

Robotski sistem za zarezovanje krušnega testa med premikanjem na tekočem traku

Maruša Petrovčič¹

Mentorji: Sebastjan Šlajpah¹, Aleš Turel², Peter Dolenc², Marko Munih¹, Matjaž Mihelj¹

¹Fakulteta za elektrotehniko, Univerza v Ljubljani, Tržaška c. 25, 1000 Ljubljana

²Gostol Gopan d.o.o., Prvomajska ulica 37, 5000 Nova Gorica

E-pošta: mp8702@student.uni-lj.si

Robotic system for dough scoring during conveyor belt operation

This paper presents the development of a robotic system for automated bread dough scoring on a moving conveyor belt. The project was undertaken in collaboration with Gostol Gopan d.o.o., a manufacturer of high-capacity baking lines. The system integrates a parallel robot Omron iX3, a conveyor belt with incremental encoders, and a laser scanner SICK TriSpectorP1000 for dough pose detection.

The laser scanner acquires point clouds of the dough on the conveyor belt. Principal component analysis is applied to calculate the optimal bounding box of the loaf, enabling the determination of the dough position and planning the trajectory of robot cuts. Scoring is performed with a blade attached to the robot while following the conveyor movement. The test results have shown that the system accurately detects the position and orientation of the dough and completes the cutting process in under 3 seconds, facilitating efficient automation of the procedure.

Kratek pregled prispevka:

V prispevku predstavljamo razvoj robotskega sistema za avtomatsko zarezovanje krušnega testa na premikajočem se tekočem traku. Projekt je bil izveden v sodelovanju s podjetjem Gostol Gopan d.o.o., ki se ukvarja s proizvodnjo linij za peko kruha. Robotski sistem sestavljajo paralelni robot Omron iX3, tekoči trak z inkrementalnima dajalnikoma pozicije ter laserski skener SICK TriSpectorP1000 za določanje lege krušnega testa.

Z laserskim skenerjem zajemamo oblake točk krušnega testa na tekočem traku. Z metodo glavnih komponent izračunamo optimalni omejitveni volumen testa, kar omogoča določitev lege objekta, ter načrtamo trajektorije rezov z robotom. Zarezujemo z nožem, pritrjenim na vrh robota, pri čemer sledimo premikanju tekočega traku. Rezultati testiranja so pokazali, da sistem z ustrezno natančnostjo zazna položaj in orientacijo testa ter izvede zarezovanje v manj kot 3 sekundah, kar omogoča učinkovito avtomatizacijo postopka.

1 Uvod

Kot odziv na problematike preskrbe s prehrano, naraščanje svetovne populacije in pomanjkanje delovne sile se kot mnoge druge panoge tudi prehranski sektor usmerja v avtomatizacijo in robotizacijo [1, 2, 3]. Na ta način se ne le zmanjša obremenitev delavcev s fizično zahtevnimi nalogami, temveč se v mnogih primerih tudi izboljša kakovost izdelkov. Ponovljivi avtomatizirani postopki predelave prehranskih izdelkov so večinoma hitrejši v primerjavi z ročnim delom, izboljšan je estetski izgled izdelkov, nenazadnje pa je ob odsotnosti človeka pogosto lažje zadostiti higienskim standardom [4, 5]. Kruh je osnovno živilo, ki ga v takšni ali drugačni obliki pozna večina svetovnih kultur in je v človeški prehrani v osnovnih oblikah prisoten že zadnjih pet tisoč let [6].

