

# Мобильная робототехника : навигация и управление движением

# Введение

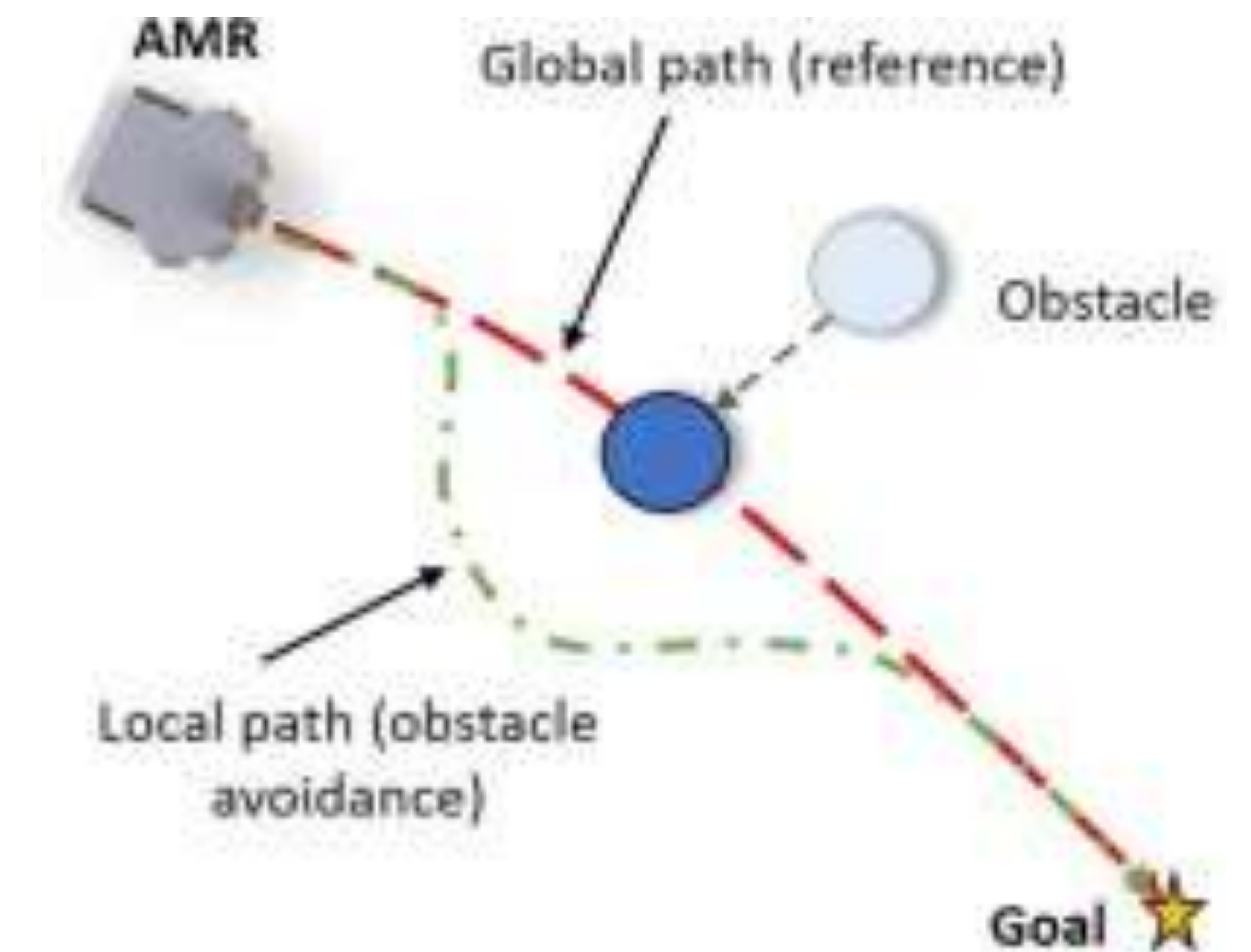
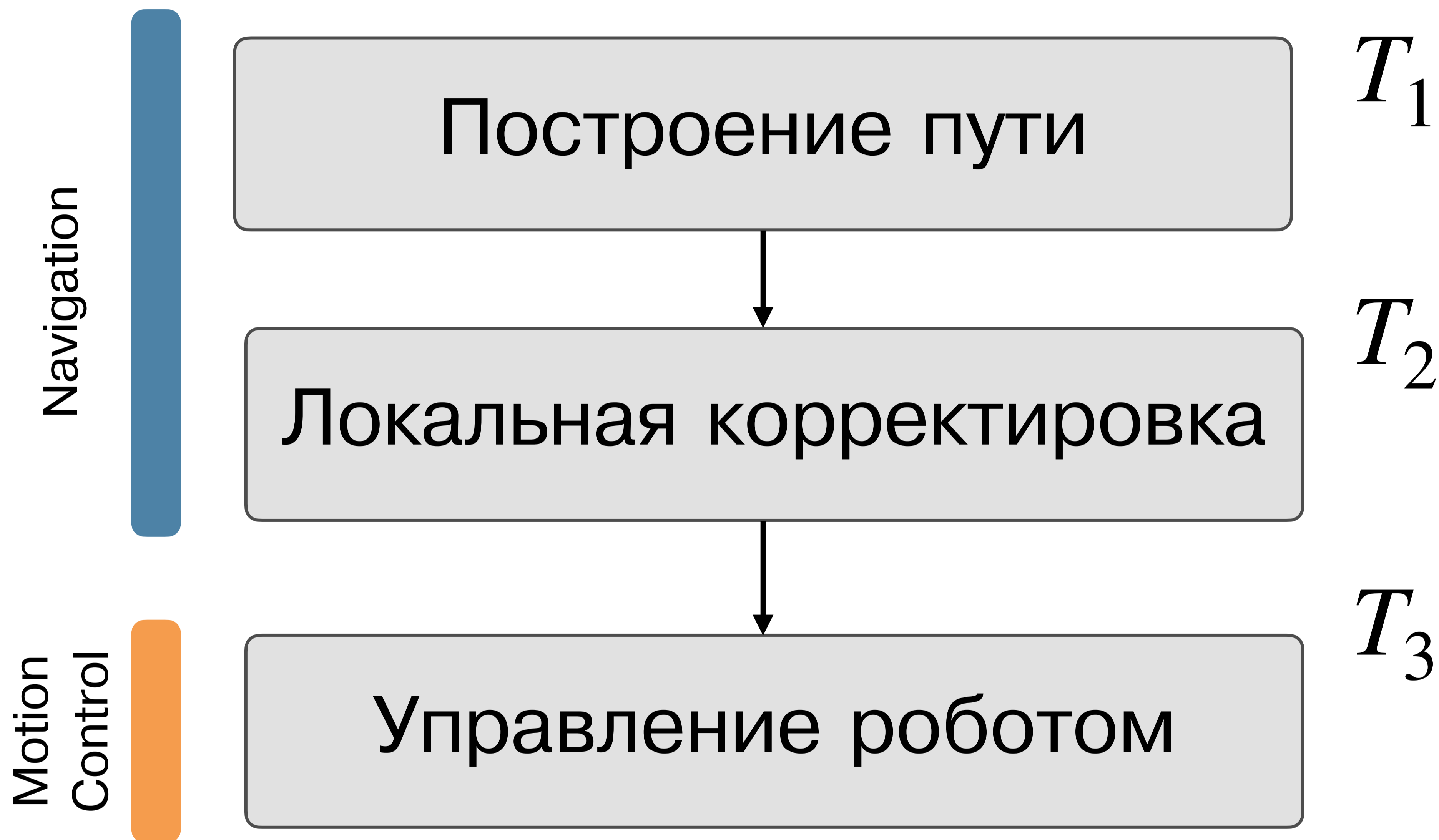
В мобильной робототехнике методы навигации и управления движением играют ключевую роль для обеспечения автономной работы роботов. Эти методы позволяют роботам ориентироваться в пространстве, избегать препятствий, планировать маршруты и выполнять задачи в различных условиях окружающей среды.

В общем виде, управление движением включает планирование траектории движения, локализацию робота относительно окружения и выполнение команд управления приводами робота.

Навигация же отвечает за определение текущего положения робота и выбор оптимального пути к цели.

# Терминология

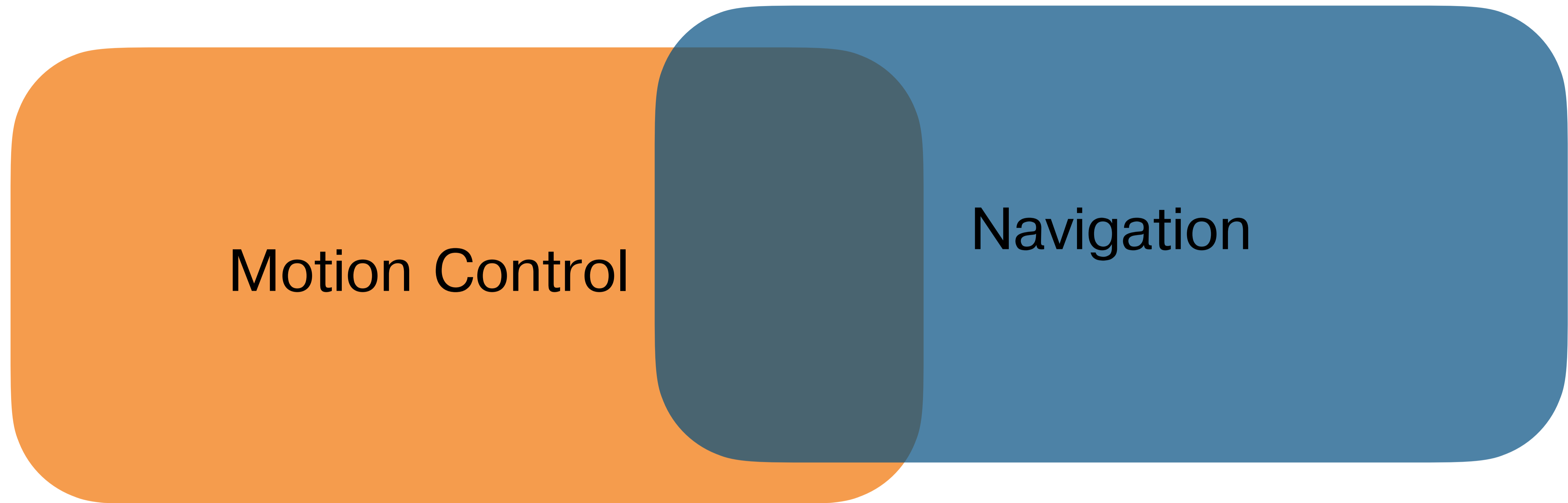
## Декомпозиция мобильного робота



$$T_1 > T_2 > T_3$$

# Терминология

## Граница между Motion Control и Navigation



- Следование намеченному пути
- Реакция на изменение состояния робота или среды

- Нахождение безопасного пути из точки А в точку Б
- Нахождение пути наименьшей длины на карте
- Перестройка части пути при получении новых данных об окружении
- Картографирование

# Классификация методов

Модель окружения (сцены)

