

Разработка метода построения программной траектории и регулятора движения по ней беспилотного наземного транспортного средства.

Выполнил: Магистрант группы ЭВМ-2н Марков Алексей Евгеньевич

Руководитель: Горобцов Александр Сергеевич

Волгоградский Государственный Технический Университет

Постановка задачи

Разработать систему построения программной траектории и регулятора движения по ней беспилотного наземного транспортного средства

Актуальность

В настоящее время идет активное развитие беспилотных автомобилей и интеллектуальных систем помощи водителю (ADAS)

Преимущества беспилотных автомобилей:

- Увеличение безопасности
- Более эффективное использование дорожной инфраструктуры

Цель

Разработать систему построения программной траектории и регулятора движения по ней беспилотного наземного транспортного средства

Задачи

- Анализ существующих подходов проектирования систем управления, выделение типичных подсистем
- Анализ методов планирования движения наземных транспортных средств
- Проектирование и реализация подсистемы формирования программной траектории
- Проектирование и реализация подсистемы движения по траектории
- Экспериментальное исследование, анализ результатов работы

1. Обзор предметной области

DARPA Urban Challenge (2007)



BOSS (Карнеги-Меллон, GM, CAT, ...)



Junior (Стэнфордский университет)



Talos (MIT)



AnnieWAY(технологический университет Карлсруэ)

Основное аппаратное обеспечение

LiDAR



- Получение облаков точек 360°
- Большая дальность
- Высокая цена

Камеры



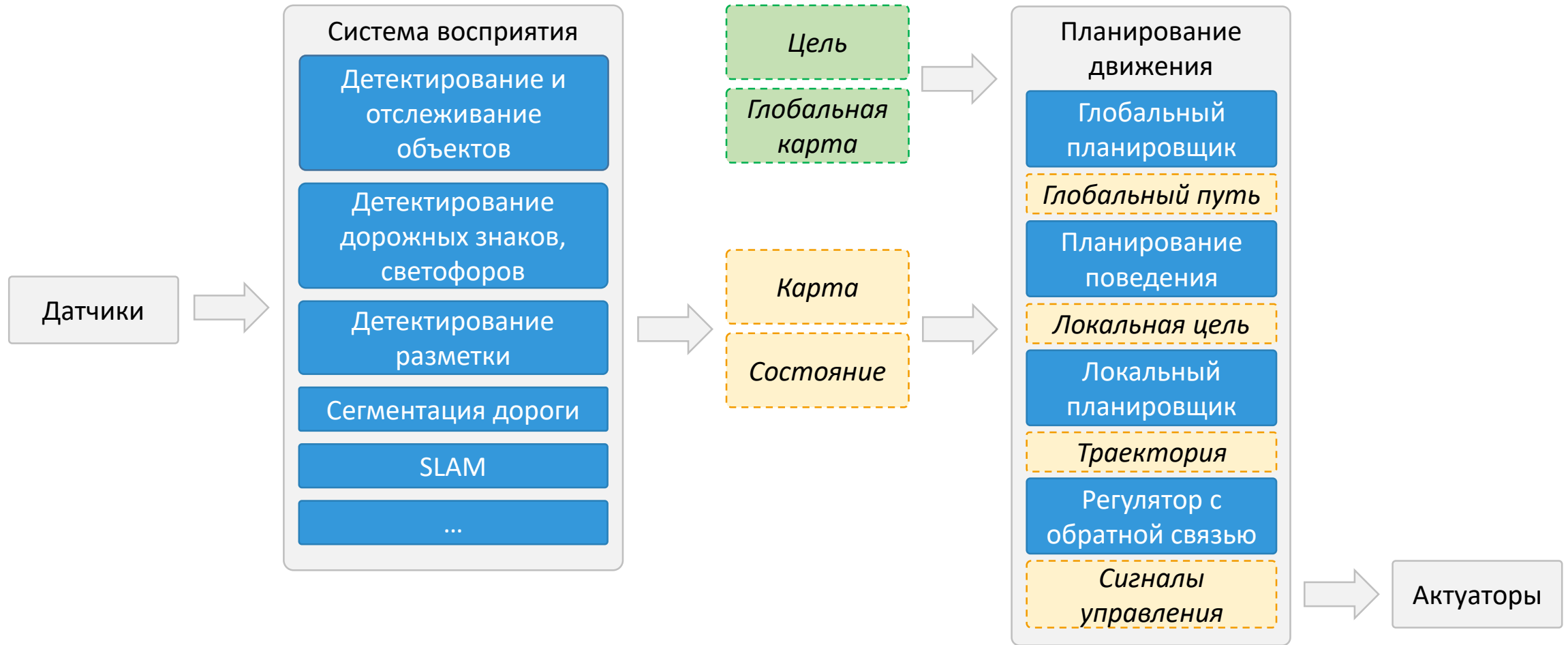
- Цветное изображение
- Высокое разрешение
- Низкая цена