2 Robotski sistem za zarezovanje krušnega testa

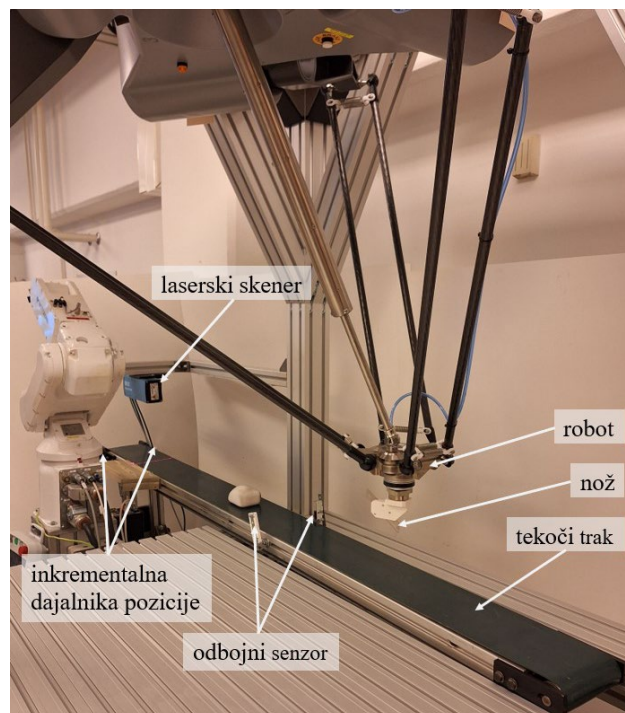
Podjetje Gostol Gopan d.o.o. razvija in proizvaja linije za peko pekovskih izdelkov s kapaciteto tudi nad 10 000 kosov na uro. Cilj raziskave je bil razvoj robotskega sistema za zarezovanje krušnega testa na premikajočem tekočem traku. Štruce testa je potrebno vizualno zaznati, določiti njihovo lego, ustrezno načrtati trajektorijo robota in izvesti zarezovanje. Ciljni čas cikla zarezovanja znaša 2,7 sekunde.

3 Uporabljena oprema

Ker je ena izmed zahtev aplikacije ustrezno kratek čas cikla, smo za izvedbo zarezovanja izbrali paralelnega robota Omron iX3 565 s krmilnikom NJ501 R300, ki ima specificiran čas cikla 0,35 sekunde pri obremenitvi 0,1 kg. Zarezovanje krušnega testa se sicer izvaja z vrtečimi rezalnimi diski, za potrebe razvoja aplikacije pa je bil na vrh robota vpet nož. Programiranje robota je potekalo v programskem okolju Automation Control Environment (ACE).

V delovnem prostoru robota se nahaja tekoči trak, opremljen z dvema inkrementalnima dajalnikoma pozicije: SICK DLS40E-S3AV01024 (povezan s krmilnikom robota) ter SICK

DFS60B-S4PC10000 (povezan z laserskim skenerjem). Za določanje lege štruce testa na tekočem traku je uporabljen laserski skener SICK TriSpectorP1000, ki deluje na osnovi laserske triangulacije in na podlagi podatkov z optičnega kodirnika tekočega traku zajete posnetke (rezine) združuje v oblake točk.



Slika 1: Robotska celica za zarezovanje krušnega testa

Začetek štruce na tekočem traku je detektiran z odbojnim senzorjem Leuze PRK 3B/66-S8. Obdelava podatkov in sinhronizacija med elementi robotskega sistema poteka na računalniku v programskem jeziku python.

Celotna robotska celica je prikazana na sliki 1.

4 Zajem oblakov točk

Pozicijo in orientacijo štruce krušnega testa na tekočem traku določamo z laserskim skenerjem, ki ga programiramo v programskem jeziku Lua v okolju SICK AppStudio. Ker se lega skenerja nad tekočim trakom ne spreminja (slika 2), so parametri zajema konstantni. Optimalno širino vidnega polja skenerja določimo glede na širino tekočega traku, dolžino posameznega zajema pa prilagodimo predvideni dolžini štruce krušnega testa. Pogostost zajemanja profilov je določena s

hitrostjo tekočega traku, ki jo merimo z inkrementalnim kodirnikom. Zajemanje skenov se začne, ko objekt (štruca) prekine laserski snop na določeni višini.