Аналитическое представление сцены

Непрерывные

Методы  
потенциальных  
полей

Алгоритмы Bug

Дискретные

Алгоритмы на  
графах

Случайная  
дорожная карта

Метод  
Силуэтов

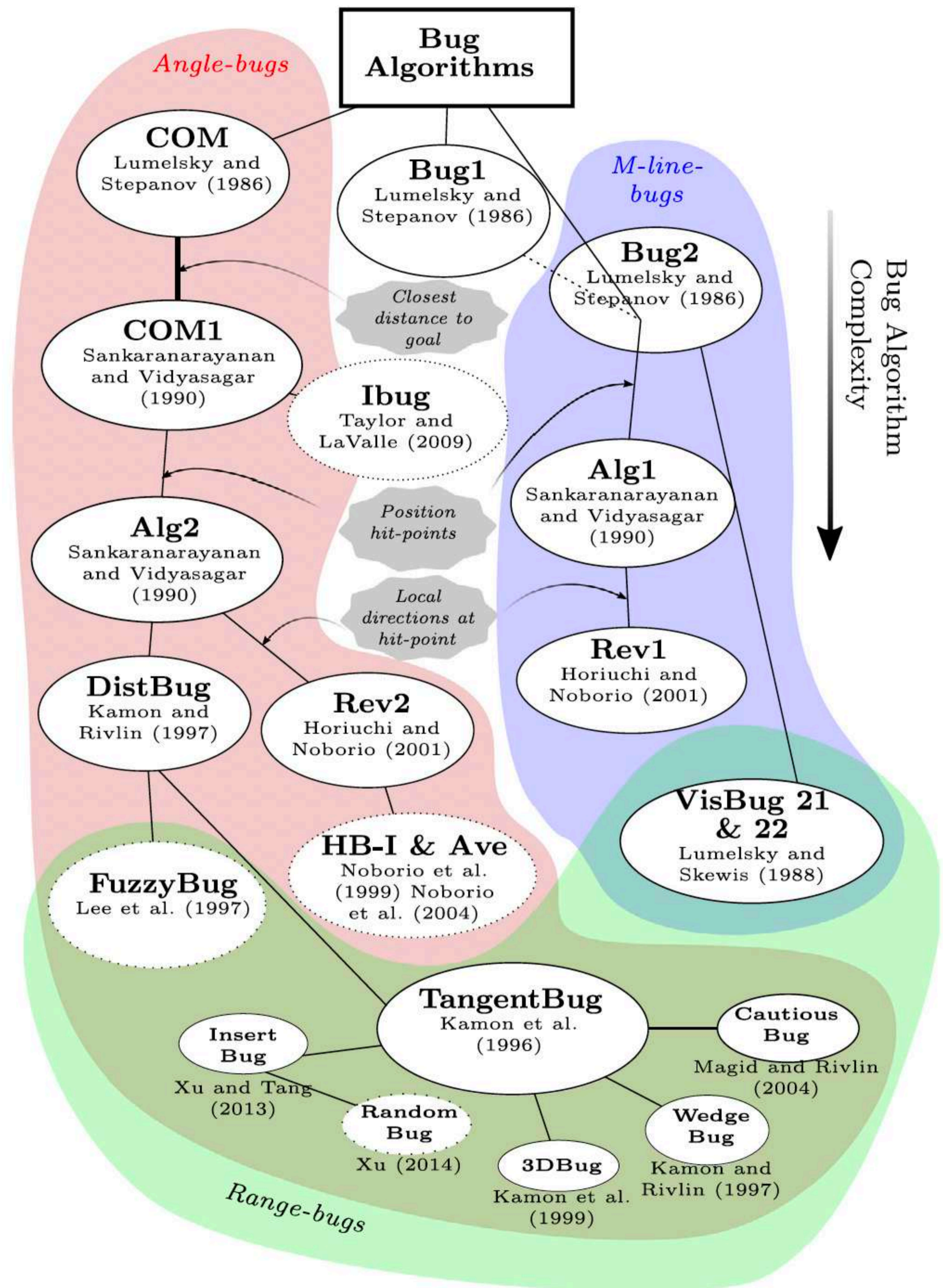
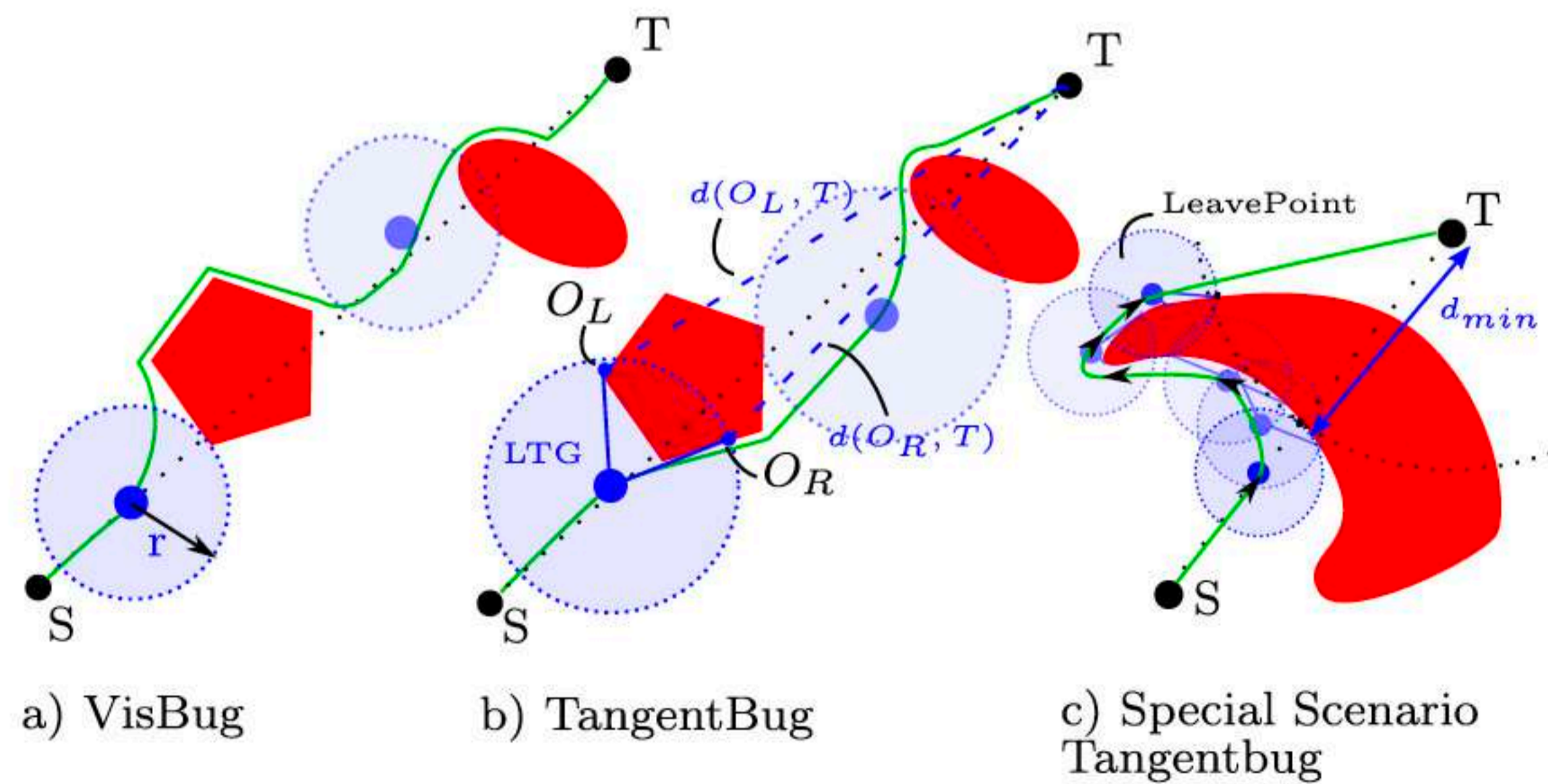
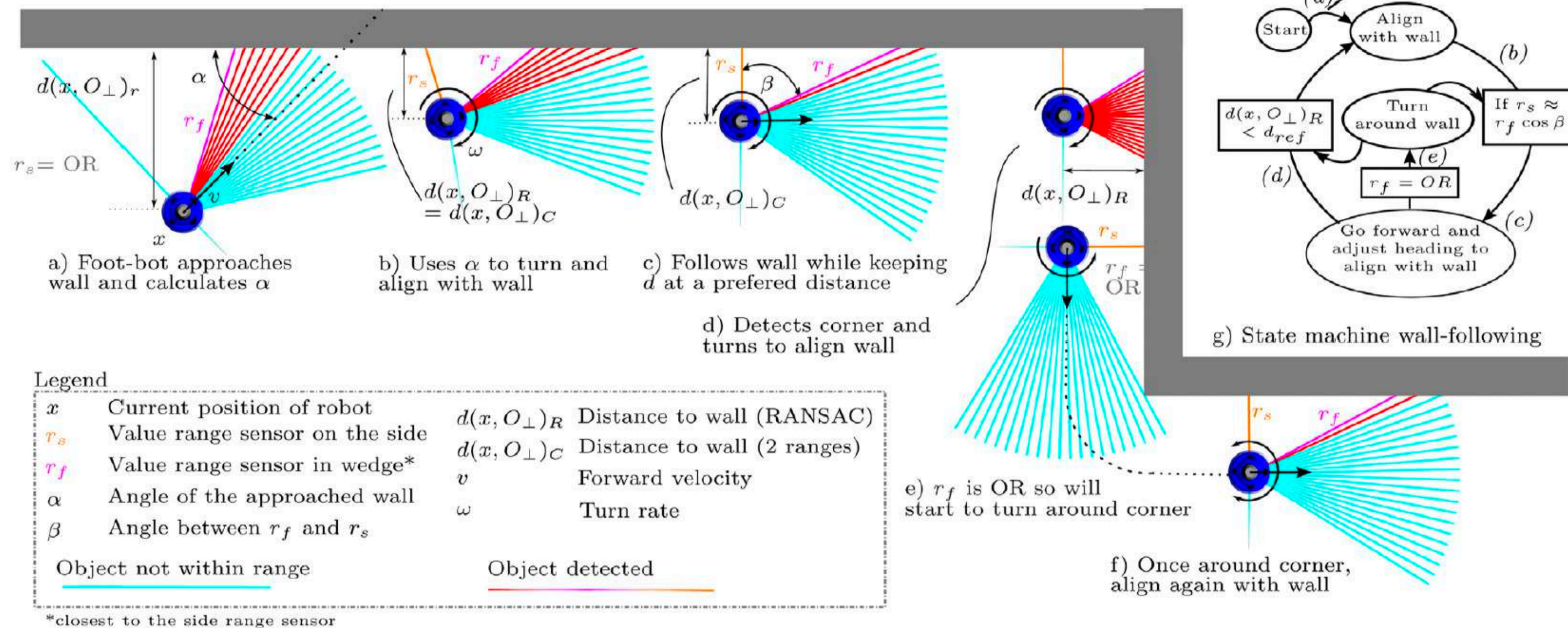
Эвристическое представление сцены

Нейросетевые алгоритмы

Нечёткая логика

# Классификация методов

## Семейство алгоритмов Bug



K.N. McGuire, G.C.H.E. de Croon and K. Tuyls / Robotics and Autonomous Systems 121 (2019)

# Отслеживание эквидистанты

## Пример реактивного регулятора

Неголономный неполноприводный мобильный планарный робот движется с постоянной скоростью  $\bar{v} = v > 0$  и управляется изменением направления движения

$$|\omega| < \bar{\omega}, \bar{\omega} > 0$$

Пободные роботы описываются моделью **машины Дубинса**

$$\begin{cases} \dot{\mathbf{r}} = \bar{v} \vec{e}(\theta) \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \\ \dot{\theta} = \omega \in [-\bar{\omega}, \bar{\omega}] \end{cases} \quad \mathbf{r}(0) = \mathbf{r}_{in}, \theta(0) = \theta_{in} \quad (1)$$

Уравнение (1) отражает способность робота двигаться вдоль пути с кривизной  $\geq R_{min} = \bar{v}\bar{\omega}^{-1}$

На плоскости располагаются  $N$  **неизвестных объектов**;  $i$  **ый** объект занимает множество  $D_i$

In its local frame and within a given **visibility distance**  $R_{vis}$ , the robot has access to the part of the scene that is in direct line of sight. Also, it is able to update the relative coordinates of the points so that to compensate for the robot's own motion, e.g., via odometry

