

ai

ÚSTAV AUTOMATIZACE  
A INFORMATIKY



VYSOKÉ UČENÍ FAKULTA  
TECHNICKÉ STROJNÍHO  
V BRNĚ INŽENÝRSTVÍ

[uai.fme.vutbr.cz](http://uai.fme.vutbr.cz)

# Využití Robotického operačního systému (ROS) pro řízení kolaborativního robota UR3

Bakalářská práce

Autor : Martin Juříček

Vedoucí : Ing. Roman Parák

# Kolaborativní robotika

# Současný stav kolaborativní robotiky



ABB SWIFTI



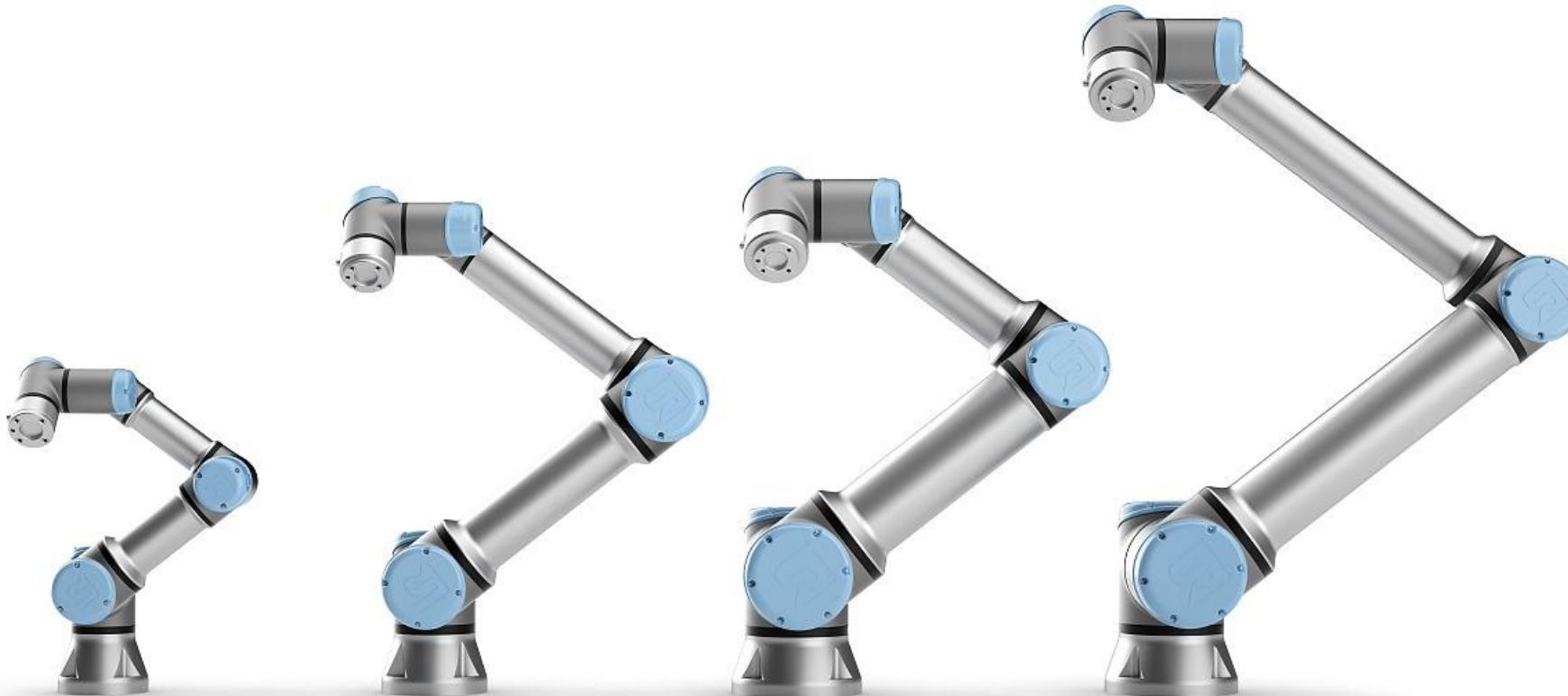
ABB GoFa

# Současný stav kolaborativní robotiky



Doosan a0509  
<https://github.com/doosan-robotics/doosan-robot>

# Současný stav kolaborativní robotiky



Universal Robots eSeries

# Současný stav kolaborativní robotiky



Universal Robots UR3



AUBO i3

# Současný stav kolaborativní robotiky

## FANUC - CR-35iA

- nosnost: 4-35 kg
- hmotnost: 990 kg
- dosah: 1813 mm
- rychlosť: 750 mm/s
- instalace: podlaha



Fanuc CR-35iA

Fanuc kontrolér

## Programování a řízení

- ovládací panel
- gesta X hlasové příkazy
- programy založené na CAD
- průmyslové programovací jazyky



