



ЦЕНТР МОЛОДЁЖНОЙ  
РОБОТОТЕХНИКИ  
МГТУ ИМ. Н.Э. БАУМАНА

# Базовое управление четвероногими роботами

# Научно-исследовательский кружок по ИИ в робототехнике

Организован при содействии  
Лаборатории Робототехники  
Сбербанка

## Шагающие роботы:

Unitree Aliengo  
Tinker

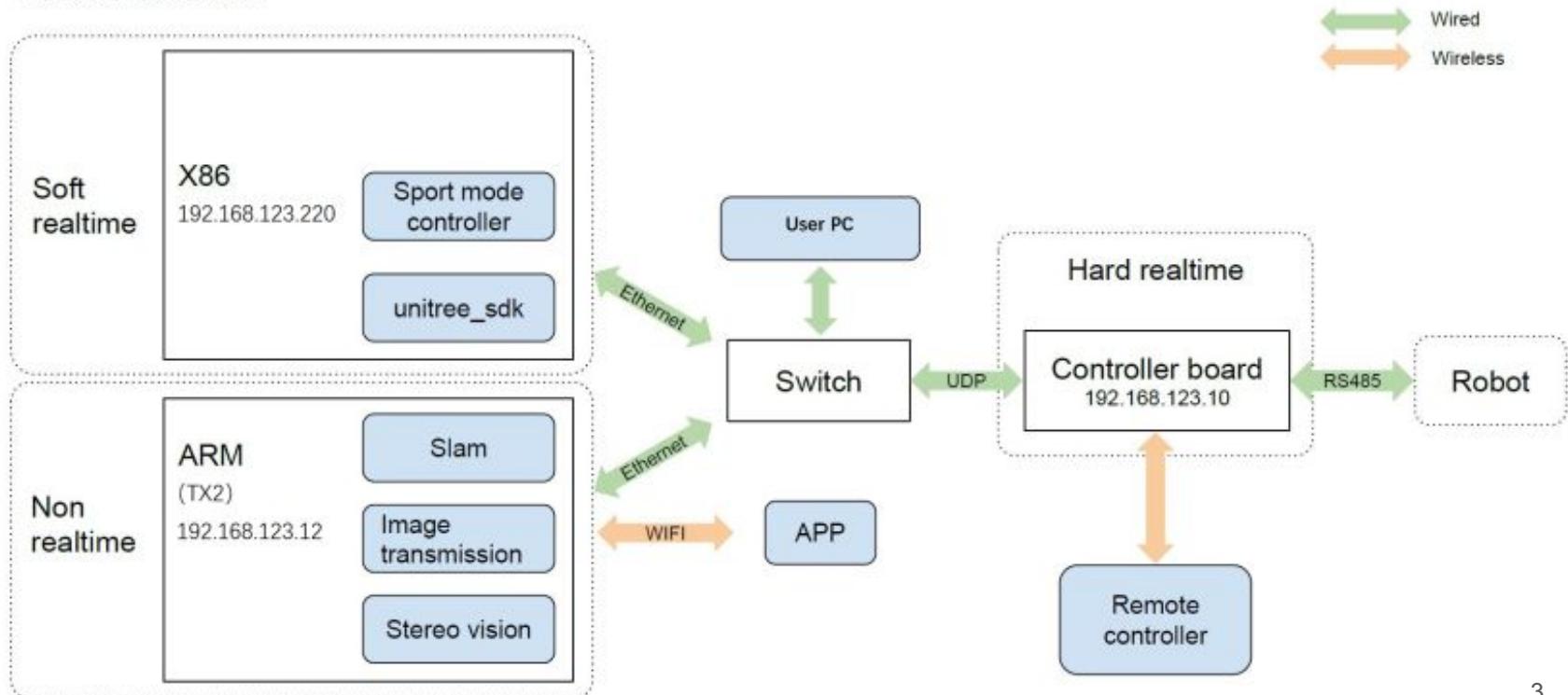
### Зачем кружок:

- Написание НИР, дипломов и статей по шагающим роботам
- Выступление на конференциях
- Создание своих проектов на основе шагающего робота



# Архитектура Unitree-робота

AlienGo architecture



## Режимы работы

- low-level (192.168.123.10) -> Управляющая плата
- high-level (192.168.123.220) -> Mini PC

Переключить в low-level:

- L2+A, L2+A, L2+B - опуститься и расслабить моторы
- L1 + L2 + Start - отключение спортивного режима

# Код проекта

Оригинальное сдк для управления unitree-роботами. Нам нужна ветка Aliengo.



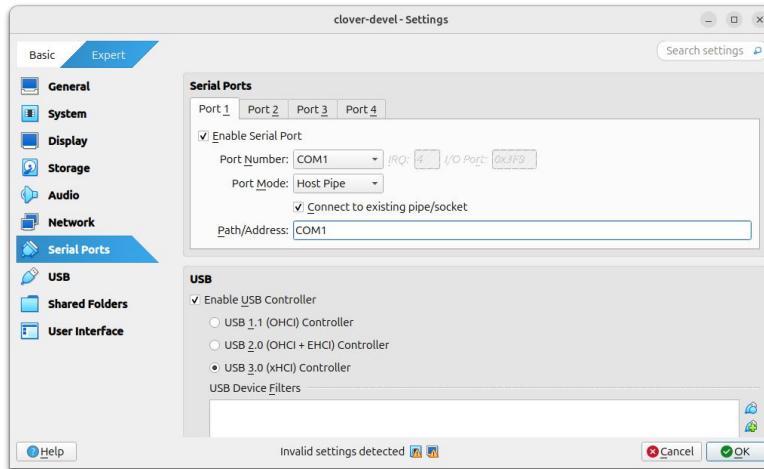
**Проект-обертка**, в котором `unitree_legged_sdk` — один из подмодулей. С помощью `rl_go1` скрипта без изменения может быть запущен как в симуляторе `Mujoco`, так и на реальном роботе