Задачей является достижение и следование границы области

$$D := \bigcup_i D_i$$

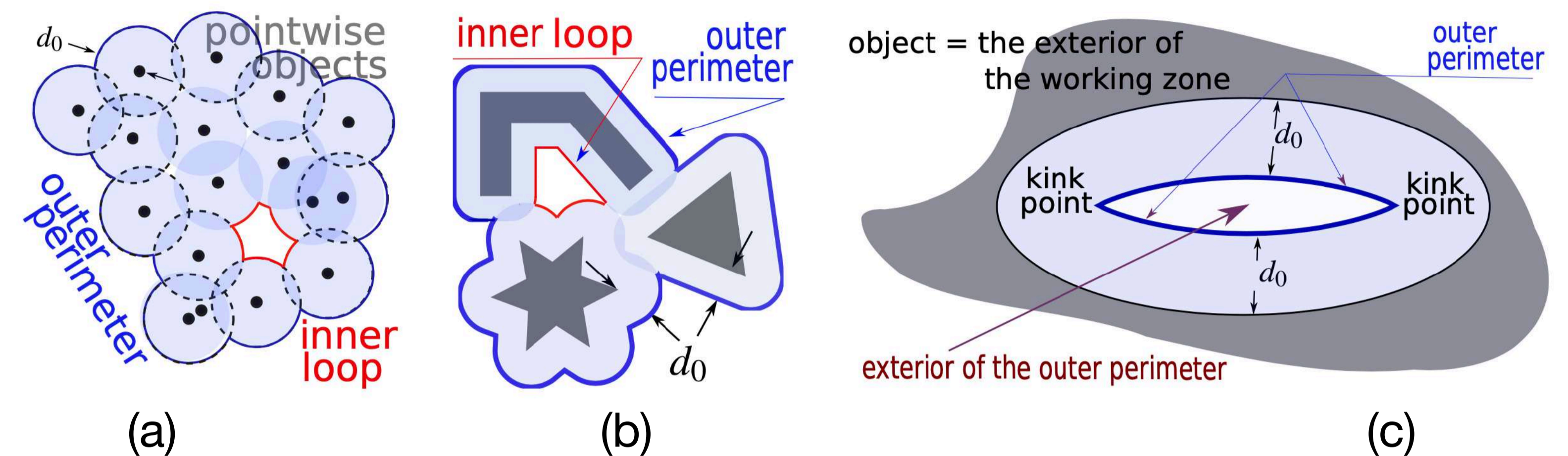


Рис.1. (a) Точечные объекты • (b) Протяжённые объекты; (c) Точки перегиба

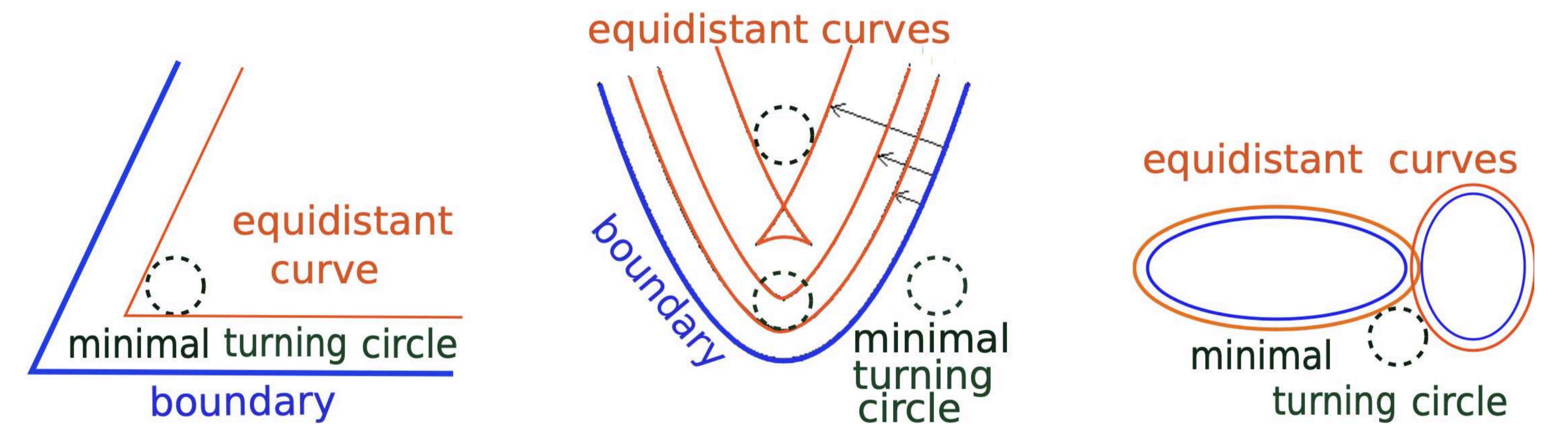


Рис.2. Проблемные эквидистанты

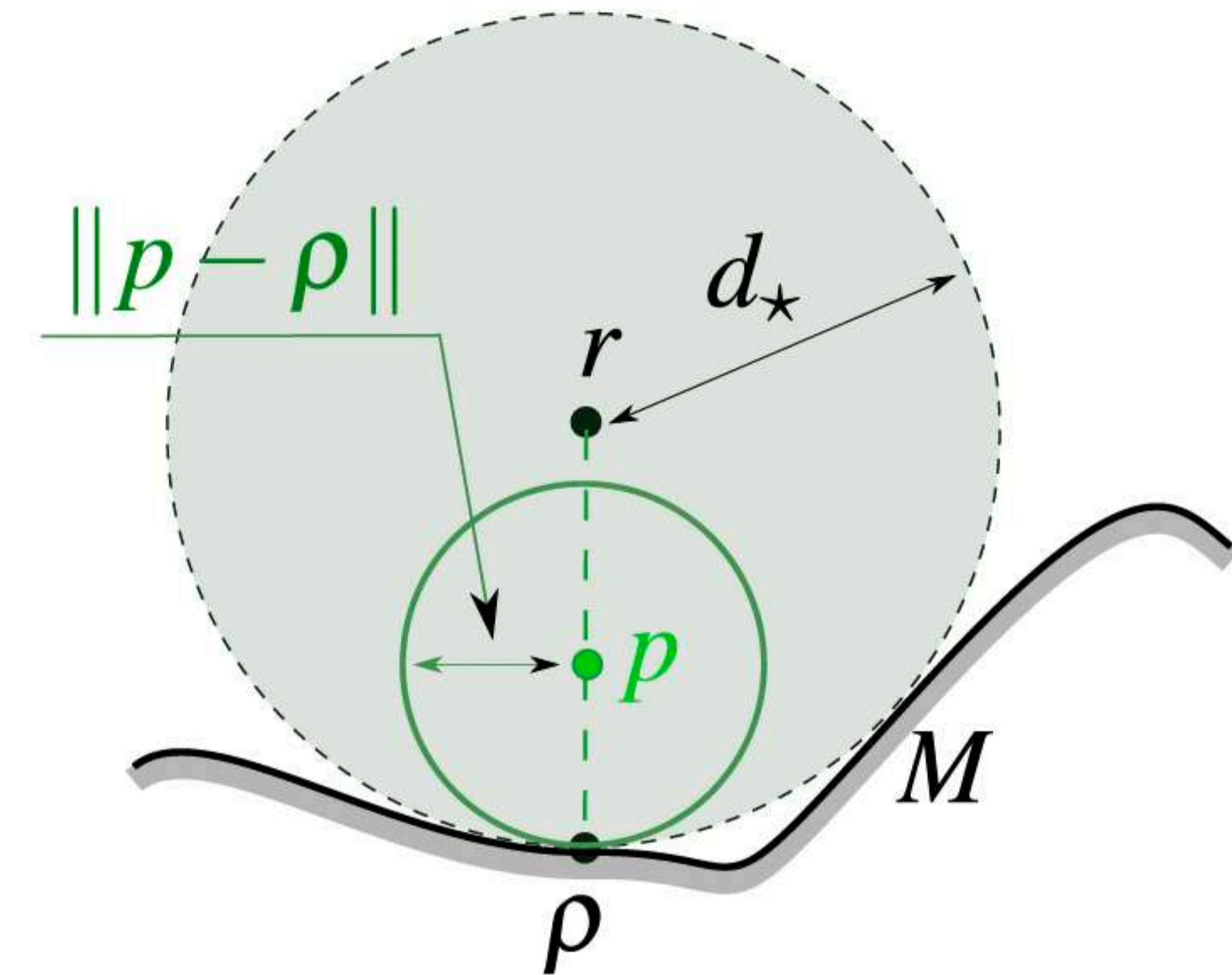
[Опр. 2.1]

Будем говорить что робот **сопровождает объект** если  $\text{dist} [\mathbf{r}(t), \mathcal{F}] \rightarrow 0$  при  $t \rightarrow \infty$  для некоторого  $R_{op}$ -fence  $\mathcal{F}$ .

# Основная идея

Переформулируем задачу следующим образом

Пусть вместе с центром робота движется диск радиусом, равному желаемому расстоянию до объектов. Тогда задача может быть сформулирована как отслеживание точки, связанной с центром диска.



Сопровождающий диск в точке возможного мгновенного центра поворота



# Допущения и соображения

## [Допущение 2.1]

Граница  $B := \partial N^{d_0}(D)$  содержит жорданову кривую  $\mathcal{P}$ , такую, что все объекты лежат в одном полупространстве

## [Допущение 3.1]

Всякое  $D_i$  либо синглтон, либо ограничено кусочно аналитической жордановой кривой. Также  $D_i \cap D_j = \emptyset \forall i \neq j$ .

## [Опр. 3.1]

Будем называть расстояние  $d_\star > 0$  **странным** если выполнено любое из условий:

1. Существует объект  $D_i$  чья граница содержит бесконечно много точек в которых  $\chi_i(s_i^\pm) = -d_\star^{-1}$ ;
2. Существует бесконечно много положений робота  $r \notin D$  в которых  $\mathbf{dist}[r, D] = d_\star$  и  $\mathbf{Pr}_D[r]$  — синглтон
3. Существует “мост” радиусом  $2d_\star$ .

## [Лемма 3.1]

Множество странных расстояний счётно или пусто.

## [Допущение 3.2]

$d_0$  и  $d_0 + R_{op}$  не являются странными

## [Лемма 3.2]

Если  $d_0 + R$  не является странным и  $R \geq R_{op}$  и **допущение 2.1, 3.1** верно, верно и следующее.

- i) Граница компонент связности  $Q_{CC}$  множества  $Q(R) := \{c \in R_{ext} : \mathbf{dist}[c, \mathcal{P}] > R\}$  представима в виде кусочно аналитической функции
- ii) объединение  $R$ -дисков с центром в  $Q_{CC}$  связно и ограничено  $(R, R_{op})$ -кривой, называемой **основной**
- iii) Всякая безопасная траектория лежит в заплатке к основной  $(R, R_{op})$ -кривой
- iv) При  $R \geq R_* \geq R_{op}$ , заплатка основной  $(R, R_{op})$ -кривой лежит в единственной  $(R_*, R_{op})$ -кривой

Последнее допущение означает, что вокруг начального местоположения робота достаточно свободного пространства. В частности, обозначим  $C_\pm^{in}$  окружность, пройденную роботом от начального состояния при постоянном управлении  $\omega \equiv \pm \bar{\omega}$ .

## [Допущение 3.3]

Множество  $Q(R) := \{c \in R_{ext} : \mathbf{dist}[c, \mathcal{P}] > R\}$  не пусто  
Так же начальное положение робота находится внутри  $Q(R)$ .

# Закон управления

Логическое переключение

[Опр. 4.1]

$$\mathfrak{P} \xrightarrow{d_R = \text{dist}[r, D] - d_0 \leq \eta(R_{op} - R_{min}) \quad t \geq 2\pi/\bar{\omega}} \mathfrak{B}$$

где  $\eta \in (0, 1)$  некоторый параметр регулятора

[Опр. 4.2]

$$\chi'(z) > 0 \quad \forall z, \quad \chi(0) = 0,$$

$$\bar{\chi} := \sup_{z \in \mathbb{R}} |\chi(z)| < \infty, \quad \bar{\chi}' := \sup_{z \in \mathbb{R}} |\chi'(z \pm)| < \infty$$

Примерами функции являются  $\chi(z) = a\chi_*(z/b)$ , где  $a, b > 0$

$$\text{и } \chi_*(z) = \arctan(z), \tanh(z), \frac{z}{\sqrt{1+z^2}}$$

[Опр. 4.3]

$$\omega(t) = \bar{\omega} \times \begin{cases} \text{sgn}\{\dot{d}_R(t) + \mu\chi[d_R]\} & \text{в режиме } \mathfrak{P}, \\ \text{sgn}\{\dot{d}_{R_{op}}(t) + \mu\chi[d_{R_{op}}]\} & \text{в режиме } \mathfrak{B} \end{cases}$$

Производная по времени  $\dot{d}_R(t)$  может быть численно вычислена для известной последовательности  $d_R(t)$ .