Fanuc ovládací panel  
ipendant-Touch

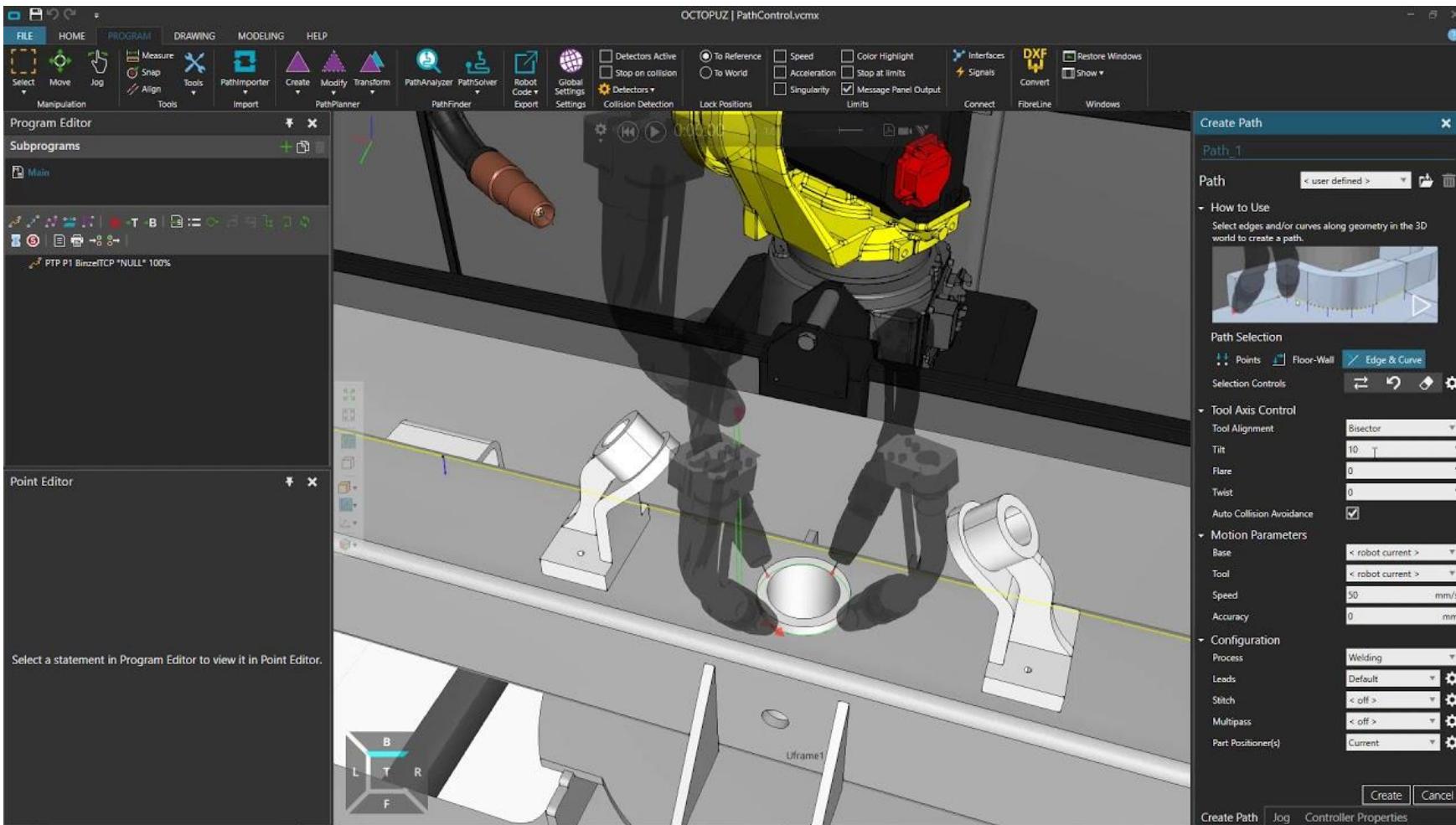
## Příklady

- FANUC iPendant-Touch
- Microsoft Speech API
- Visual Components
- FANUC Karel

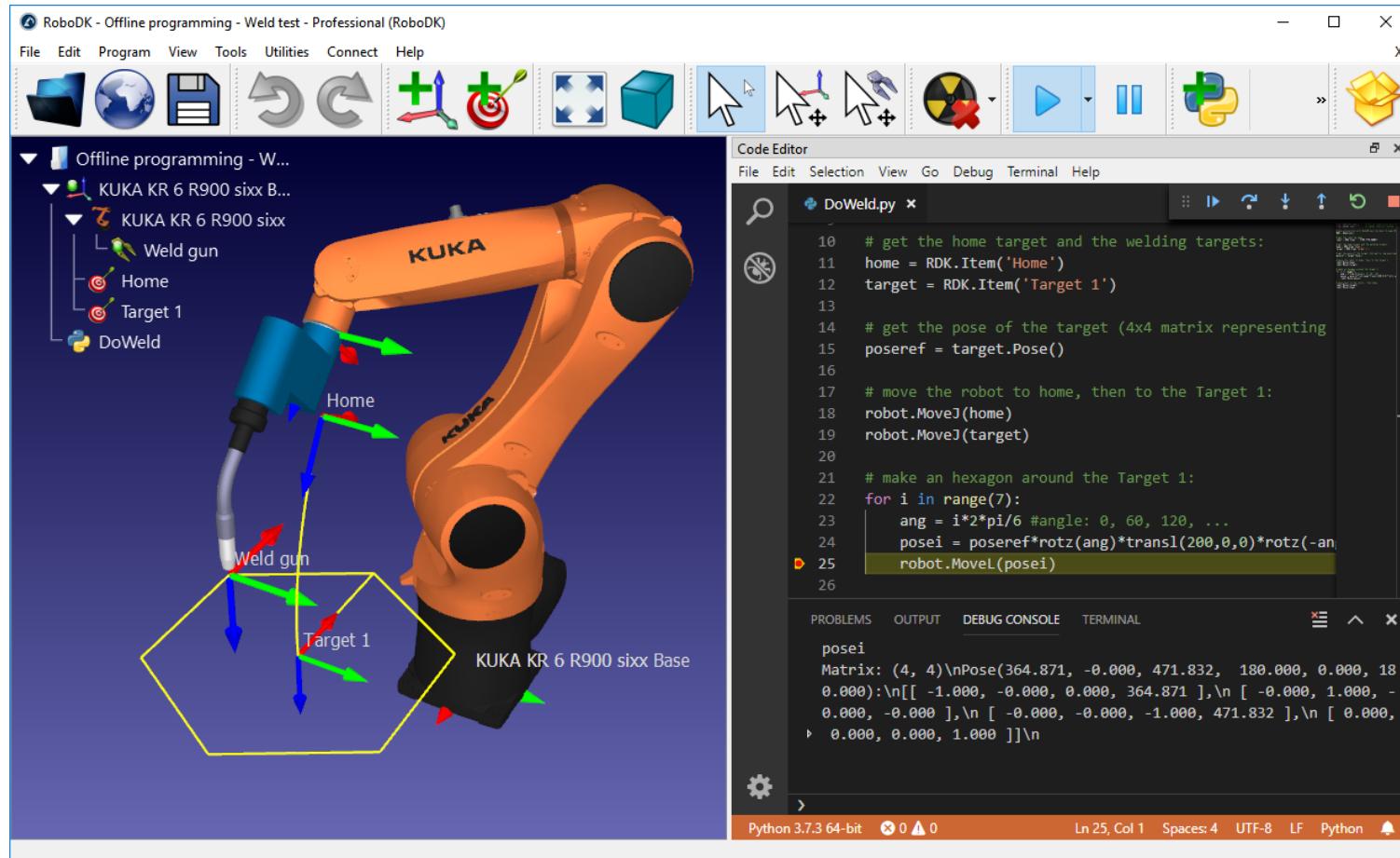


Náhled do programu Visual Components

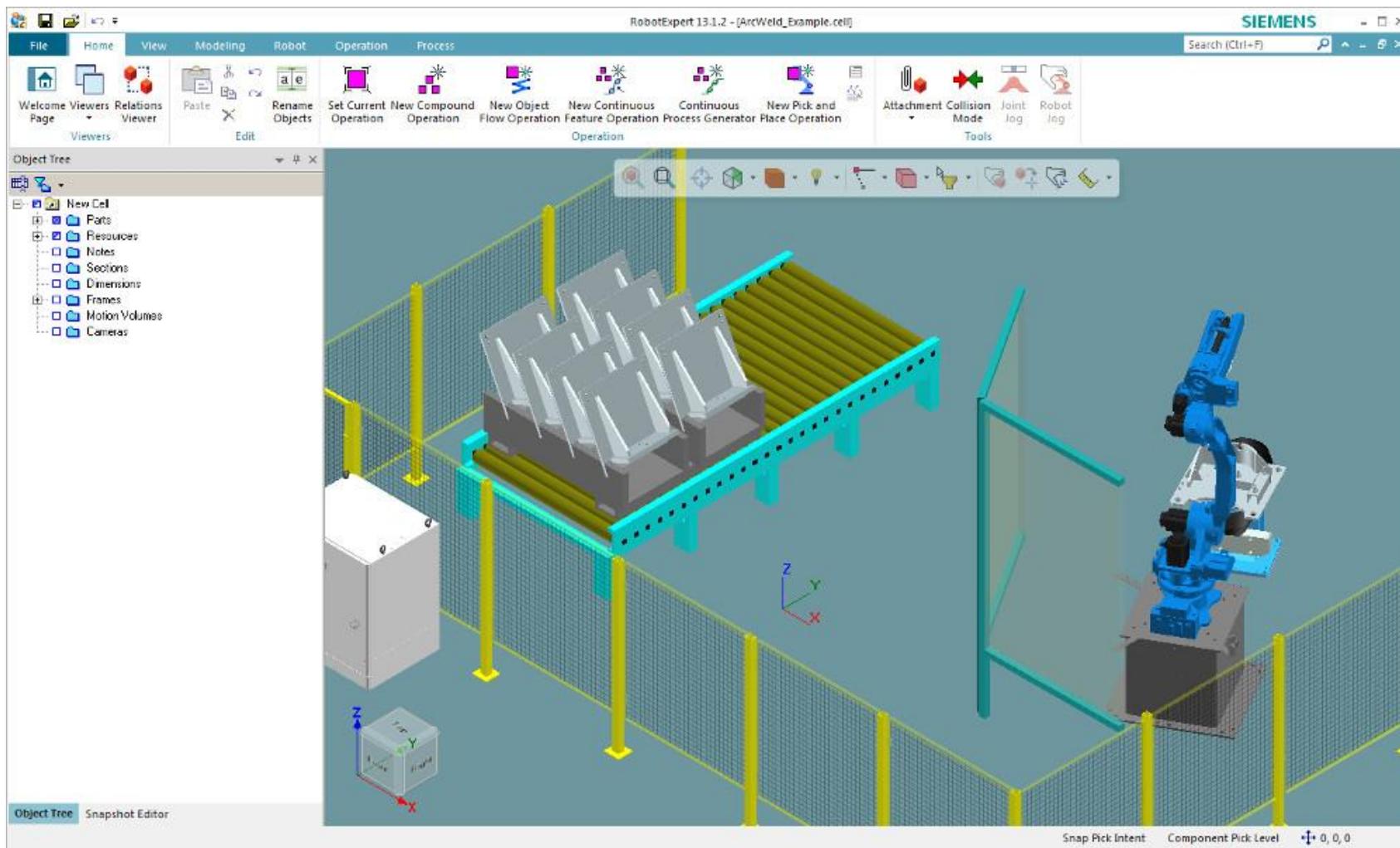
# Současný stav kolaborativní robotiky



# Současný stav kolaborativní robotiky



# Současný stav kolaborativní robotiky



# Aplikace kolaborativních robotů

## Aplikace

- zdravotnictví
- potravinářský průmysl
- strojní průmysl
- elektro průmysl
- farmaceutický průmysl
- chemický průmysl



Universal Robots: kobot při manipulaci se vzorkem krve

## Kladené podmínky

- bezpečnost
- přesnost a preciznost
- flexibilita
- jednoduchá ovladatelnost
- *hygiena*



AOT CARLO: osteotomický nástroj

## Příklady

- Inspekce
- Bin-picking
- Pick-and-place
- ADAMO ROBOT
- KUKA KMR iiwa



ADAMO: fyzioterapeutická aplikace

# Aplikace kolaborativních robotů



KUKA ROBERT

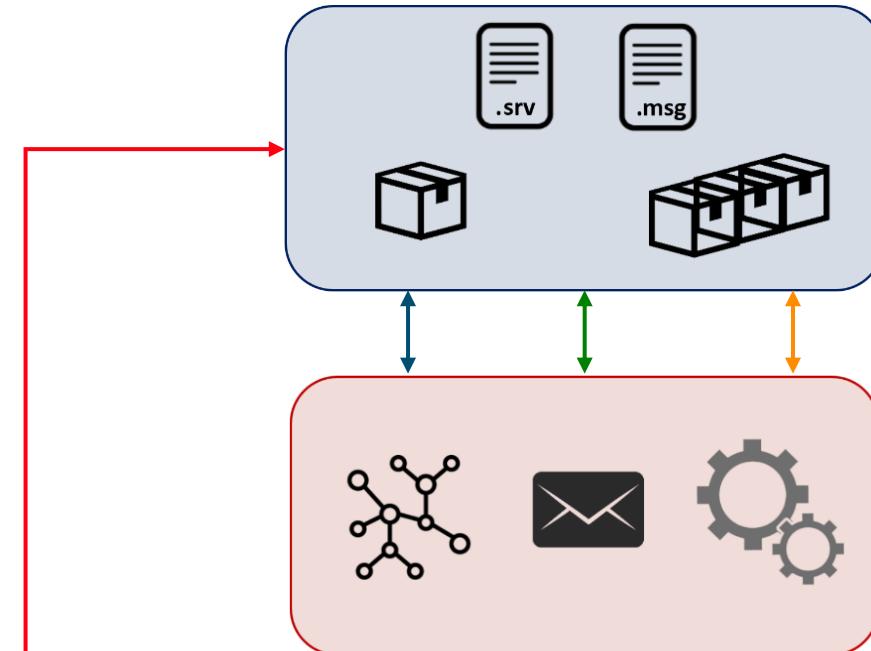


KUKA KMR iiwa

# ROS - Robotický Operační Systém

## Úroveň souborového systému (filesystem level)

- balíky => metabalíky
- typ zpráv X typ služeb

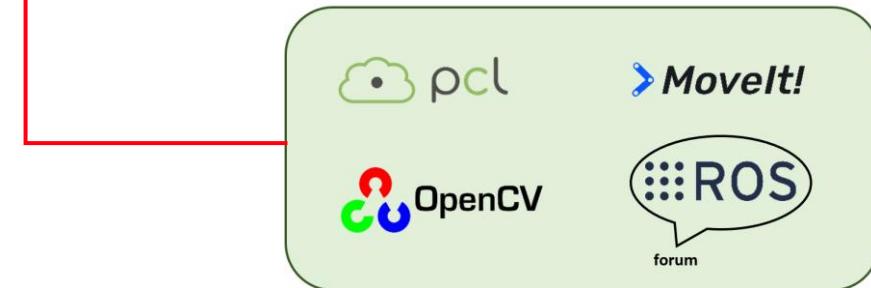


## Grafová výpočtová úroveň (computation graph level)

- uzly
- zprávy X služby
- master
- téma

## Úroveň komunity (community level)

- fóra
- ROS wiki
- knihovny



úrovně ROS

# ROS - Robotický Operační Systém

## Distribuce ROS

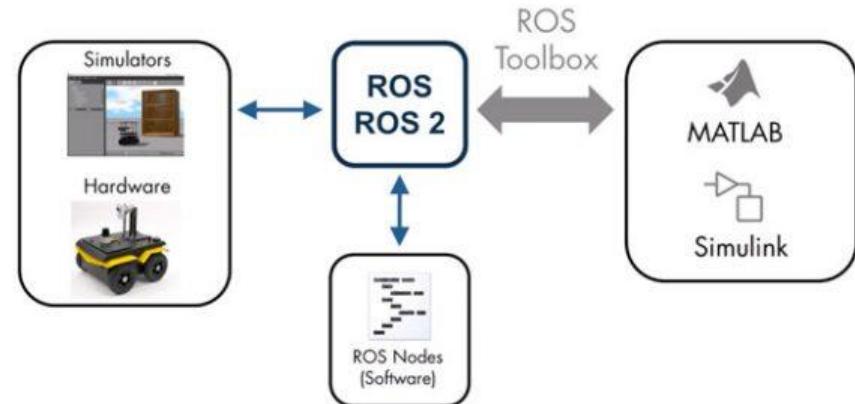
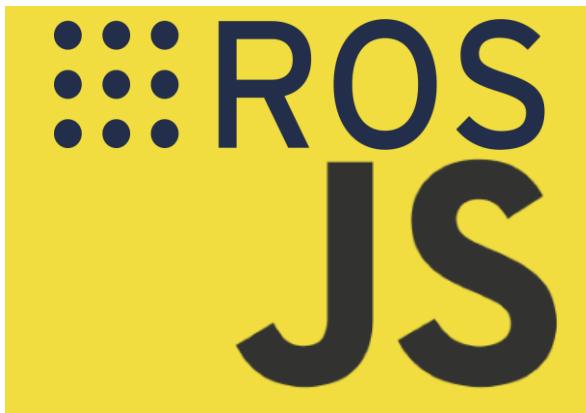
- Box Turtle
- Kinetic Kame -> Melodic Morenia



plakáty ROS distribucí



# ROS - Robotický Operační Systém



# ROS – Simulační prostředí

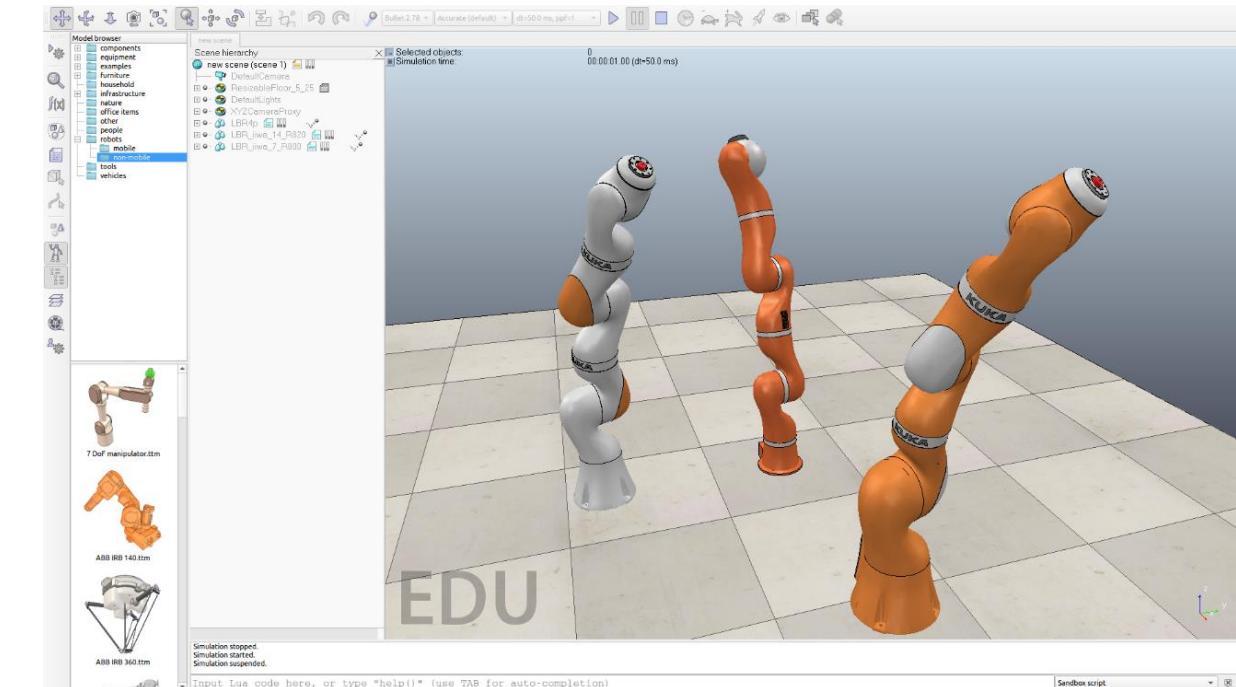
## Gazebo

- enginy: ODE, Bullet, Simbody, DART, OGRE
- programování: C++ pluginy, ROS



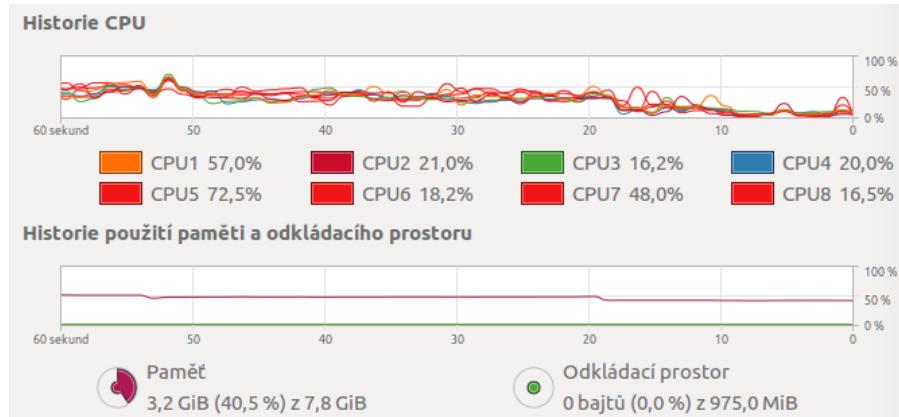
## V-REP CoppeliaSim

- enginy: ODE, Bullet, Vortex
- programování: integrované skripty, Remote API, ROS

