МИНОБРНАУКИ РОССИИ
Федеральное государственное образовательное учреждение высшего образования
«**МИРЭА – Российский технологический университет**»
РТУ МИРЭА
Институт кибернетики
Кафедра автоматических систем

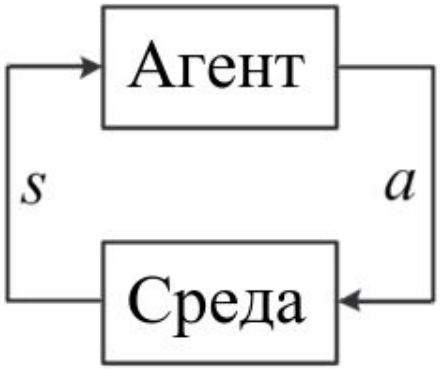
ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ

Докладчик: **Барашков Алексей Андреевич**,
программист, ООО Викрон,
выпускник кафедры Автоматических систем.

Научный руководитель: Филимонов Александр Борисович,
профессор кафедры Автоматических систем,
д.т.н., с.н.с.

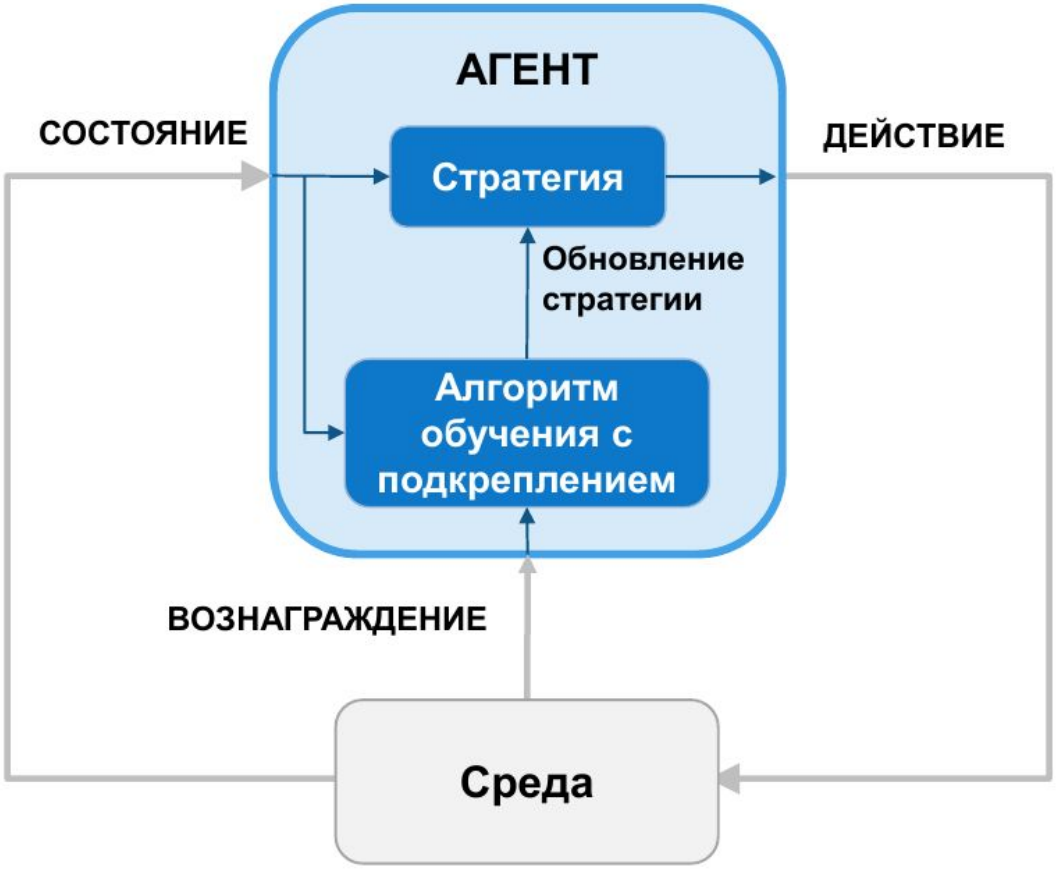
МОСКВА 2019

Машинное обучение с подкреплением

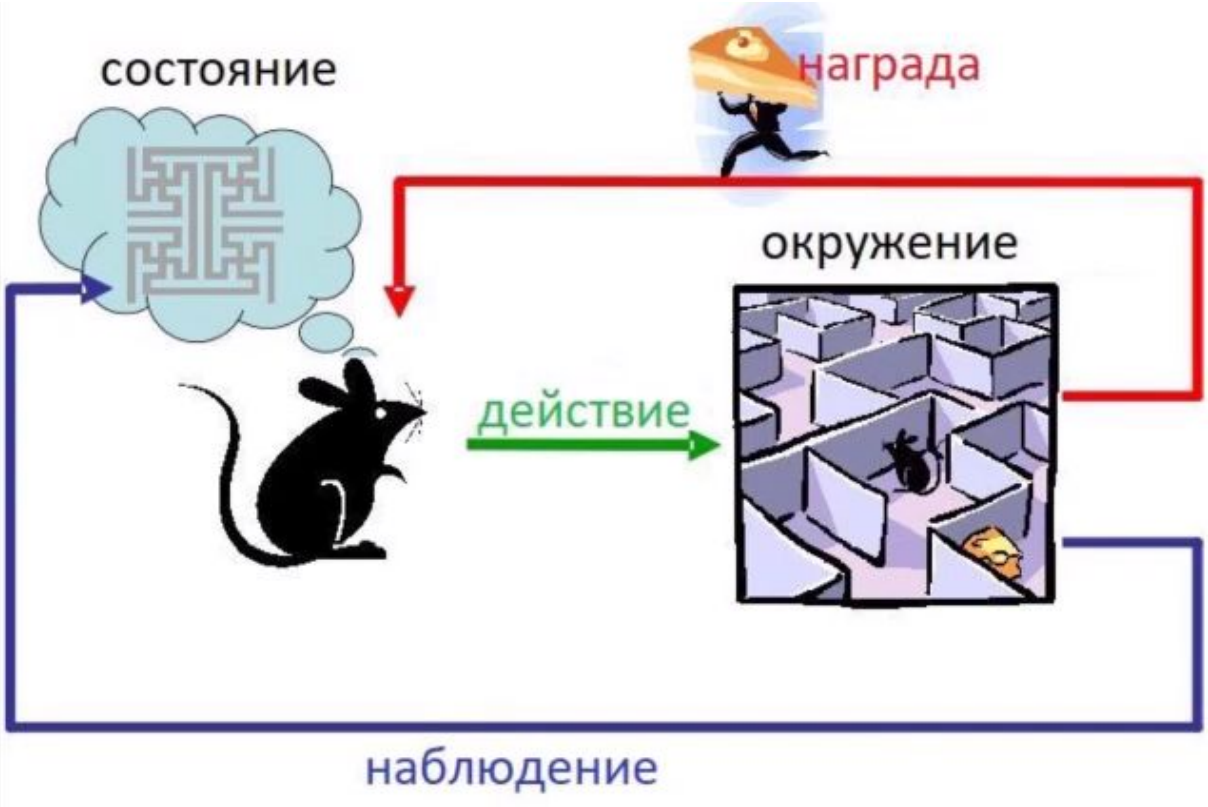


- Агент действует в некой среде.
- Агент с помощью датчиков определяет состояние *s*, в котором находится
- Агент совершает действие *a*.
- Агент переходит в новое состояние *s'*.
- Агент оценивает, на сколько данное действие было полезным при помощи награды *r*.

Развёрнутая схема обучения с подкреплением



Наглядная схема



Опыт