# Запуск через VirtualBox

В виртуальной машине:

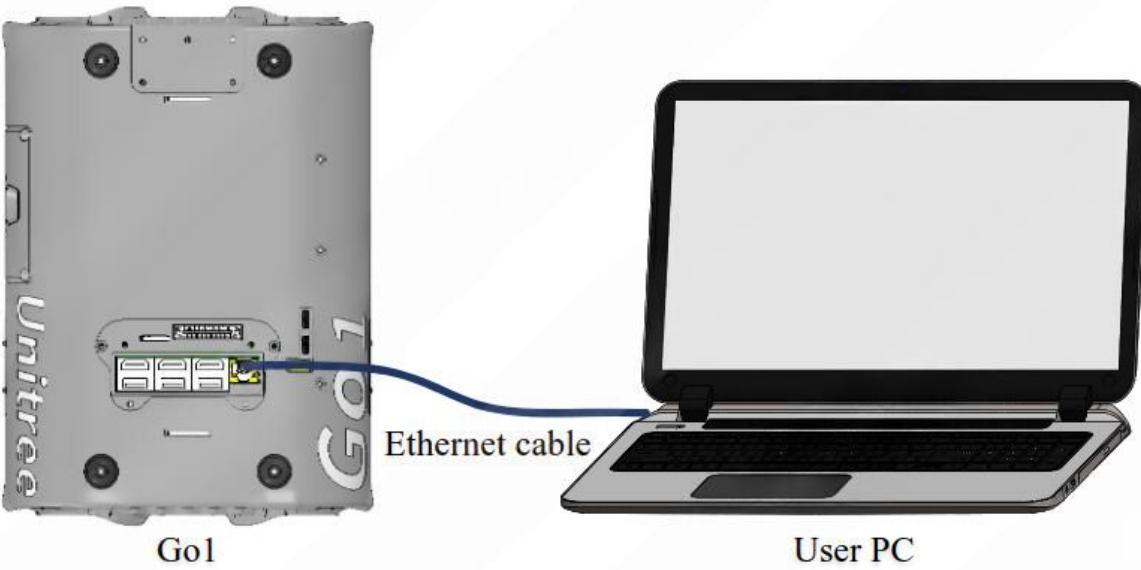
- VSCode
- Проект rl\_gol
  - unitree\_legged\_sdk
  - mujoco (симулятор)



Внутри контейнера выполнить:

```
conda activate rl_gol
cd rl_gol
python scripts/standup.py
```

# Управление через СДК



```
sudo ifconfig eth0 down # ethxxx is your PC port  
sudo ifconfig eth0 192.168.123.200/24  
sudo ifconfig eth0 up  
ping 192.168.123.10
```