Slika 2: Postavitev laserskega skenerja nad tekočim trakom

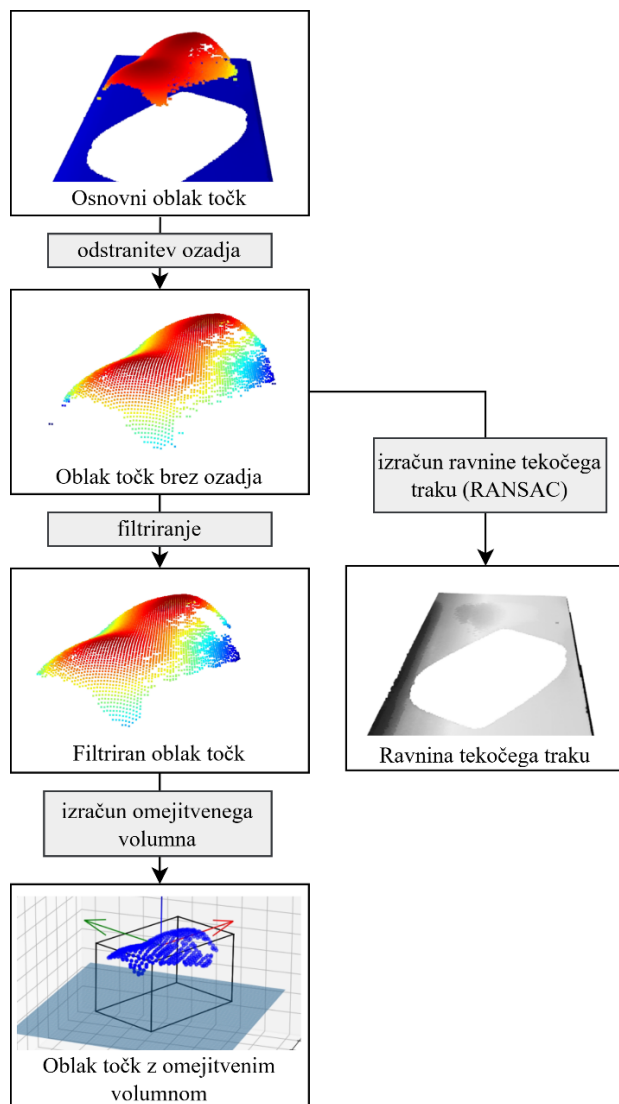
5 Obdelava oblakov točk

Podatke, zajete z laserskim skenerjem, je potrebno primerno obdelati, da lahko za vsako štruco načrtamo reze robota. Za delo z oblaki točk je uporabljena prosto dostopna knjižnica *Open3D*. Od oblaka točk odstranimo dele oblaka, ki niso predmet zanimanja, odstranimo osamelce (funkcija *remove_statistical_outlier*) in znižamo število vzorcev (angl. *downsampling*).

Ker predpostavljamo, da v oblaku točk največjo ravno površino predstavlja tekoči trak, lahko za določitev parametrov ravnine uporabimo funkcijo *segment_plane*, ki deluje na podlagi metode RANSAC (angl. *random sample consensus*). Funkcija naključno izbere n točk iz oblaka, na podlagi katerih nato določi ravnino ter za ostale točke glede na podano prazno vrednost preveri, ali pripadajo izbrani ravnini. Ravnina, ki ji pripada največ točk iz oblaka, je iskana ravnina tekočega traku. Ker se lega laserskega skenerja ne spreminja, parametre ravnine traku določimo samo enkrat in tako pohitrimo proces obdelave podatkov.

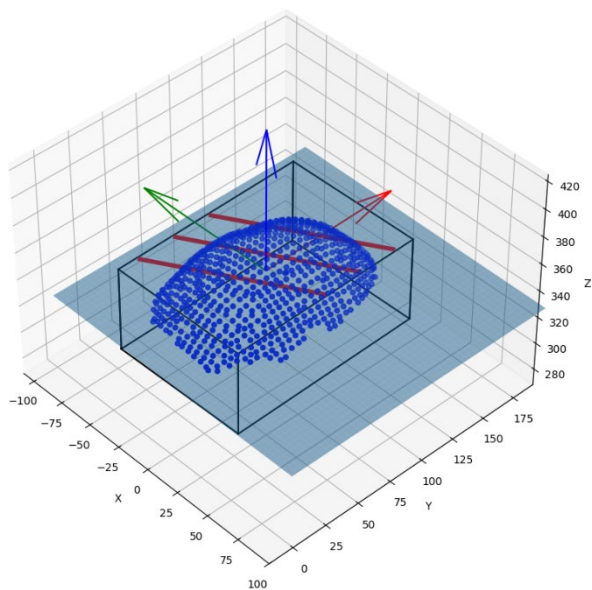
Za določitev lege štruca krušnega testa na tekočem traku izračunamo njen omejitveni volumen (angl. *bounding box*). Os x štruca

določimo z metodo glavnih komponent z uporabo funkcije *PCA* (angl. *principal component analysis*) iz knjižnice *scikit-learn*. Točke iz oblaka projiciramo na ravnino tekočega traku ter poiščemo smer, v kateri imajo točke največjo varianco, in le-ta ustreza osi x .



Slika 3: Diagram poteka obdelave oblaka točk

Os z sovpada z normalo ravnine tekočega traku, os y pa določimo tako, da je koordinatni sistem desnosičen. Ko poiščemo skrajne točke vzdolž določenih lastnih osi štruca, lahko izračunamo oglišča omejitvenega volumna in višino štruca. Na zgornjo ploskev omejitvenega volumna nato načrtamo poljubno število rezov pod poljubnim kotom, ki so določeni vsak z začetno in končno točko, kot je prikazano na sliki 4. Dolžino reza določata širina omejitvenega volumna ter kot reza.