- За счёт совершения различных действий в среде агент набирается опыта.
- **Опыт – в каком состоянии было совершено какое действие, какая награда была за это получена и в какое новое состояние в результате агент попал.**

$$\langle s, a, r, s' \rangle$$

- Опыт должен быть максимально разнообразным: желательно побывать в наибольшем числе состояний и попробовать в каждом из них как можно больше различных действий.

Награда

Агент оценивает ситуацию – пару «состояние-действие» при помощи скалярной награды (действительного числа).

$$r(s, a)$$

Награда показывает, насколько полезно было совершить определённое действие в данном состоянии

Задание инженером правильного метода формирования награды играет определяющую роль в успехе обучения

Стратегия

Агент руководствуется некоторой стратегией действий.

Стратегия определяет в каком состоянии будет совершено какое действие.

$$\pi : S \rightarrow A$$

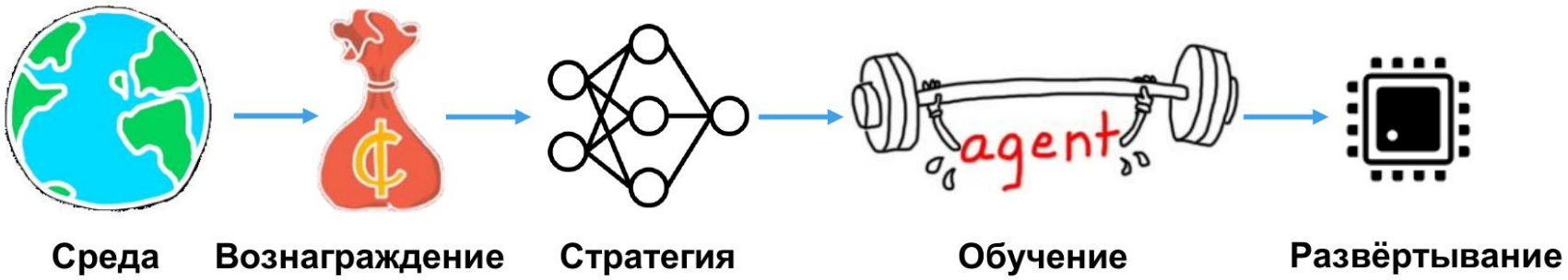
$$a = \pi(s)$$

Обучение

- За счёт использования полученного опыта **обновляется стратегия** поведения агента.