# Управление через СДК

Cancel **Wired** Apply

Details Identity **IPv4** IPv6 Security

**IPv4 Method**

Automatic (DHCP)  Manual  Link-Local Only  
 Shared to other computers  Disable

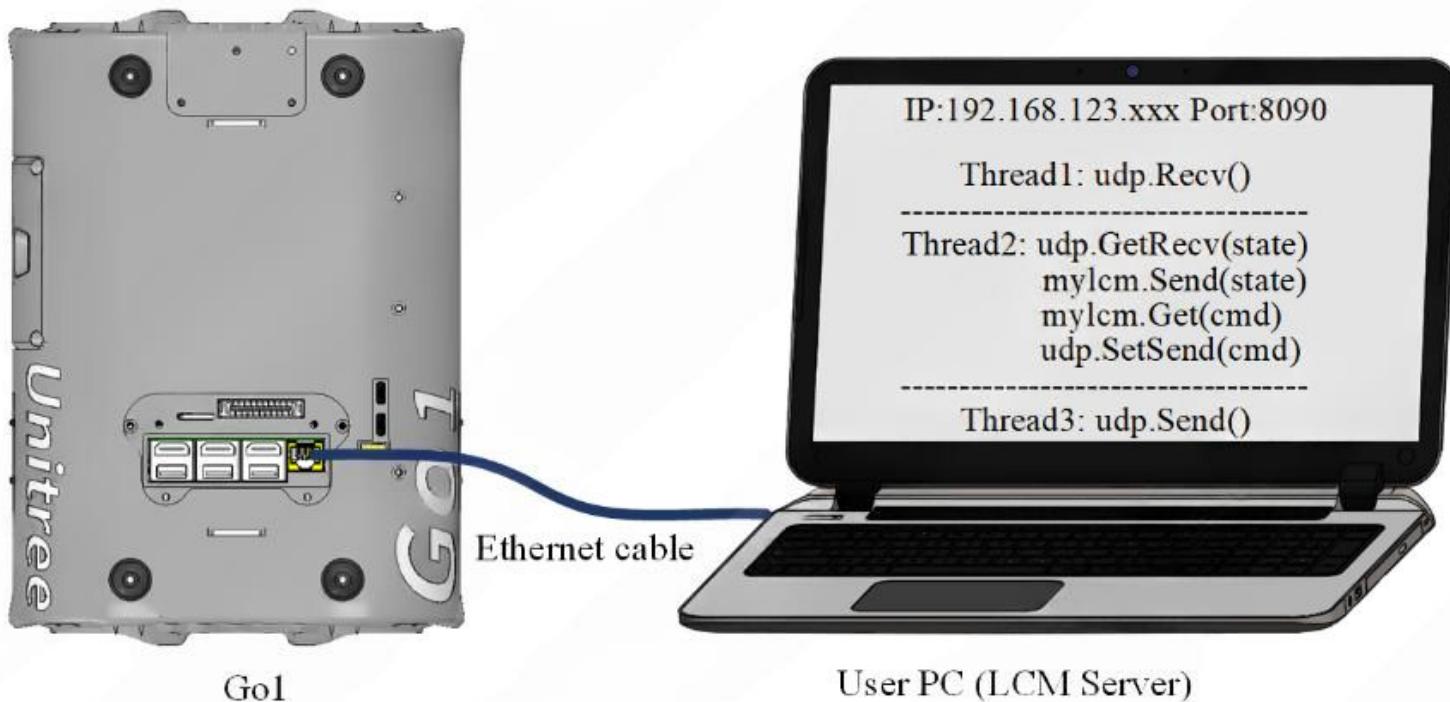
**Addresses**

Address	Netmask	Gateway
192.168.123.200	255.255.255.0	

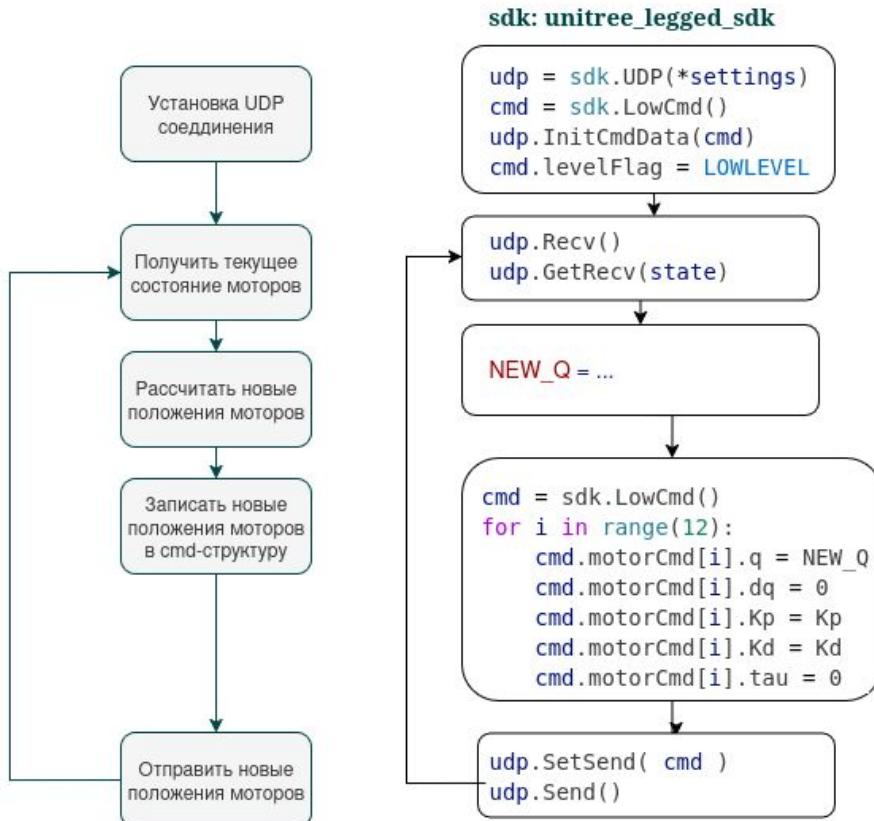
**DNS** Automatic

Separate IP addresses with commas

# Структура кода



# Структура кода



Нужно установить UDP-соединение с роботом, а затем в цикле получать текущие состояния в структуре `LowState`, рассчитывать на их основе новые — и отправлять их роботу в структуре `LowCmd`.

# Управление моторами

- Угловое положение ротора —  $q$
- Угловая скорость ротора —  $dq$
- Крутящий момент ротора —  $\tau_{\text{au}}$
- Пропорциональный коэффициент ПД-регулятора (коэффициент жесткости) —  $K_p$
- Дифференциальный коэффициент ПД-регулятора (коэффициент затухания) —  $K_d$

Position mode:

$$\begin{aligned} q &= q(\text{time}) \\ dq &= 0 \\ K_p &= K_p_{\text{const}} \\ K_d &= K_d_{\text{const}} \\ \tau_{\text{au}} &= 0 \end{aligned}$$

Speed mode:

$$\begin{aligned} q &= \text{PosStopF} \\ dq &= dq(\text{time}) \\ K_p &= 0 \\ K_d &= K_d_{\text{const}} \\ \tau_{\text{au}} &= 0 \end{aligned}$$



Torque mode:

$$\begin{aligned} q &= \text{PosStopF} \\ dq &= \text{VelStopF} \\ K_p &= 0 \\ K_d &= 0 \\ \tau_{\text{au}} &= \tau_{\text{au}}(\text{time}) \end{aligned}$$