Slika 4: Oblak točk testa z načrtanim omejevalnim volumnom in rezi robota

6 Robotski program

Program robota je napisan kot strukturirano besedilo (angl. *structured text*, ST) v programskem okolju ACE. Ob zagonu aplikacije se najprej vzpostavi povezava robota s TCP/IP strežnikom na računalniku, nato pa poženemo tekoči trak.

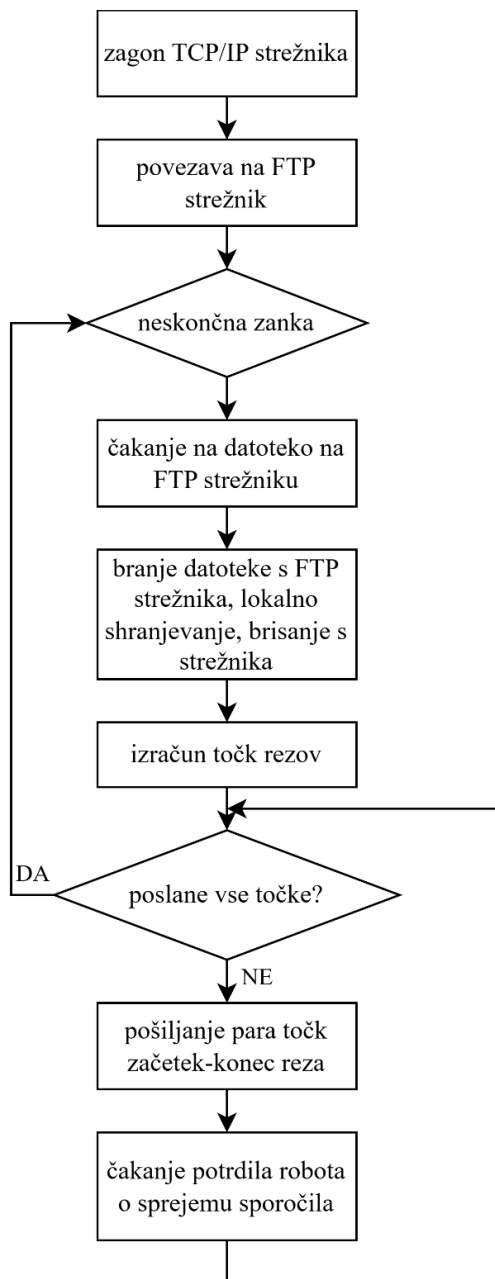
Robot preko TCP/IP protokola sprejema pare koordinat začetnih in končnih točk reza ter vsakič potrdi prejem, dokler ne dobi znaka za konec pošiljanja. Nato se pomakne nad tekoči trak in čaka, da štruca testa prekine žarek odbojnega senzorja. Vrh robota orientiramo tako, da je rezilo noža usmerjeno vzdolž načrtovane trajektorije reza. Upoštevajoč hitrost tekočega traku koordinatam začetka in konca reza med izvajanjem prištevamo vmesni premik traku, tako da robot med rezanjem sledi štruci na tekočem traku.

7 Integracija sistema

Medsebojno delovanje posameznih enot robotskega sistema za zarezovanje usklajuje program v jeziku python, ki teče na računalniku. Slika 5 prikazuje zaporedje korakov pri izvedbi procesa zarezovanja.

Ko poženemo program, se najprej zažene TCP/IP strežnik, pri katerem računalnik

predstavlja strežniško stran, in čaka na povezavo z robotom, ki je klient. Nato se vzpostavi še povezava s FTP strežnikom, ki teče na računalniku in s katerim je že povezan laserski skener.