Радар

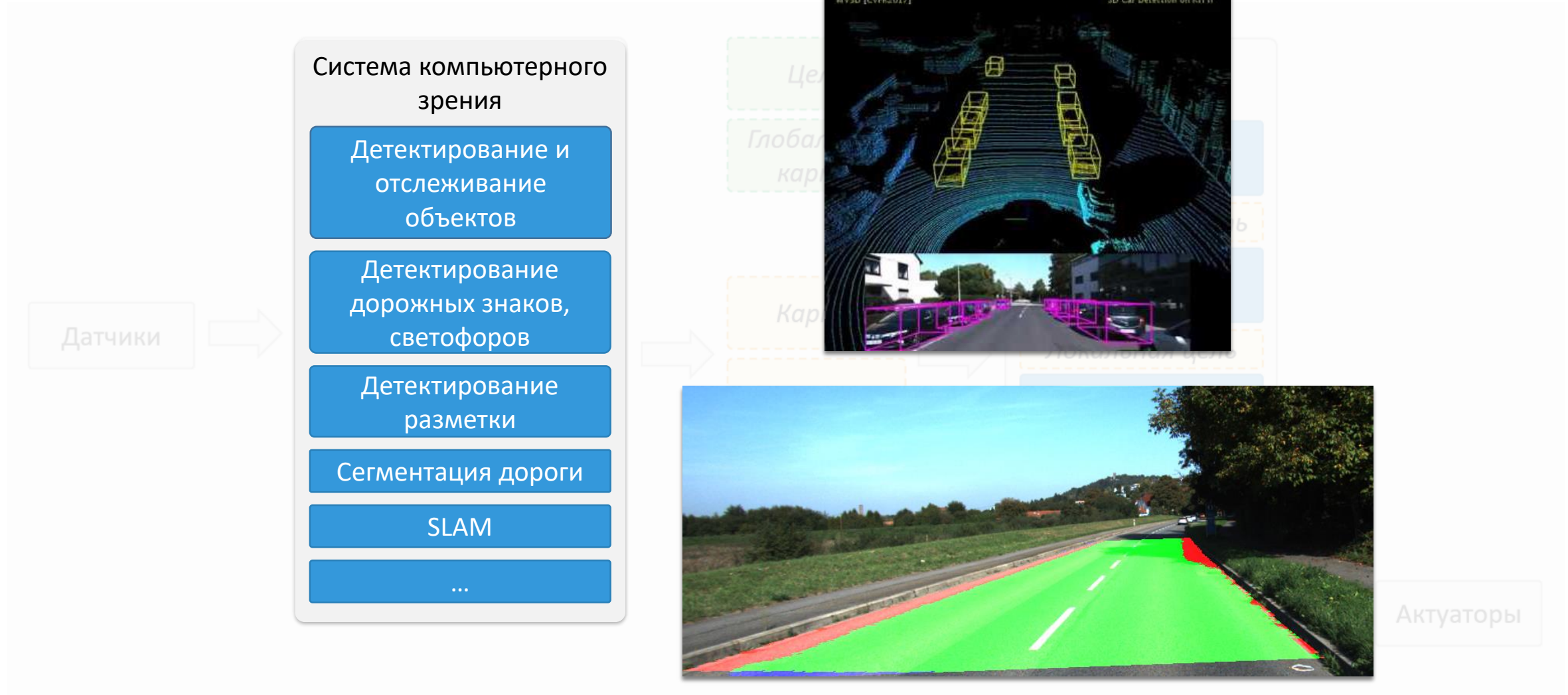


- Точность обнаружения объектов
- Определение скорости объектов

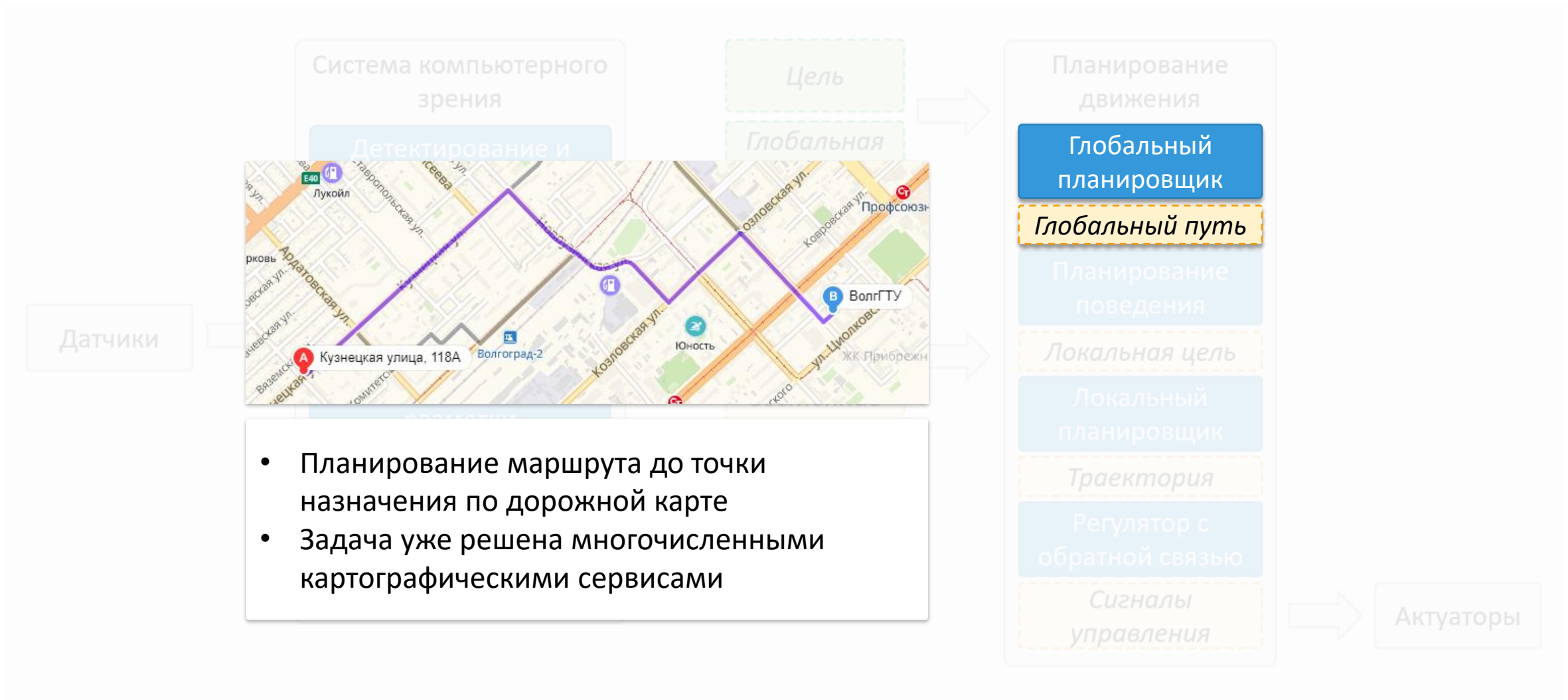
Общая структура системы управления



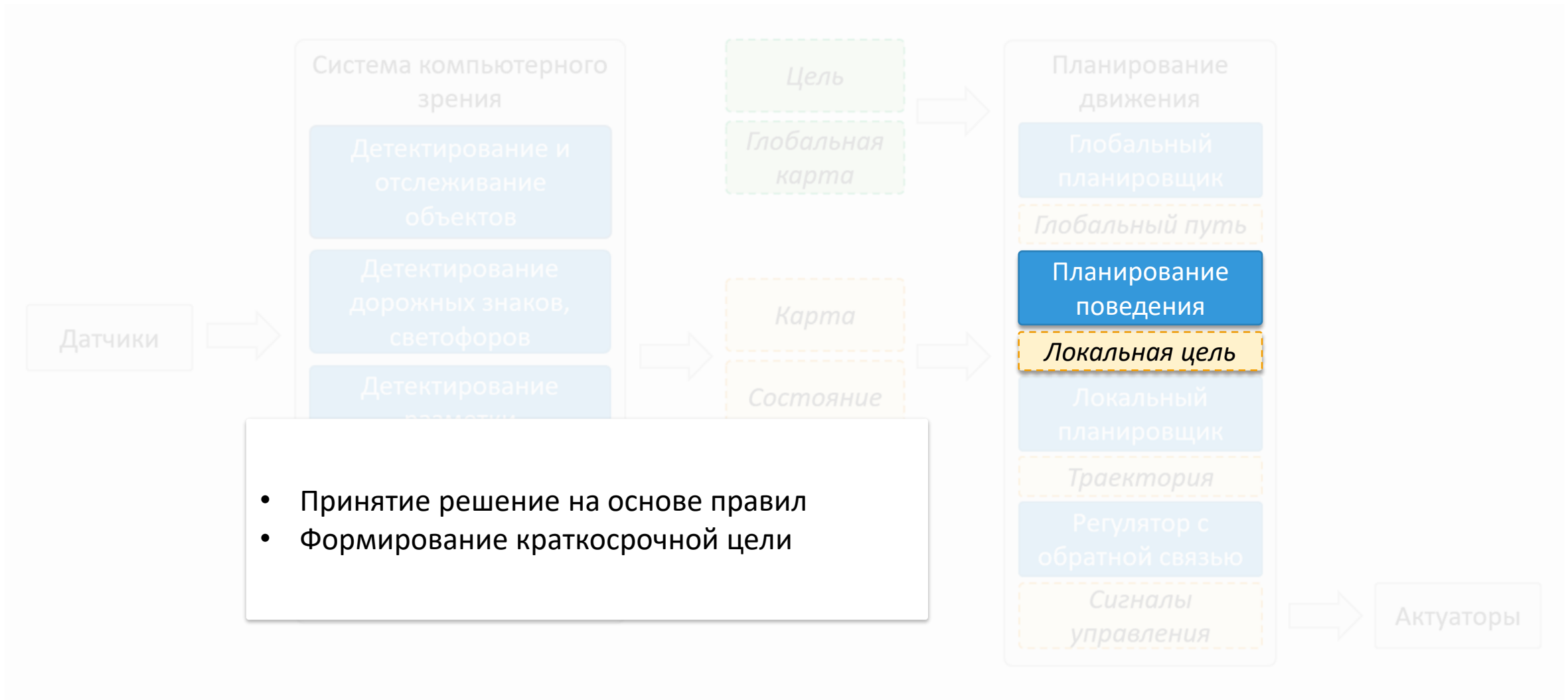
Общая структура системы управления



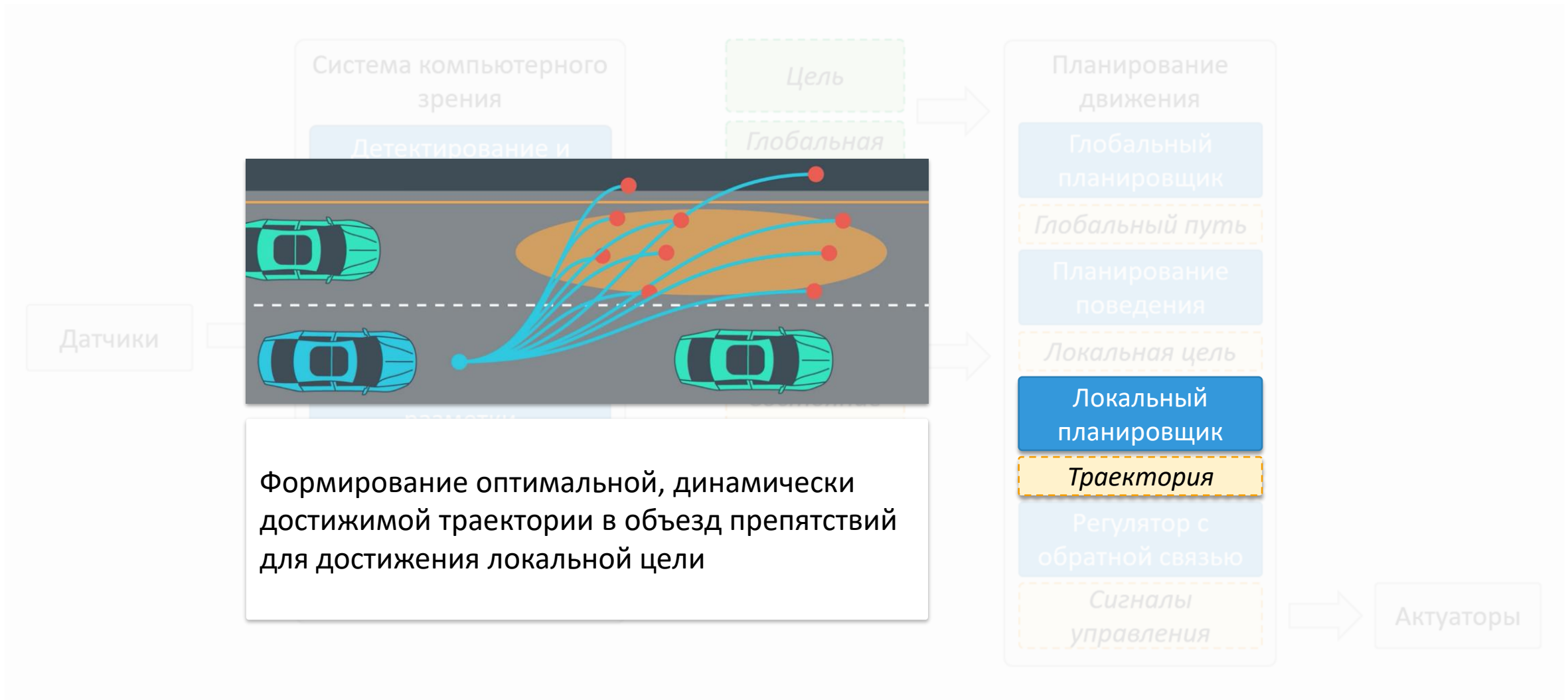
Общая структура системы управления



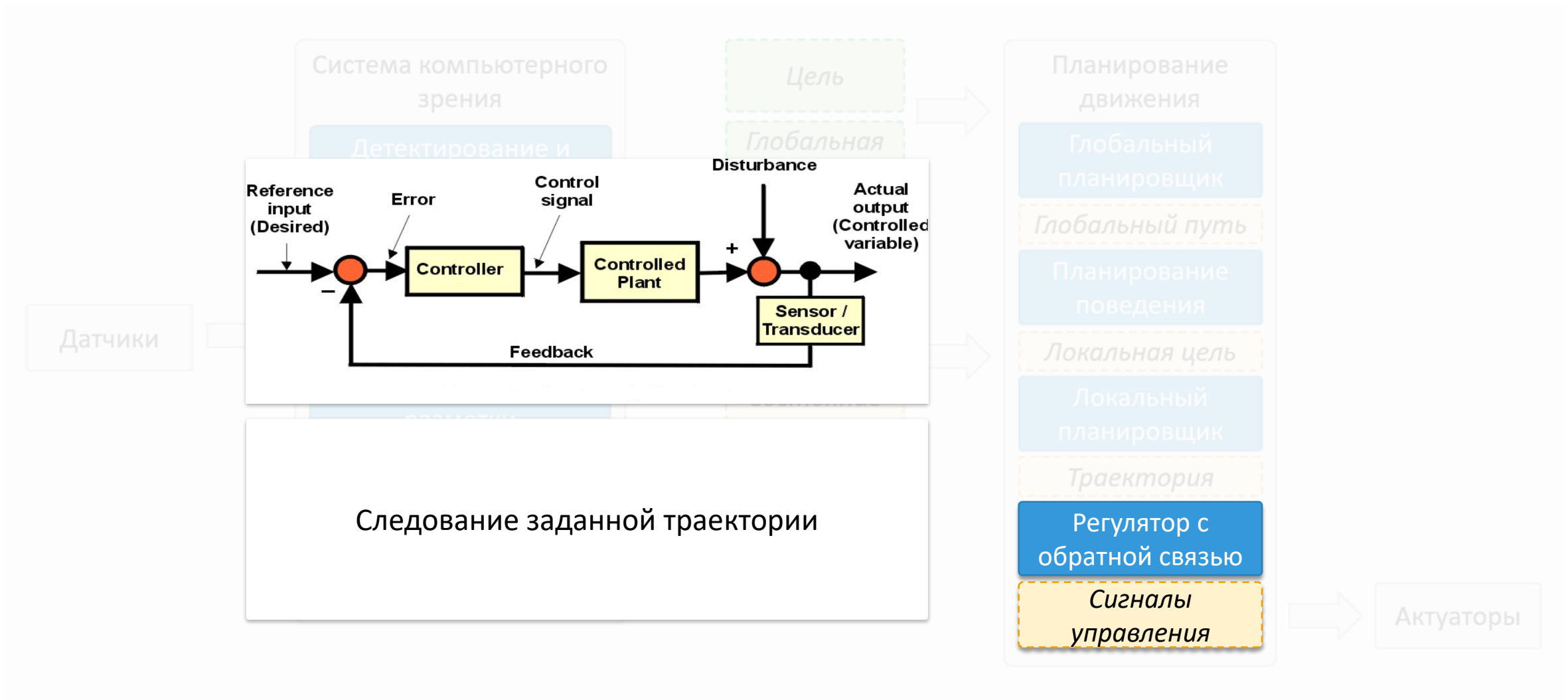
Общая структура системы управления



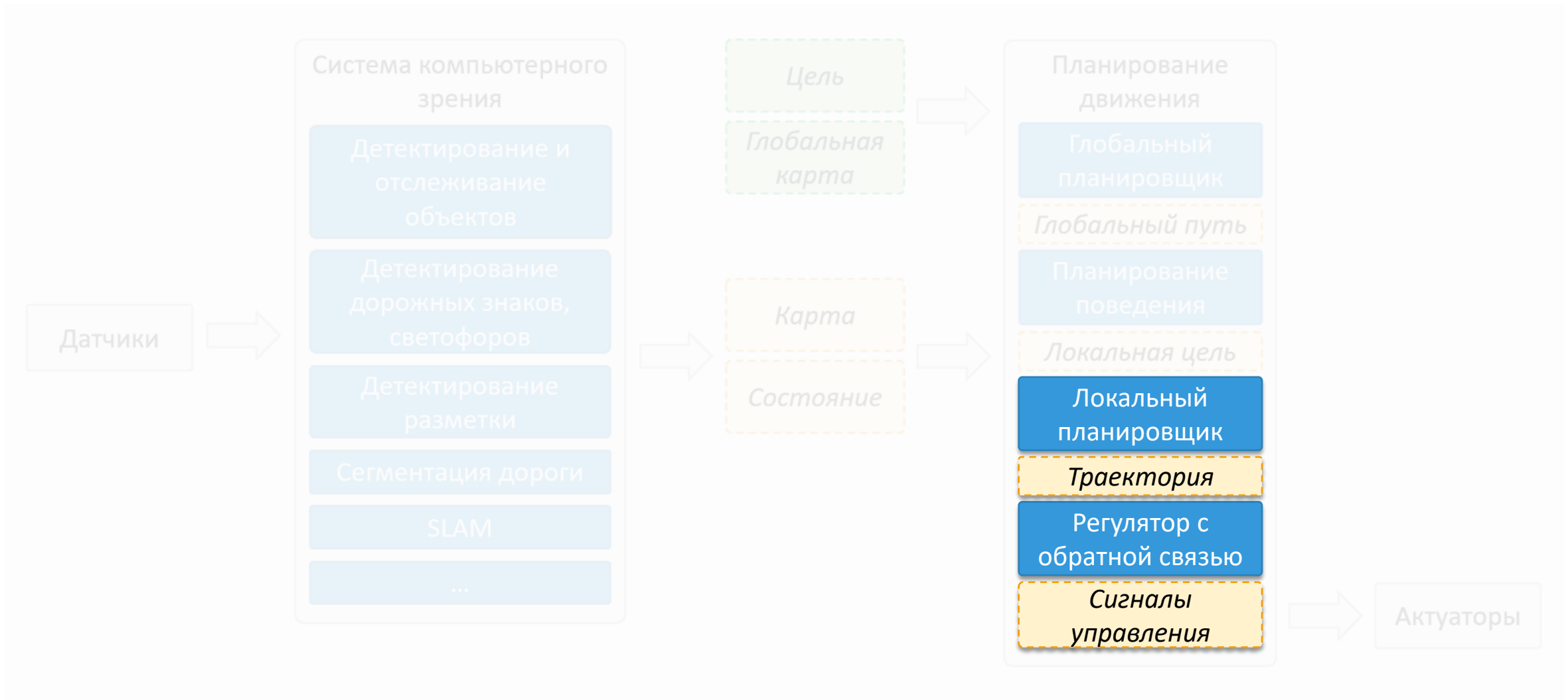
Общая структура системы управления



Общая структура системы управления



Общая структура системы управления



Планирование траектории

Основные методы:

- Поиск на графе
- Случайные методы (sample-based methods)
- Методы интерполяции кривыми
- ...

Планирование траектории

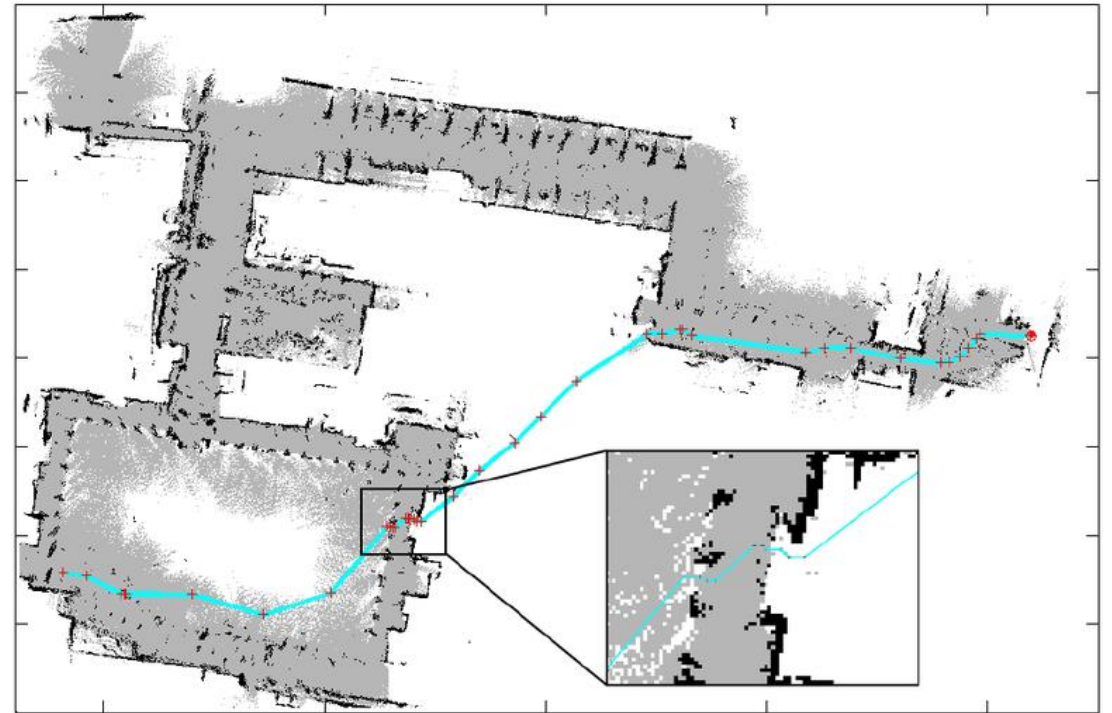
Методы поиска на графах

1. Представить пространство в виде графа
2. Найти оптимальный маршрут на графе (A^* , Дейкстра и т.д.)

Планирование траектории

Методы поиска на графах: Occupancy Grid