[Теорема 4.1]

Пусть допущения 2.1, 3.1, 3.3 выполнены, а параметры  $R_{op}, R, \mu, \eta$  регулятора удовлетворяют допущению 3.2 и

$$\mu\bar{\chi} < v, \quad \eta \in (0, 1),$$

$$\frac{\mu^2 \bar{\chi} \bar{\chi}'}{v\sqrt{v^2 - \mu^2 \bar{\chi}^2}} < \frac{\Delta}{R_{min}(R_{min} + \Delta)}, \text{ где}$$

$$\Delta := \min\{R - R_+, (1 - \eta)(R_{op} - R_{min})\}$$

Тогда робот под действием данного закона управления сопровождает объекты в смысле Опр. 2.2, не покидает  $\mathcal{R}_{ext}$  и движется безопасно.

Расстояние  $d_R$  вычисляется как

$$d_R(t) := \begin{cases} d(t) - d_0 & \text{в режиме } \mathfrak{C} \\ R - \|r(t) - v[A(t)]\| & \text{в режиме } \mathfrak{B} \end{cases}$$

Закон переключения логических мод таков

$$\mathfrak{C} \mapsto \mathfrak{B} \quad \text{если } Pr_D[\mathcal{C}(t)] \text{ не синглтон,}$$

$$\mathfrak{B} \mapsto \mathfrak{C} \quad \text{в момент покидания } A(t)$$

Где  $A(t)$  — область между двумя лучами, запоминаемая роботом в его СК; постоянное обновление внутренних переменных должно осуществляться, например, с помощью одометрии.

# Экспериментальные результаты

## Движение через область с сингулярностью

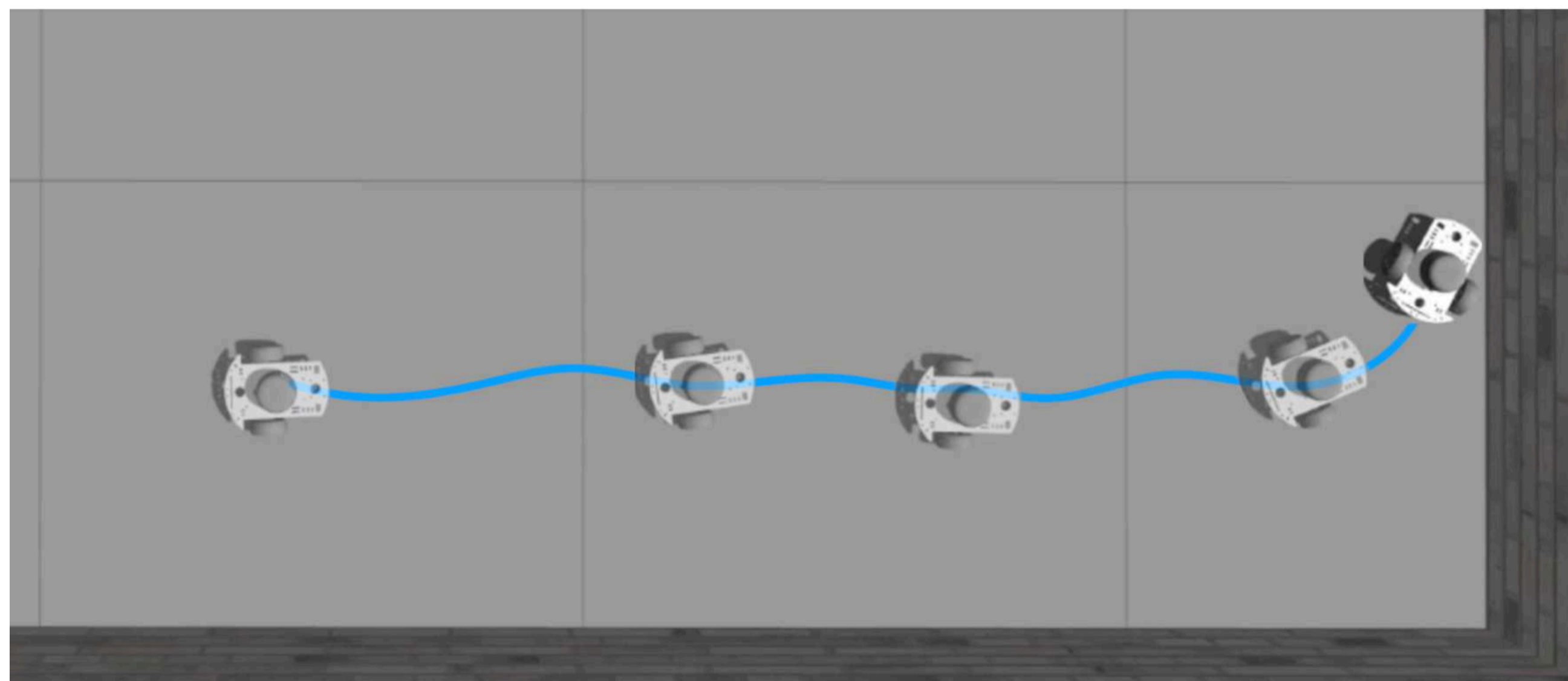
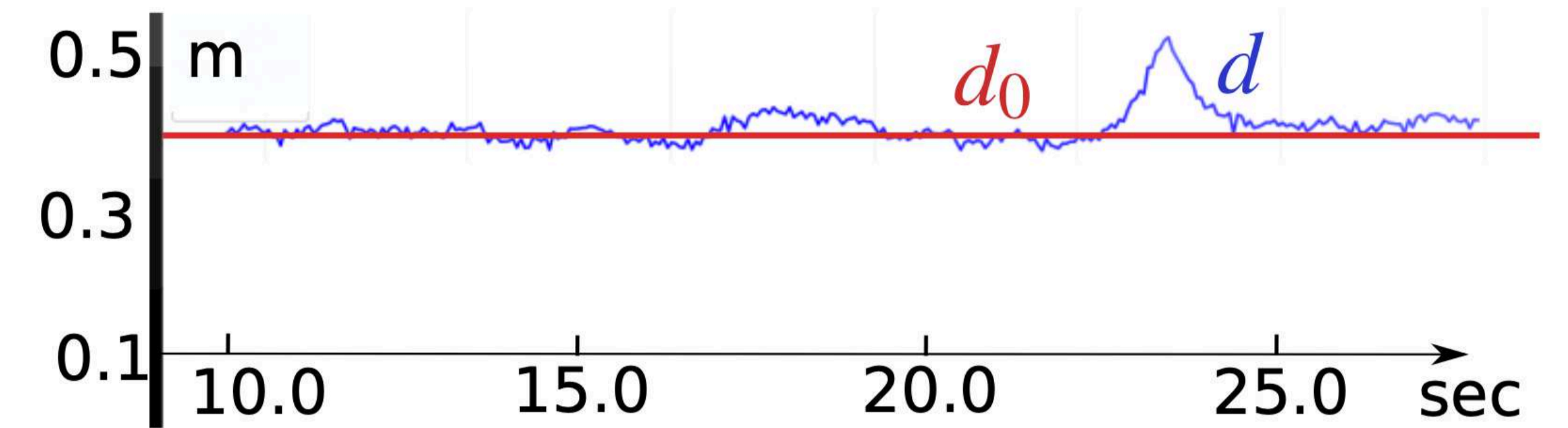
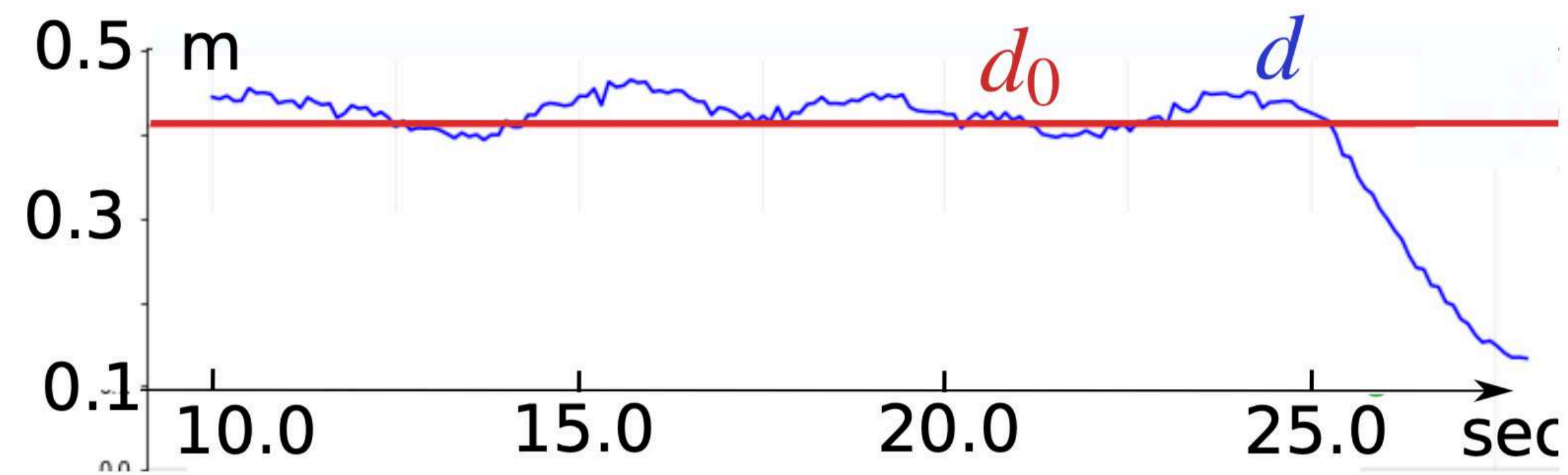


Рис.3. Мобильный робот следует по эквидистантной кривой без переключения режимов управления