- После завершения обучения агент может действовать, используя полученную стратегию.

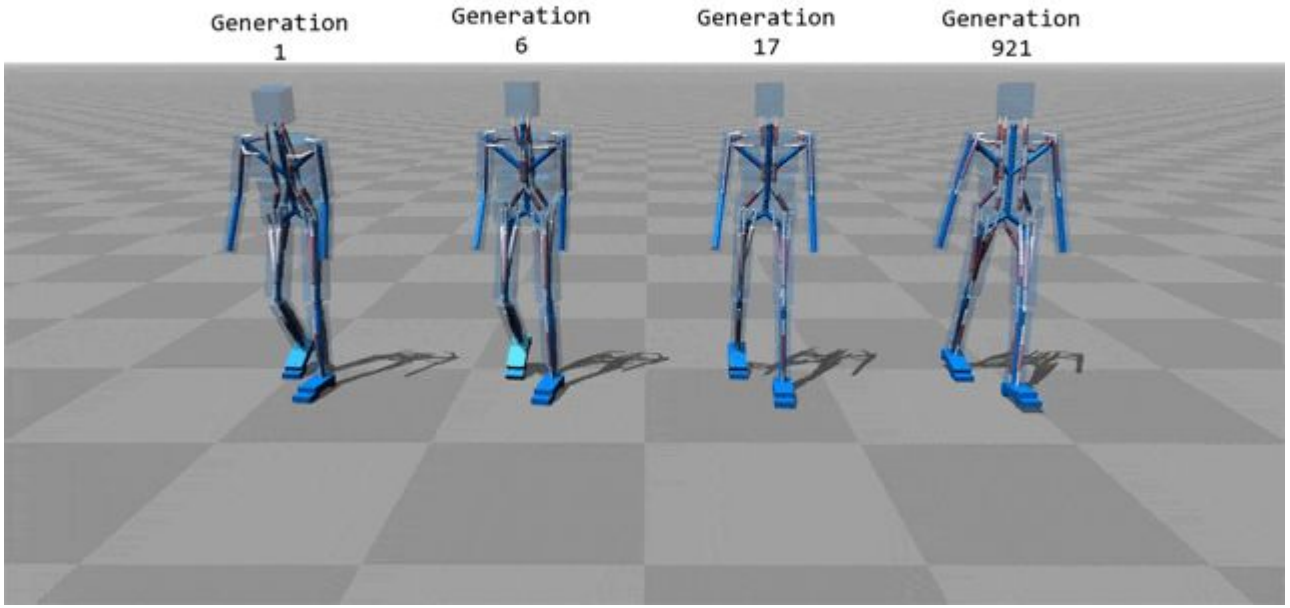
Этапы рабочего процесса при использовании обучения с подкреплением



Пример: Обучение беспилотного автомобиля

- Бортовой компьютер обучается вождению...
  (агент)
- с помощью данных с датчиков (камеры и LIDAR),...
  (состояние)
- которые отображают дорожные условия, положение автомобиля,...
  (среда)
- генерирует команды рулевого управления, торможения и газа, ...
  (действие)
- и, согласно соответствию «состояние-действие», ...
  (стратегия)
- пытается оптимизировать комфорт водителя и эффективность расхода топлива...
  (вознаграждение)
- Алгоритм действия обновляется методом проб и ошибок с помощью алгоритма **обучения с подкреплением**

Популярный пример: обучение ходьбе роботов



Q-обучение

- Самый простой популярный алгоритм обучения с подкреплением.
- В основе лежит определение оценки функции полезности (*Q*-функции) для конечного числа действий.

Функция полезности действия

- Каждое действие в каждом состоянии можно оценить при помощи **функцией полезности** $Q^{\pi}(s, a)$ – ожидаемой суммой наград при совершении агентом действия a в состоянии s и совершении последующих действий в соответствии со стратегией π.
- **Процесс обучения** – определение функции полезности в процессе функционирования агента.

Функция полезности действия

- Функция полезности показывает, насколько большую награду можно получить за определённое действие, **а также насколько данное действие является перспективным.**
- Т.е. **сколько ещё наград можно будет собрать в будущем**, если при движении из нового состояния, используя текущую стратегию.
- На сколько сильно будет учитываться перспектива получения наград в будущем, инженер задаёт с помощью коэффициента дисконтирования γ :

$$0 < \gamma < 1$$

Стратегия действий агента при Q-обучении

- **Стратегия действий** – выбор действия с максимальной текущей оценкой полезности.