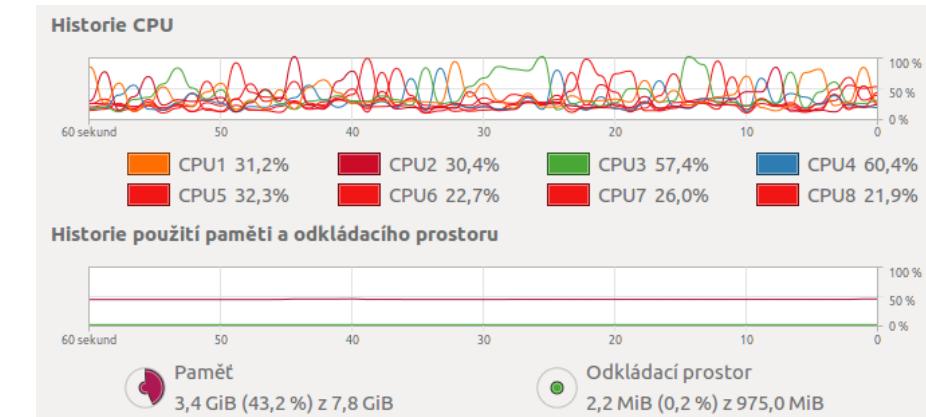


# ROS – Simulační prostředí

## Gazebo



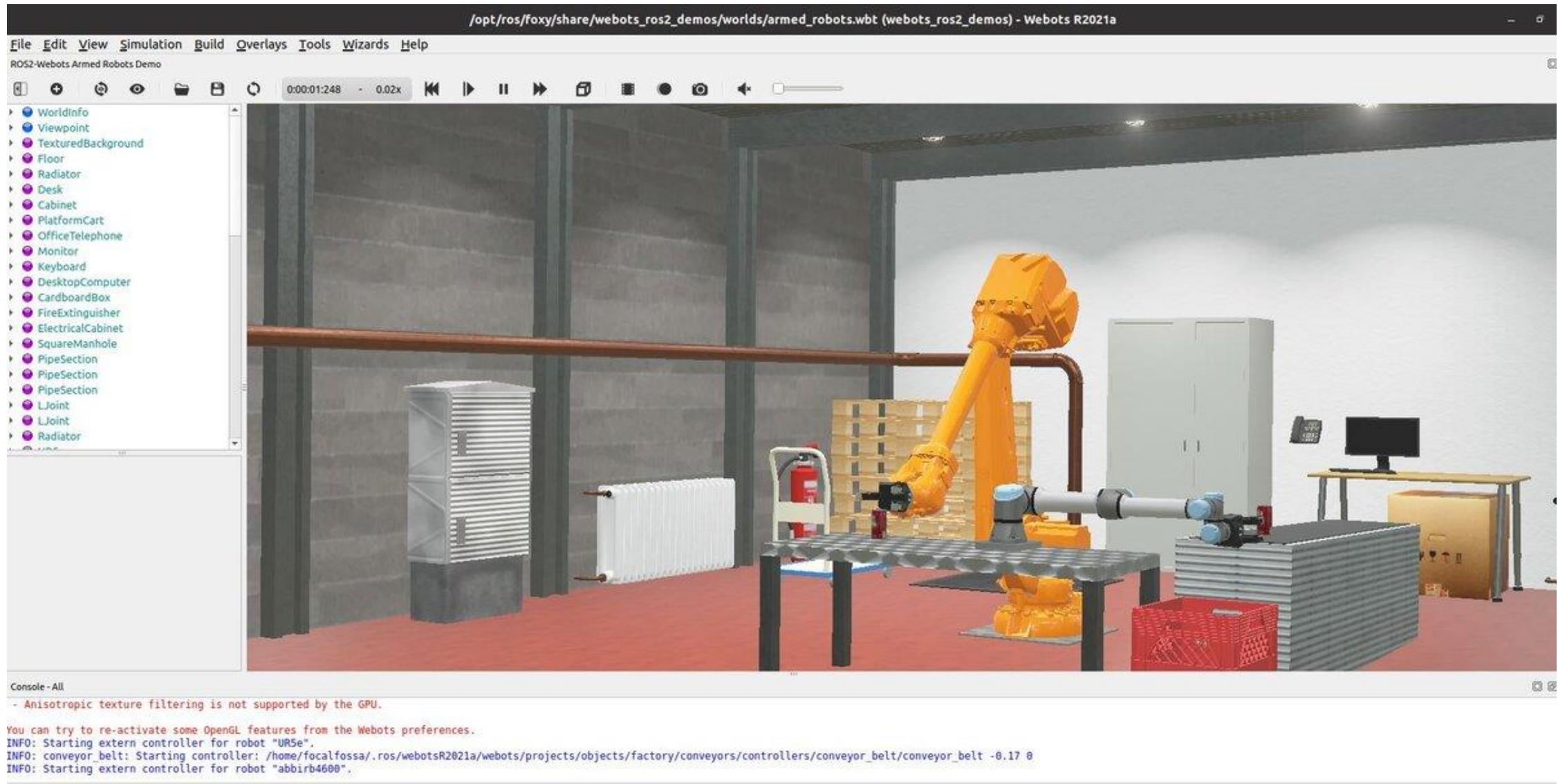
## V-REP CoppeliaSim



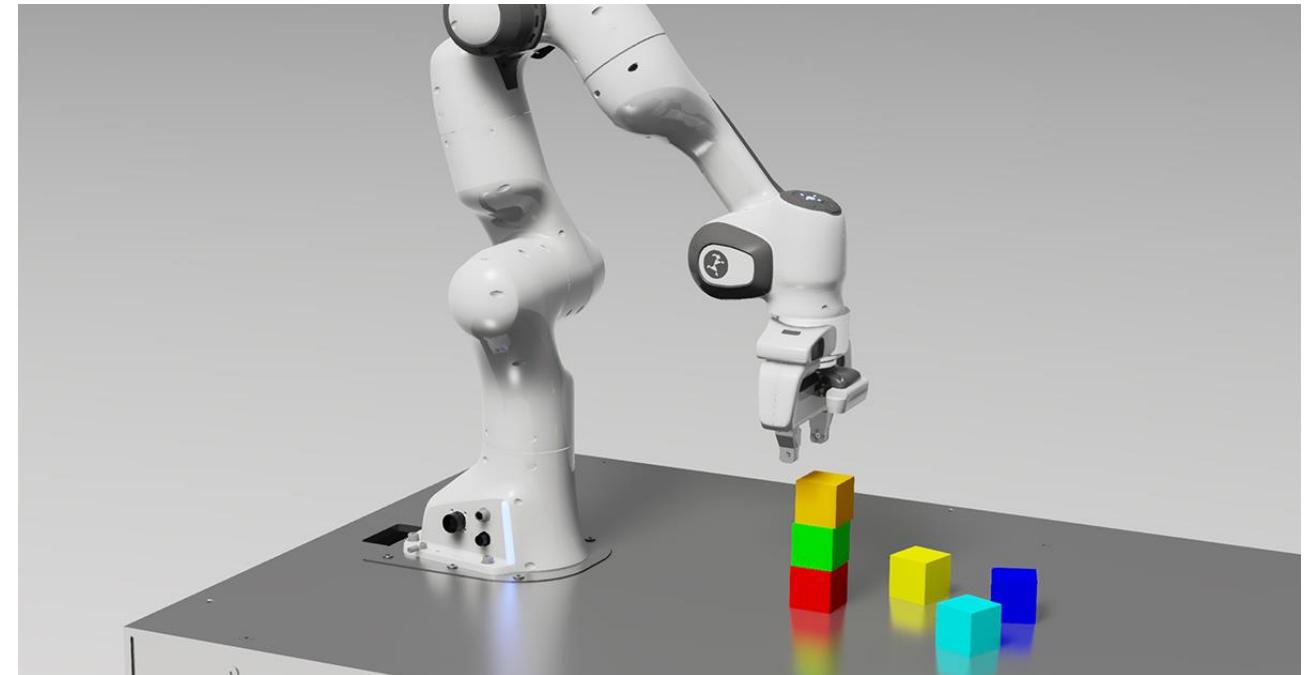
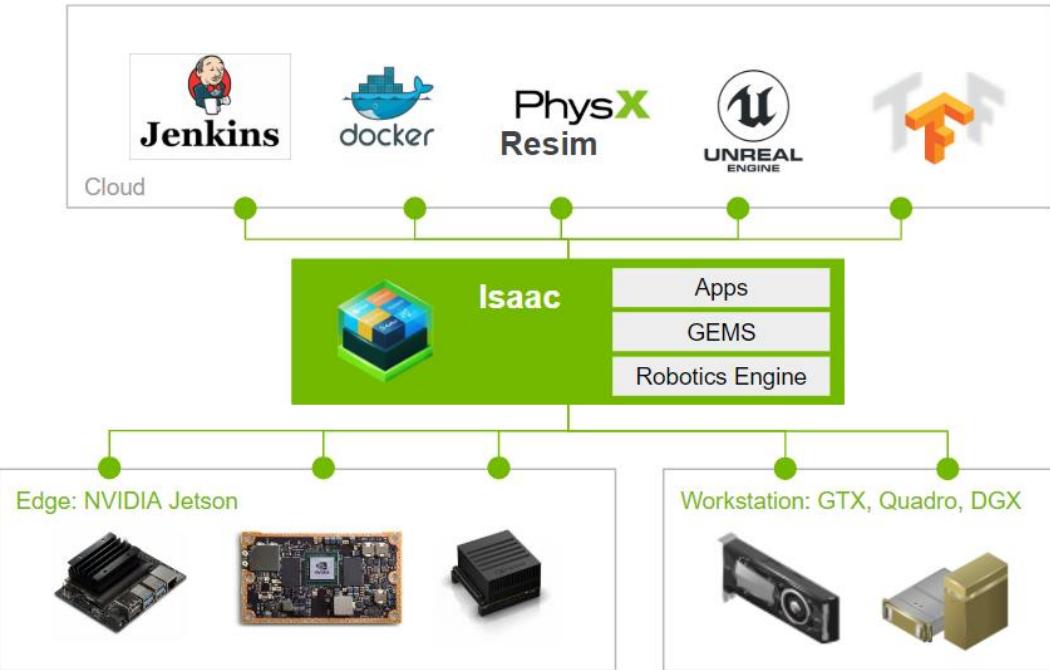
```
+-----+-----+
| NVIDIA-SMI 384.130        Driver Version: 384.130 |
| GPU  Name Persistence-M| Bus-Id Disp.A  Volatile Uncorr. ECC |
| Fan  Temp Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util Compute M. |
|=====+=====+=====+=====+=====+=====+=====+=====+=====
|   0  GeForce GTX 105... Off | 00000000:09:00.0 On |          N/A |
| 45% 31C    P0  ERR! / 75W | 664MiB / 4038MiB | 44% Default |
+-----+-----+
```

```
+-----+-----+
| NVIDIA-SMI 384.130        Driver Version: 384.130 |
| GPU  Name Persistence-M| Bus-Id Disp.A  Volatile Uncorr. ECC |
| Fan  Temp Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util Compute M. |
|=====+=====+=====+=====+=====+=====+=====+=====+=====
|   0  GeForce GTX 105... Off | 00000000:09:00.0 On |          N/A |
| 45% 37C    P0  ERR! / 75W | 380MiB / 4038MiB | 54% Default |
+-----+-----+
```

