

**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

Sortirni robotski manipulator

Stjepan Molnar

Osijek, 2025.

SADRŽAJ

1. Uvod.....	1
2. Opis radnog okruženja i robotske ruke	1
3. Implementacija	2
4. Zaključak	5

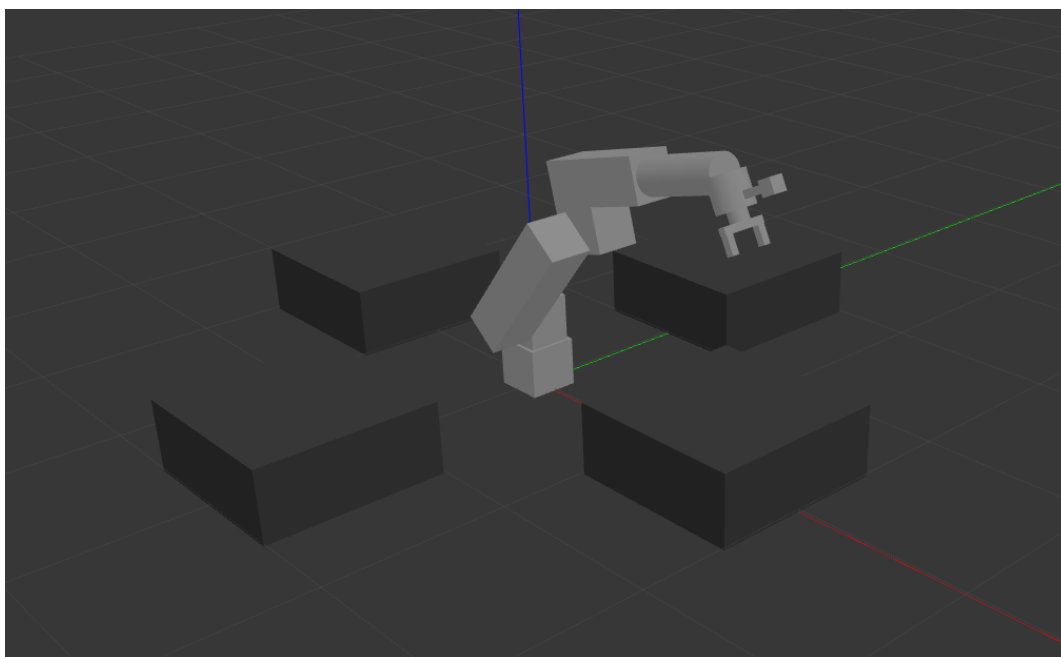
1. Uvod

Sortiranje je učestala radnja u proizvodnom procesu. Korištenje robotske ruke olakšava i ubrzava procese sortiranja.

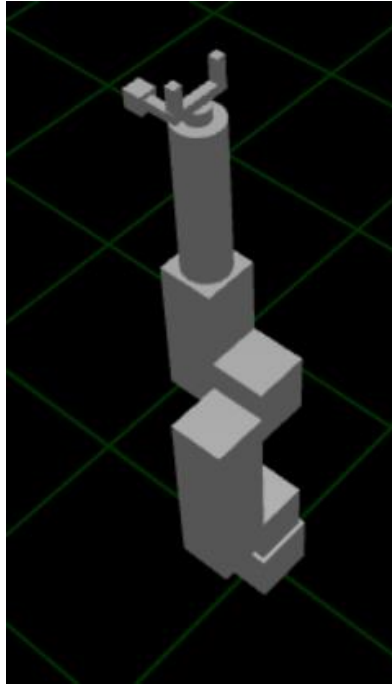
ROS (engl. Robot Operating System) je skup biblioteka i alata dizajniran za objedinjenje računalnih klastera korištenih u robotskom manipulatoru. Pomoću ROS-a je implementirano sortiranje.

2. Opis radnog okruženja i robotske ruke

Radno okruženje robota se sastoji od robotske ruke i četiri stola, jedan stol na kojem se stvaraju kocke različitih boja i tri na kojem se postavljaju kocke sortirane po bojama. Prikaz okruženja je na sljedećoj slici.



Robotski manipulator je dizajniran na temelju Yaskawa MS80 robota. Dodana su kliješta sa dva članka te kamera. Prikaz robota je na sljedećoj slici.



Upravljanje robota je razdvojeno na dva dijela. Upravljanje robotske ruke kojom se upravlja pomoću položaja izvršnog članka. Korišten je „position_controllers/JointTrajectoryController“. Upravljanje po položaju je moguće zadavanjem pozicije i orijentacije izvršnog članka, zadavanje kutova zglobova robota ili zadavanje predefiniranim položajima robota.