Хранение оценок полезности действий в таблице

	a_1	a_2	…	a_n
s_1	$Q(s_1, a_1)$	$Q(s_1, a_2)$	…	$Q(s_1, a_n)$
s_2	$Q(s_2, a_1)$	$Q(s_2, a_2)$	…	$Q(s_2, a_n)$
…	…	…	…	…
s_m	$Q(s_m, a_1)$	$Q(s_m, a_2)$	…	$Q(s_m, a_n)$

Глубокое Q-обучение

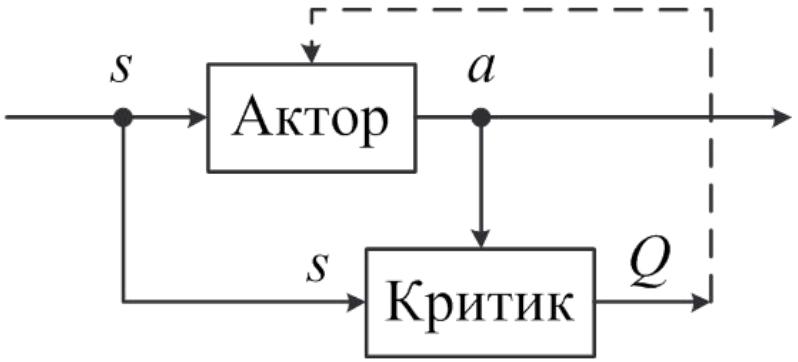
- Для аппроксимации функции полезности в непрерывном пространстве состояний используется нейронная сеть.
- Т.е. если состояний бесконечно много, нейронная сеть позволяет правильно определить полезность состояний, находящихся близко к уже исследованным.
- Глубокая нейронная сеть позволяет не производить предварительную обработку информации о состоянии. Например, на вход нейронной сети может подаваться изображение с камеры.

Системы адаптивной критики

- Более сложный алгоритм, чем *Q*-обучение.
- Нет ограничений на количество действий (например, действие - угол поворота руля на любой угол от -90° до +90°).
- Используется два блока: актор и критик.
- Позволяет настраивать управляющее устройство (актор) таким образом, чтобы предлагаемое им действие в каждом состоянии имело максимальную полезность.
- Актор может иметь различную структуру.
- Критик, как правило, реализуется с помощью нейронной сети.

Системы адаптивной критики

- **Критик** – блок системы управления, который оценивает качество её работы.
- Задачей **критика является аппроксимация функции полезности действий *Q***.
- **Актор** – блок системы управления, задающий действия этой системы.
- **Задача актора – выбор наилучших с точки зрения критика действий.**
- Актор и критик можно реализовать при помощи **нейронных сетей**.



Авторы - Данил Валентинович Прохоров, Дональд С Вунш II, Миссурийский университет науки и технологий, 1997.

*В IT-сообществе широко известна небольшая модификация метода под названием **DDPG**, 2015.*

Формулы

Определение функции полезности:

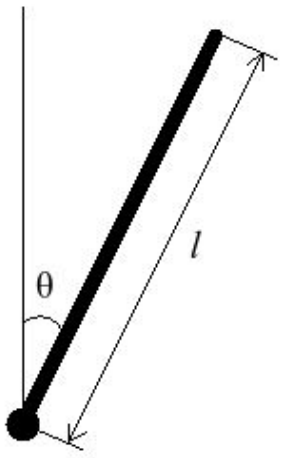
$$Q^{\pi}(s,a) = M\left[\sum_{t=0}^{t_F} \gamma^t r\big(s(t),a(t)\big)\,\middle|\, s(0)=s,\ a(0)=a\right].$$

Формула вычисления целевых значений для обучения критика:

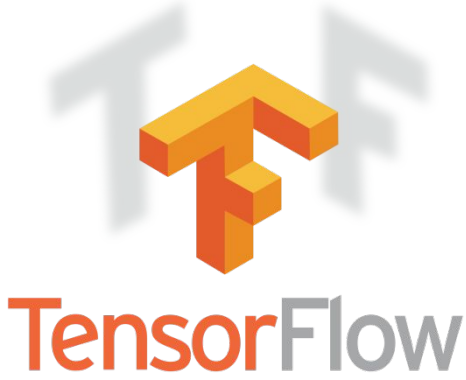
$$\overline{q}_{s,a} = Q_t(s,\ a) + \alpha\big(r(s,\ a) + \gamma Q_t(s',\ a') - Q_t(s,\ a)\big)$$

Задача о перевёрнутом маятнике

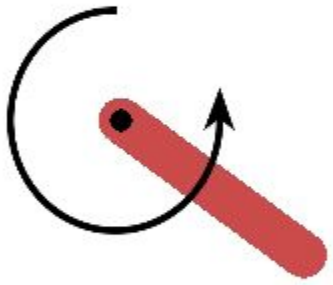
- Простая задача для апробации методов обучения с подкреплением.
- Целевое состояние маятника: стабилизация в вертикальном положении (нулевой угол отклонения от вертикальной оси, нулевая угловая скорость).
- Чем ближе положение маятника к вертикальному, больше награда.
- В точке подвеса – мотор. Действие - управляющий момент, создаваемый мотором.