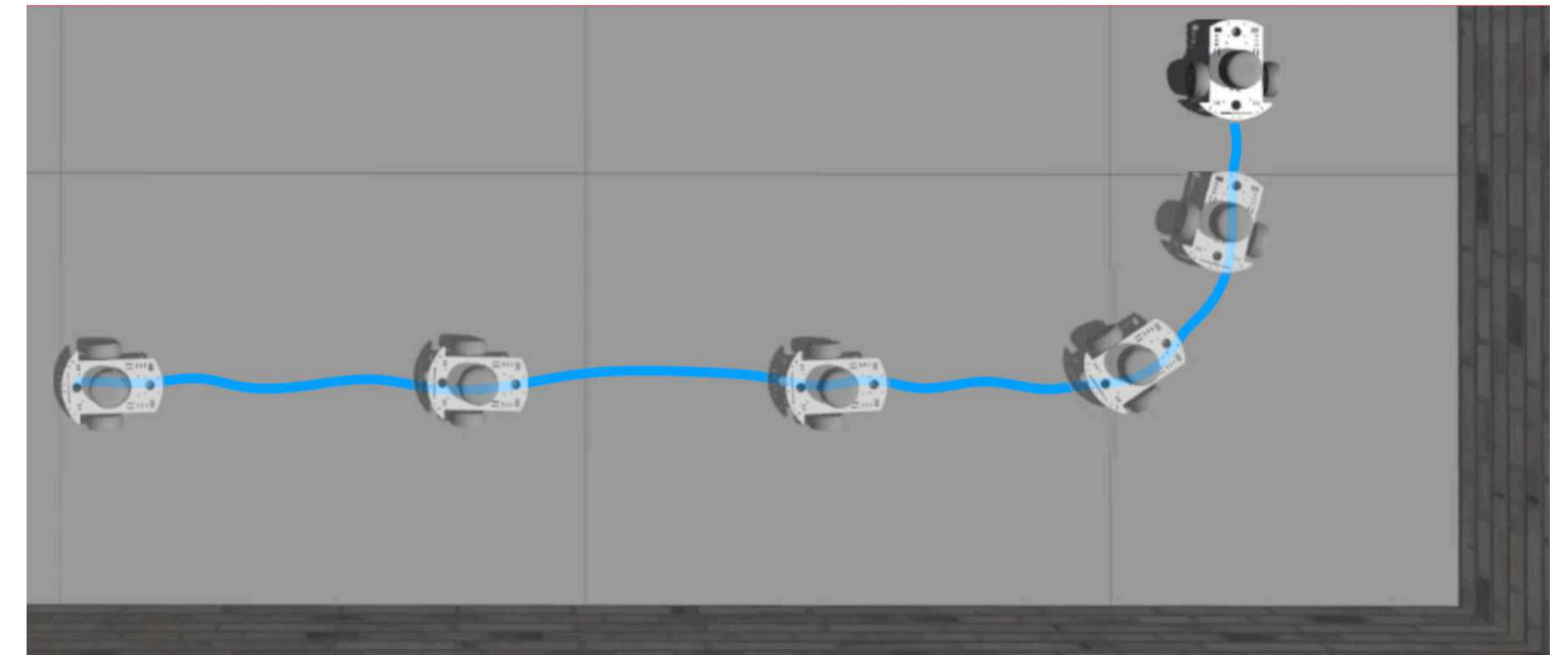


Рис.4. Мобильный робот следует по эквидистантной кривой с переключением режимов управления

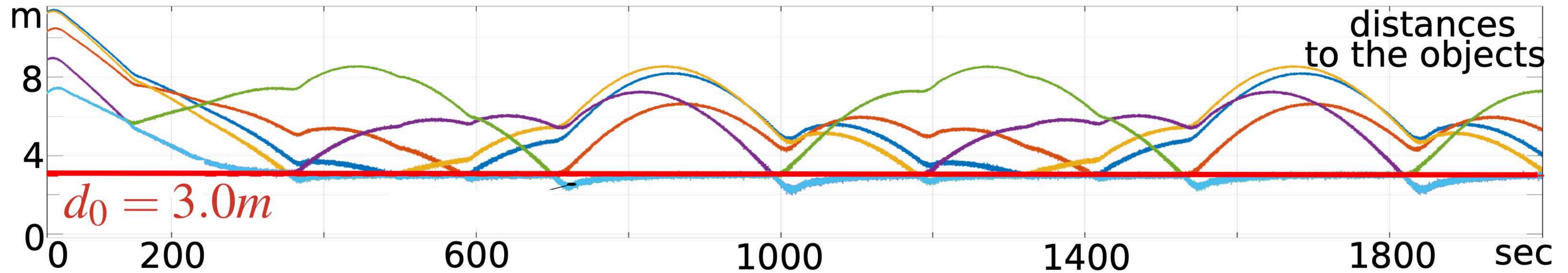
Первый тест был проведен с помощью симулятора робототехники Gazebo 11 с пакетом ROS DiffBot. Цель - сравнить эффективность двух законов навигации при движении вдоль стены.

Закон без переключателей приводит робота к столкновению со стеной, поскольку он начинает учитывать свою «переднюю часть» только тогда, когда ближайшая точка стены перемещается в эту часть. Оказывается, что это слишком поздно для предотвращения столкновения.

Our simulation tests were carried out with  $\bar{v} = 0.4 \text{ m/s}$ ,  $\bar{\omega} = 0.6 \text{ 1/s}$ ,  $d_0 = 0.41 \text{ m}$ ,  $\mu = 1$ . The control update period was  $0.1 \text{ s}$  and  $\chi(\cdot)$  was the linear function  $\chi(z) = 0.025z \text{ 1/s}$  saturated at the threshold  $0.1 \text{ m}$ .

# Экспериментальные результаты

## Круговое движение вокруг точечных объектов



Расстояние между парами объектов не превосходит  $2d_0 = 6.0\text{ m}$ , так что целевая эквидистанта связна.

Поскольку эквидистантная кривая не является гладкой, ожидаемо, что в моменты, когда ближайший объект изменяется из-за движения робота, наблюдаются выбросы ошибки расстояния  $d(t) - d_0$ .

Тем не менее, эти выбросы малы по времени и величине, и робот хорошо справляется с задачей.

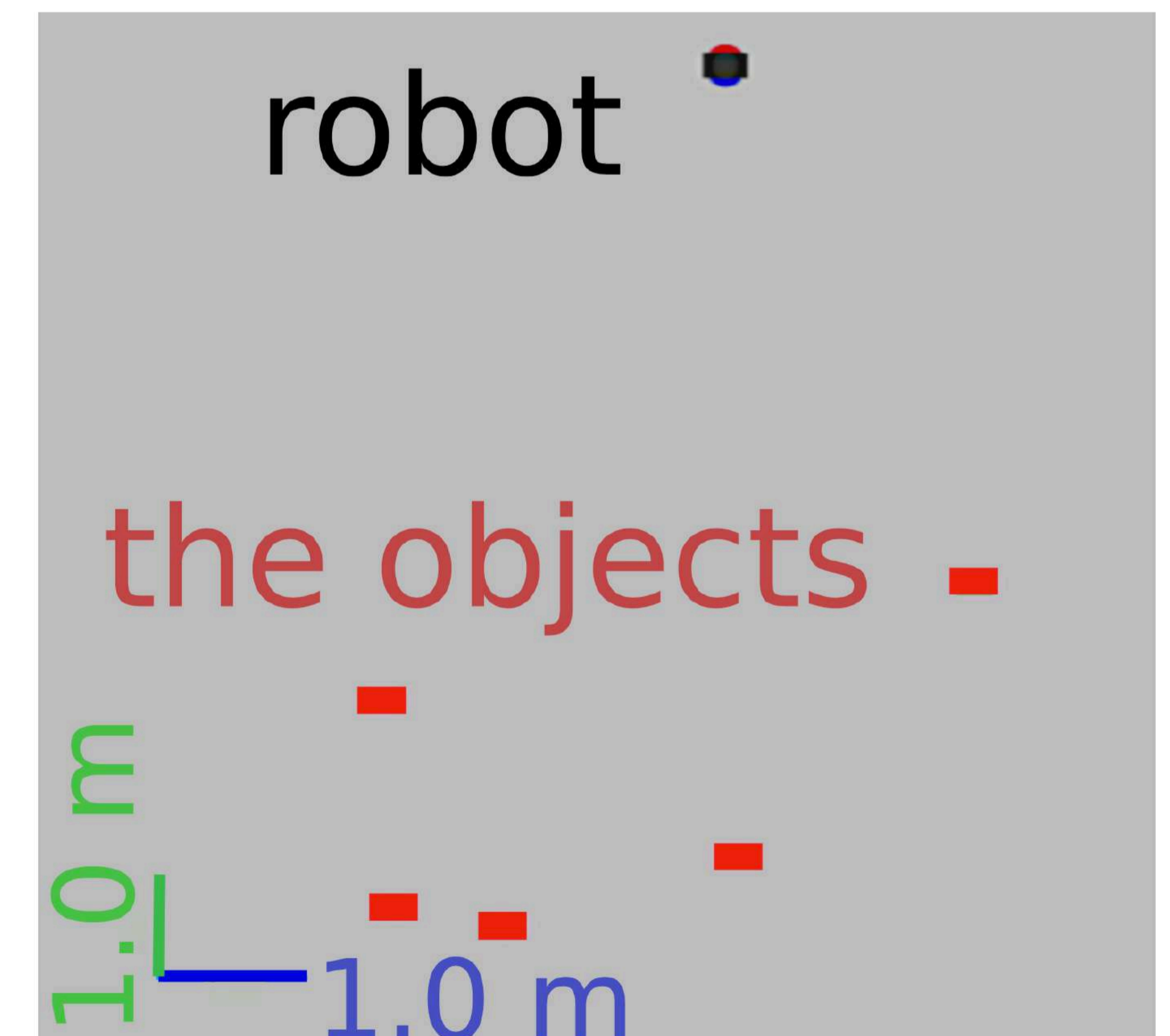
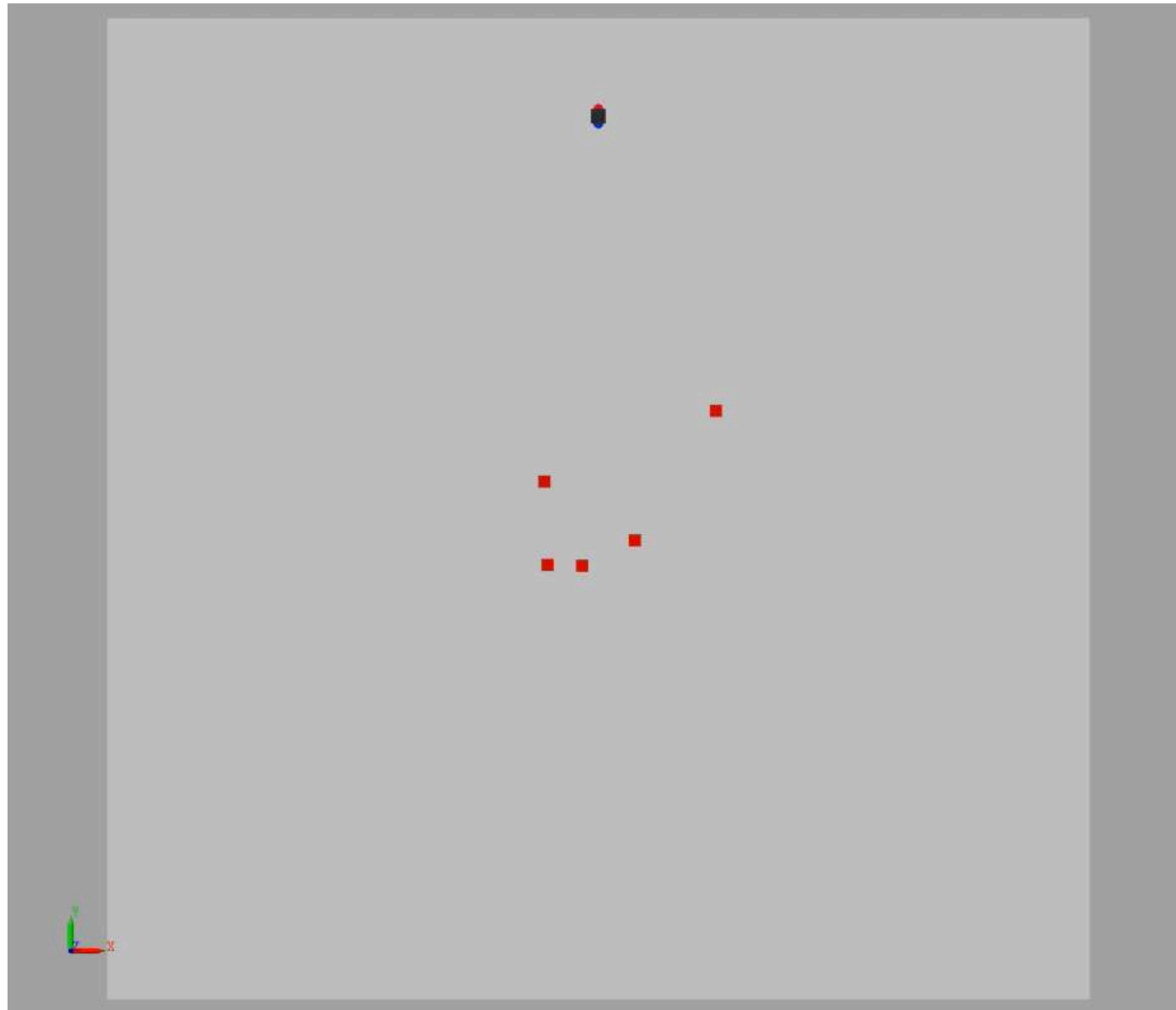