- + Простота реализации
- Не учитывает динамические ограничения
- Дискретность состояний
- Объем резко возрастает при уменьшении дискретизации, увеличении области и размерности

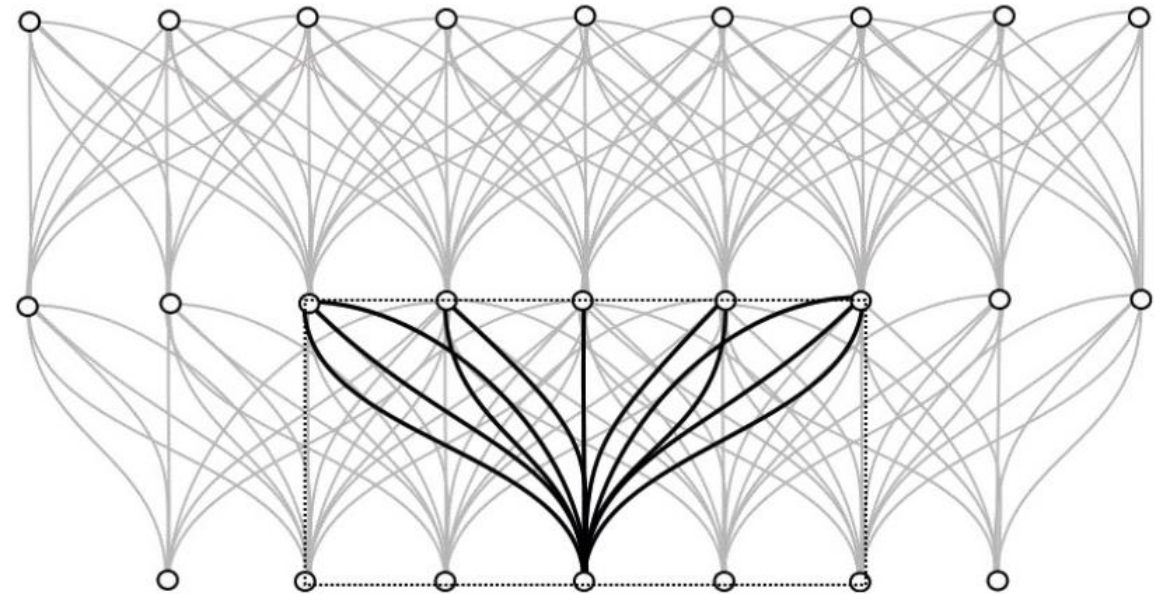


Планирование траектории

Методы поиска на графах: State Lattice

Регулярная решетка, в которой все узлы динамически достижимы

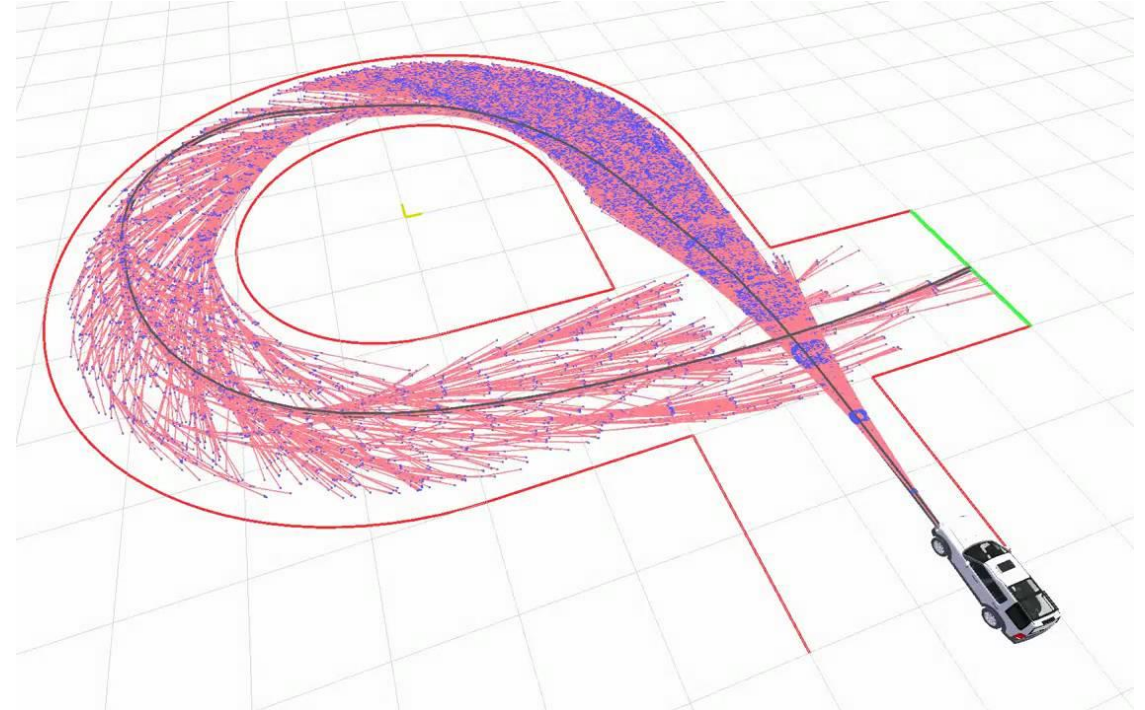
- + Построение динамически достижимых траекторий
- + Уменьшение вычислительной сложности за счет предварительного расчета
- Дискретность состояний
- Объем резко возрастает при уменьшении дискретизации, увеличении области и размерности



Планирование траектории

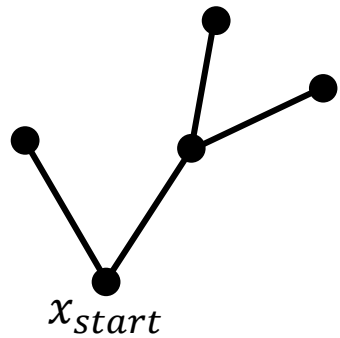
Sample-based methods

- + Rapidly Exploring Random Trees (RRT) и вариации
- + Возможность планирования сложных маневров
- + Построение динамически достижимых траекторий
- + Применимость к высоким размерностям
- Высокая вычислительная сложность
- Трудности с поиском оптимального решения