Используемый инструментарий



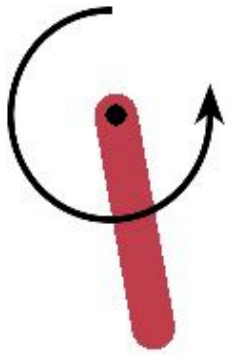
До обучения



Результаты обучения маятника

Время обучения – порядка 5 – 10 минут

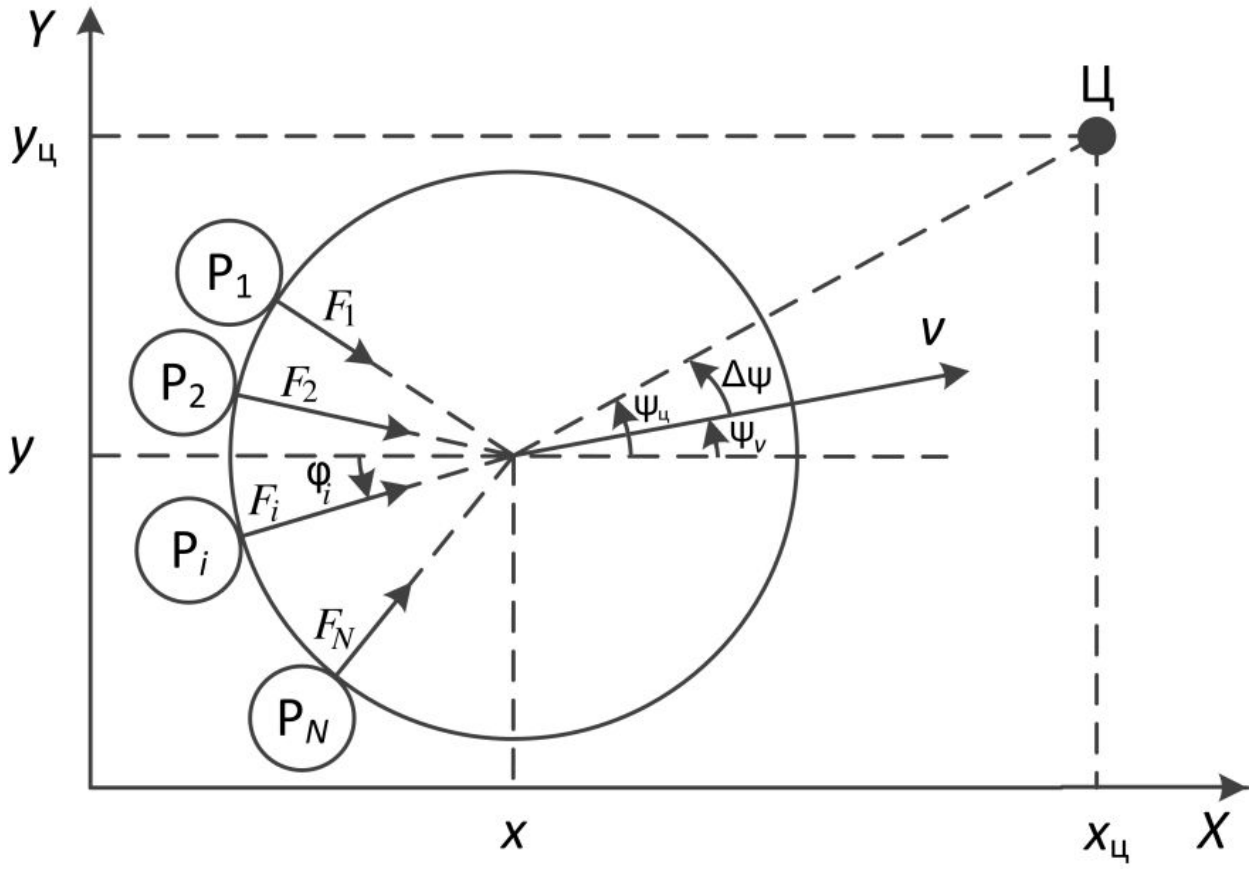
Результаты обучения маятника



Мультиагентное обучение с подкреплением

- Наиболее актуальная на настоящее время область исследований.

Задача перемещения твёрдого тела группой роботов
(отсутствие прямой информационной связи)



Постановка задачи

- В разных точках вдоль периметра цилиндра находятся роботы, давящие на него с разной силой.
- Роботы не могут друг с другом обмениваться сообщениями.
- Роботам необходимо переместить цилиндр к удалённой точке, находящейся на расстоянии порядка сотен метров.
- Каждый робот обучается самостоятельно. Остальные роботы для него – неизвестные факторы окружающей среды.