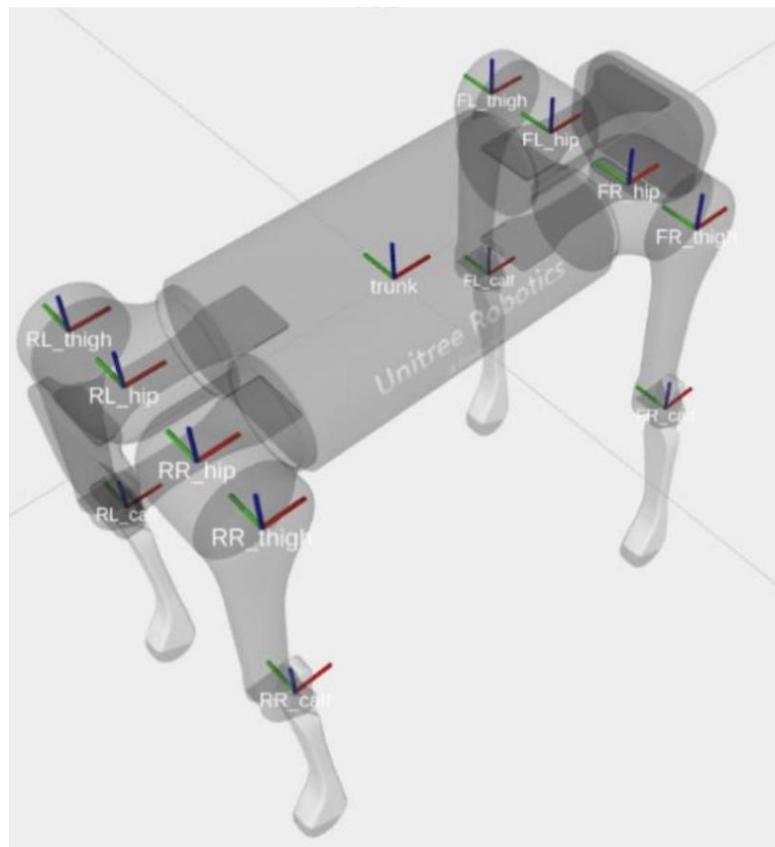
# Управление моторами

Суммарный крутящий момент:

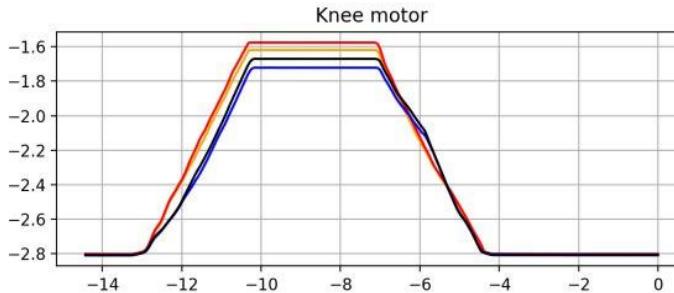
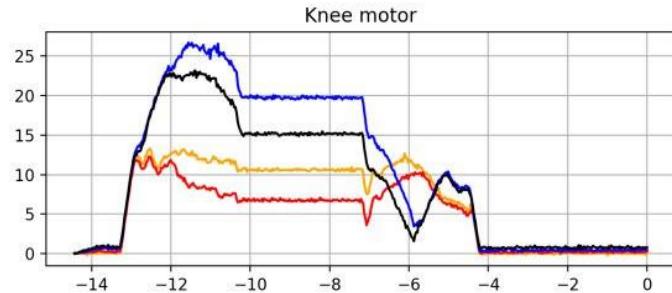
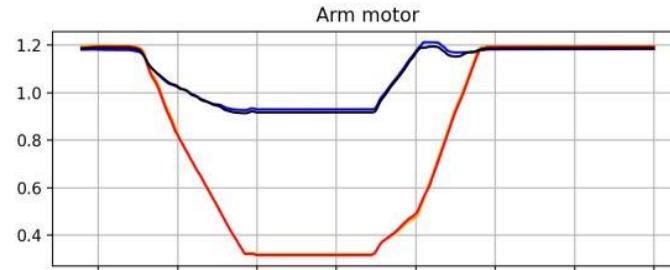
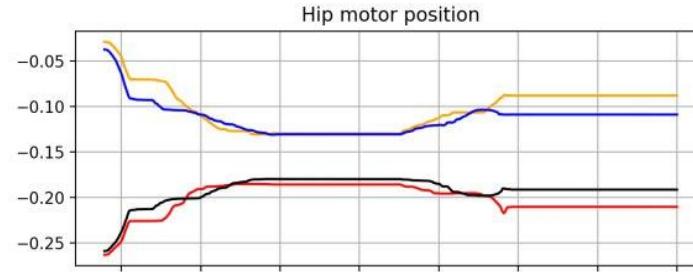
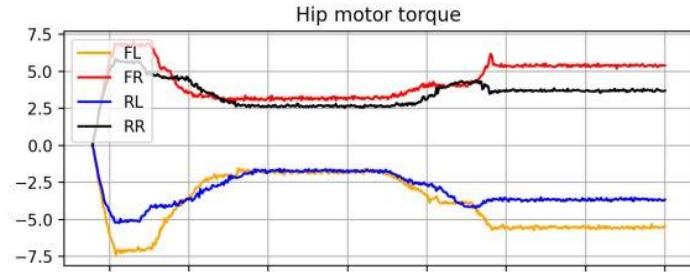
$$\text{out} = \tau + K_p * (q_{\text{des}} - q) + K_d * (dq_{\text{des}} - dq)$$

Состоит из:

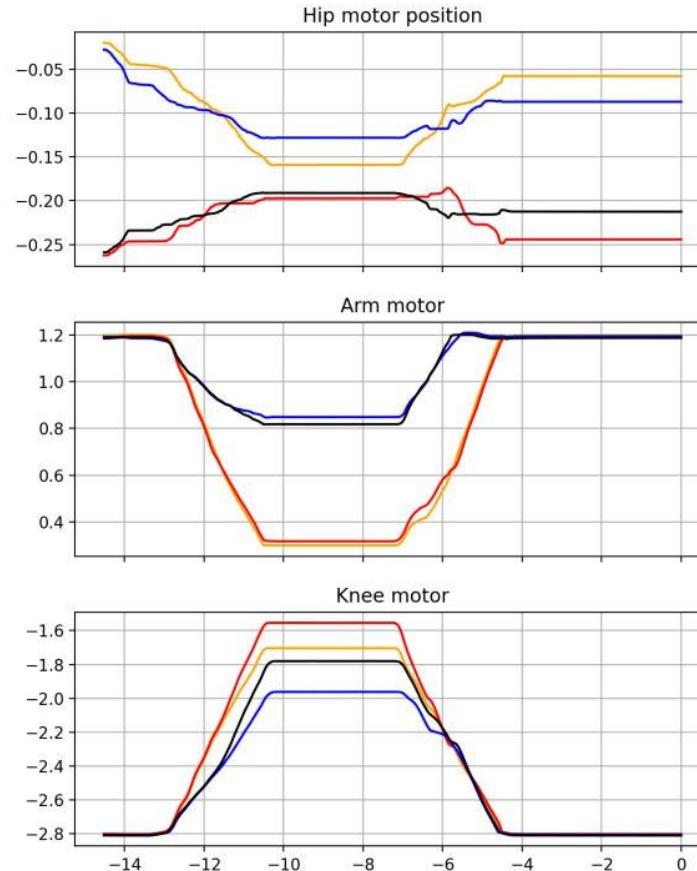
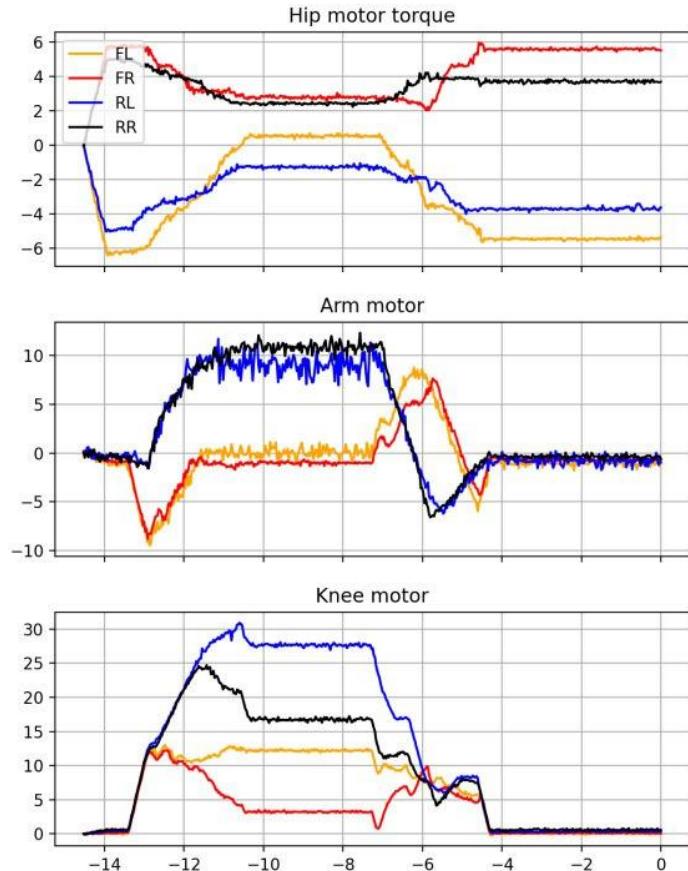
1.  $\tau$  - уменьшает ошибку регулятора, компенсируя известные силы
2.  $K_p * (q_{\text{des}} - q)$  - Пропорциональная составляющая
3.  $K_d * (dq_{\text{des}} - dq)$  - Дифференциальная составляющая