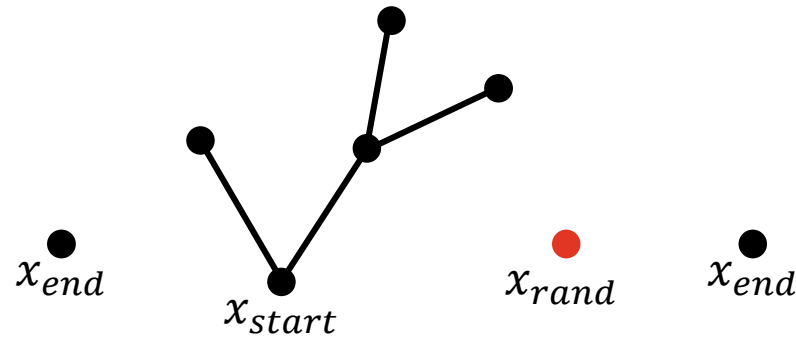


Планирование траектории

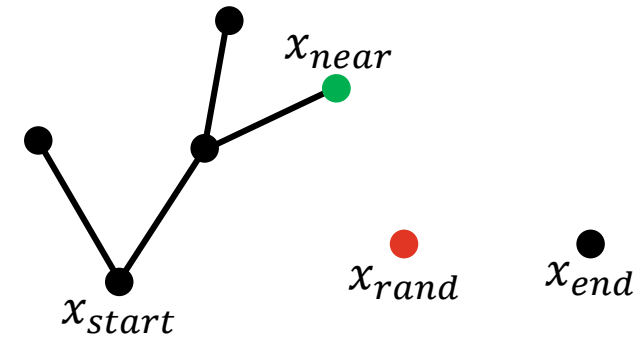
Sample-based methods



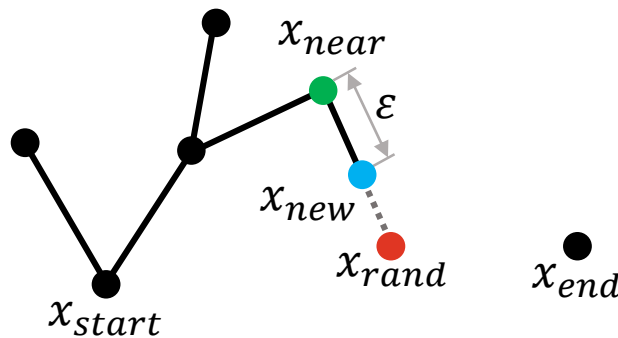
1. Исходный граф



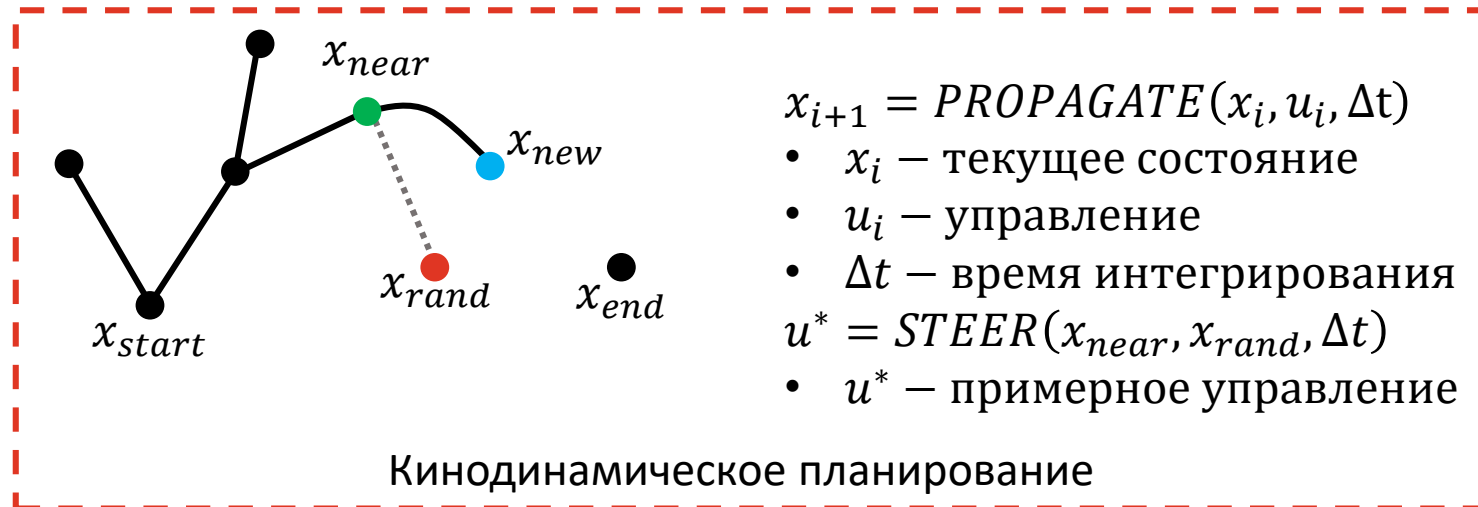
2. Выбираем случайную точку



3. Находим ближайшую к ней



3. Если нет препятствий, добавляем



$$x_{i+1} = \text{PROPAGATE}(x_i, u_i, \Delta t)$$

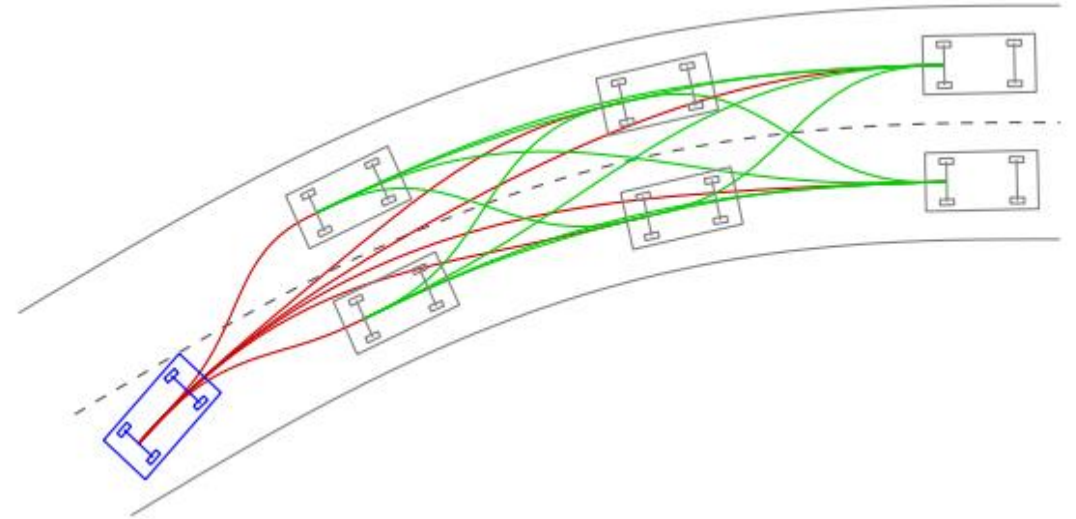
- x_i – текущее состояние
 - u_i – управление
 - Δt – время интегрирования
- $$u^* = \text{STEER}(x_{\text{near}}, x_{\text{rand}}, \Delta t)$$
- u^* – примерное управление

Кинодинамическое планирование

Планирование траектории

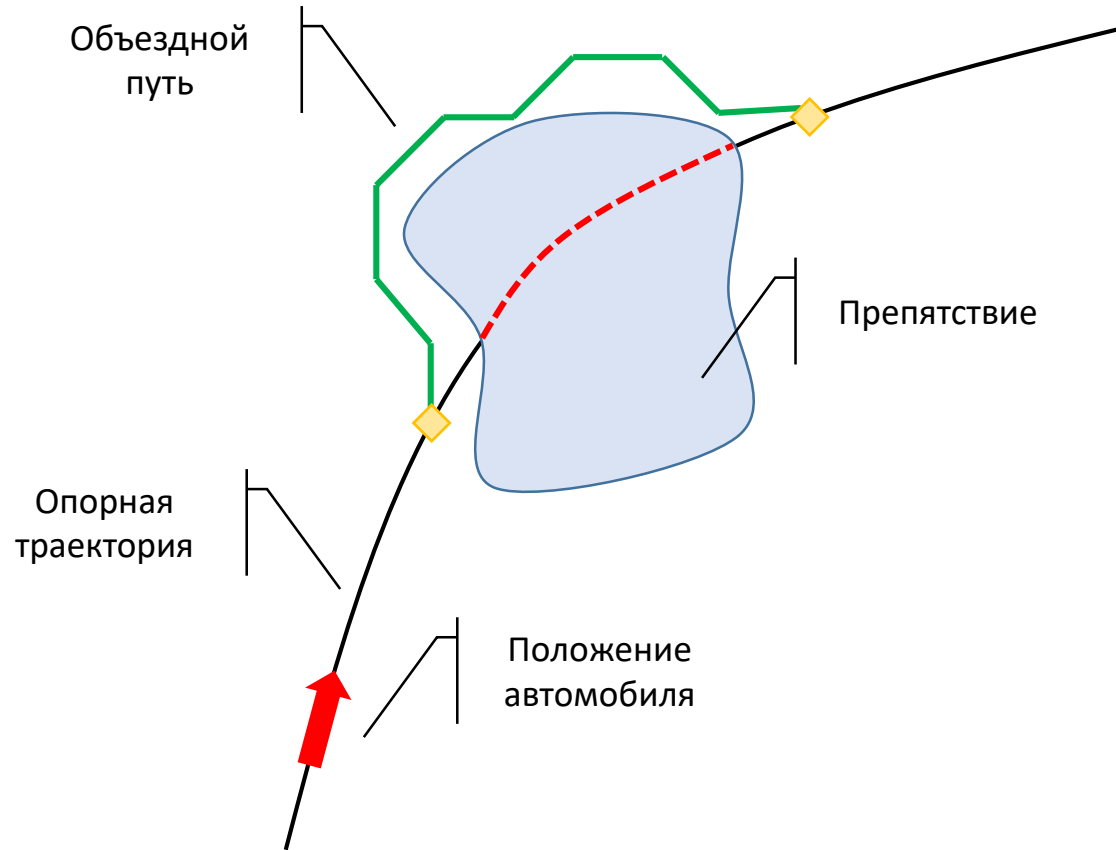
Методы интерполяции кривыми

- + Вычислительная сложность ниже, чем у RRT
- + Возможно построение динамически достижимых траекторий (с некоторыми ограничениями)
- + Возможно нахождение оптимальной траектории
- Не полноценный учет динамики
- Ограниченная область применимости



2. Проектирование системы управления

Вариант №1: A*



- Не учитывает кинематические и динамические ограничения транспортного средства
- Нестабилен

Вариант №2: интерполяция кривыми

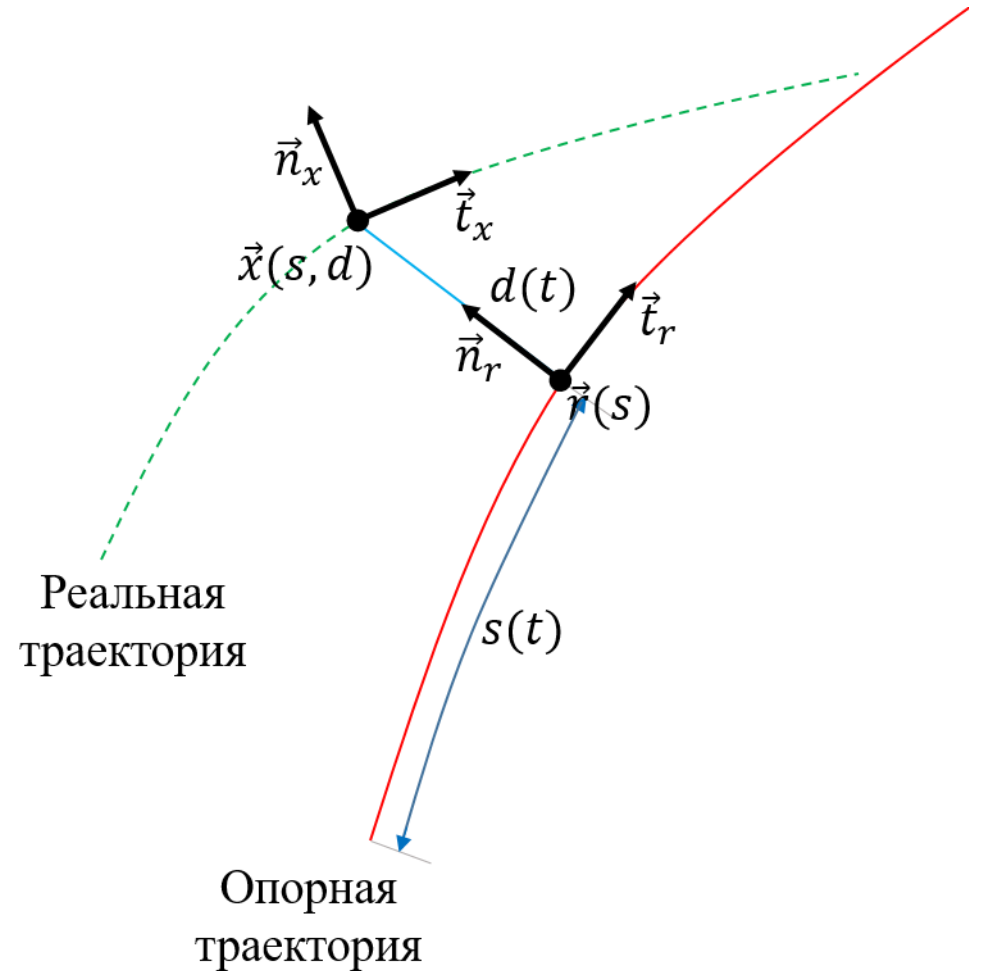
- Выбран метод интерполяции кривыми
 - За основу взят подход команды Junior из DARPA Urban Challenge
-
- M. Werling, J. Ziegler, S. Kammel and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenét Frame," *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK, 2010, pp. 987-993.
 - Montemerlo M. et al. (2009) Junior: The Stanford Entry in the Urban Challenge. In: Buehler M., Iagnemma K., Singh S. (eds) *The DARPA Urban Challenge*. Springer Tracts in Advanced Robotics, vol 56. Springer, Berlin, Heidelberg