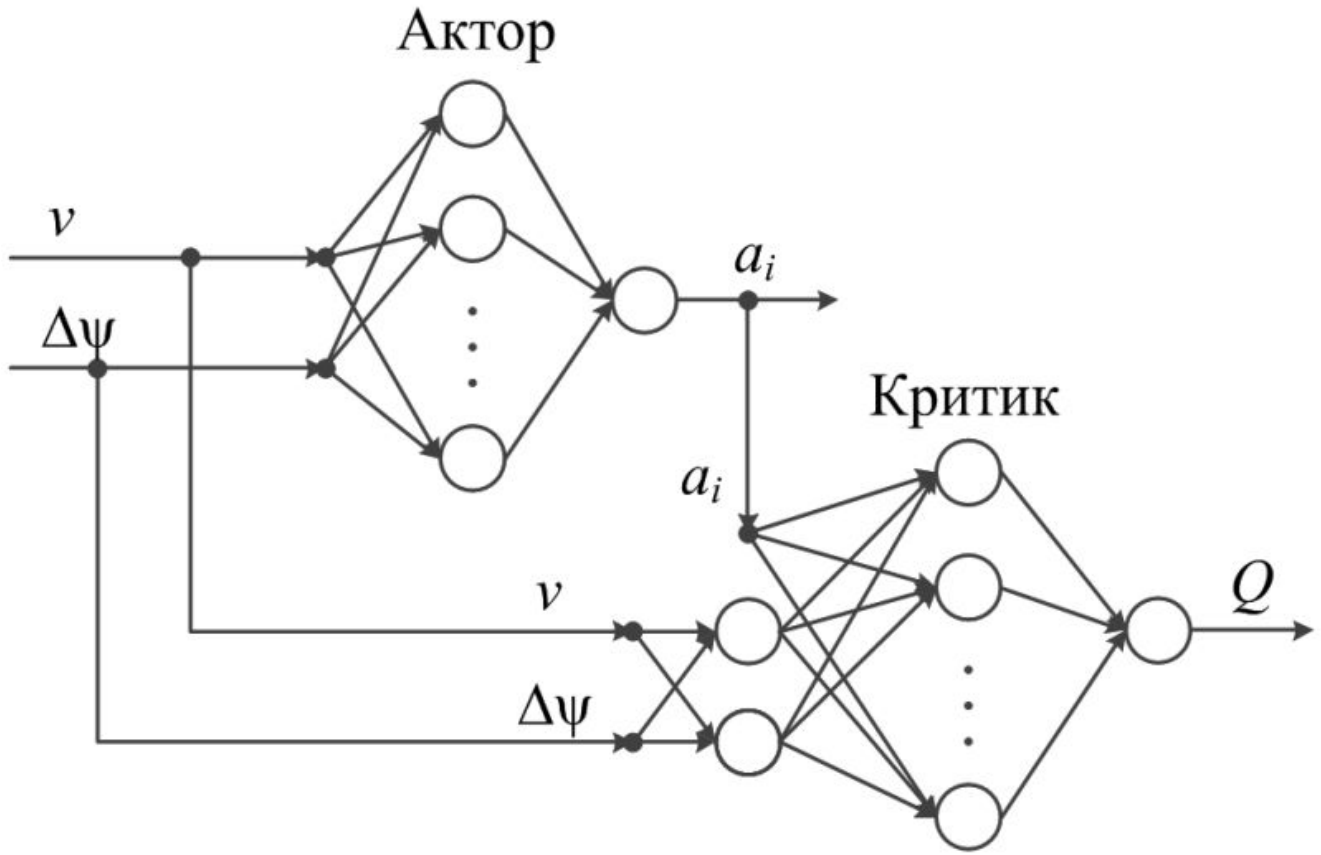
Подход к решению задачи

- В каждом роботе используется независимая система адаптивной критики.
- Обучение происходит полностью за время движения.
- Каждый робот в результате обучения получает уникальную роль в коллективе.