# ROS – Simulační prostředí

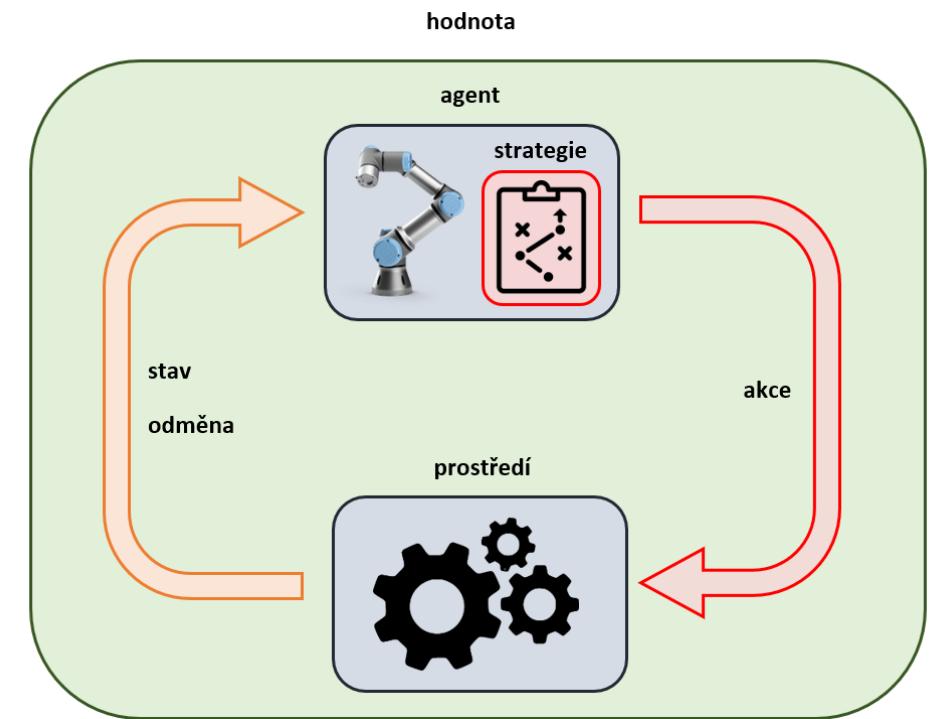


# ROS – Simulační prostředí



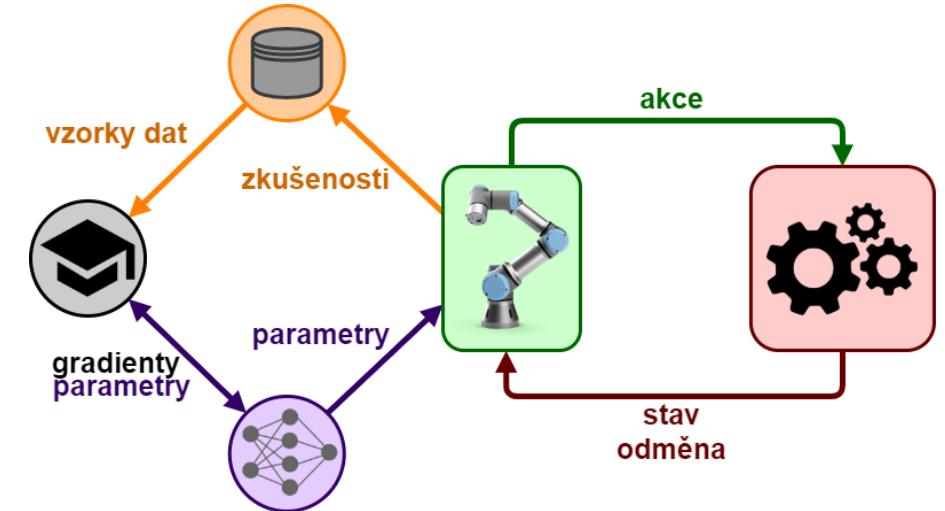
# Zpětnovazební učení

- agent: model, provádí akce
- stav
- prostředí: charakter scénáře
- strategie: jádro agenta
- odměna je posílána prostředím agentovi
- hodnota: určuje dlouhodobé cíle



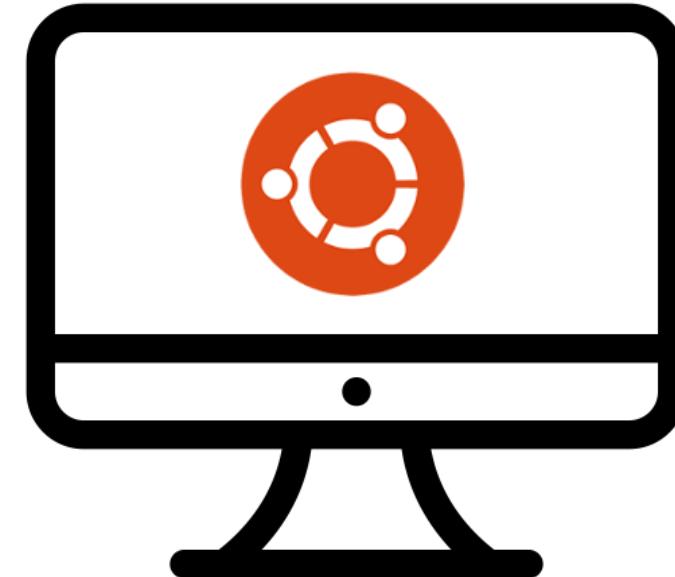
# Hluboké zpětnovazební učení

- agent: model, provádí akce
- stav
- prostředí: charakter scénáře
- strategie: jádro agenta
- odměna je posílána prostředím agentovi
- hodnota: určuje dlouhodobé cíle
- hluboká neuronová síť



# Praktická aplikace

# Instalace a konfigurace ROS

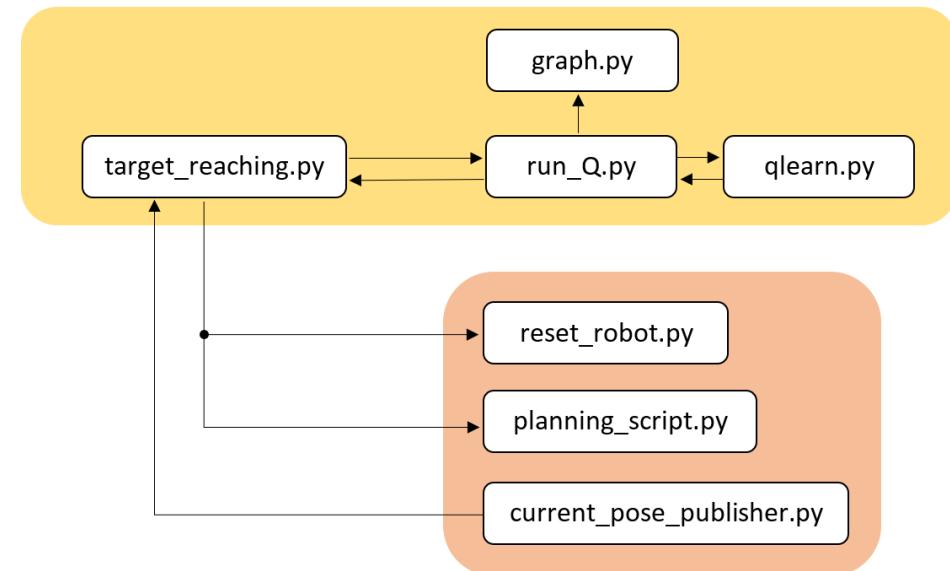


> **MoveIt**



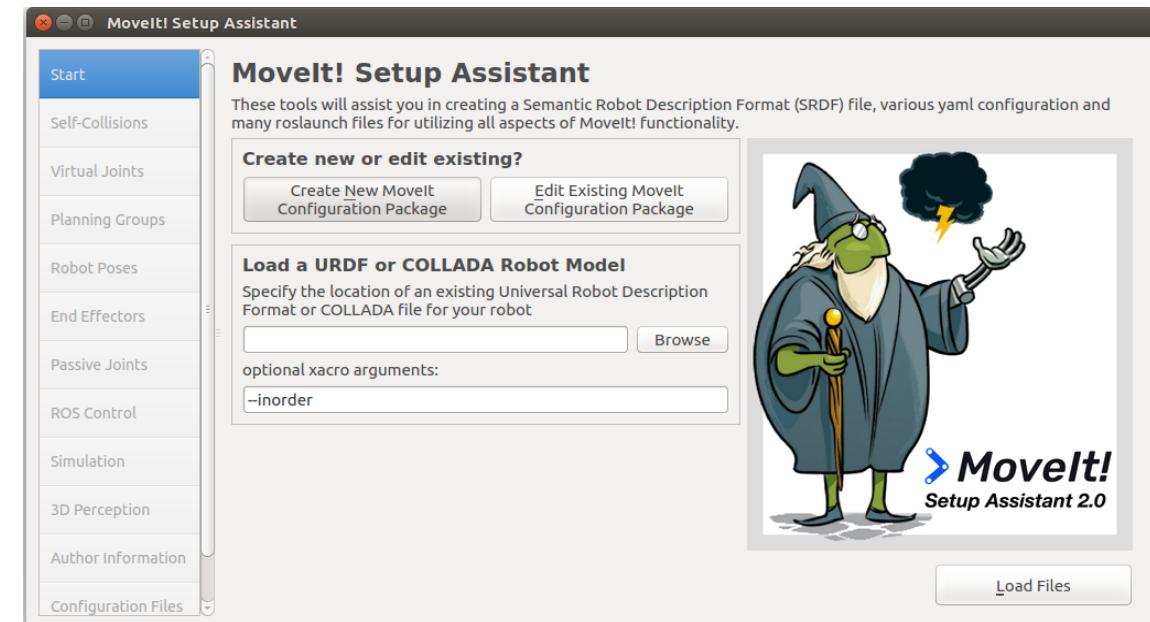
GAZEBO

1. návrh řešení
2. vytvoření balíku ROS
3. adaptace balíku k řešenému problému
4. aktualizace a instalace knihoven
5. implementace návrhu



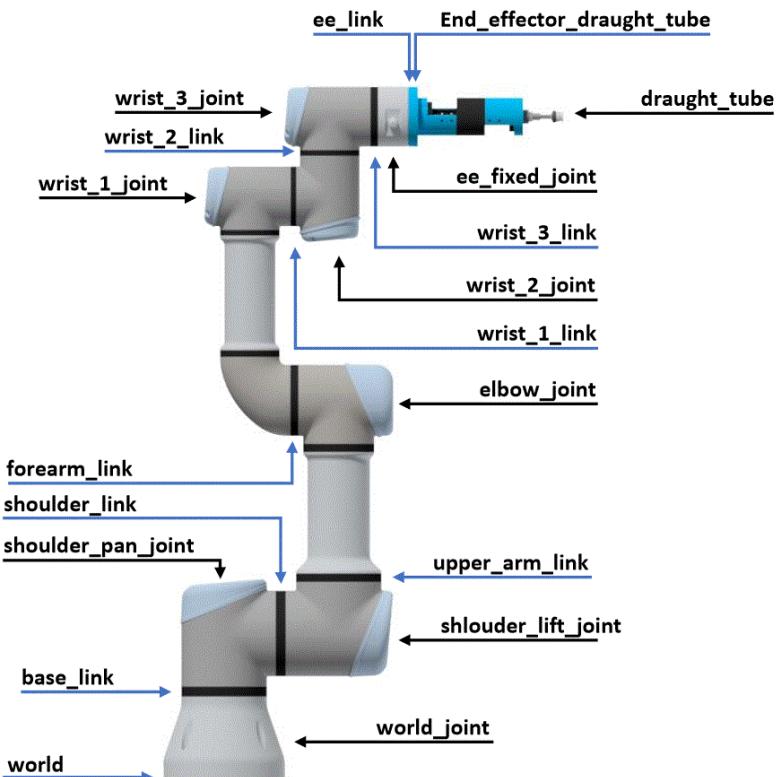
# Aplikace zpětnovazebních algoritmů

1. návrh řešení
2. vytvoření balíku ROS
3. adaptace balíku k řešenému problému
4. aktualizace a instalace knihoven
5. implementace návrhu



# Aplikace zpětnovazebních algoritmů

1. návrh řešení
2. vytvoření balíku ROS
3. adaptace balíku k řešenému problému
4. aktualizace a instalace knihoven
5. implementace návrhu

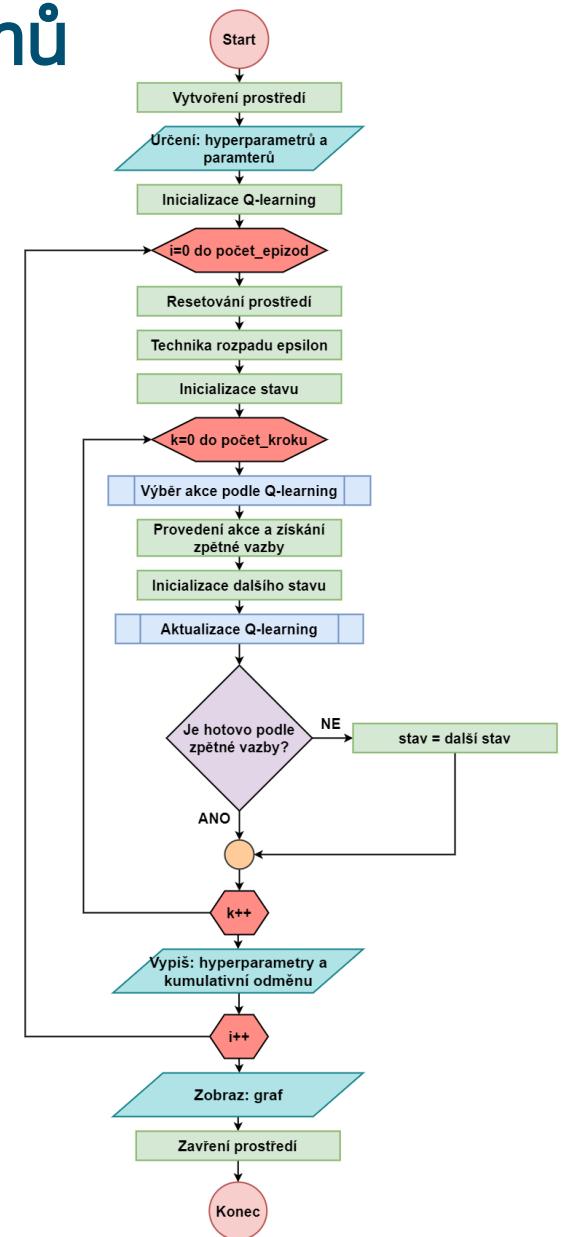
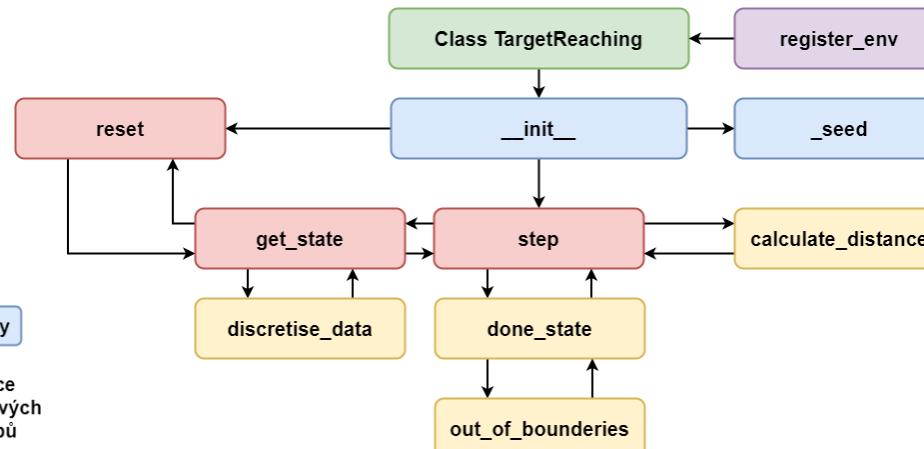
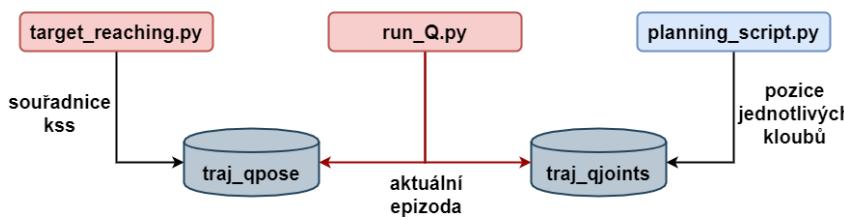