Frenet Frame

Планирование в системе координат, движущейся вместе с автомобилем по опорной траектории

Поэтому движение можно представить в виде комбинации **продольного** и **поперечного**

- $s(t)$ – покрытая длина дуги
- \vec{t}_r – тангенциальный вектор
- \vec{n}_r – нормальный вектор
- $\vec{x}(s, d)$ – положение автомобиля
- $d(t)$ – отклонение от идеальной траектории



Функционал стоимости

За основу взята минимизация рывков (jerk) \dot{a} :

$$J = \int_0^T \ddot{x}^2(t) dt$$

- + Построение траекторий с более плавными маневрами и меньшим количеством маневров
- + Минимизация рывков способствует комфорту пассажиров

Функционал стоимости

Поперечное движение

$$C_d = K_{dj} \int_0^T \ddot{d}(t)^2 dt + K_d d(T)^2 + K_{dt} T$$

Продольное движение

$$C_s = K_{sj} \int_0^T \ddot{s}(t)^2 dt + K_s (s(T) - S_1)^2 + K_v (\dot{s}(T) - \dot{S}_1)^2 + K_{st} T$$

Объединенная

$$C = K_{lat} C_d + K_{lon} C_s$$

Полиномы пятого порядка

Оптимальные траектории могут быть найдены в форме полиномов пятого порядка

$$x(t) = a_0 t^5 + a_1 t^4 + a_2 t^3 + a_3 t^2 + a_4 t + a_5$$

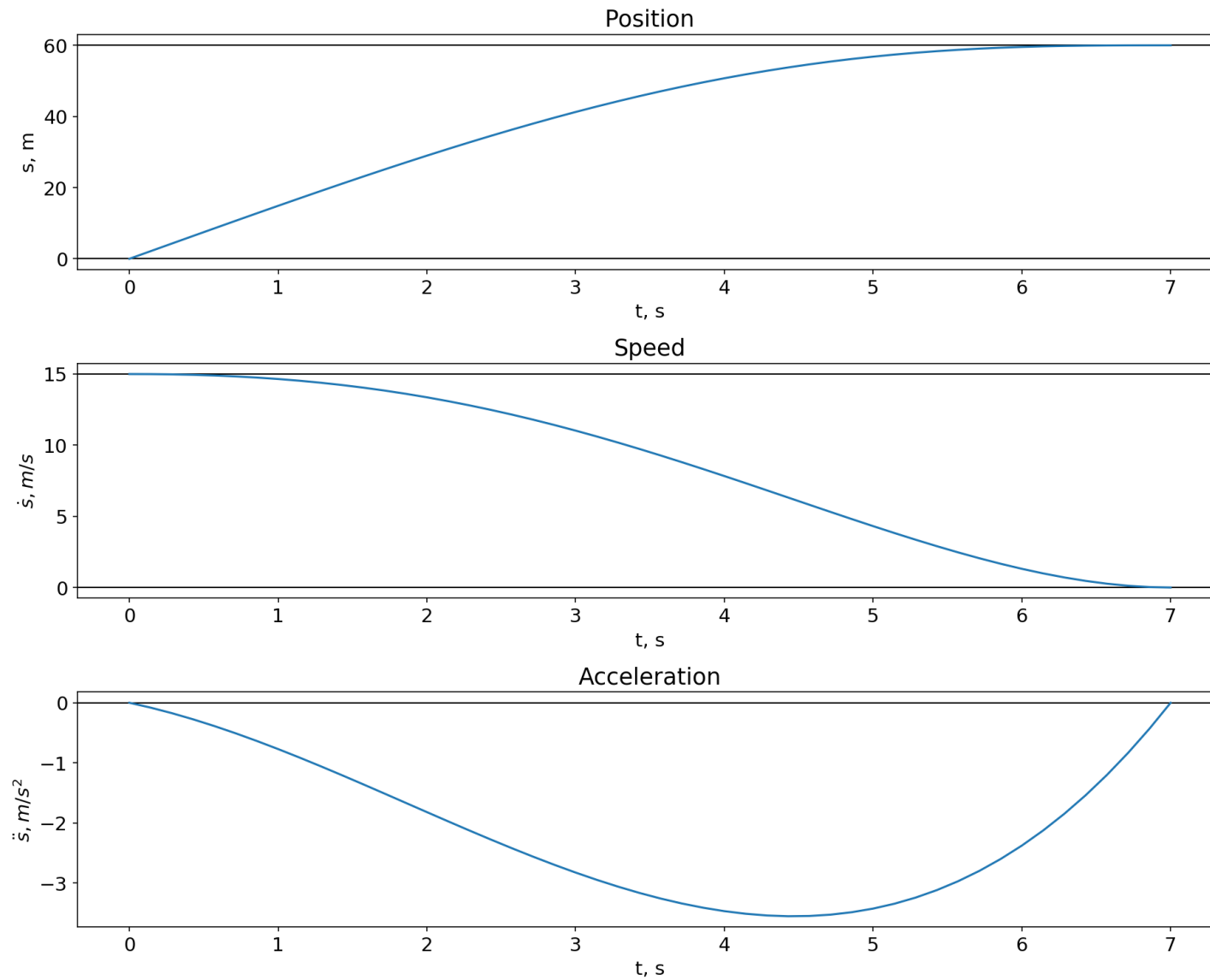
Расчет:

- Начальное состояние $P_0 = [x(0), \dot{x}(0), \ddot{x}(0)]$
- Конечное состояние $P_1 = [x(T), \dot{x}(T), \ddot{x}(T)]$
- Длительность маневра T

+ Гладко стыкуются $x(t)$ и $\dot{x}(t)$

+ Оптимизирует функцию стоимости

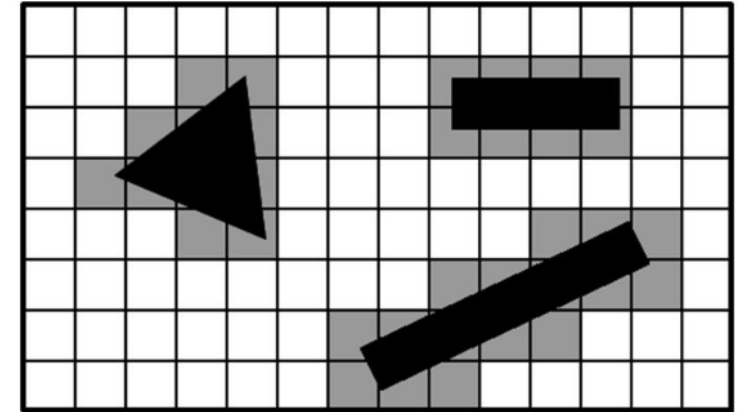
Полиномы пятого порядка



Оптимизация и удовлетворение ограничениям

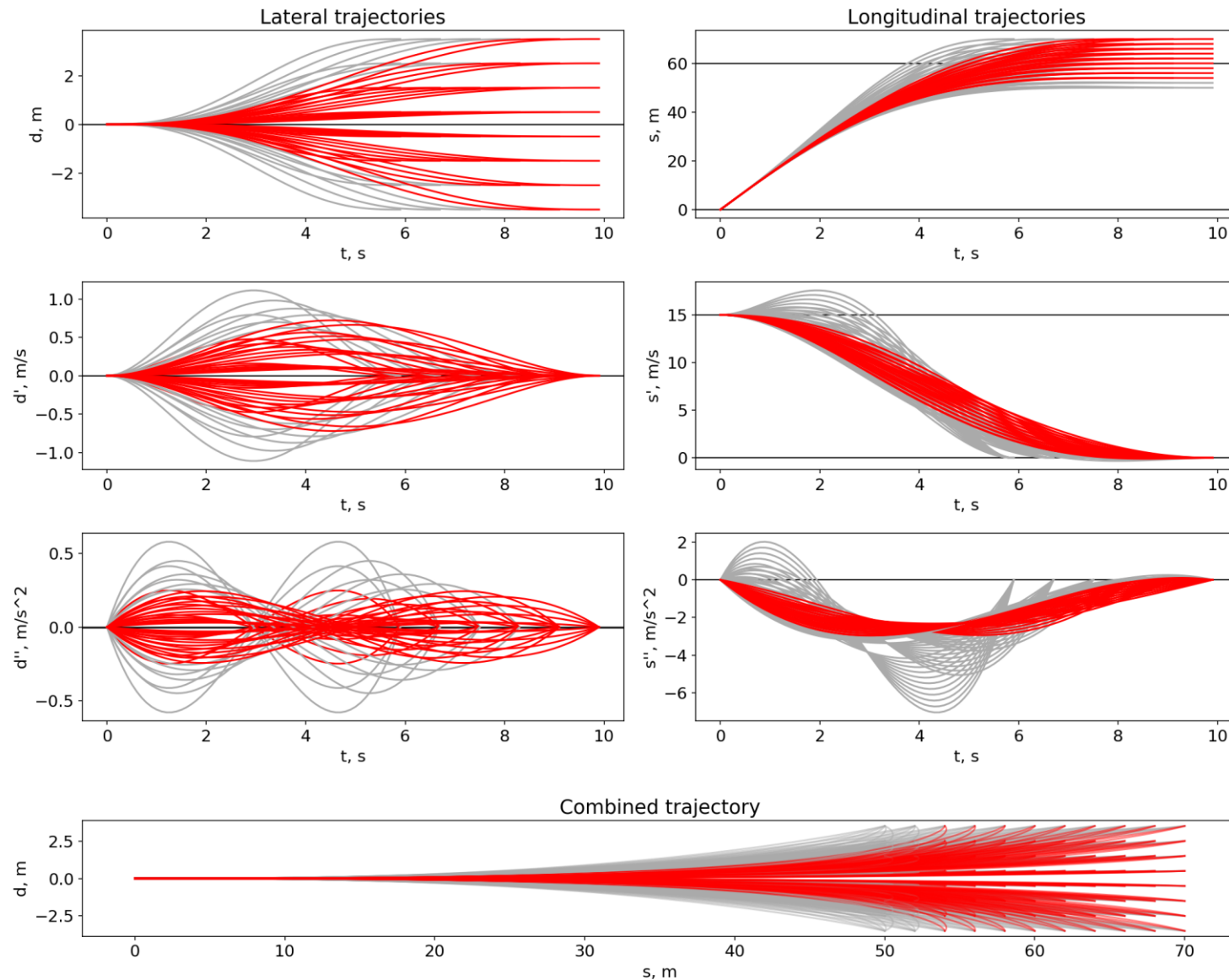
Требуется найти траекторию, которая:

- Минимизирует функционал стоимости
- Удовлетворяет ограничениям
 - Максимальная продольная скорость, максимальное продольное и поперечное ускорение, минимальный радиус кривизны
- Не пересекается с препятствиями, представленными в виде Occupancy Grid



Решение: сформировать набор траекторий-кандидатов и выбрать оптимальную среди тех, которая удовлетворяет ограничениям

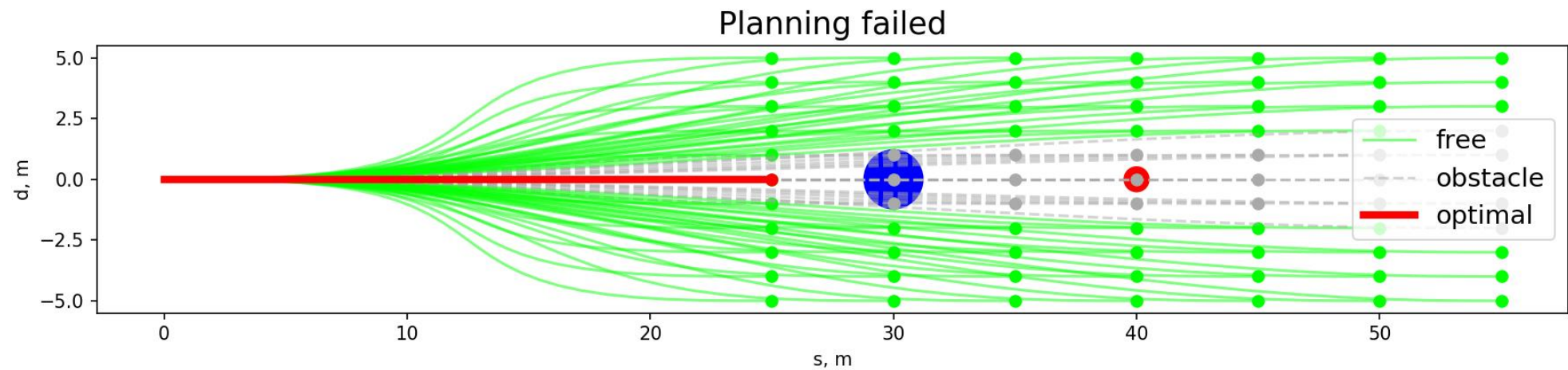
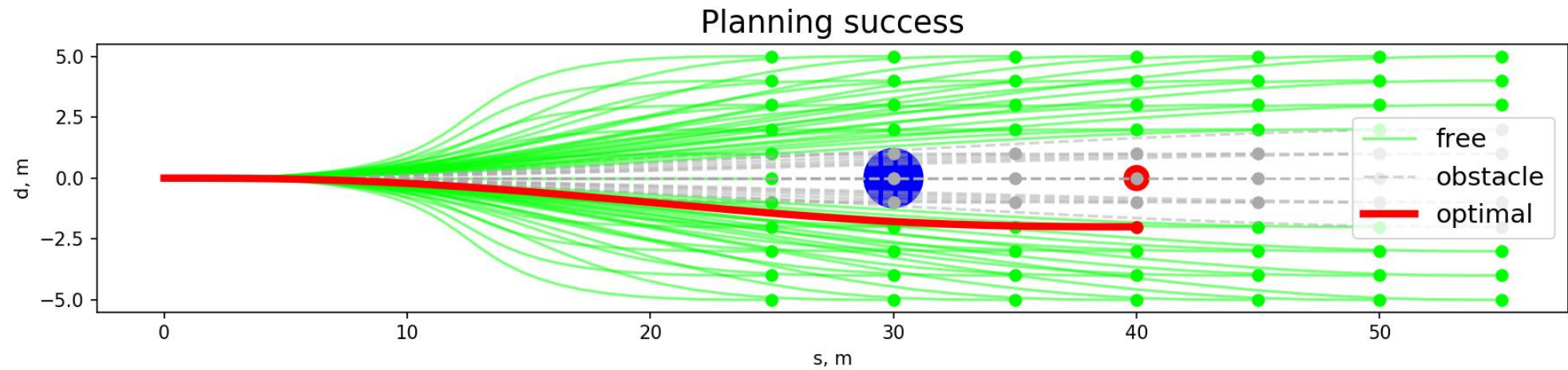
Оптимизация и удовлетворение ограничениям



Выявленные недостатки

- Сильная зависимость от весовых коэффициентов в функционале стоимости
- Проблема локального минимума: «предпочитает» выбирать оптимальные в данный момент траектории тем, которые позволяют лучше достичь цели

Выявленные недостатки



● Препятствие
● Цель

Вариант №3: Вариант №2 + граф

Планирование на несколько шагов вперед позволит выбрать траекторию, которая лучше достигает цели, даже если в данный момент она не оптимальна

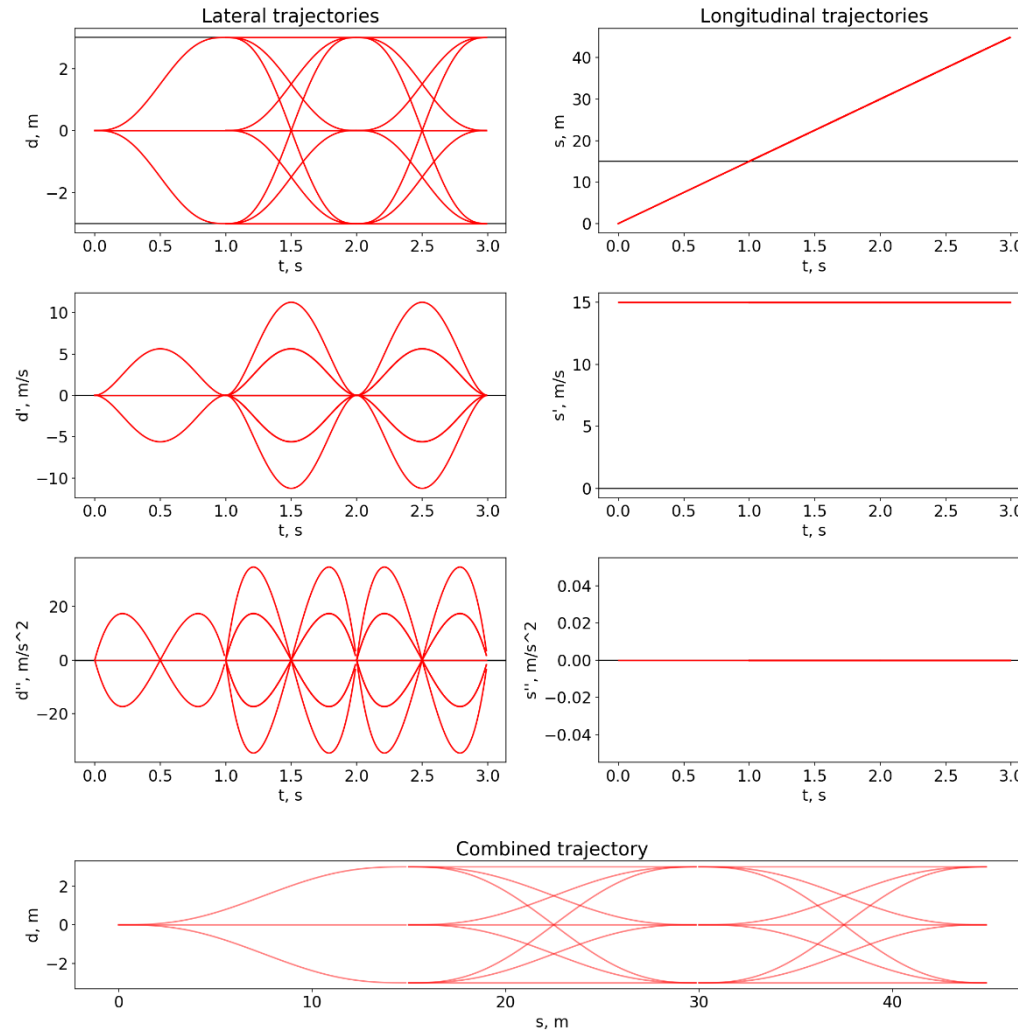
1. Построить граф, где вершины – состояния, ребра – траектории в форме полиномов пятого порядка
2. Найти оптимальную вершину и кратчайший путь

$$C = C_{path} + C_{vert}$$

$$C_{path} = K_{lon}K_{sj} \int_0^T \ddot{s}(t)^2 + K_{lat}K_{dj} \int_0^T \ddot{d}(t)^2 dt$$

$$C_{vert} + K_{lon} \left[K_s (s(T) - s_1)^2 + K_v (\dot{s}(T) - \dot{s}_1)^2 + K_{st}T \right] + K_{lat} [K_d d(T)^2 + K_{dt}T]$$

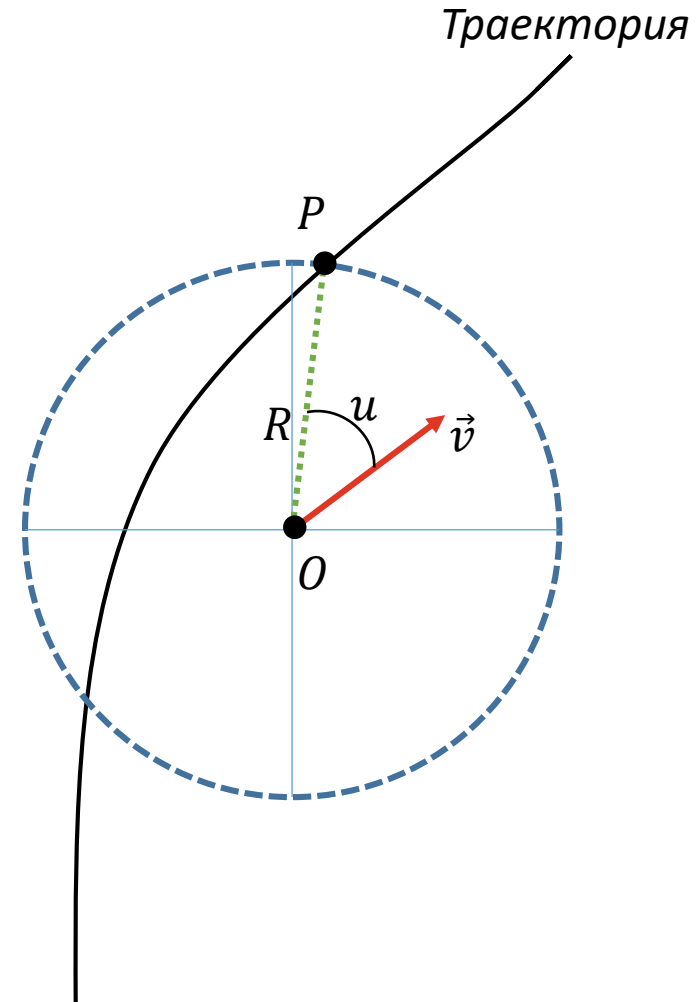
Вариант №3: Вариант №2 + граф



Регулятор с обратной связью

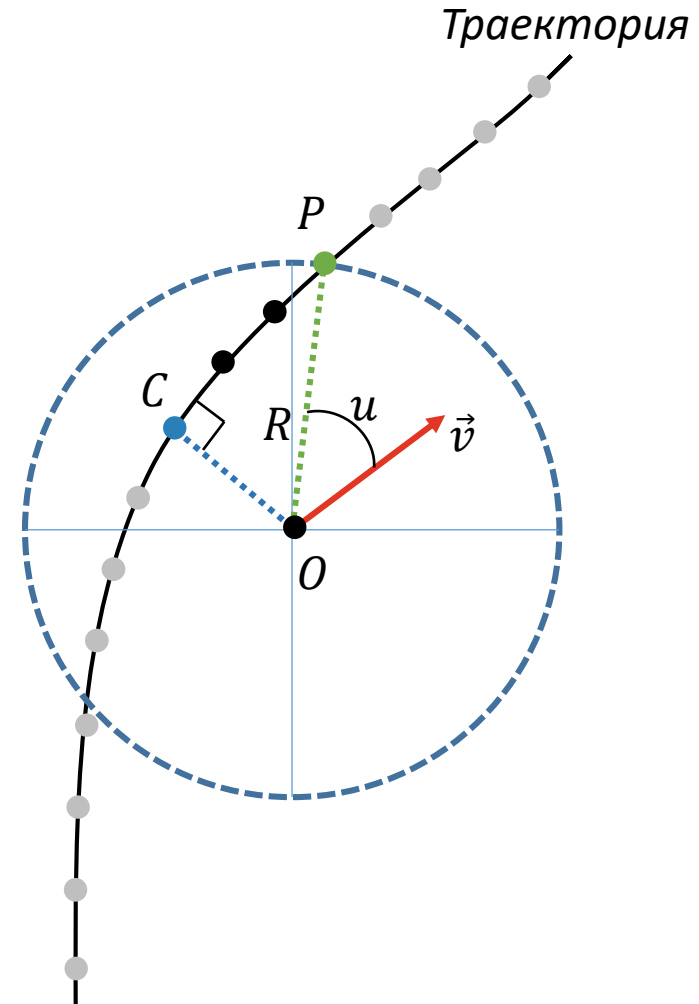
Регулятор основан на модель
МакАдама

- Выбирается точка P впереди по траектории на расстоянии R от текущего положения O
- Управление - угол между текущим вектором скорости \vec{v} \overrightarrow{OP}
- Применяется пропорциональный (П) регулятор