Рис.5. сцена симуляции в Matlab

# Экспериментальные результаты

## Круговое движение вокруг точечных объектов



# Экспериментальные результаты

## Круговое движение вокруг протяжённого объекта



Рис.7. Дифбот

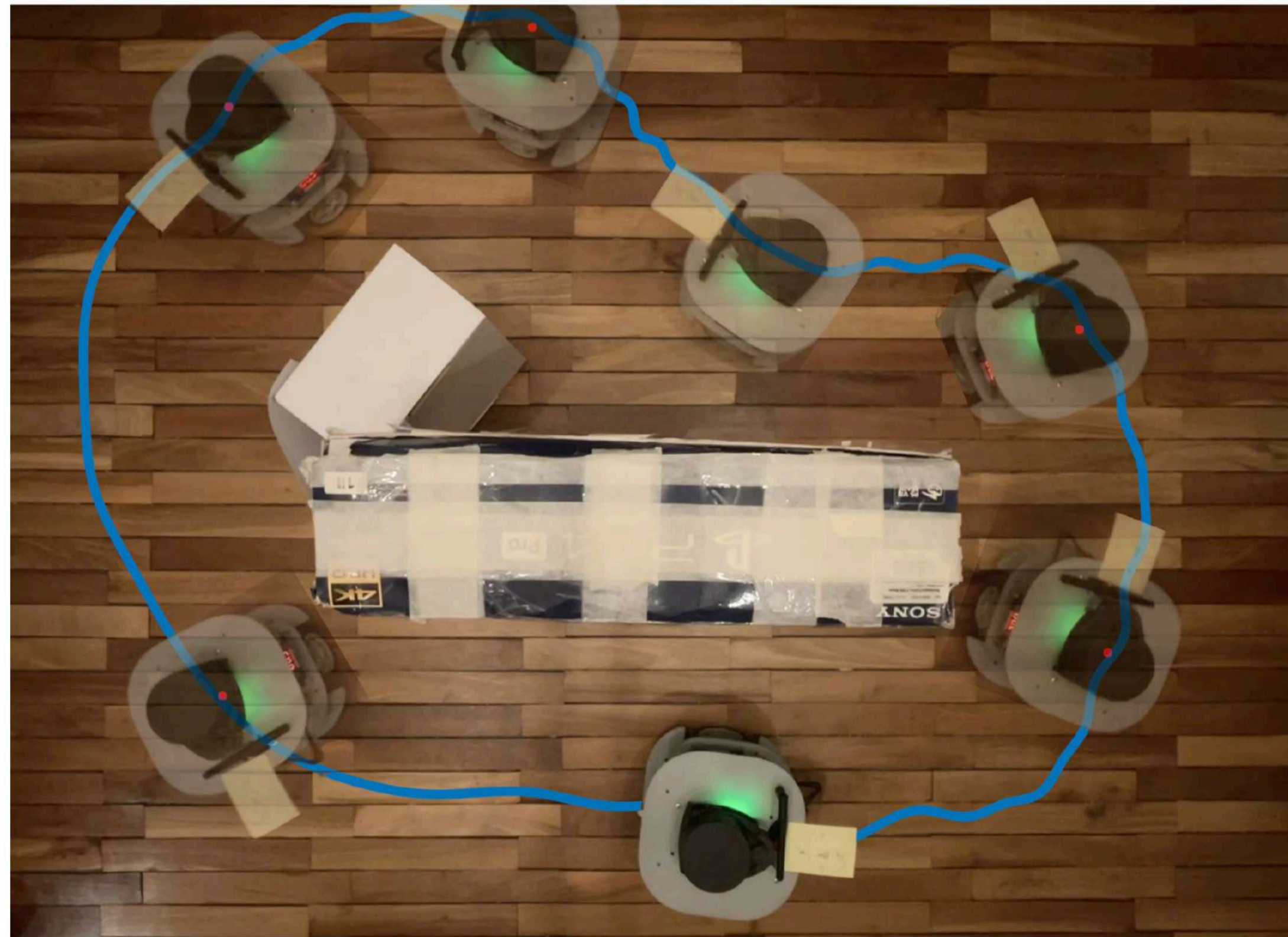


Рис.8. Путь робота

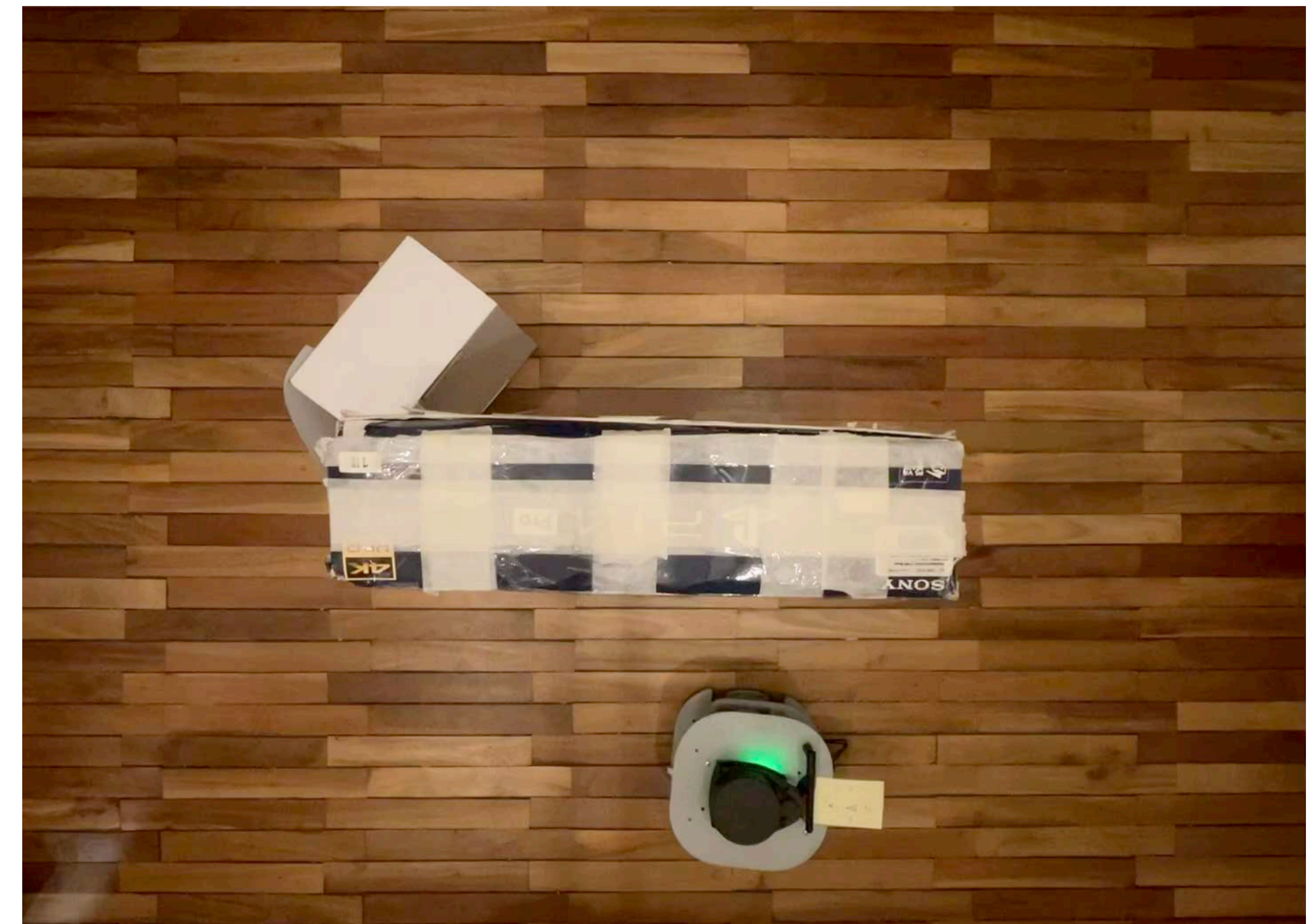


Рис.9. Движение робота по эквидистанте к протяжённому объекту с сингулярностью

Мобильный робот Дифбот (рис. 7) с дифференциальным приводом, оснащенный датчиком LIDAR RPLidar A1 начального уровня. Данные с датчиков обрабатываются и преобразуются в управляющее воздействие на двигатели в соответствии с описанной стратегией управления на одноплатном компьютере Raspberry Pi 3B+, работающем под управлением Ubuntu. Параметры контроллера, используемые в экспериментах, такие же, как и в описанных ранее симуляциях.

Единственное различие в алгоритмах заключается в методах регистрации и обработки данных датчиков: если в наших симуляционных тестах все облако точек LIDAR обновляется полностью в каждый момент выборки датчиков, то в лабораторном роботе точки обновляются по одной. LIDAR установлен на верхней части робота, чтобы избежать слепых зон. Ни датчик ToF, ни карта сцены не использовались.

# DDP

## Иллюстрация MPC

Вообще говоря, последовательные подходы сначала вычисляют полную траекторию управления  $u$  на каждой итерации, а затем моделируют нелинейную систему в разомкнутом контуре с этой траекторией  $u$ , чтобы получить состояния  $x$  для следующей точки линеаризации.

**Differential Dynamic Programming** — разновидность метода MPC (фактически — задачи оптимального управления дискретной системы на конечном временном интервале) В отличие от этого, дифференциальное динамическое программирование (DDP) использует изменяющийся во времени закон аффинной обратной связи  $u_i^0(x_i) = k_i + K_i x_i$  из рекурсии Риккати для моделирования нелинейной системы вперед во времени

# DDP — Постановка задачи

Планирование траектории на примере шагающих роботов

$$\min_{x(\cdot), u(\cdot), \lambda(\cdot)} \text{Cost}(x(\cdot), u(\cdot), \lambda(\cdot))$$

$$\text{subject to } M(q)\dot{v} + C(q, v)v + \tau_g(q) = S^T \tau + J^T(q)\lambda \quad \forall t \in [t_0, t_1]$$

$$\text{ContactConstraints}(x(t), \lambda(t), u(t), Env)$$

$$\text{KinematicsConstraints}(x(t))$$

$$\text{InputConstraints}(u(t), x(t))$$

$$\text{TaskConstraints}(x(t), u(t), \lambda(t))$$

$$\text{TaskConstraints}(x(t), u(t), \lambda(t))$$



# DDP

## Планирование траектории на примере шагающих роботов

Функция Гамильтона:  $H(x, \lambda, t) = I(x, u) + \lambda^T f(x, u)$

Функция Гамильтона:  $u(x, \lambda) = \min \quad s.t. \quad g(x, u) \leq 0$

состояние системы  $\dot{x} = f(x, u), x(t_0) = x_0$

сопряжённое состояние  $\dot{\lambda} = -\nabla_x H(x, \lambda, u), \lambda(t_N) = I_N(x_N)$

# DDP

## Планирование траектории на примере шагающих роботов

Воспользуемся линейно-квадратичной аппроксимацией гамильтониана

$$H_k(\cdot) = \frac{1}{2} \begin{bmatrix} 1 \\ \delta x_{k+1} \end{bmatrix}^T \begin{bmatrix} 0 & V_{x_{k+1}}^T \\ V_{x_{k+1}} & V_{xx_{k+1}} \end{bmatrix} \begin{bmatrix} 1 \\ \delta x_{k+1} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 1 \\ \delta x_k \\ \delta u_k \end{bmatrix}^T \begin{bmatrix} 0 & I_{x_k}^T & I_{u_k}^T \\ I_{x_k} & I_{xx_k} & I_{xu_k} \\ I_{u_k} & I_{xu_k} & I_{uu_k} \end{bmatrix} \begin{bmatrix} 1 \\ \delta x_k \\ \delta u_k \end{bmatrix}$$