$$K_p = 90, K_d = 2$$



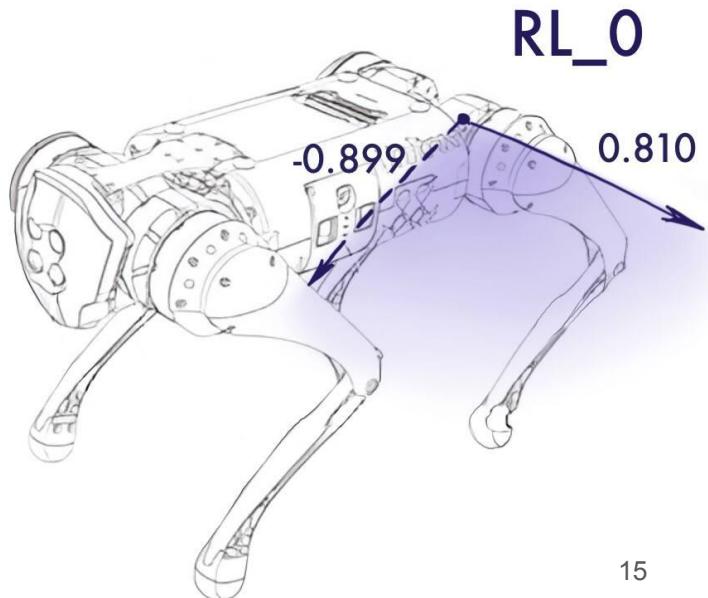
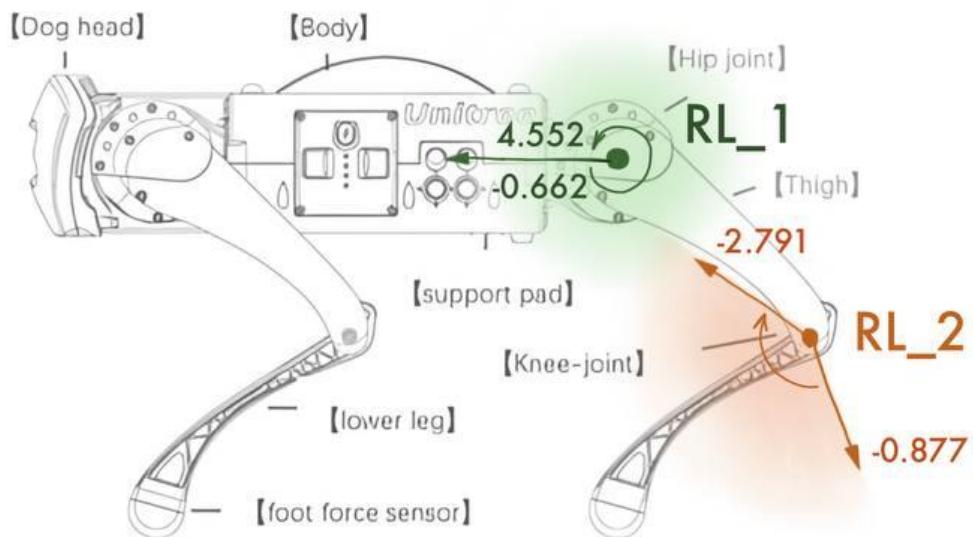
$$K_p = 60, K_d = 2$$



# Границы значений моторов

```
...  
('RL_0', -0.899, 0.810), # hip -70°~70°  
('RL_1', -0.662, 4.552), # thigh -360°~360°  
('RL_2', -2.791, -0.877) # calf -159°~37°
```

...



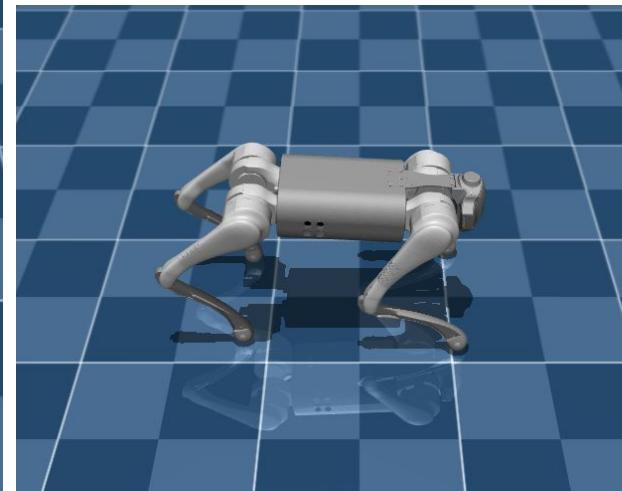
Фаза 1



Фаза 2



Фаза 3



# Интерполяция

Нельзя за одну секунду резко изменить положение моторов - моторы сгорят.

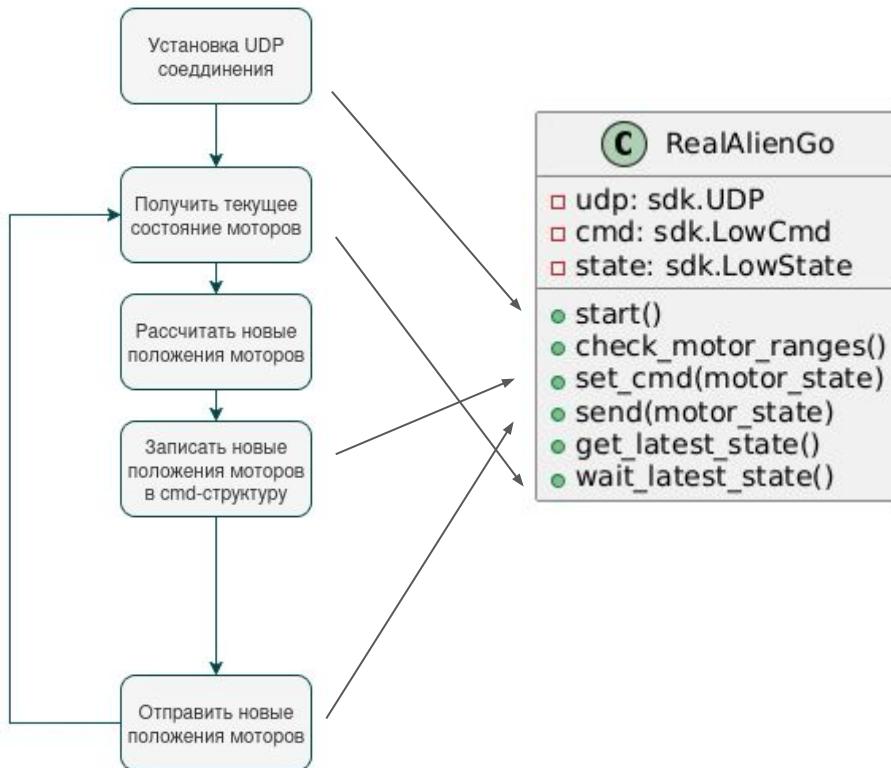
Нужно разбить изменение на маленькие шаги, например, с помощью линейной интерполяции.

```
def jointLinearInterpolation(initPos, targetPos, rate):  
  
    rate = min(max(rate, 0.0), 1.0)  
  
    p = initPos*(1-rate) + targetPos*rate  
    return p
```

