

PROJEKTNI ZADATAK
za školsku 2025./2026. godinu iz predmeta

INDUSTRIJSKA ROBOTIKA

verzija 0.9 (preliminarna)
30.6.2026.

Opšti deo

Zadatak opisan u ovom dokumentu odnosi se na sve studente koji su slušali predmet u školskoj 2025./2026. godini. Na zadatku se radi u timovima sa dva člana. U posebnom slučaju, uz pristanak profesora, tim može imati i trećeg člana ili se može raditi i samostalno. Početak rada na projektu, kao i sastav tima, potrebno je prijaviti predmetnom nastavniku pre početka rada na projektu. Zadatak je u osnovi isti za sve studente.

Vreme rada na projektu nije posebno ograničeno osim što prijavljeni projekat treba završiti do objavljivanja sledećeg zadatka naredne školske godine. Ukoliko se ne završi do tada, podrazumeva se da važi novi zadatak čak i ako je student predmet slušao u ranijoj školskoj godini.

Rešenje projekta podrazumeva kompletno izrađen softver i prateće fajlove potrebne za demonstraciju u softverskom okruženju ROS2. Funkcionalnost rešenja obavezno je prikazati u simulatoru Gazebo. Rešenje i demonstracija treba da budu praćeni pisanom dokumentacijom koja razmatra i objašnjava sve aspekte rešenja. Rad na projektu završava se odbranom pred komisijom odobrenom od strane predmetnog nastavnika. U odbrani učestvuju svi članovi tima i od njih se očekuje da potvrde poznavanje svih detalja projekta. Nakon odbrane svi članovi tima dobijaju konačnu ocenu projekta koja ne mora biti ista za sve članove.

Postavka i zadatak

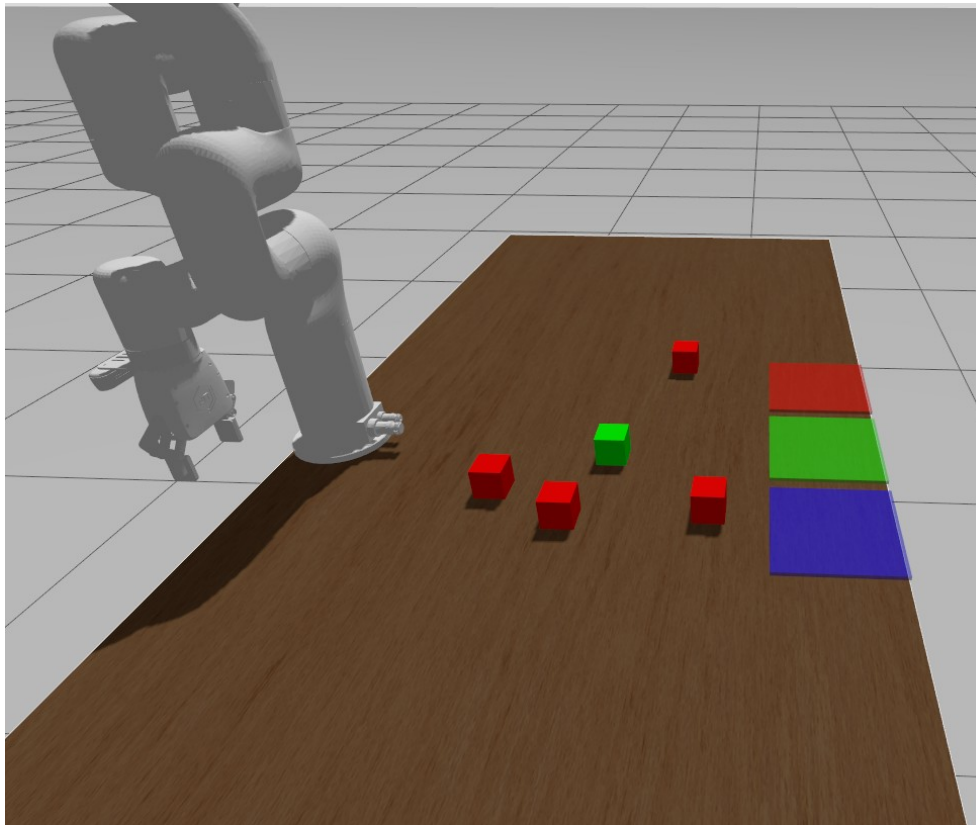
Dat je kolaborativni robot, **Ufactory xArm7**, sa dvoprstnim hvatačem i **Realsense D435i** kamerom u Gazebo simulacionom okruženju. Baza robota se nalazi na stolu. Pored robota na stolu se nalaze:

- kockice različitih boja (crvena, plava i zelena)
- tri površi različitih boja (crvena, plava i zelena)

Kockice imaju ivice dimenzija 4 cm, dok su površi dimenzija 15 x 15 cm.

Svako pokretanje simulacije nasumično postavlja boje i položaje kockica, dok su površi statičke, a samo im se boja nasumično određuje. Uvek su prisutne sve tri boje površi, dok za kockice to ne mora da važi.

Primer jedne postavke vidi se na sledećoj slici:



Potrebno je izvršiti sortiranje kockica tako da se svaka kockica nalazi na površi iste boje. Za upravljanje robotom na raspolaganju je **MoveIt** paket iz ROS2 frejmworka. Preporuka je koristiti **MoveItPy** za planiranje i izvršavanje kretanja.

Objašnjenja

Radi izbegavanja problema sa simulacijom fizike prilikom kontakta, za hvatanje kocke ne zatvarati dvoprsti hvatač, nego koristiti sledeći pristup.

Prilikom pokretanja `project_2026 → task.launch.py` pokrenuće se Gazebo simulacija, **ROS2** kontroleri, **moveit** kontroleri. Dodatno, pokreće se i `cube_tracker_node` koji pokreće servis: `/get_closest_cube` koji prima `target_pose` pri pozivu, a vraća `cube_id` string i `bool found` ukoliko je kocka pronađena. `target_pose` predstavlja koordinatu oko koje proveravamo da li je kocka u blizini, odnosno u radijusu od 2 cm. Ako jeste, `found=True`, `cube_id` vraća naziv kocke. Voditi računa da se baza robota nalazi na:

```
ROBOT_BASE_X = -0.2  
ROBOT_BASE_Y = -0.5  
ROBOT_BASE_Z = 1.021
```

pa je potrebno izvršiti adekvatnu transformaciju pozicije TCP-a pre prosleđivanja servisu `/get_closest_cube`. Pozvati ovaj servis nekoliko puta, radi sigurnosti da je dobra kocka pronađena.

```
geometry_msgs/Pose target_pose
---
string cube_id
bool found
```

Pozu TCP-a možemo dobiti po ovom šablonu.

```
moveit_py = MoveItPy(node_name="moveit_py_tcp_pose")

# Get the active planning component (e.g., your robot arm)
arm_component = moveit_py.get_planning_component("panda_arm")

# Get the current robot state
robot_state = arm_component.get_start_state()

# Alternatively, fetch the latest state from the Planning Scene #
robot_state =
moveit_py.get_planning_scene_monitor().get_planning_scene().getCurrentState()

# Define your end-effector link name (e.g., "tool0" or "panda_link8")
eef_link = "panda_link8"

# Retrieve the current pose as a Transform (position & quaternion)
tcp_transform = robot_state.get_global_link_transform(eef_link)
```

Pored ovoga, dostupne su teme `/gz/cube_{id}/attach` i `/gz/cube_{id}/detach` tipa `std_msgs.msg.Empty`. Slanjem jedne poruke na odgovarajuću temu, data kocka se spaja na alat, odnosno odvaja od alata robota. Na ovaj način je potrebno simulirati hvatanje.

Voditi računa da se na ove teme ne pošalje poruka pre nego što alat priđe kocki na dovoljnu razdaljinu, odnosno slati poruke samo onda kada je `found=True` za `attach` i kada smo prišli na odgovarajuću površ za ostavljanje.