Upravljanje kliještima je odrađeno pomoću „effort_controllers/JointTrajectoryController“. Ovaj pristup omogućuje da koristimo silu za prihvat objekata. Iako se koristi „plugin“ koji fiksira objekt između članaka, korištenje sile povećava šansu da „plugin“ dobro radi.

Kocke koje robot sortira su identičnog oblika ali različite boje, program ih stvara iznad sortirnog stola, pozicija nije egzaktna nego sadrži slučajan element.

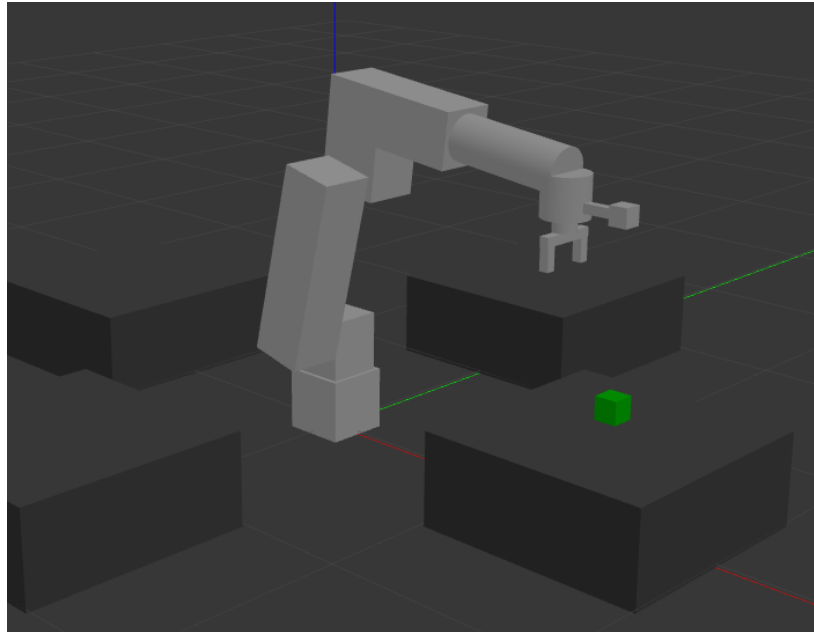
3. Implementacija

Simulacija započinje stvaranjem robotske ruke, koja je stvorena pomoću generiranih „Move-it“ datoteka.

Stvaranje statičkog dijela okružja i stvaranje kocke je odrađeno pomoću servisa (engl. Service). Poslužitelj je u ovom slučaju sami „gazebo“ simulator, on omogućuje stvaranje objekta na sceni, potrebno je poslati „SpawnModelRequest“ koji se sastoji od naziva objekta, putanje do urdf datoteke koja opisuje objekt i položaja objekta u prostoru s obzirom na referentni koordinatni sustav. Za stvaranje stolova je korišten identičan model samo je položaj drugačiji.

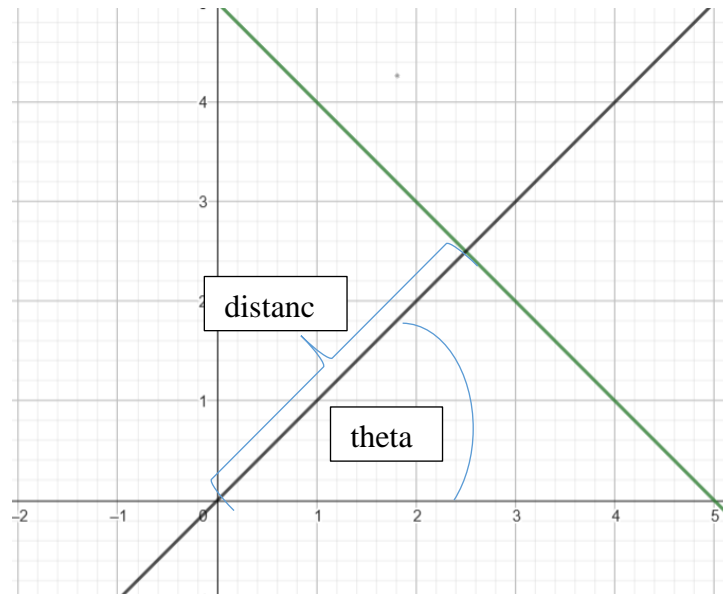
Slučajnim odabirom je odabrana jedna boje kocke, za svaku boju je jedna urdf datoteka. Kocka se ne stvara na istome mjestu nego se stvara sa disperzijom od $[-0.15, 0.15]$ po x i y osi od centra stola.