1. návrh řešení
2. vytvoření balíku ROS
3. adaptace balíku k řešenému problému
4. aktualizace a instalace knihoven
5. implementace návrhu

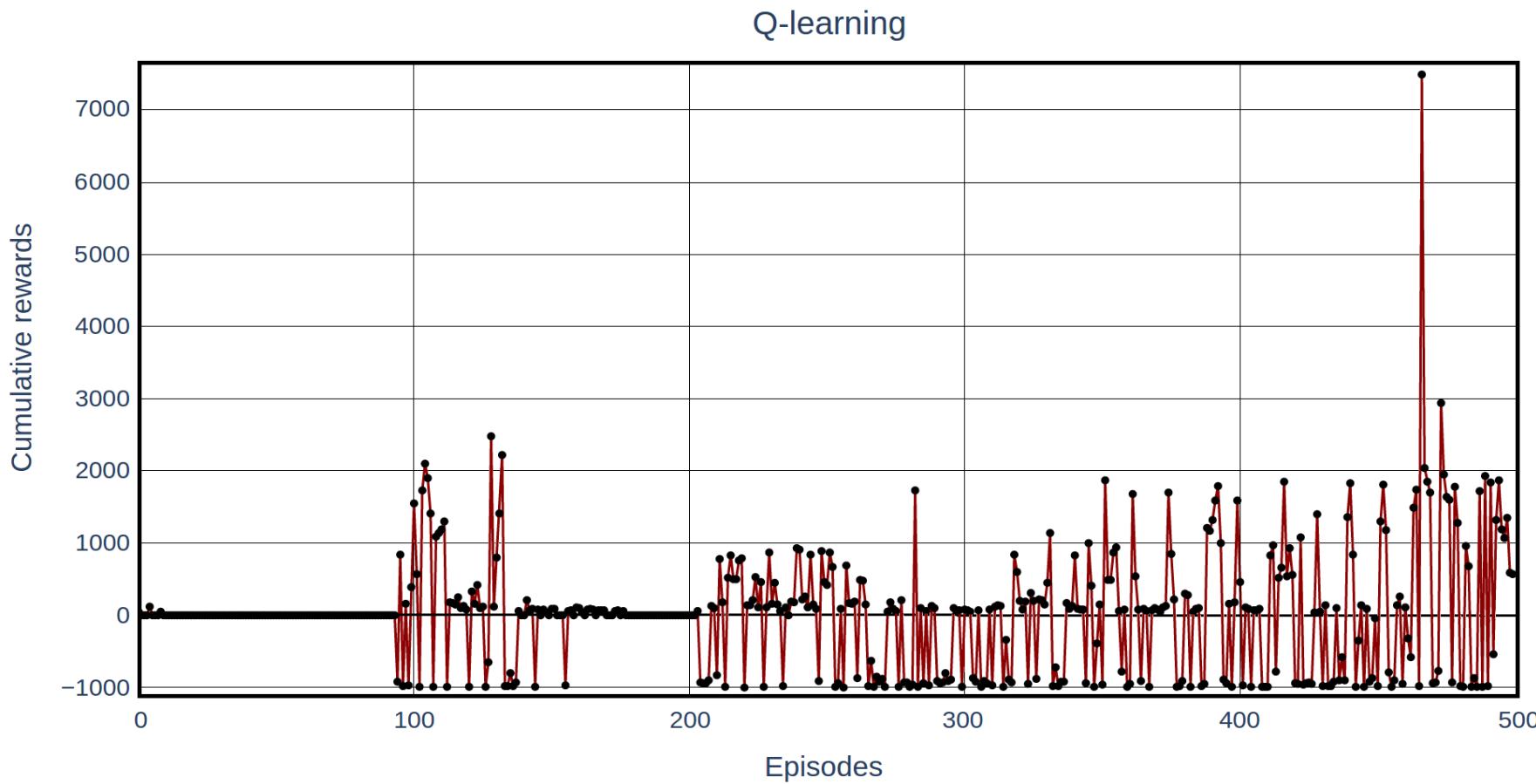


# Aplikace zpětnovazebních algoritmů

1. návrh řešení
2. vytvoření balíku ROS
3. adaptace balíku k řešenému problému
4. aktualizace a instalace knihoven
5. implementace návrhu

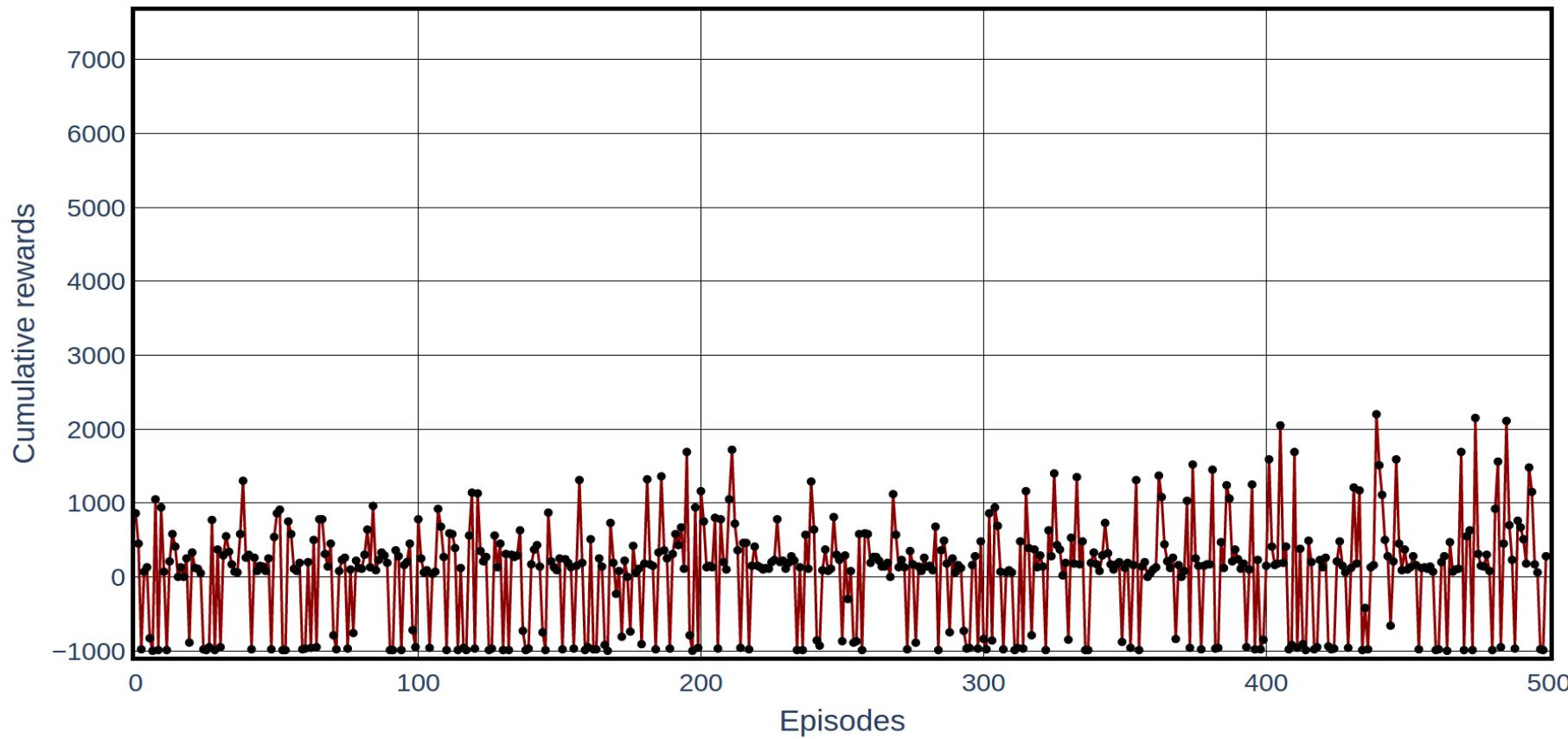


## 6. testování

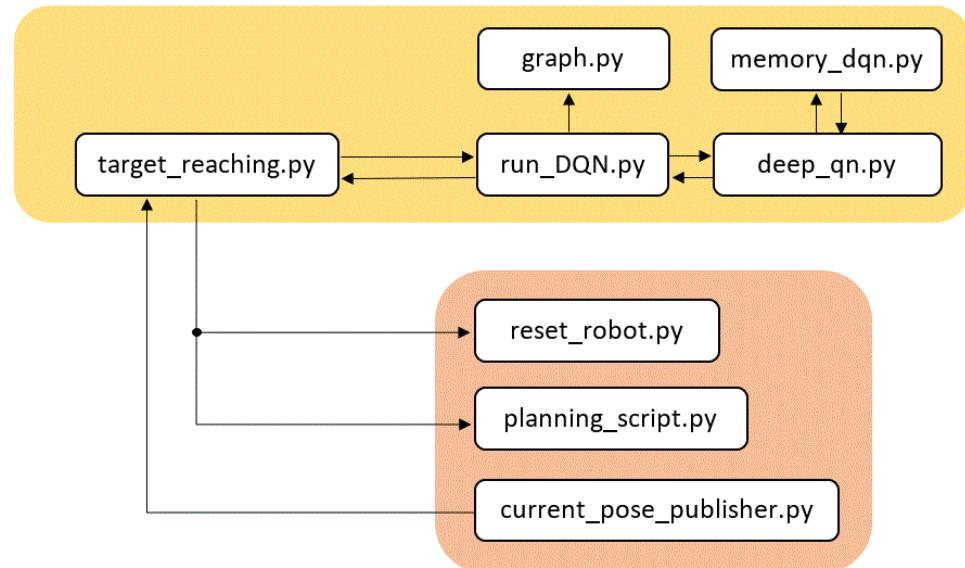


## 6. testování

SARSA



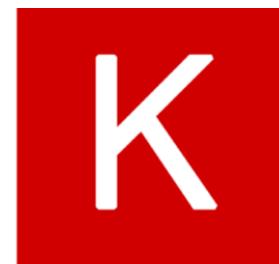
1. návrh řešení
2. využití již vytvořeného balíku ROS
3. aktualizace a instalace knihoven
4. implementace návrhu



1. návrh řešení
2. využití již vytvořeného balíku ROS
3. aktualizace a instalace knihoven
4. implementace návrhu

# Aplikace algoritmů hlubokého zpětnovazebního učení

1. návrh řešení
2. využití již vytvořeného balíku ROS
3. aktualizace a instalace knihoven
4. implementace návrhu