# Структура проекта



# Структура кода



**C RealAlienGo**

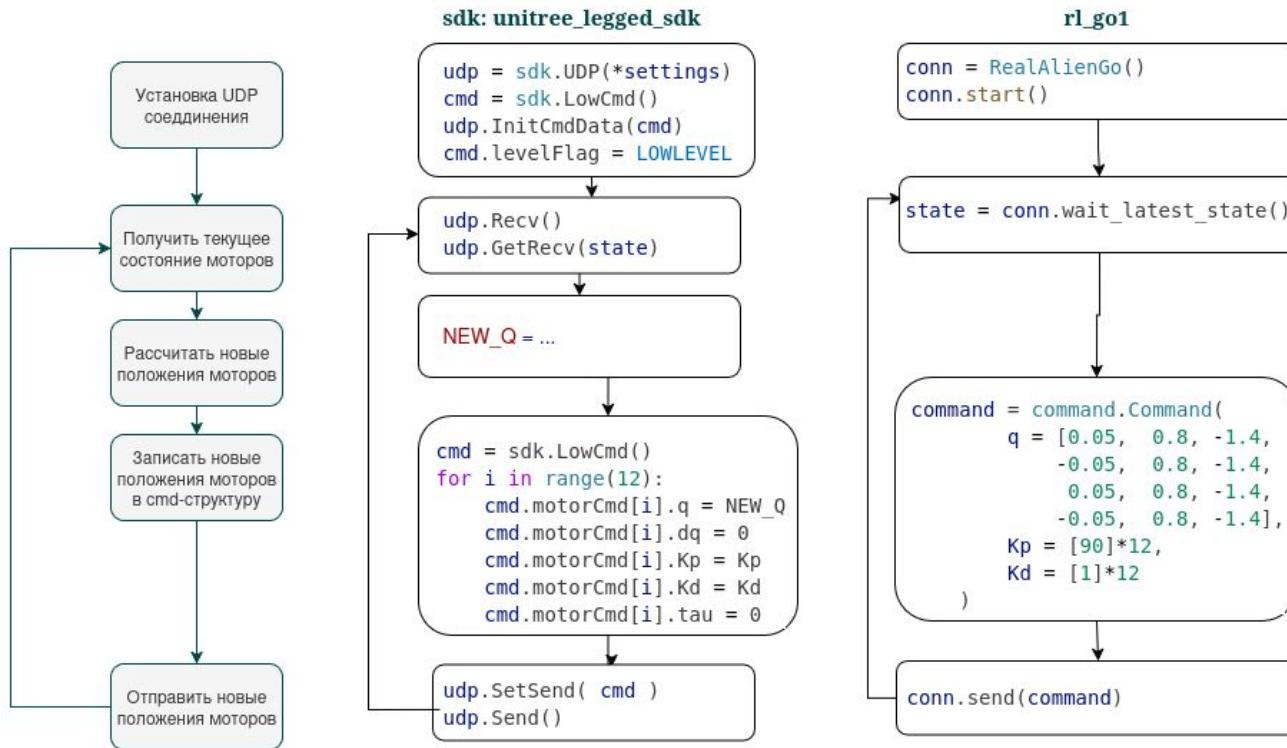
- ❑ udp: sdk.UDP
- ❑ cmd: sdk.LowCmd
- ❑ state: sdk.LowState
- start()
- check\_motor\_ranges()
- set\_cmd(motor\_state)
- send(motor\_state)
- get\_latest\_state()
- wait\_latest\_state()

**C Simulation**

- ❑ mj\_model: mujoco.MjModel
- ❑ mj\_data: mujoco.MjData
- ❑ config: Any
- ❑ states: deque
- ❑ cmd: Any
- ❑ viewer: Any
- \_\_init\_\_(config)
- start()
- control(cmd)
- send\_impl(cmd)
- get\_states\_impl()



# Структура кода



С помощью rl\_go1 скрипт без изменения может быть запущен как в симуляции, так и на реальном роботе

# Напишите свой скрипт

Измените скрипт `standup` — измените целевые значения q моторов, количество фаз. Например, можно научить робособаку сидеть.

Протестируйте скрипт в тијосо.