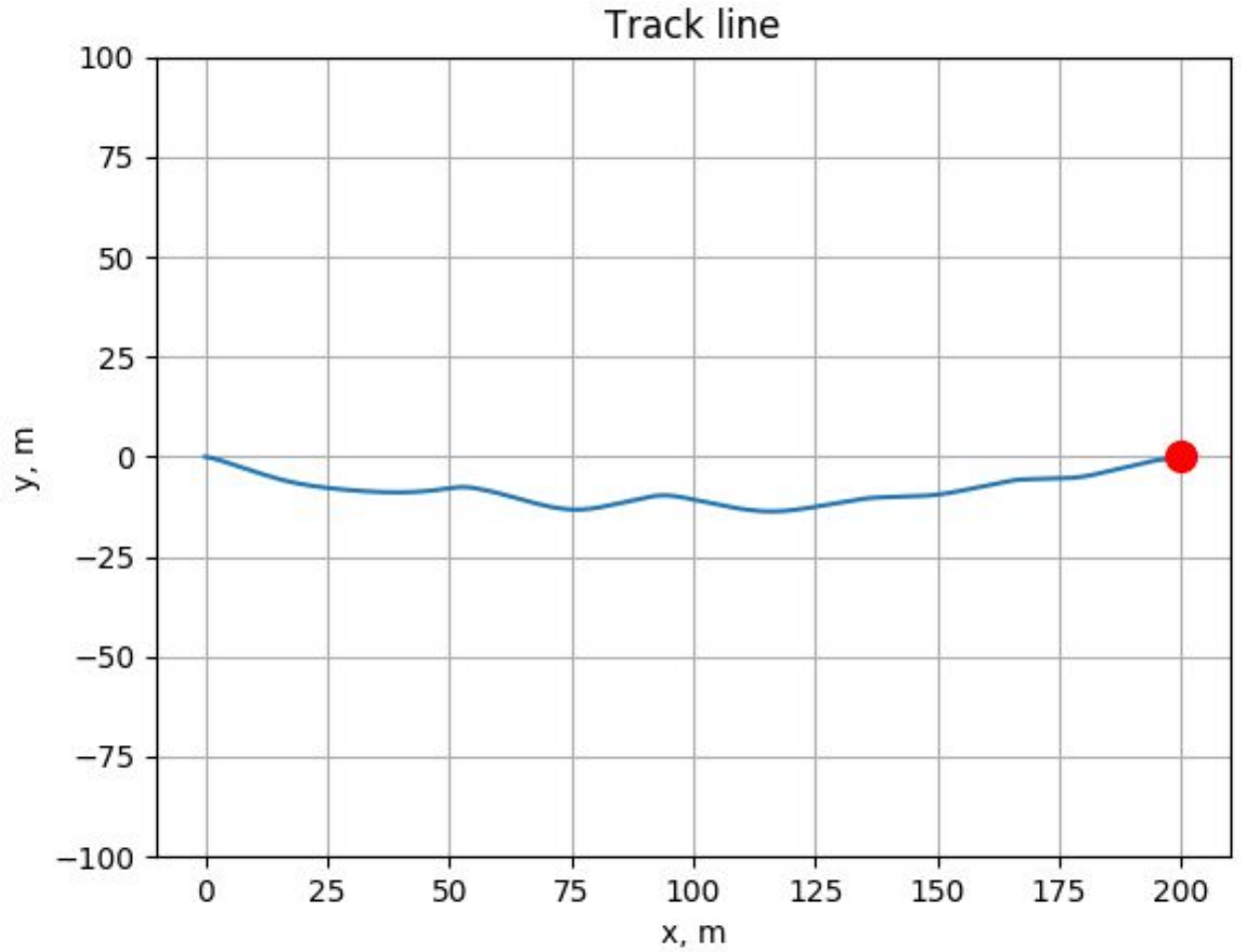
Подход к решению задачи

- Робот измеряет **скорость движения и угол отклонения направления движения от направления к цели**. Эти данные характеризуют **состояние**.
- **Действие** робота – **величина силы**, с которой он действует на цилиндр.
- **Награда** тем больше, чем меньше **отклонение угла** направления движения от направления к цели и немного возрастает при увеличении скорости.