Robotska ruka odlazi u predefinirani „home“ položaj u koji će se vraćati nakon svake sortirane kocke. Nakon toga kocka odlazi u položaj za pregled. Prikaz robota u položaju za pregled je na sljedećoj slici.



Kamera u tom položaju uslikava stol i kocku te pomoću algoritma pronalazi poziciju kocke u prostoru slike te transformira to u prostor kamere. Toj jednoj inverznoj transformaciji se pridružuje transformacija iz baze robota u kameru te zajedno čine položaj kocke u prostoru.

Korišten je *Canny Edge detection* metoda za otkrivanje rubova kocke. Nakon toga je korišten *Hough line detection* koji detektira linije na slici, u kombinaciji sa metodom *hough_line_peaks* izvučene su četiri dominantne linije kocke. *Hough line* metoda vraća vrijednosti *theta* i *distance*, te dvije vrijednosti su vrijednosti u polarnom zapisu linearne funkcije.



Taj oblik zapisa je korišten zato što pokriva slučajeve kada je nagib funkcije 90 stupnjeva. Pomoću $y = ax+b$ to ne bi bilo moguće.

Zapis linearne funkcije u ovom obliku je

$$distance = x \cos(theta) + y \sin(theta)$$

Određivanje sjecišta pojedinih pravaca je dobiveno tako da se izjednači dvije funkcije te se rješavanjem tog sustava jednadžbi dobiva točka sjecišta ta dva pravca. Postupak se ponavlja za sve pravce koji imaju sjecište na slici.

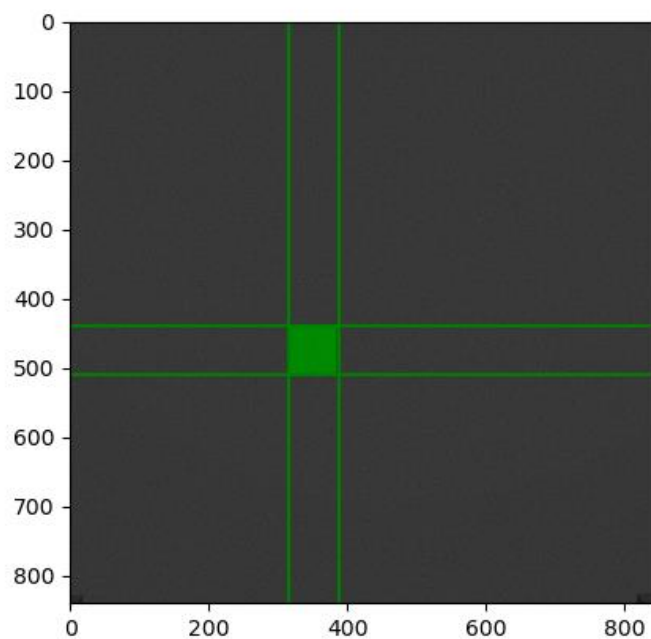
Položaj pravokutnika se dobiva kao prosjek svih x, y točaka(odvojeno za x pa za y). Izračunata pozicija je u prostoru slike, potrebno je pomoću formule tu poziciju preslikati na poziciju kamere u prostoru, to se izračunava pomoću sljedećih izraza:

$$u = f \frac{x}{z} + u_c$$

Pozicija u prostoru kamere je označena sa u i v. Vrijednosti optičkog centra su označeni sa u_c i v_c . Vrijednost f predstavlja žarišnu duljinu kamere izražene u pixelima. Vrijednosti z osi je zadana tako da je potrebno izračunati x i y vrijednosti.

$$v = f \frac{y}{z} + v_c$$

Podatak o boji kocke je dobiven iz vrijednosti piksela u izračunatom središtu kocke, koja boja dominira na „rgb“ kanalu, te boje je kocka te sukladno tome je sortirana. Prikaz slike koju je kamera uslikala, te dodane *Hough-ove* linije je na sljedećoj slici .



Nakon detekcije pozicije kocke u prostoru, robot uhvati kocku te ju postavlja na predefinirani stol ovisno o boji. Ovaj postupak se ponavlja 10 puta čime program završava.

4. Zaključak

Omogućeno je korištenje robotske ruke za sortiranje kocki ovisno o boji. Algoritam za prepoznavanje pozicije kocke u prostoru dovoljno dobro i precizno određuje poziciju kocke, čime je omogućeno dinamičko hvatanje kocke.