Регулятор с обратной связью

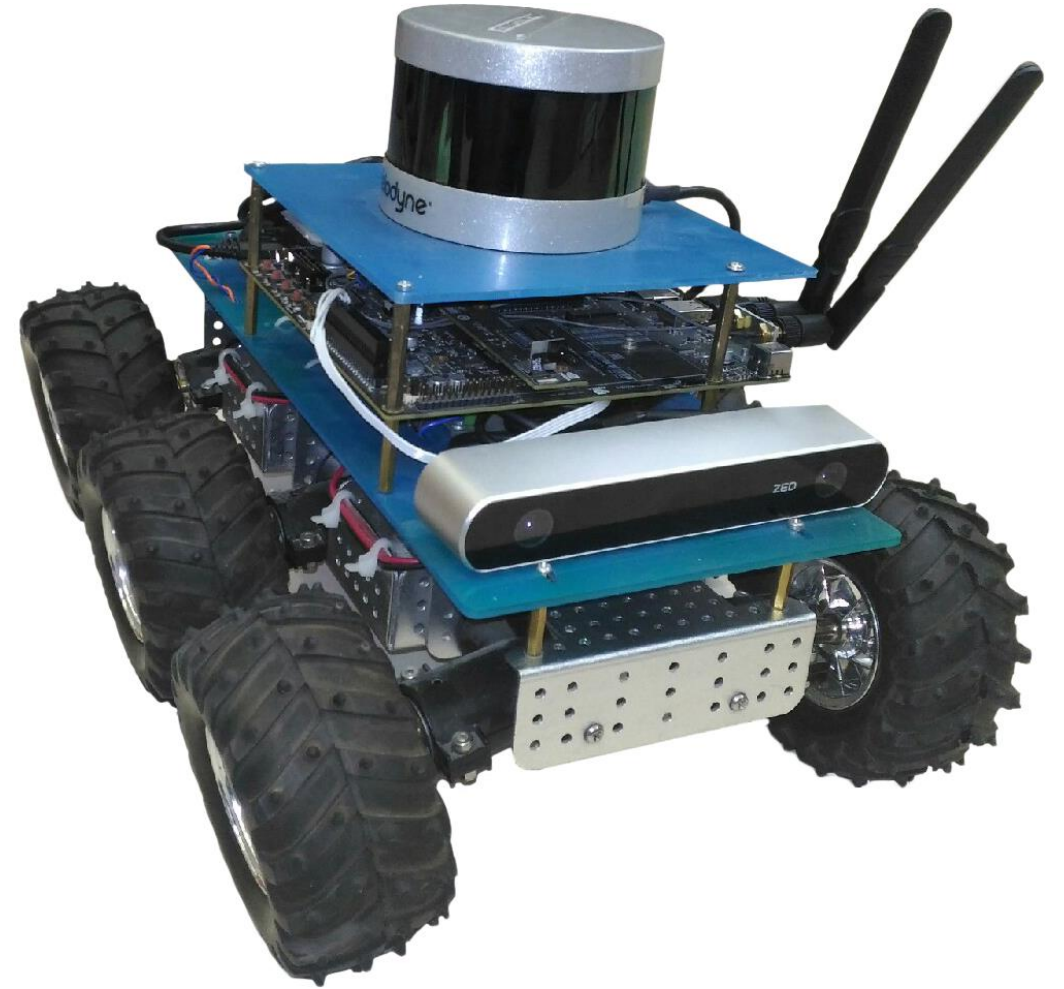
- Траектория представлена в виде большого количества точек
- Выбирается точка C ближайшая к текущему положению автомобиля O
- От точки C вперед по траектории отсчитываются точки, до тех пор, пока расстояние $|\overrightarrow{OP}| < R$
- Найдена точка P такая, что $|\overrightarrow{OP}| \sim R$



3. Реализация системы управления

Экспериментальная платформа

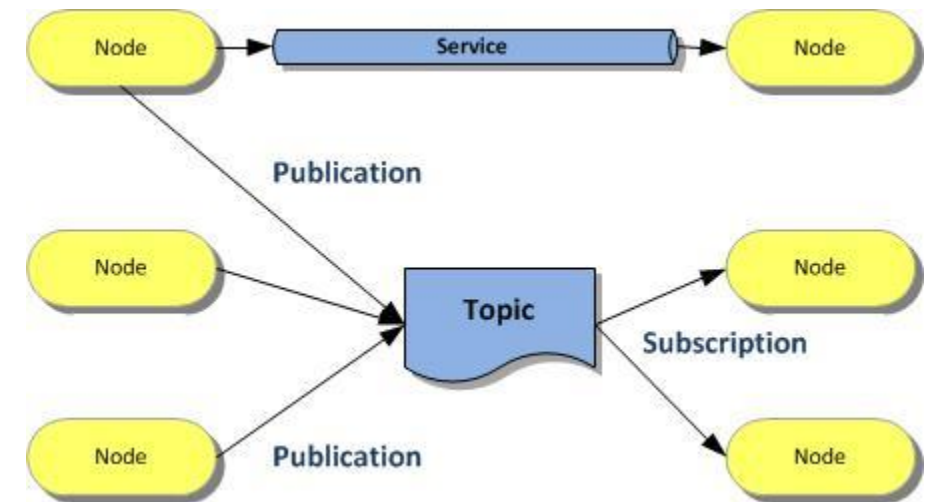
- NVidia Jetson TX2
- ZED-камера
- Velodyne VLP-16



ROS

ROS – фреймворк для разработки программного обеспечения для роботов.

- Распределенный модульный дизайн
- Готовая архитектура взаимодействия между модулями (топики)
- Множество готовых библиотек для управления, компьютерного зрения и визуализации
- Подробная документация
- Обширное сообщество разработчиков