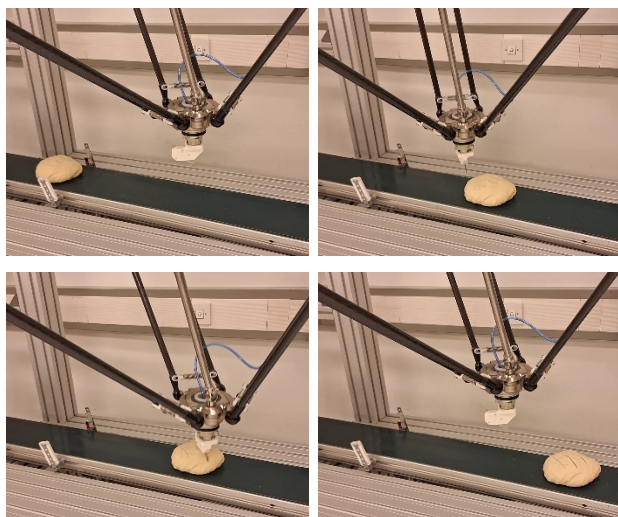
Slika 5: Diagram poteka korakov celotne aplikacije zarezovanja

Program nato vstopi v neskončno zanko, ki se izvaja do prekinitve s strani operaterja. Program čaka, da laserski skener shrani datoteko z oblakom točk na FTP strežnik. Datoteko prebere, jo lokalno shrani ter jo izbriše s strežnika.

Lokalno shranjeni oblak točk obdelamo in načrtamo točke rezov robota, kot je opisano v poglavju 5. Posamezne pare točk začetka in konca reza nato prek TCP/IP komunikacije pošljemo robotu, pri čemer je zadnji poslani par posebej označen, robot pa prejem vsakega para točk potrdi. Na tej točki se izvede postopek zarezovanja štruce, program pa že preverja prisotnost novih datotek na FTP strežniku.

8 Rezultati

Slika 6 prikazuje nekaj izsekov iz postopka zarezovanja testne štruce, rezultat zarezovanja pa je prikazan na sliki 7. Skupni čas obdelave oblaka točk in izračuna rezov robota znaša 0,3 sekunde.



Slika 6: Izvedba zarezovanja krušnega testa

Čas zarezovanja testa, brez upoštevanja časa obdelave podatkov, znaša manj kot 3 sekunde. Potrebno je omeniti, da zaradi uporabe noža namesto rezalnih diskov, rezov nismo izvajali pri robotovi maksimalni hitrosti. Ob višjih hitrostih zarezovanja nož namreč ni lepo zarezal skozi testo, zaradi česar se je štruca odlepila od traku in zarezovanje ni bilo uspešno. Ob uporabi rezalnih diskov se ta problem najverjetneje ne bi pojavil.

Preizkusili smo različne lege štruc testa na tekočem traku. Orientacija štruce je bila pravilno zaznana v vseh primerih, pravilno so bili določeni tudi rezi robota. Pri polovici maksimalne hitrosti robota je bila tudi izvedba zarezovanja uspešna.



Slika 7: Končni rezultat zarezovanja

9 Zaključek

Predstavljeni sistem za zarezovanje zarezuje štruco testa v poljubni legi z rezi pod poljubnim kotom. Ključno slabost trenutne aplikacije predstavlja relativno dolg čas izvedbe celotnega procesa. Le-ta je v veliki meri posledica dolgih časov komunikacije med elementi sistema. Najšibkejši člen zagotovo predstavlja prenos podatkov med laserskim skenerjem in računalnikom, ki je bil zaradi težav pri vzpostavitvi komunikacije izveden preko protokola FTP in bi ga bilo bolje izvesti s protokolom TCP/IP. Kot že omenjeno, bi nam tehnologija zarezovanja z rezalnimi diski namesto z nožem omogočila zvišanje hitrosti gibov robota in s tem skrajšanje časa samega zarezovanja.

10 Literatura

- [1] Derossi, A., et al., *Avenues for non-conventional robotics technology applications in the food industry*, Food Research International, str. 113265, 2023.
- [2] Wakchaure, Y. B., B. K. Patle, and Sachin Pawar, *Prospects of robotics in food processing: an overview*, Journal of Mechanical Engineering, Automation and Control Systems, vol. 4, no. 1, str. 17-37, 2023.
- [3] Davis, Steve, et al., *Robot prototyping in the design of food processing machinery*, Industrial Robot: An International Journal, vol. 34, no.2, str. 135-141, 2007.
- [4] Drozd, Radosław, *The role of robotisation in improving bakery product quality*, Aparatura Badawcza i Dydaktyczna, str. 261-267, 2020.
- [5] Sancar, Hüseyin, *Automatic Dough Draw Robot System Design*, Gazi University Journal of Science Part A: Engineering and Innovation, vol. 6, no. 4, str. 93-103, 2019.
- [6] Zagorzewski, Timothy, *(Re)making bread: industrial technologies and the skills of food industry workers*, Diss. University of Saskatchewan, 2008.