Keras

&



TensorFlow



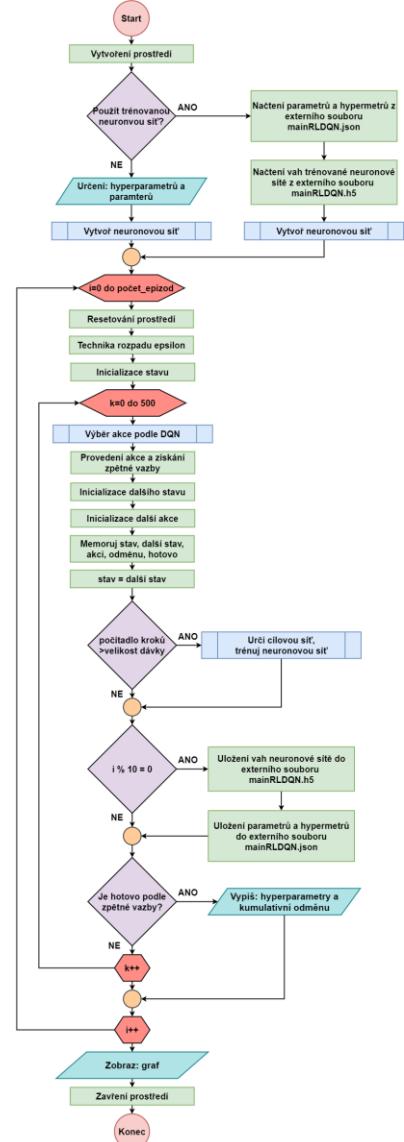
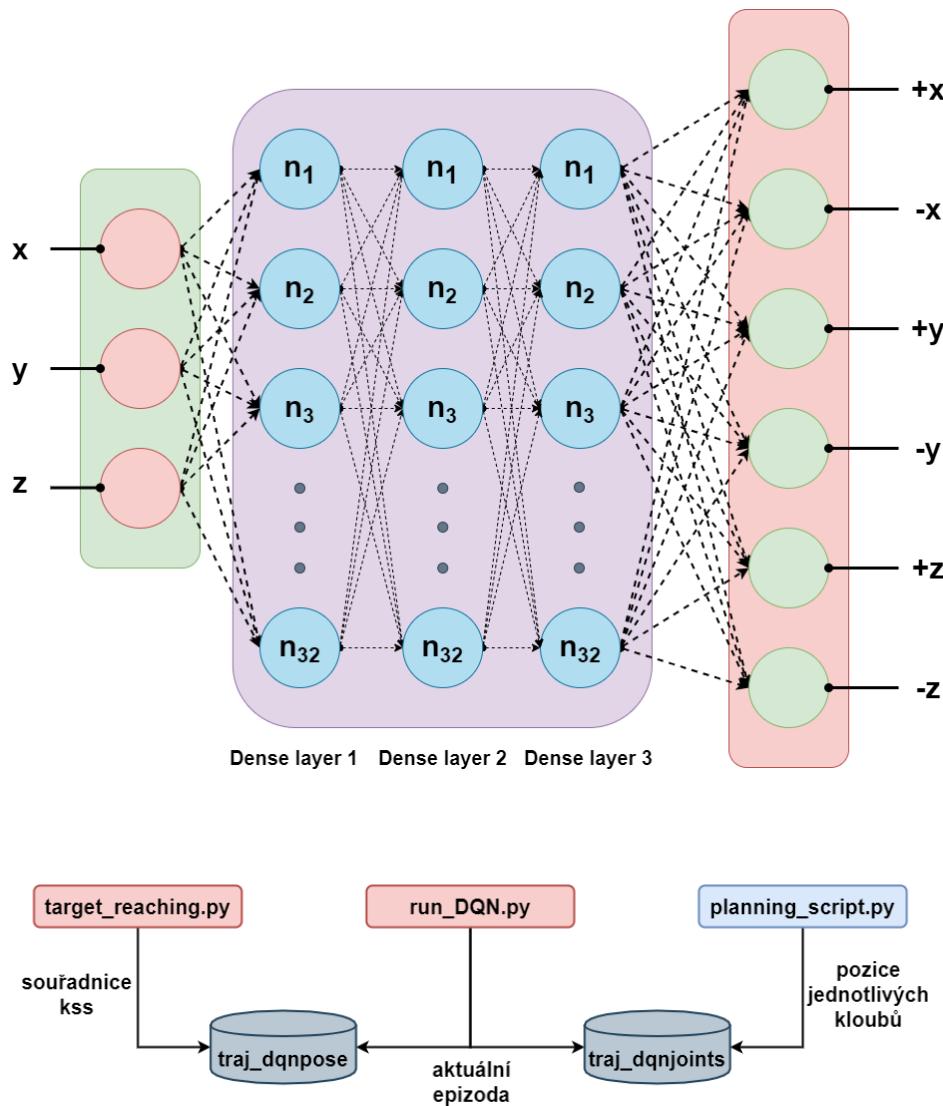
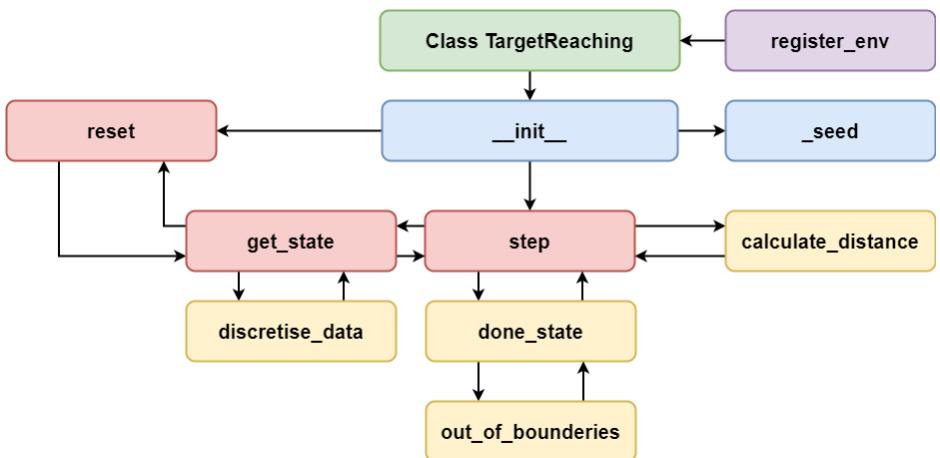
OpenAI



plotly

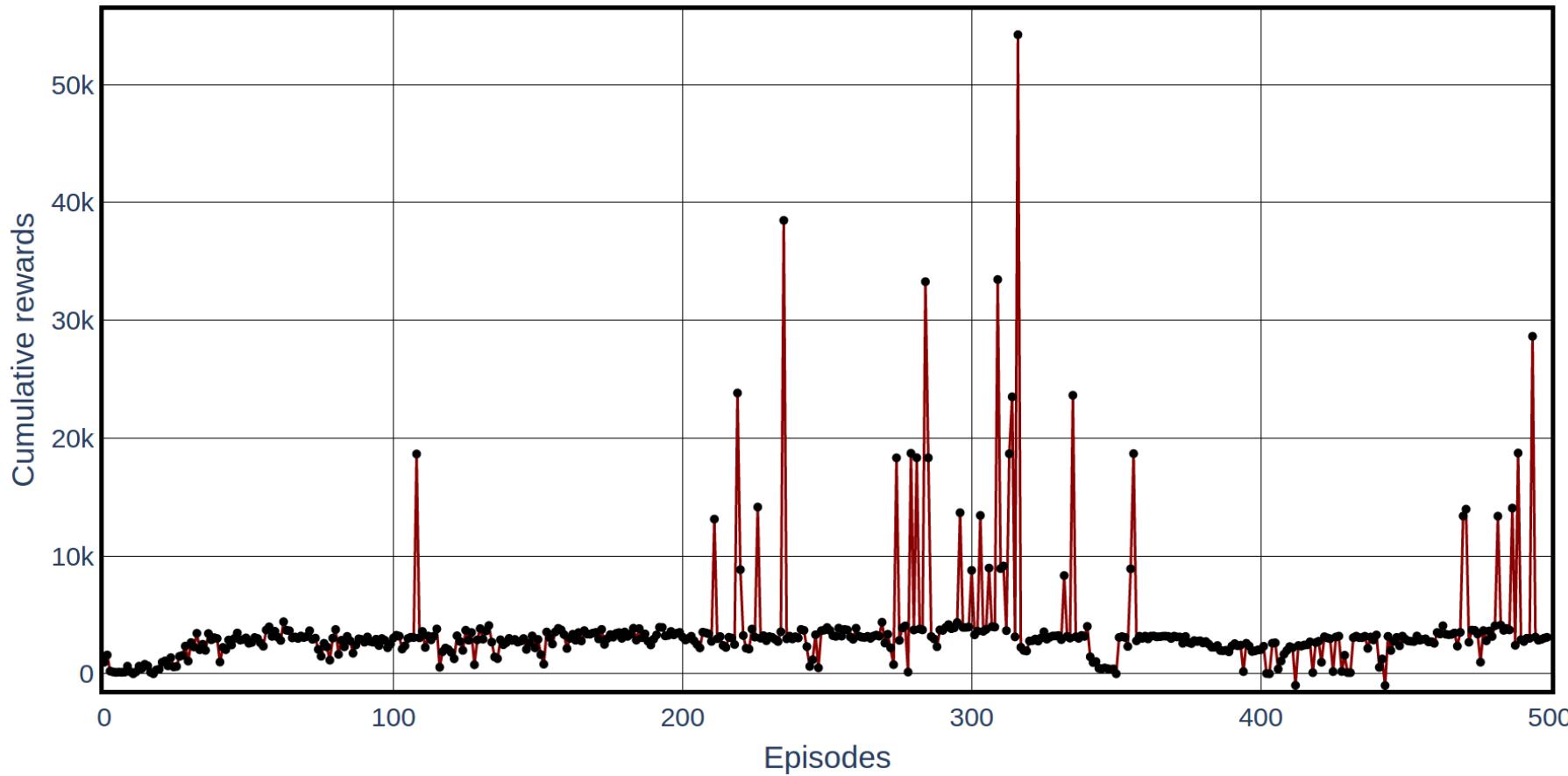
# Aplikace algoritmů hlubokého zpětnovazebního učení

1. návrh řešení
2. využití již vytvořeného balíku ROS
3. aktualizace a instalace knihoven
4. implementace návrhu



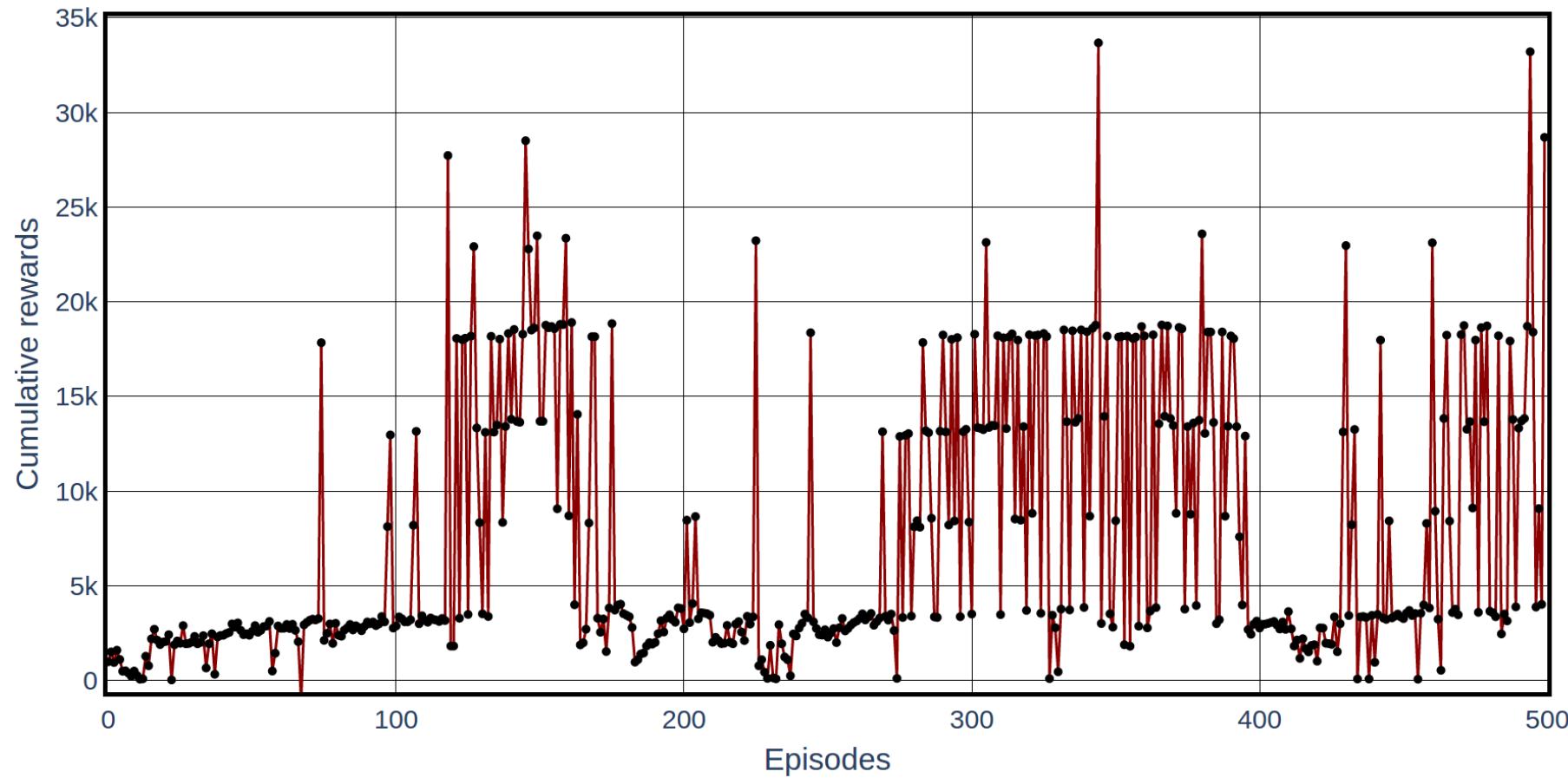
## 5. testování

DQN

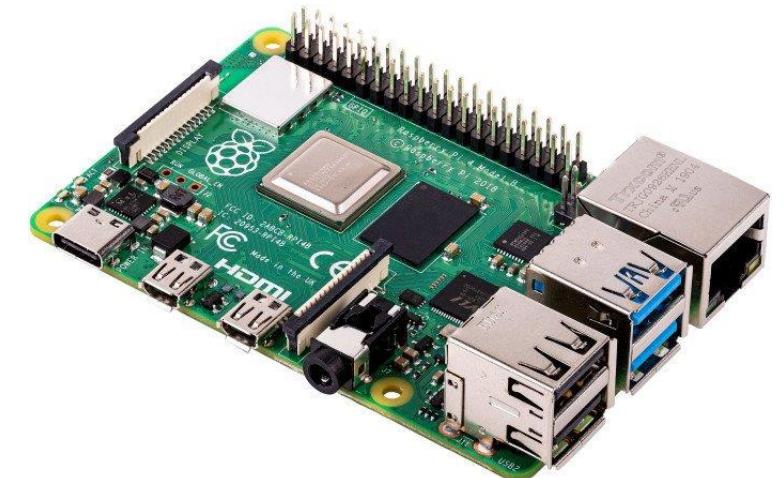


## 5. testování

DSARSA

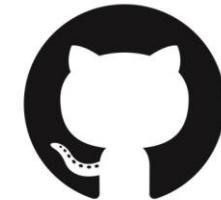


1. určení a připravení řídící jednotky
2. vytvoření catkin pracovního prostoru
3. implikace požadovaných balíků
4. simulace pohybu (framework MoveIt)
5. propojení reálného robota s ROS
6. ovládání kolaborativního robota pomocí Rviz  
ovládání kobota pomocí řídícího programu