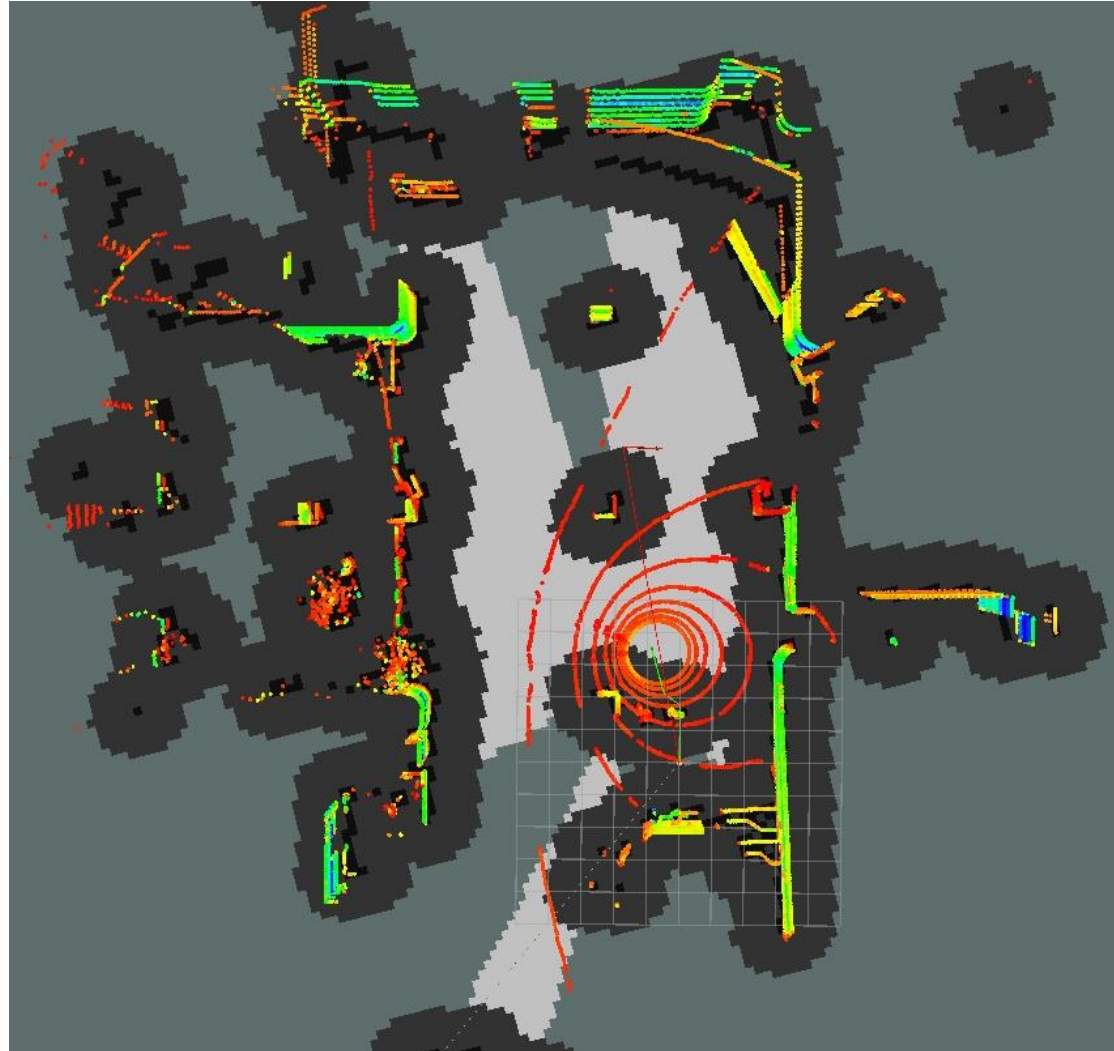


Подсистема обнаружения препятствий

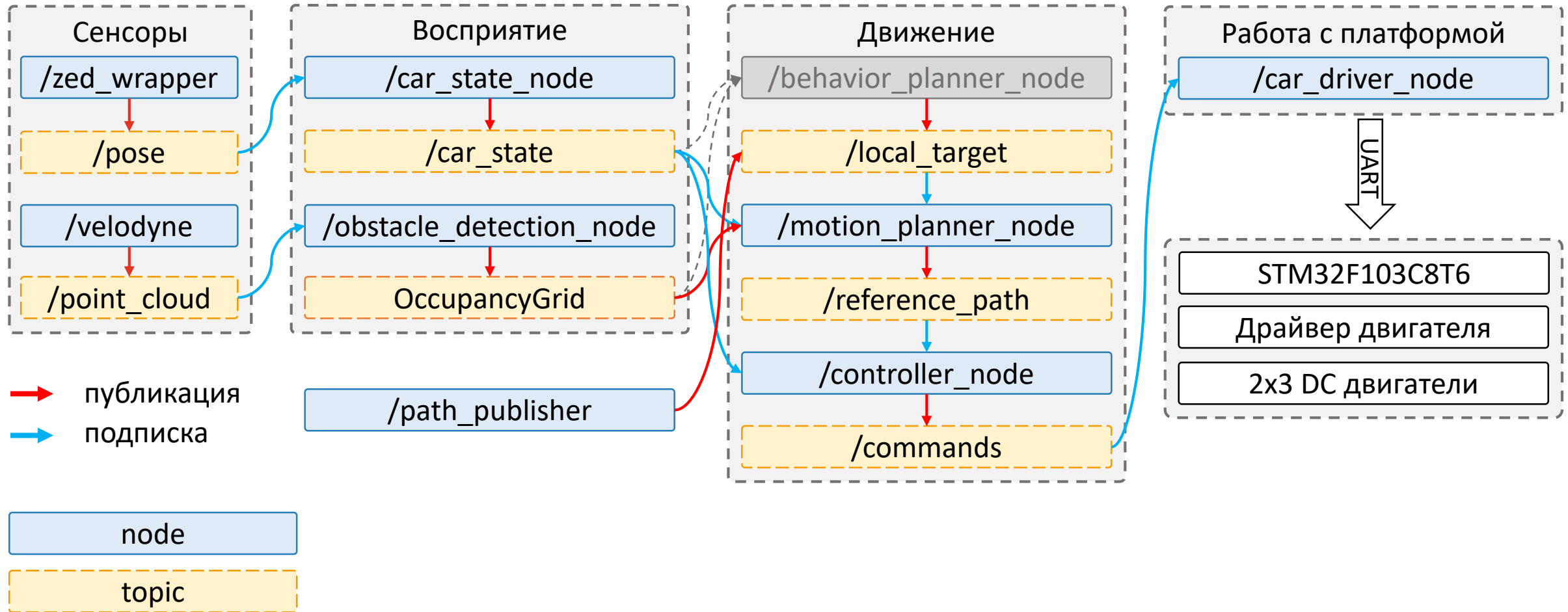
Алгоритм:

1. Получить облако точек
2. Выделить плоскость с помощью алгоритма RANSAC
3. Представить в виде OccupancyGrid
4. Произвести трассировку пути с помощью алгоритма Брезенхема для растрирования линии
5. Расширить каждую клетку препятствия кругом заданного размера с помощью алгоритма Брезенхема для растрирования круга

Подсистема обнаружения препятствий

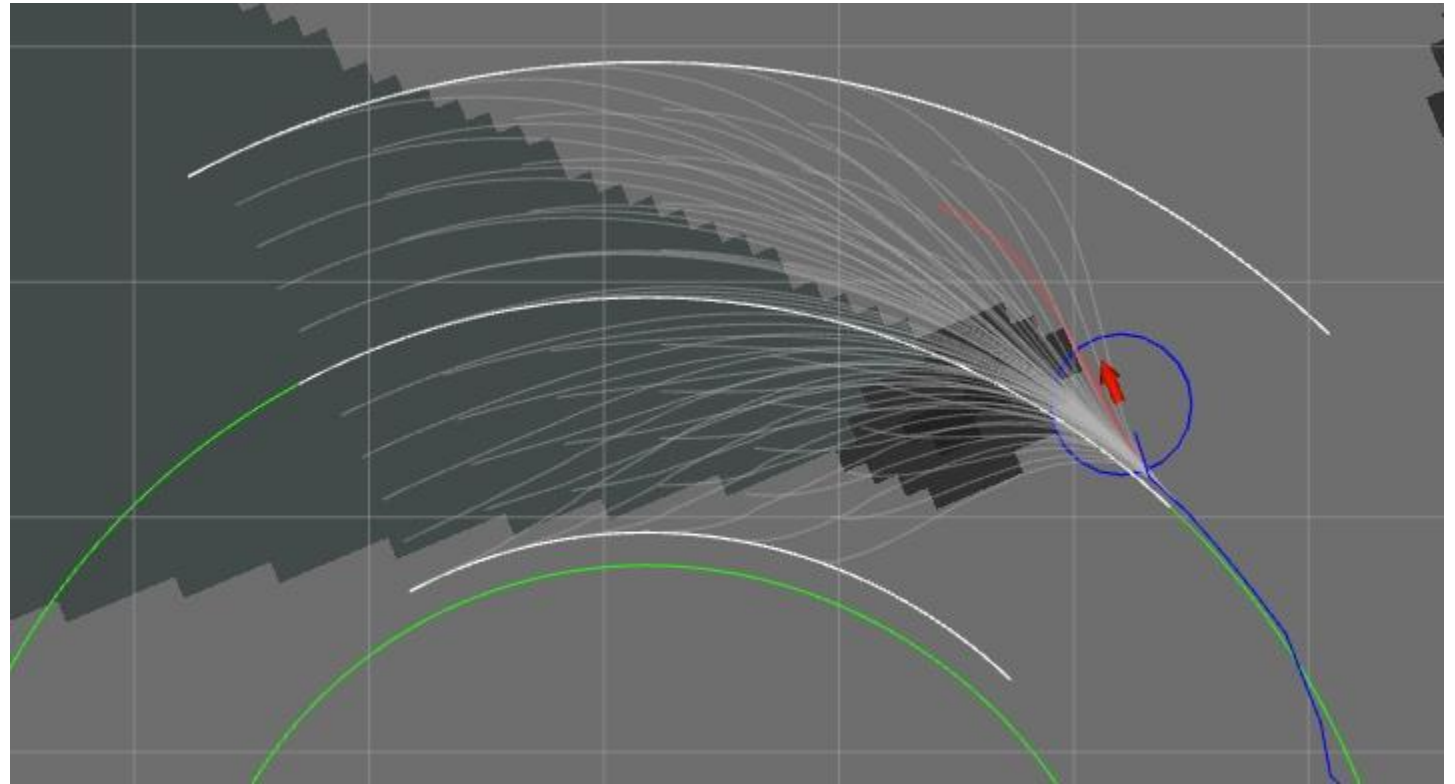
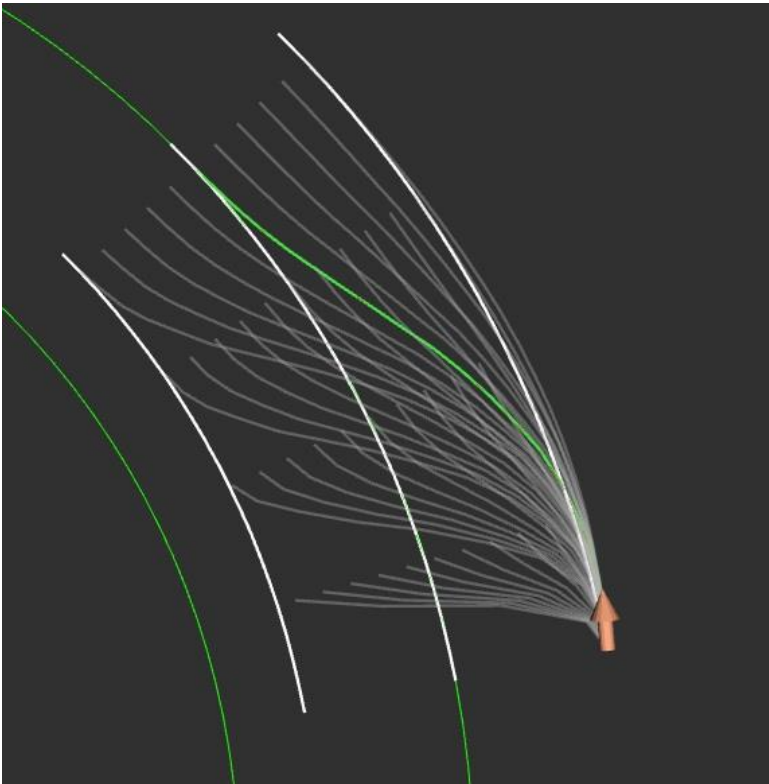


Структура программы управления

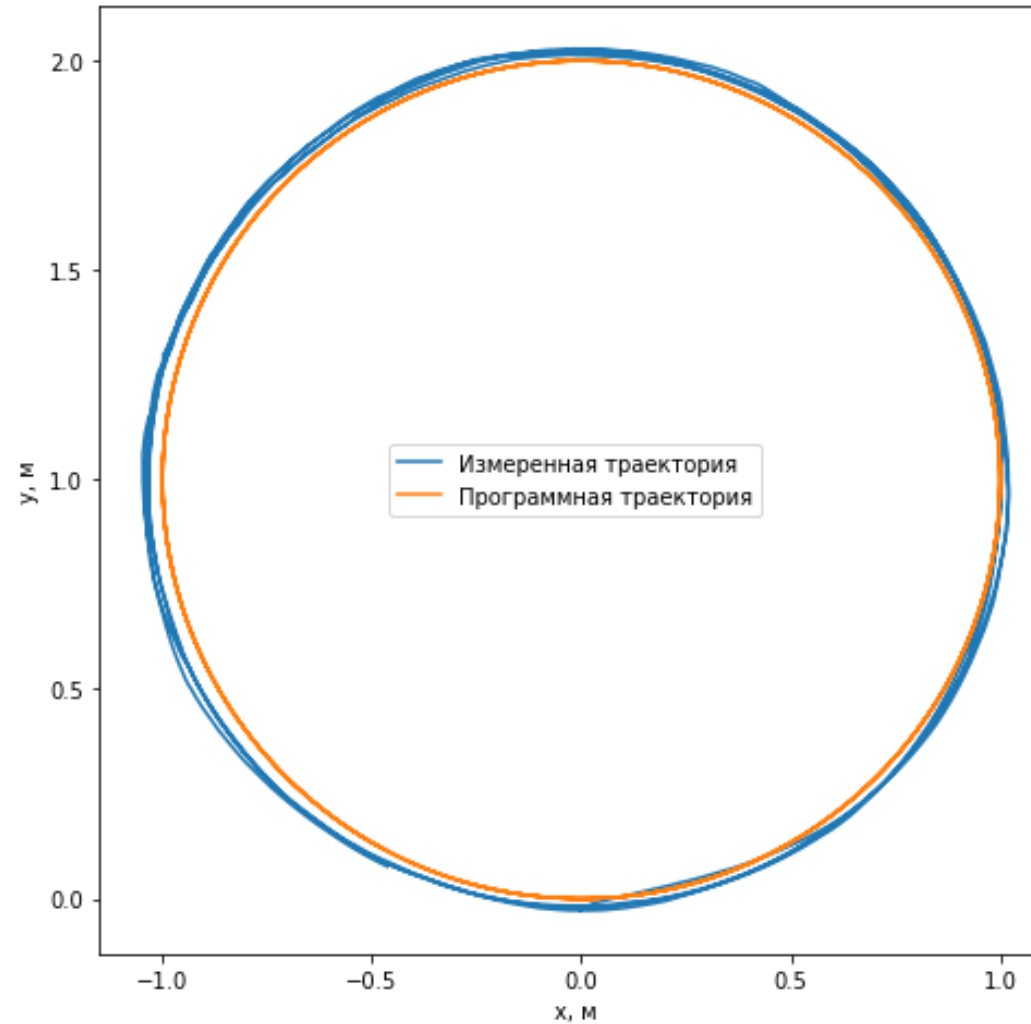


4. Результаты

Планирование траектории



Следование по траектории



Научная новизна

1. Разработан алгоритм построения программной траектории для наземного беспилотного транспортного средства, осуществляющий построение траектории в форме полиномов пятого порядка, отличающийся от существующего добавлением поиском кратчайшего пути на графе состояний, что позволяет осуществлять планирование на несколько шагов вперед
2. Разработан регулятор для движения по программной траектории с использованием обратной связи от системы SLAM

Спасибо за
внимание!