Оптимальное управление на данном шаге удовлетворяет соотношению

$$\delta u_k^*(\delta x_k) = H(\delta x_k, v_k, v_{xx_k}, \delta u_k, k)$$

$$\min_{\delta u_k} \frac{1}{2} \begin{bmatrix} 1 \\ \delta x_k \\ \delta u_k \end{bmatrix}^T \begin{bmatrix} 0 & Q_{x_k}^T & Q_{u_k}^T \\ Q_{x_k} & Q_{xx_k} & Q_{xu_k} \\ Q_{u_k} & Q_{xu_k} & Q_{uu_k} \end{bmatrix} \begin{bmatrix} 1 \\ \delta x_k \\ \delta u_k \end{bmatrix} \quad s.t. \quad g(x_k \oplus \delta x_k, u_k + \delta u_k) \leq 0$$

# DDP

## Планирование траектории на примере шагающих роботов

Прямая подстановка

оптимальное решение на шаге  $k$

эволюция системы

$$u_k^{i+1} = u_k^i + \delta u_k^i$$

$$x_{k+1}^{i+1} = f(x_k^i, u_k^i)$$

Обратная подстановка

якобиан сопряжённого состояния

гессиан сопряжённого состояния

изменение сопряжённого состояния

$$V_{x_i} = Q_{x_i} - Q_{x u_i} Q_{u u_i}^{-1} Q_{u_i}$$

$$V_{x x_i} = Q_{x x_i} - Q_{x u_i} Q_{u u_i}^{-1} Q_{u x_i}$$

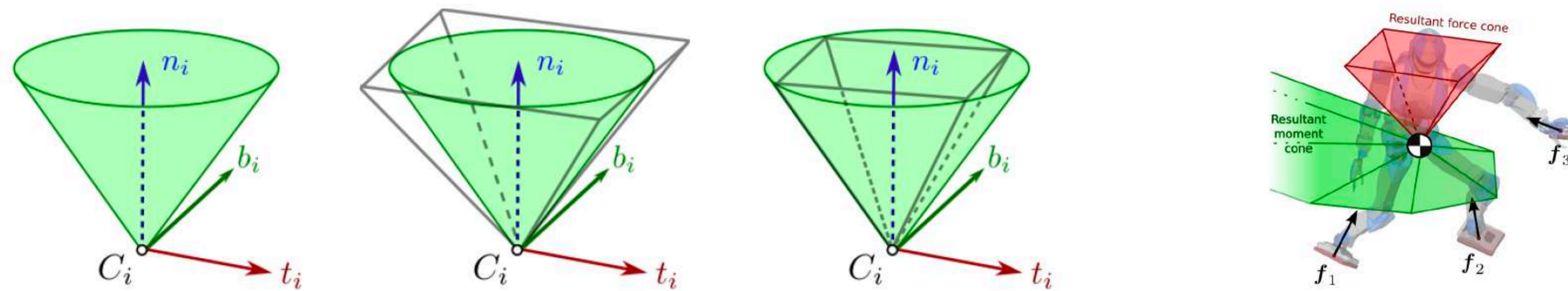
$$dV = -\frac{1}{2} Q_{u_i}^T Q_{u u_i}^{-1} Q_{u_i}$$

# DDP — Модели контакта и робота

## Планирование траектории на примере шагающих роботов

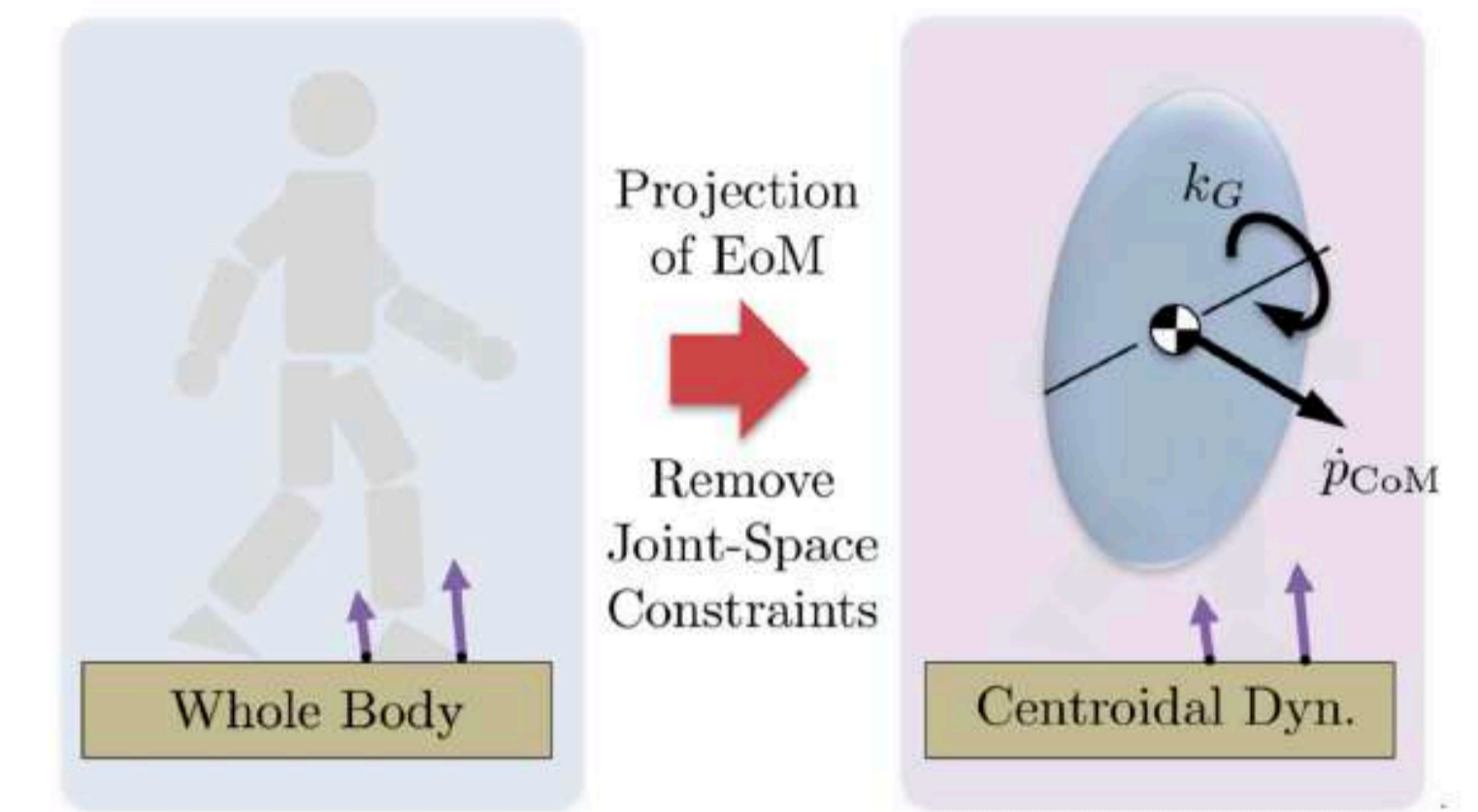
Вернёмся к шагающим робота

Представление конуса трения



Подход центроидной динамики предлагает рассмотрение упрощённой динамики шагающего робота как АТТ с соответствующим моментом инерции и положением центра масс

Кино-динамика рассматривает же динамику движения каждого из звеньев в кинематической цепи робота



**Centroidal**

$$\dot{h}_g = \begin{bmatrix} \sum_{k=1}^K f_k + mg \\ \sum_{k=1}^K (r_k^* - c) \times f_k + \tau_k \end{bmatrix}$$

$$x = [c \quad h_g]$$

$$u = \lambda$$

**Kino-dynamics**

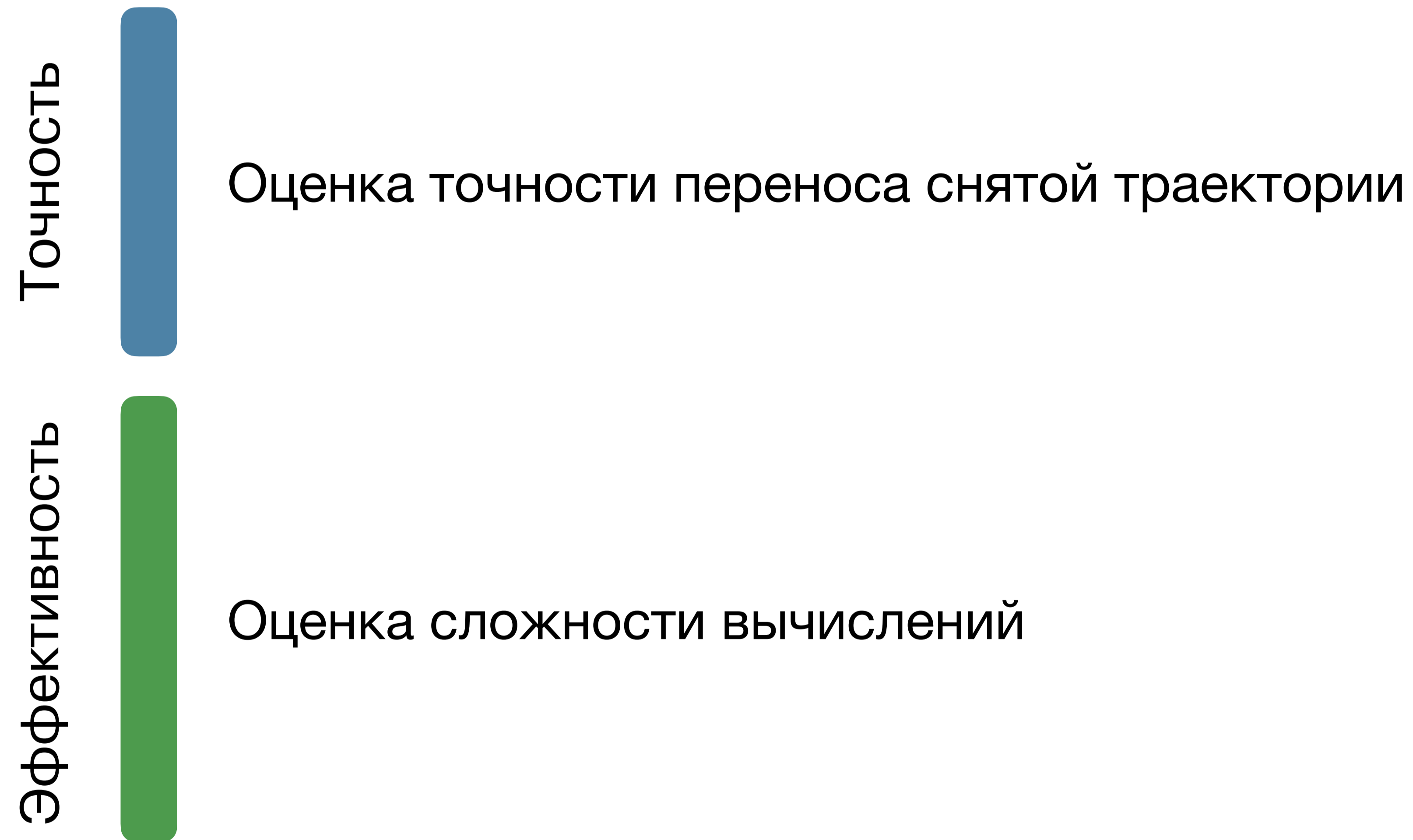
$$\dot{h}_g = \begin{bmatrix} \sum_{k=1}^K f_k + mg \\ \sum_{k=1}^K (r_k(q) - c(q)) \times f_k + \tau_k \end{bmatrix}$$

