



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Irányítástechnika és Informatika Tanszék

Robot operációs rendszerek és fejlesztői ökoszisztémák

Palettázás LBR iisy 11 robottal

Házi feladat

KÉSZÍTETTE

Simon Bertalan, Szalay Szabolcs

Tartalomjegyzék

1 Feladat leírása.....	3
2 Feladat megoldása.....	4
2.1 Előkészületek.....	4
2.2 Kezdeti lépések.....	4
2.3 Futtatása.....	4
2.4 Működése	5
3 Továbbfejlesztési lehetőségek.....	7

1 Feladat leírása

Vizsgálják meg a MoveIt2 több tervező módszerét is. Egy képzeletbeli palettázási folyamat esetére vizsgálják meg a ciklusidőket. Válasszák meg egy 4x4-es méretű paletta ideális helyét a robot körül úgy, hogy a ciklusidő a lehető legkisebb legyen. A robotnak ismert méretű, téglatest alakú dobozokat kell palettáznia.

2 Feladat megoldása

2.1 Előkészületek

Az előadáson elhangzottak alapján a kroschu által készített kuka_drivers repository volt a kiinduló alap. Mivel ez ros2-t használ méghozzá annak a Humble Hawksbill verzióját így mi is ebben dolgoztunk tovább, méghozzá egy-egy linux 22.04 LTS ubuntu verziójú virtuális gép segítségével. Ebben a repository-ban található iiqka_moveit_example példa package, ami hasonló robotkarral dolgozik, mint ami számunkra elő van írva.

2.2 Kezdeti lépések

Egy saját csomagot hoztunk létre a példa alapján, ami át lett nevezve iiqka_moveit_palletizing-ra ennek megfelelően a package.xml-t és a cmake fájlt is módosítani kellett a launch fájlban csak a robotmodellt kellett kicserélni az iisy11_rk1300-ra a feladatnak megfelelően. Maga a palettázást végző kód a példacsomag MoveitDepalletizingExample.cpp nevű példája alapján készült.

2.3 Futtatása

A megfelelő függőségek telepítése után, amik megtalálhatóak a projekt [github repojában](#) először buildelni kell a csomagot, amit az alábbi ábra mutat be.

```
ros2@ros2-vm:~/iiqka_moveit_palletizing$ colcon build
Starting >>> iiqka_moveit_palletizing
Finished <<< iiqka_moveit_palletizing [20.0s]

Summary: 1 package finished [21.2s]
```

1. Csomag build-elése

Ezután a launch fájl elindítása: „ros2 launch iiqka_moveit_palletizing moveit_palletizing.launch.py” paranccsal történik. Majd egy másik terminálban a robot_manager node-ot kell elindítani az alábbiak szerint.

```
ros2@ros2-vm:~/iiqka_moveit_palletizing$ ros2 lifecycle set robot_manager configure
Transitioning successful
ros2@ros2-vm:~/iiqka_moveit_palletizing$ ros2 lifecycle set robot_manager activate
Transitioning successful
```

2. Ros lifecycle aktiválása

Végül magát a palettázást végző node-ot indítjuk el a következőképp.

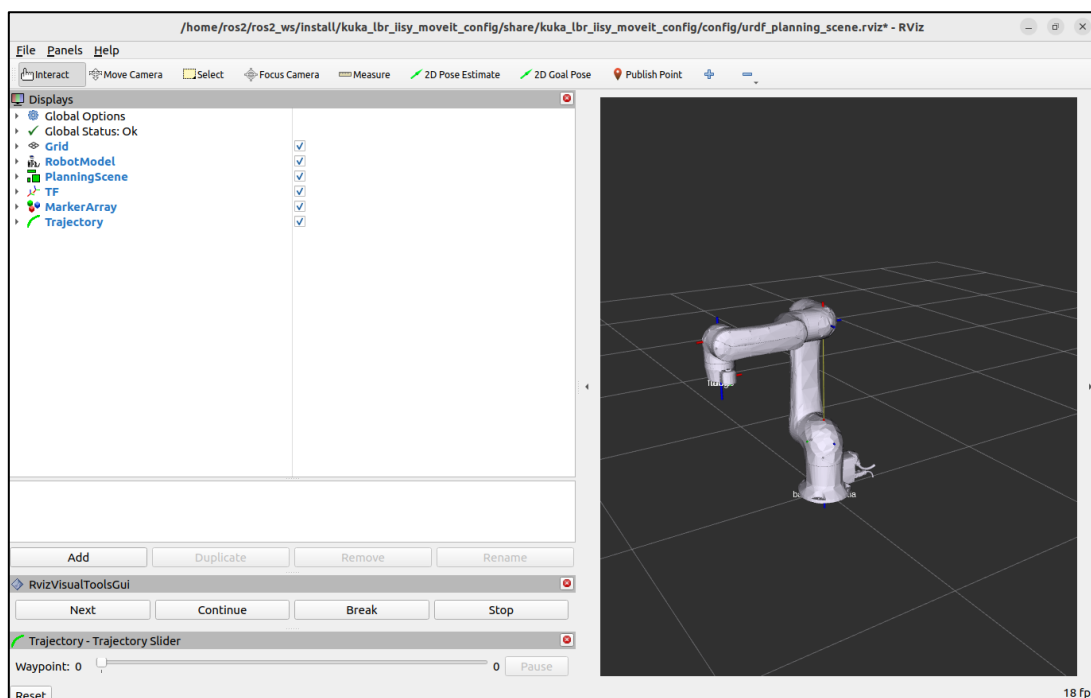
```

ros2@ros2-vm:~/iiqka_moveit_palletizing$ source install/setup.bash
ros2@ros2-vm:~/iiqka_moveit_palletizing$ ros2 run iiqka_moveit_palletizing iiqka_moveit_palletizing

```

3. Példakód futtatása

Ezután már az RViz grafikus felületén követhetjük az eseményeket és léptethetjük a programot, aminek a túldoldalon található az ábrája.