1. určení a připravení řídící jednotky
2. vytvoření catkin pracovního prostoru
3. implikace požadovaných balíků
4. simulace pohybu (framework MoveIt)
5. propojení reálného robota s ROS
6. ovládání kolaborativního robota pomocí Rviz  
ovládání kobota pomocí řídícího programu

1. určení a připravení řídící jednotky
2. vytvoření catkin pracovního prostoru
3. implikace požadovaných balíků
4. simulace pohybu (framework MoveIt)
5. propojení reálného robota s ROS
6. ovládání kolaborativního robota pomocí Rviz  
ovládání kobota pomocí řídícího programu



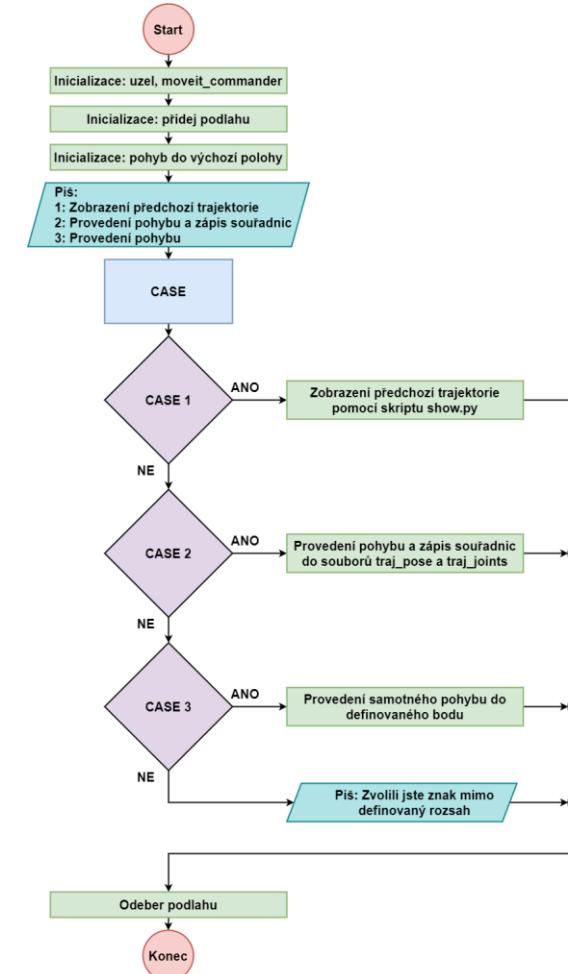
## GitHub

[Steigner / ROS\\_UR3\\_control](#)

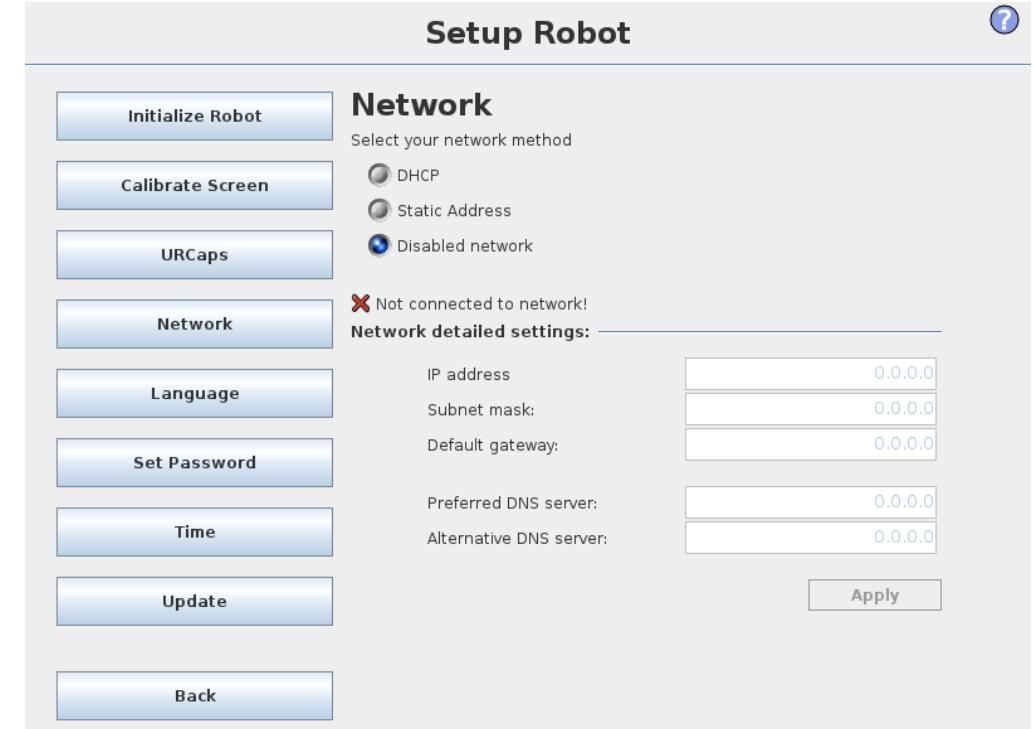
[ros-industrial / ur\\_modern\\_driver](#)

[ros-industrial / universal\\_robot](#)

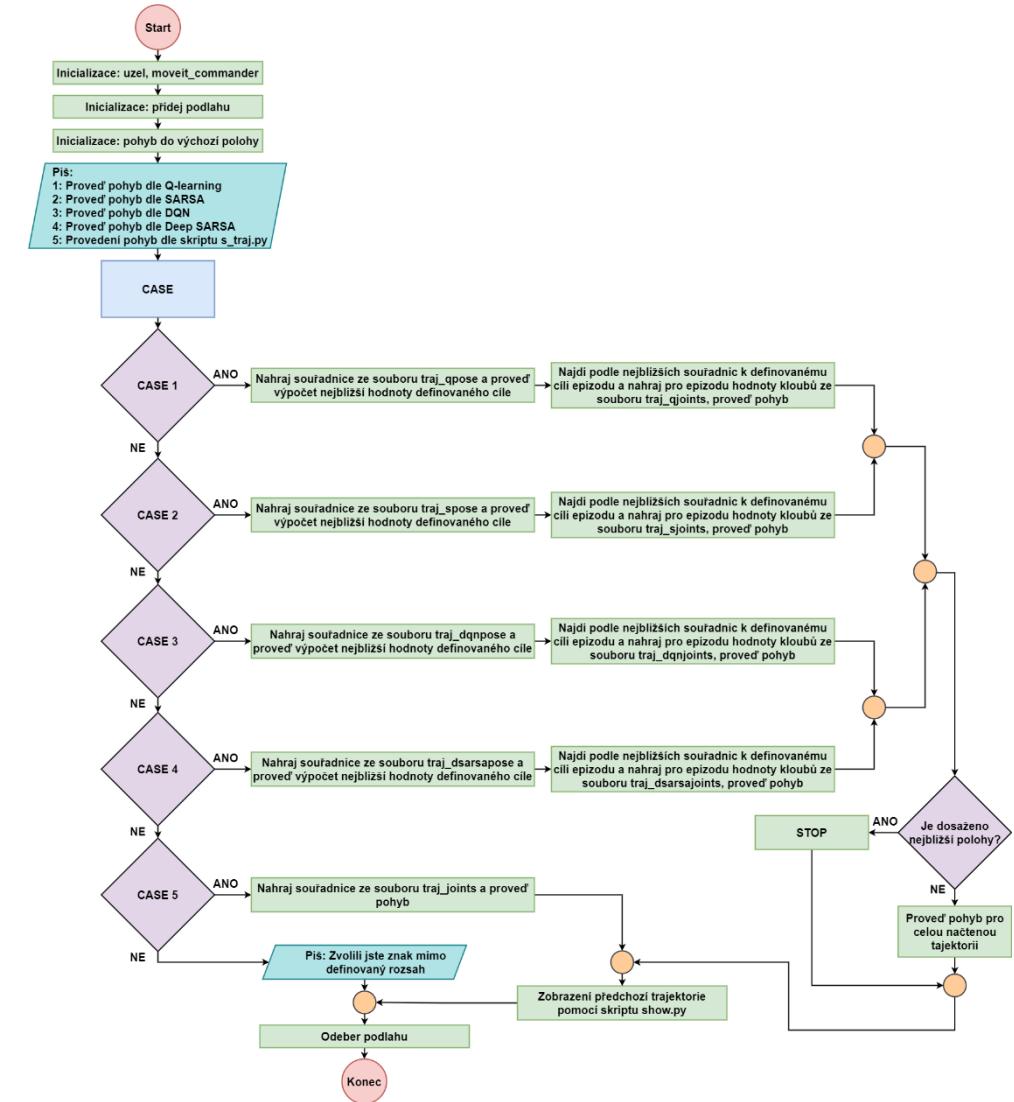
1. určení a připravení řídící jednotky
2. vytvoření catkin pracovního prostoru
3. implikace požadovaných balíků
4. simulace pohybu (framework MoveIT)
5. propojení reálného robota s ROS
6. ovládání kolaborativního robota pomocí Rviz  
ovládání kobota pomocí řídícího programu



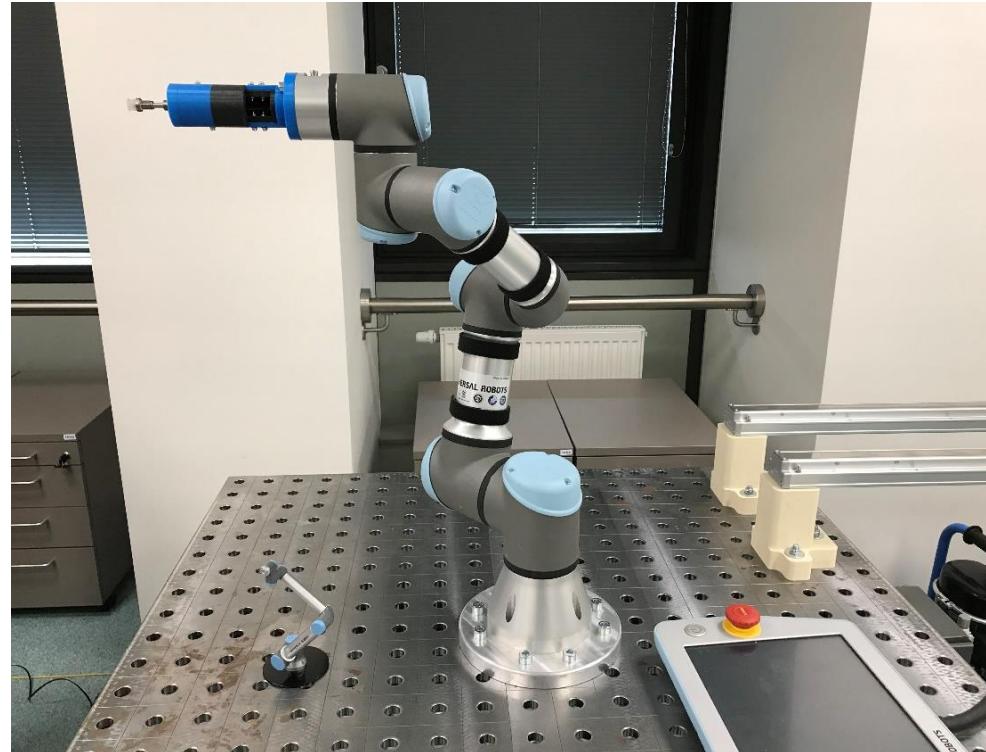
1. určení a připravení řídící jednotky
2. vytvoření catkin pracovního prostoru
3. implikace požadovaných balíků
4. simulace pohybu (framework MoveIt)
5. propojení reálného robota s ROS
6. ovládání kolaborativního robota pomocí Rviz  
ovládání kobota pomocí řídícího programu



1. určení a připravení řídící jednotky
2. vytvoření catkin pracovního prostoru
3. implikace požadovaných balíků
4. simulace pohybu (framework MoveIT)
5. propojení reálného robota s ROS
6. ovládání kolaborativního robota pomocí Rviz  
ovládání kobota pomocí řídícího programu

