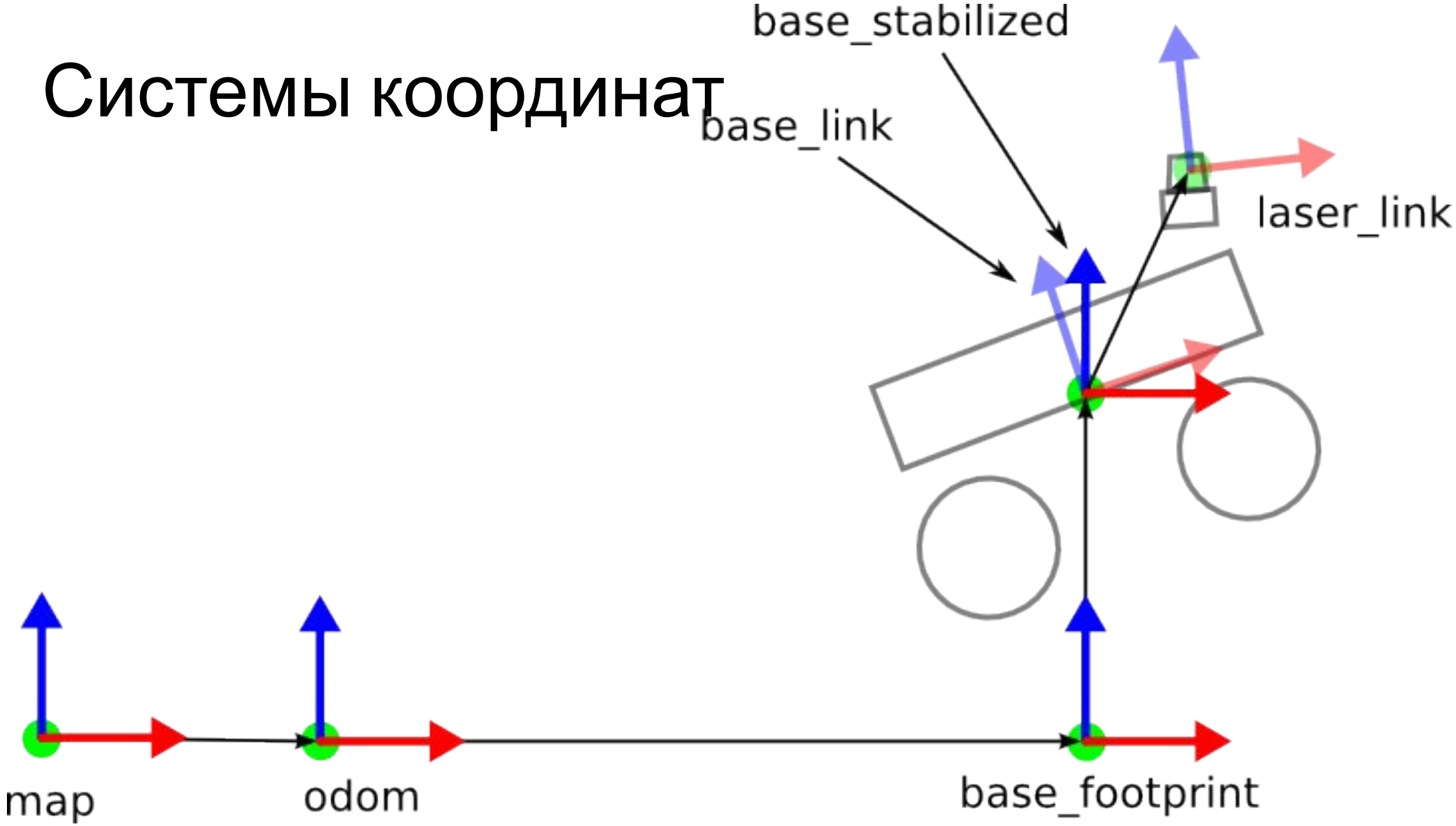


Список использованных источников

- http://hrl.informatik.uni-freiburg.de/
  Много работ с антропоморфным роботом Nao.
  В частности автономная навигация и коррекция движений, поступающих от костюма оператора
- http://wiki.ros.org/rtabmap_ros
  Пакет в ROS для rtabmap. Популярная реализация SLAM алгоритма
- http://moveit.ros.org/
  Пакет MoveIt по планированию движений для роботов
- http://wiki.ros.org/footstep_planner
  2D планер шагов по OccupancyGrid
- http://wiki.ros.org/vigir_footstep_planning
  3D планер шагов в облаках точек. Идейный продолжатель пакета выше
- http://wiki.ros.org/navigation
  Navigation stack в ROS
- http://www.ais.uni-bonn.de/humanoidsoccer/ws12/slides/HSR12_Slides_Hornung.pdf
  Search – based footstep planning
- http://www.probabilistic-robotics.org/
  Потрясающая книга ProbablisticRobotics, рассказывающая про SLAM алгоритмы
- https://www.youtube.com/channel/UCQoNsqW4v8uvrpWxnIabStg
  Качественные курсы по SLAM алгоритмам и навигации.
  Автор предоставляет лекции и практические работы по написанию своих алгоритмов с 0.
  Также качественно разобрана математическая сторона вопроса

Вопросы

Системы координат



Armin Hornung

Armin Hornung

Подписаться

Sick AG

Human-Robot collaboration, 3D Vision, Artificial Intelligence, Robotics, Humanoid Robotics

Подтвержден адрес электронной почты в домене informatik.uni-freiburg.de - Главная страница

Название 1–20	Процитировано	Год
OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees A Hornung, KM Wurm, M Bennewitz, C Stachniss, W Burgard Autonomous Robots	523	2013
OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems KM Wurm, A Hornung, M Bennewitz, C Stachniss, W Burgard Proc. of the ICRA 2010 workshop on best practice in 3D perception and	355	2010

Google Академия

Создать свой профиль

Индексы цитирований	Все	Начиная с 2012 г.
Статистика цитирования	1466	1391
h-индекс	15	15
i10-индекс	18	18