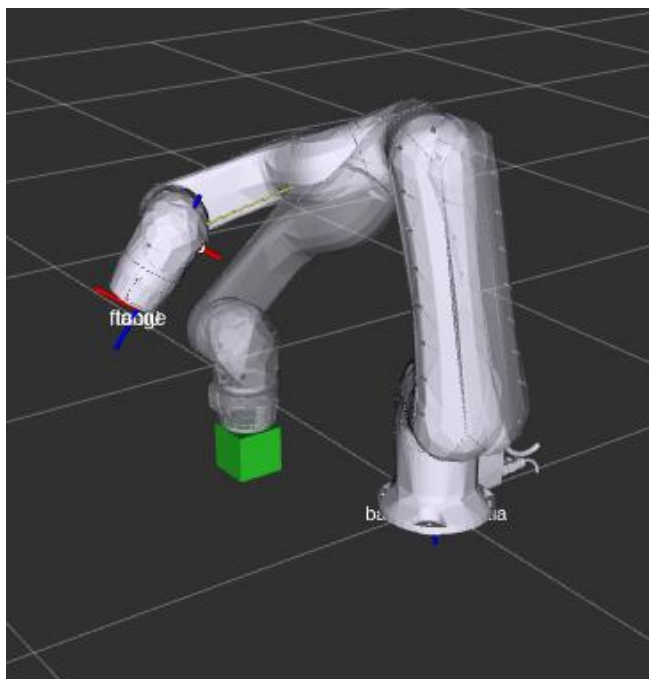
4. RViz gui felület a konfigurált robottal

2.4 Működése

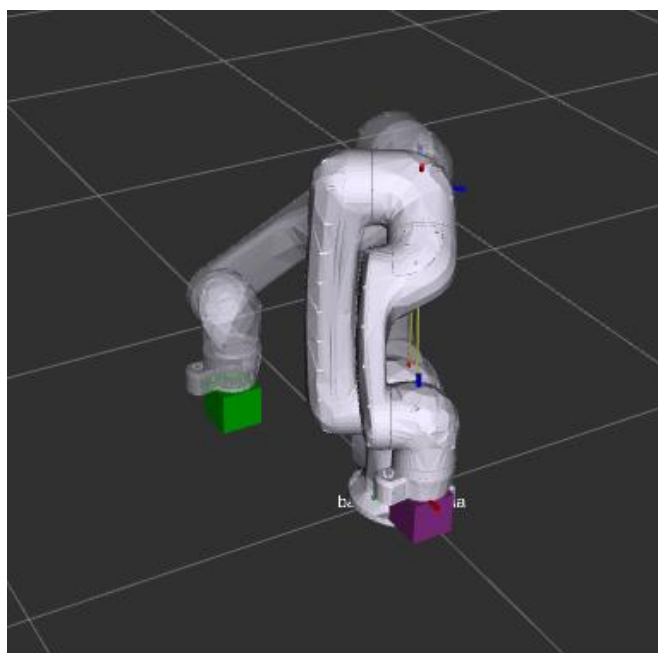
A pilz_industrial_motion_planner-t használjuk és egy egyszerű point to point pályatervezést hajtunk végre a palettázásra, ami során egy fix helyen érkező dobozokhoz megy oda a robotkar (robotkar_1) megfogja azokat majd a 4x4-es raklap éppen megfelelő helyére teszi, elengedi majd egy újabb dobozért megy amíg nem telik a raklap.

A palettázás során a lerakott téglalapokat eltávolítjuk a planner-ből, hogy így az útvonaltervezésnél ne kelljen foglalkozni plusz ütközések kezelésével. Valódi robot esetén külön pozíciókat kellene betanítani a lerakásokhoz és a lerakás előtti pozíciókhoz.

A moveit több útvonaltervező algoritmusát is kipróbáltuk a pilz_industrial_motion_planner és az ompl könyvtárakból, de csak a PTP tervezés működött valamennyi pozíció esetén stabilan, így ezt választottuk ki.



5. Útvonaltervezés, és végrehajtás a lerakott csomaghoz



6. Lerakott csomag felemelése, útvonaltervezés és végrehajtás a képzeletbeli raklaphoz

A több szimulációt futtatva azt állapítottuk meg, hogy a ciklusidő egyrészt úgy a legkisebb, ha nincs nagy távolság a csomagok érkezési pontja, és a célpaletta között. másrészt pedig azt, olyan pályákkal a legeffektívebb a mozgás, melyeknél a csuklópálya értékei keveset változnak (például nem 5° -ról megy $-5^\circ = 355^\circ$ -ra egy csukló, hanem 15° -ról 5° -ra), így a robot és paletta elhelyezésénél, tervezésnél ennek a szempontnak a figyelembevétele is kritikus fontosságú.

3 Továbbfejlesztési lehetőségek

Ebben a feladatban sok szabad paraméter van dobozok mérete a paletta mérete, helye és a pályatervező algoritmus a ciklusidőt ezek mind befolyásolják, általában a doboz mérete és a felvételi pontja adott és a paletta maga is, így a következő lépés az lenne ezeket lerögzítjük egy olyan értékre, amelyiknél stabilan működnek a planner algoritmusok, és különböző pályatervezőkkel és paletta elhelyezkedésekkel optimalizáljuk a ciklusidőt.