$$x = [q \quad \dot{q}]$$

$$u = [\ddot{q} \quad \lambda]$$

# DDP — Метрики

## Планирование траектории на примере шагающих роботов

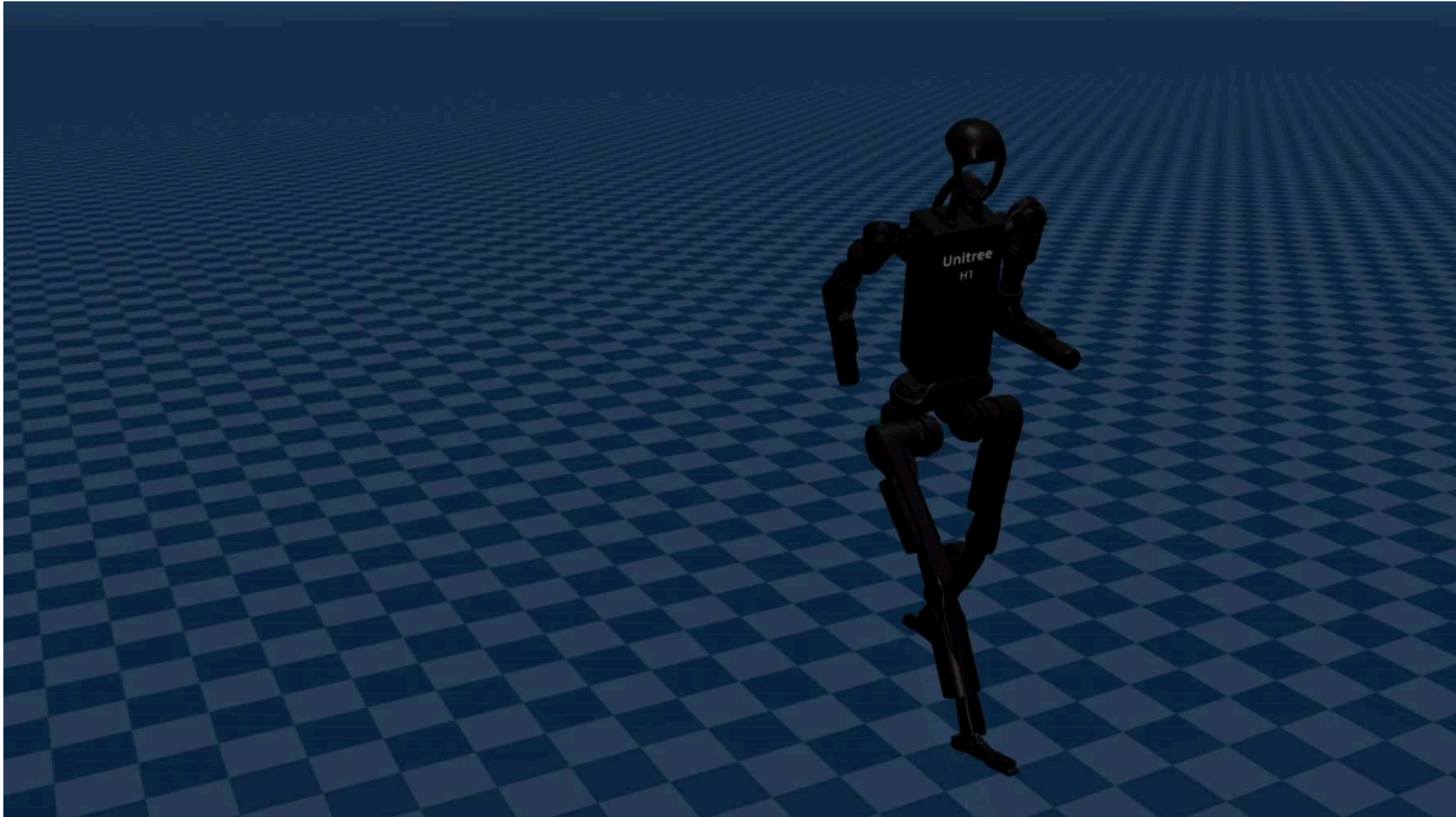


$$I_{ref}^x = ||x - x_{ref}||_{w_{ref}}$$

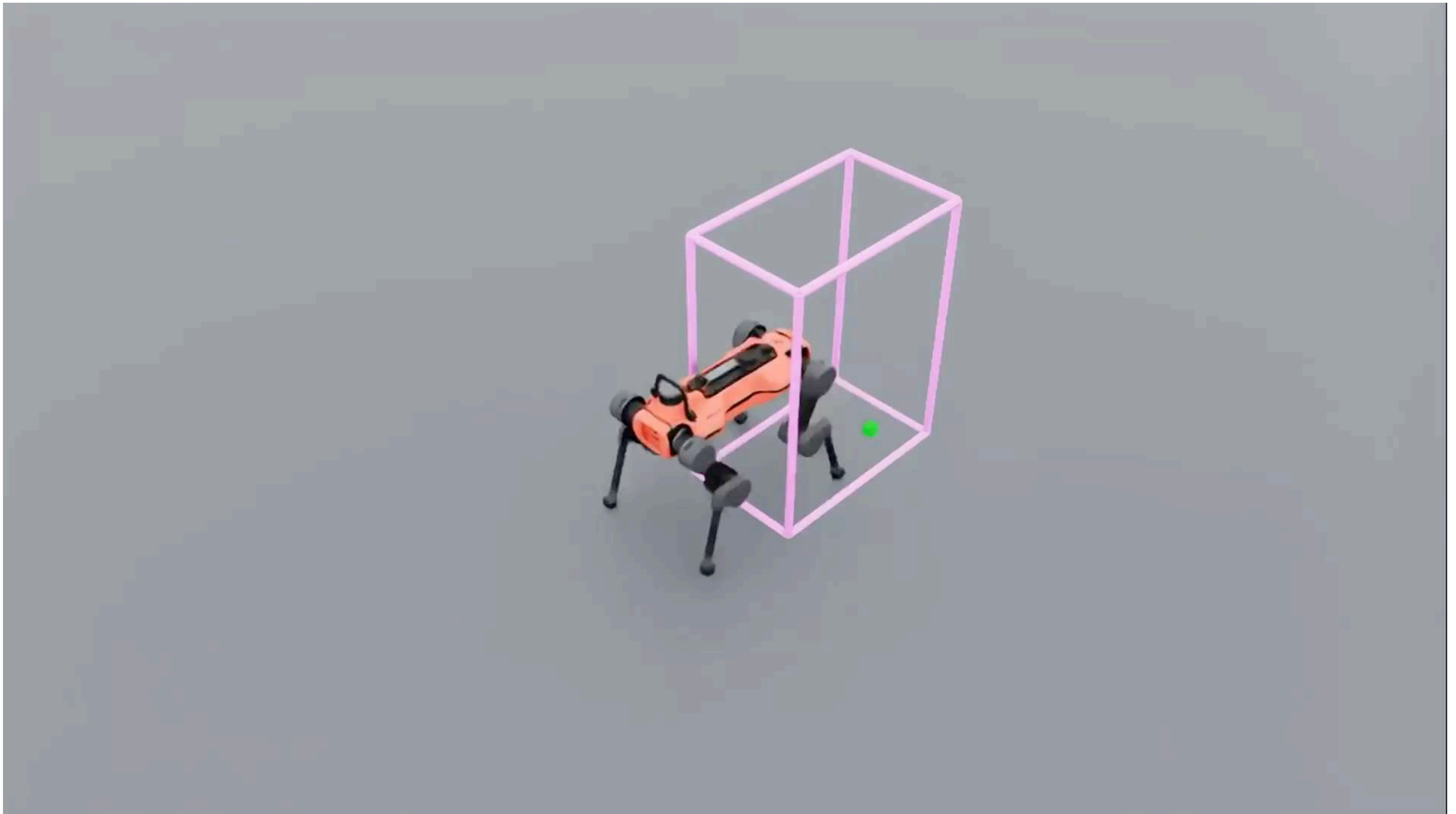
$$T_{sol}$$

# DDP

## Планирование траектории на примере шагающих роботов



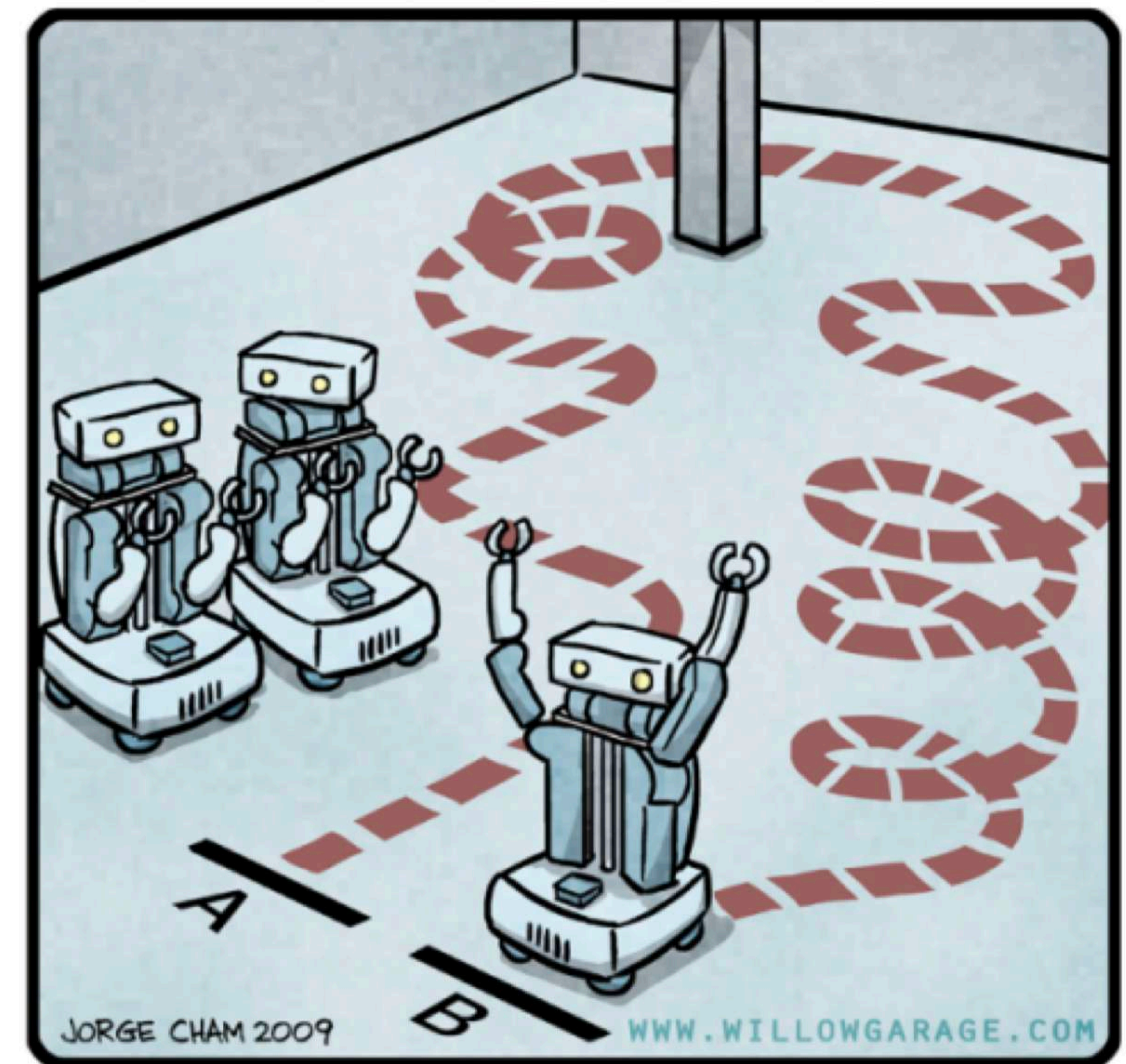
# Что день грядущий нам готовит?



Arm P. et al. Pedipulate: Enabling manipulation skills using a quadruped robot's leg //41st IEEE Conference on Robotics and Automation (ICRA 2024). – 2024.

# Спасибо за внимание

R.O.B.O.T. Comics



"HIS PATH-PLANNING MAY BE  
SUB-OPTIMAL, BUT IT'S GOT FLAIR."