## Управление локомоцией

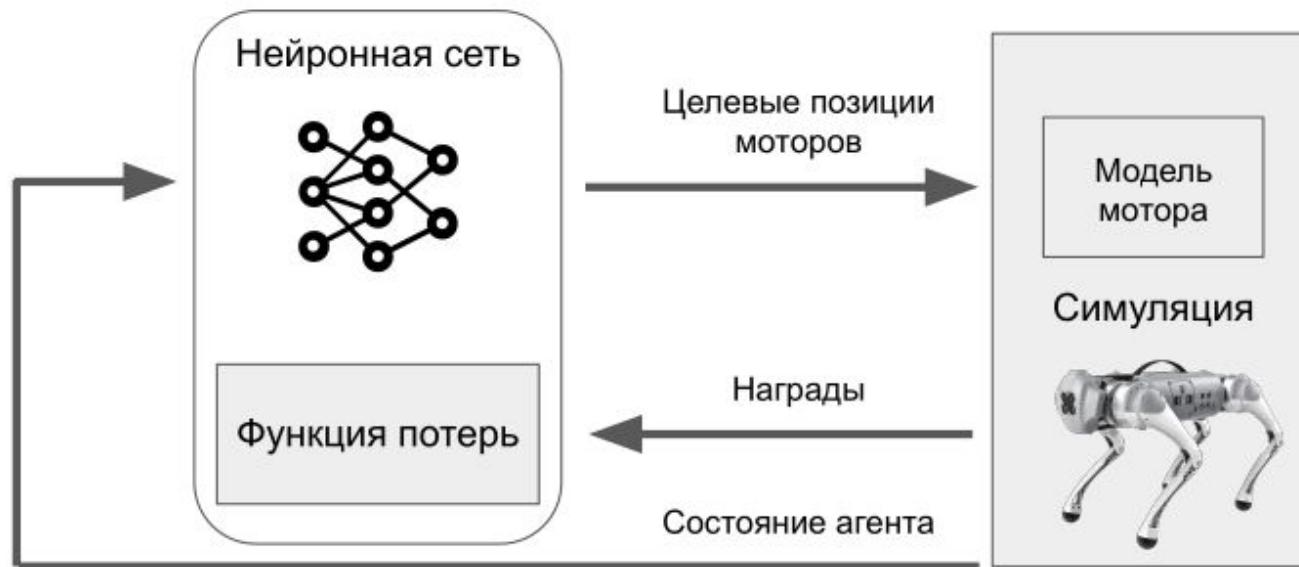
Методы оптимального  
управления

Гибридные методы

Методы глубокого обучения  
с подкреплением (RL)



# Обучение с подкреплением

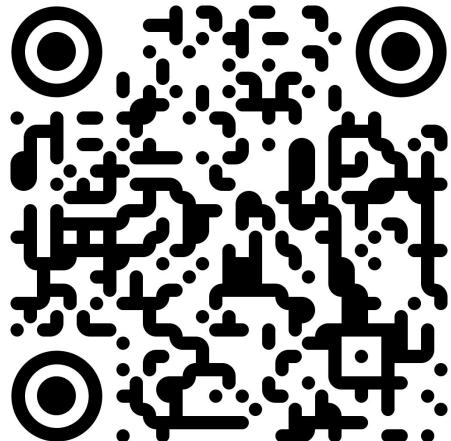


# Дополнительные материалы

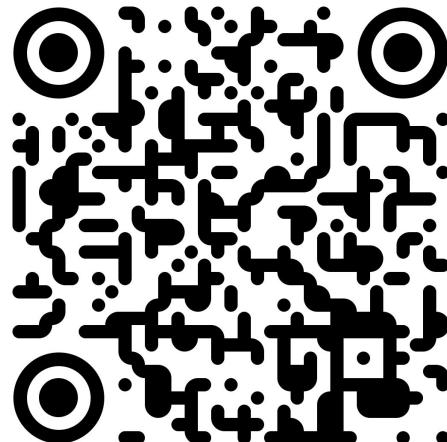
Сайт с гайдами и полезными ссылками  
про Unitree Aliengo и RL.

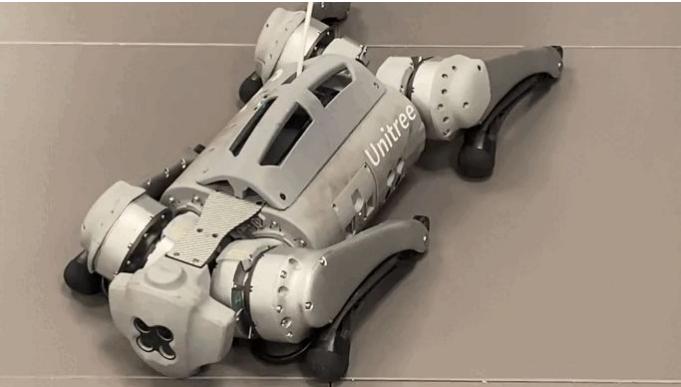
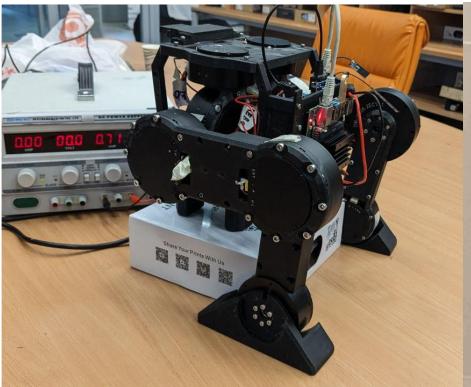
СДК для управления Aliengo через ROS  
[Noetic](#). В проекте есть docker для быстрой  
сборки.

Документация



unitree\_ros\_to\_real



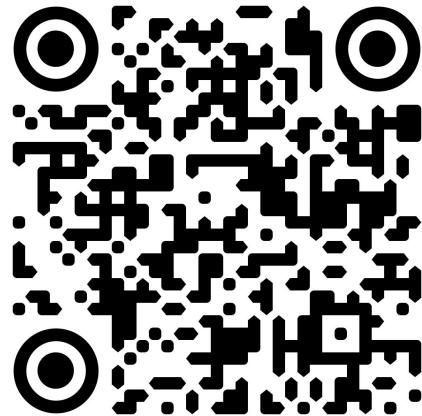


# Кружок шагающих роботов

Сборка и программирование  
шагающих роботов

Проект от Центра Робототехники  
Сбербанка

Статья на хабре



Спасибо за внимание