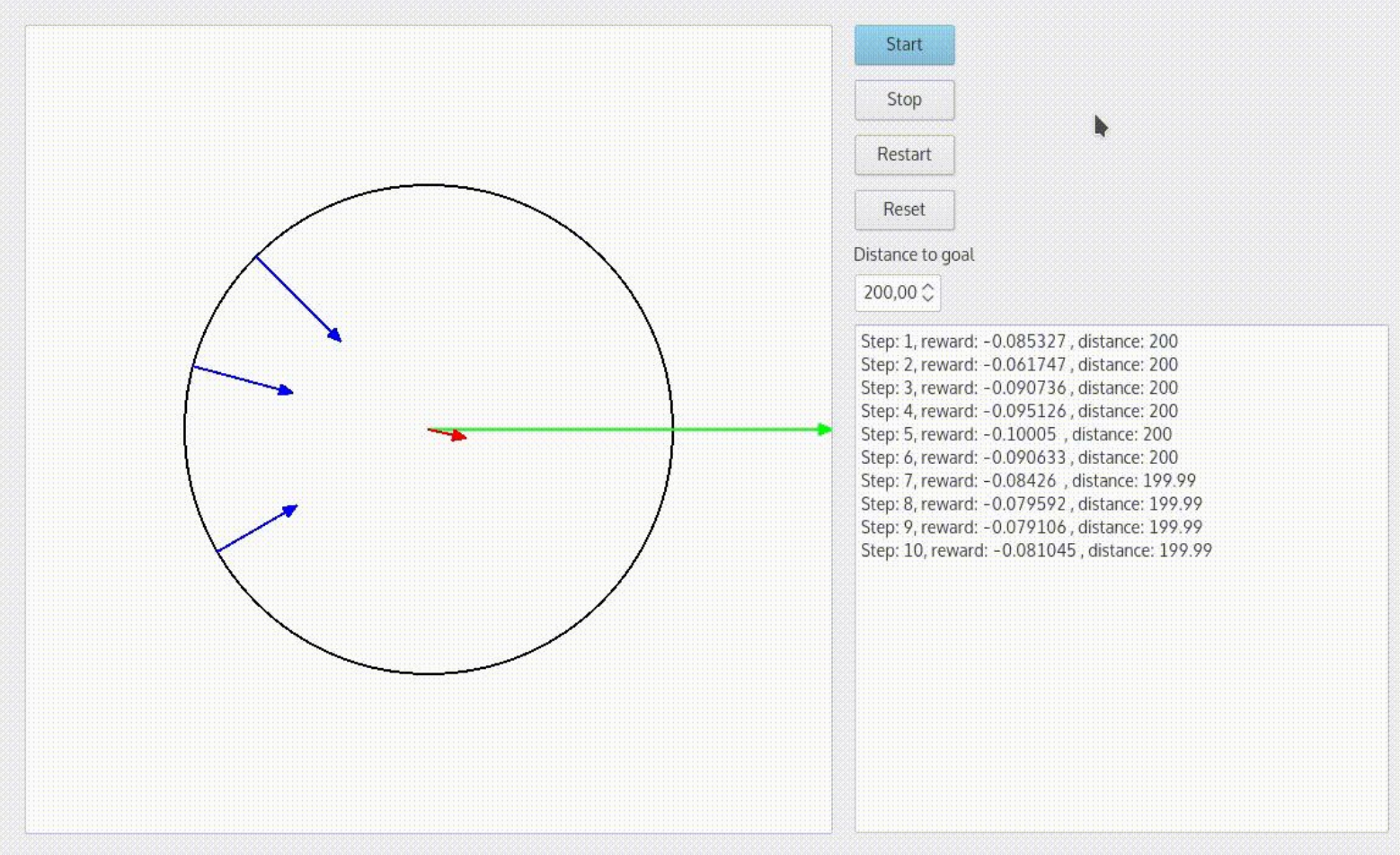
Структура актора и критика



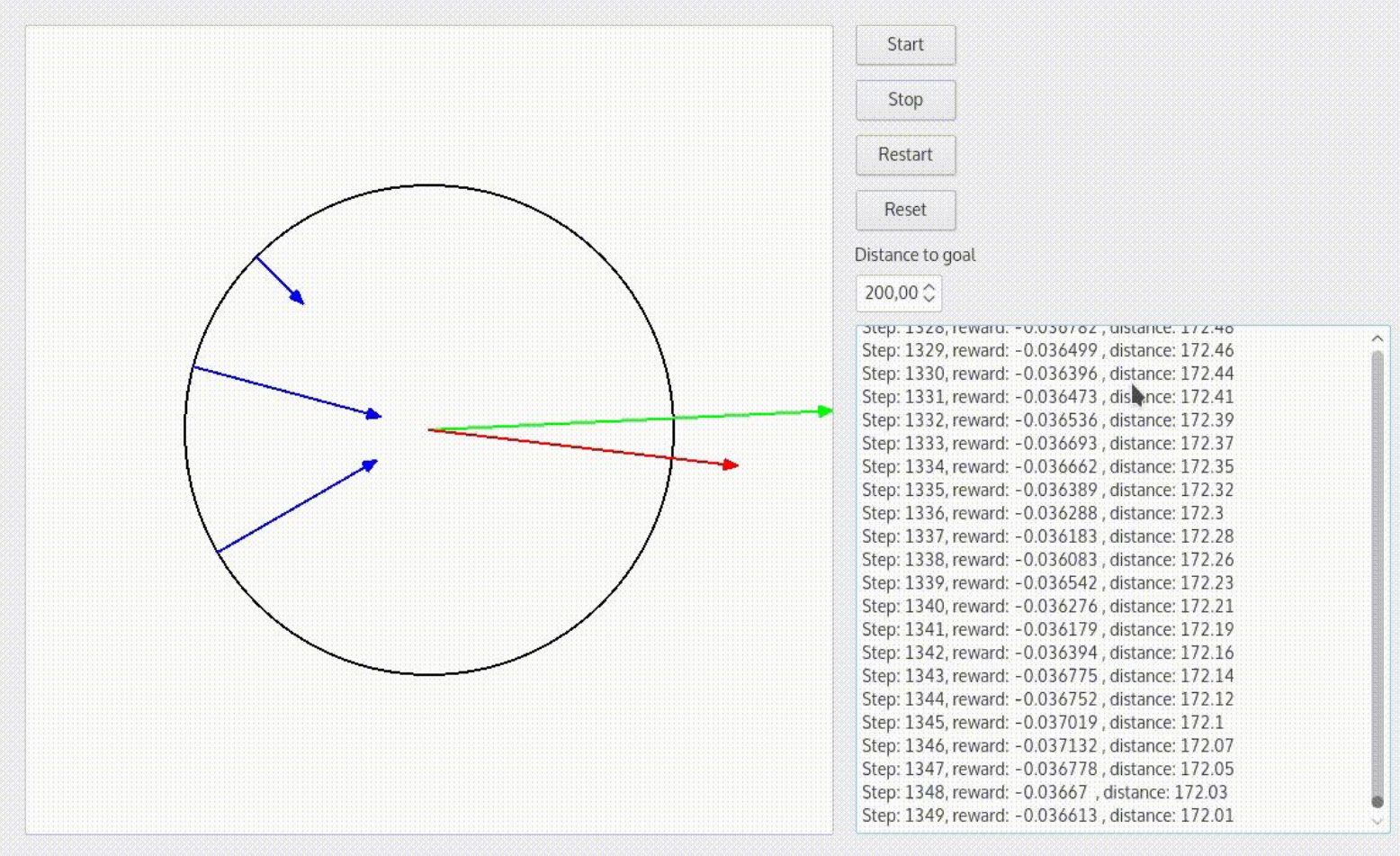
Вычислительный эксперимент – траектория перемещения тела



Визуализация работы трёх роботов



Визуализация работы трёх роботов



Благодарю за внимание!