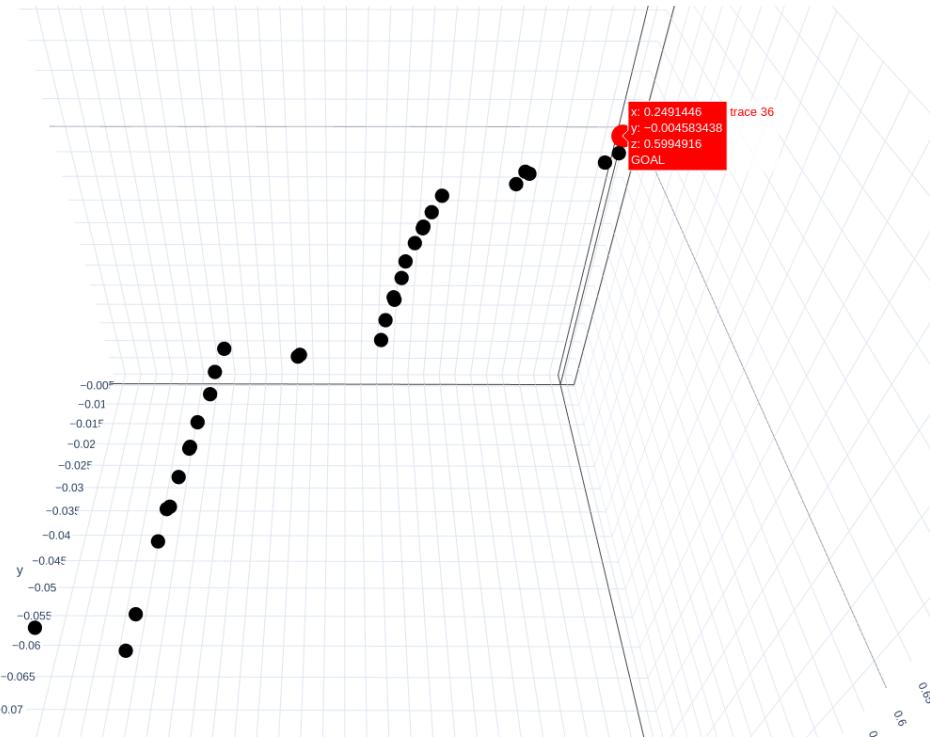


# Řízení kolaborativního robota UR3

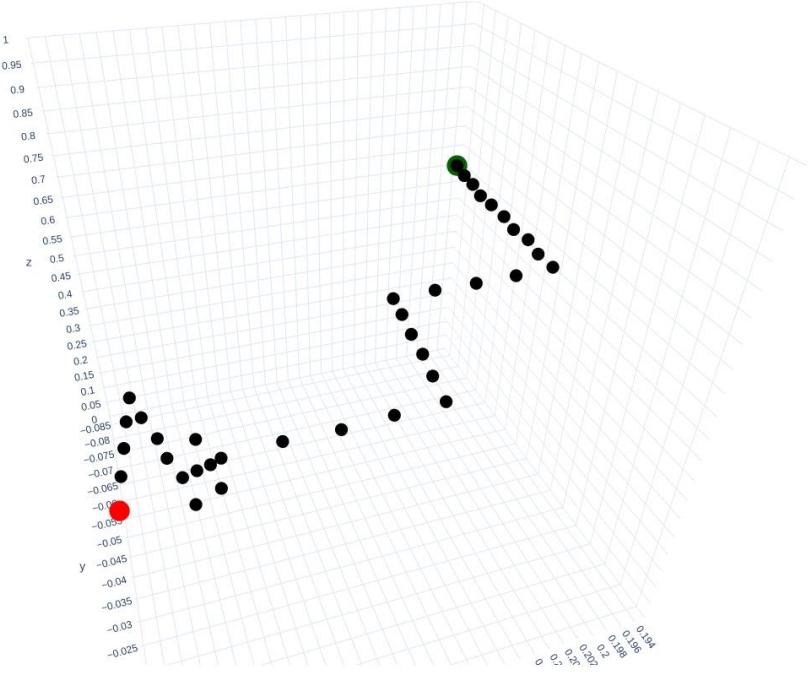


# Testování řídícího programu v laboratoři

Trajectory: Q-learning

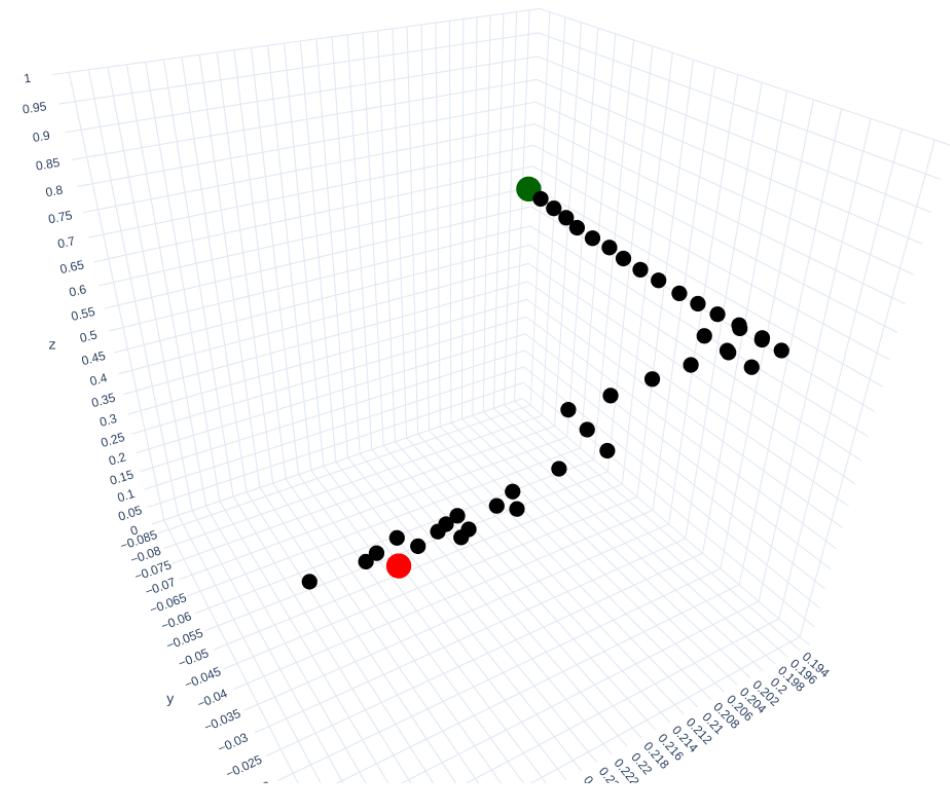


Trajectory: DQN

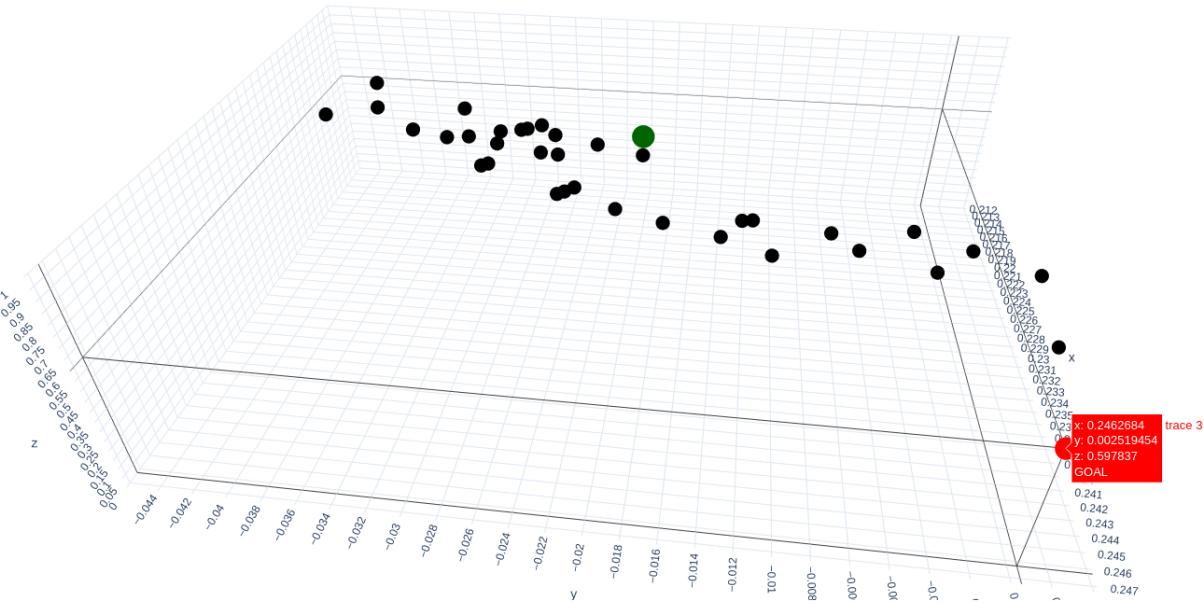


# Testování řídícího programu v laboratoři

Trajectory: DEEP SARSA

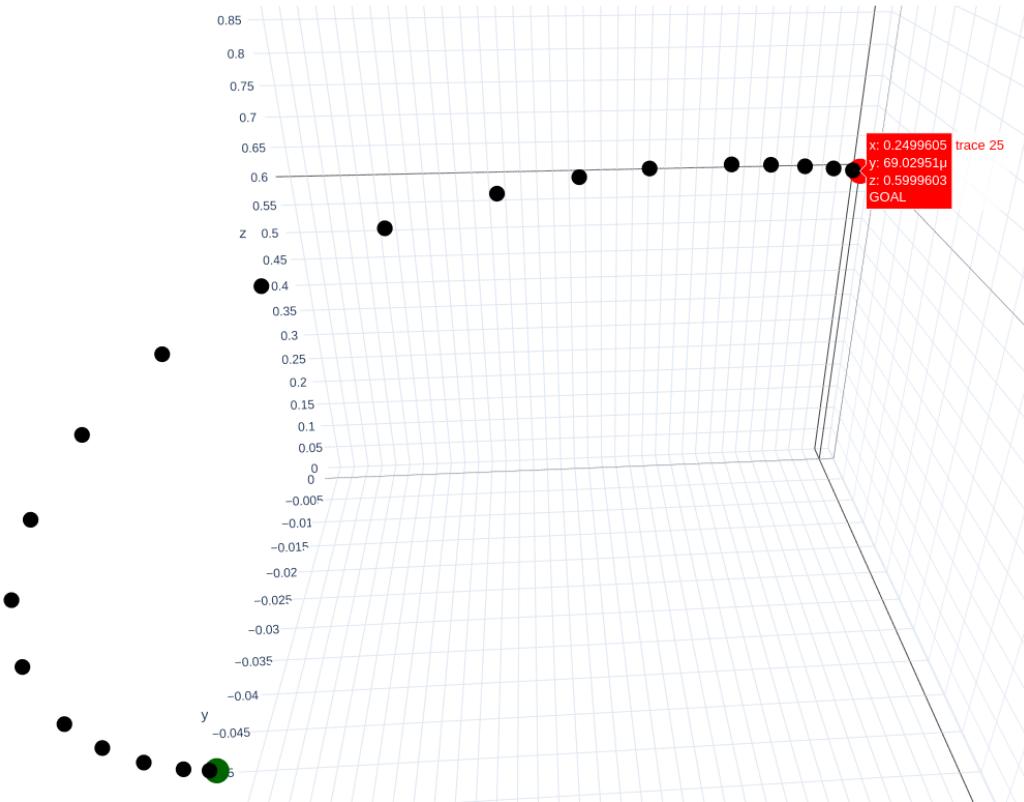


Trajectory: SARSA



# Testování řídícího programu v laboratoři

Trajectory: MoveIT planner



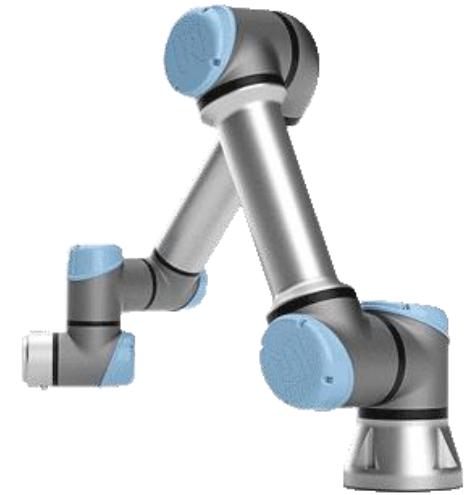
# Závěr

## Rozšíření o další algoritmy

- DDQN
- A3C
- DDPG
- TD3
- SAC

## Rozšíření o řešení dosahování bodu v plném rozsahu pohybu kobota

- kvaterniony ( $x, y, z, qx, qy, qz, qw$ )
- trénovaní sítě -> dosahování libovolného bodu v rozsahu pohybu



# Děkuji za pozornost.

Autor : Martin Juříček

Vedoucí : Ing. Roman Parák

ai

ÚSTAV AUTOMATIZACE  
A INFORMATIKY



VYSOKÉ UČENÍ FAKULTA  
TECHNICKÉ STROJNÍHO  
V BRNĚ INŽENÝRSTVÍ

[uai.fme.vutbr.cz](http://uai.fme.vutbr.cz)