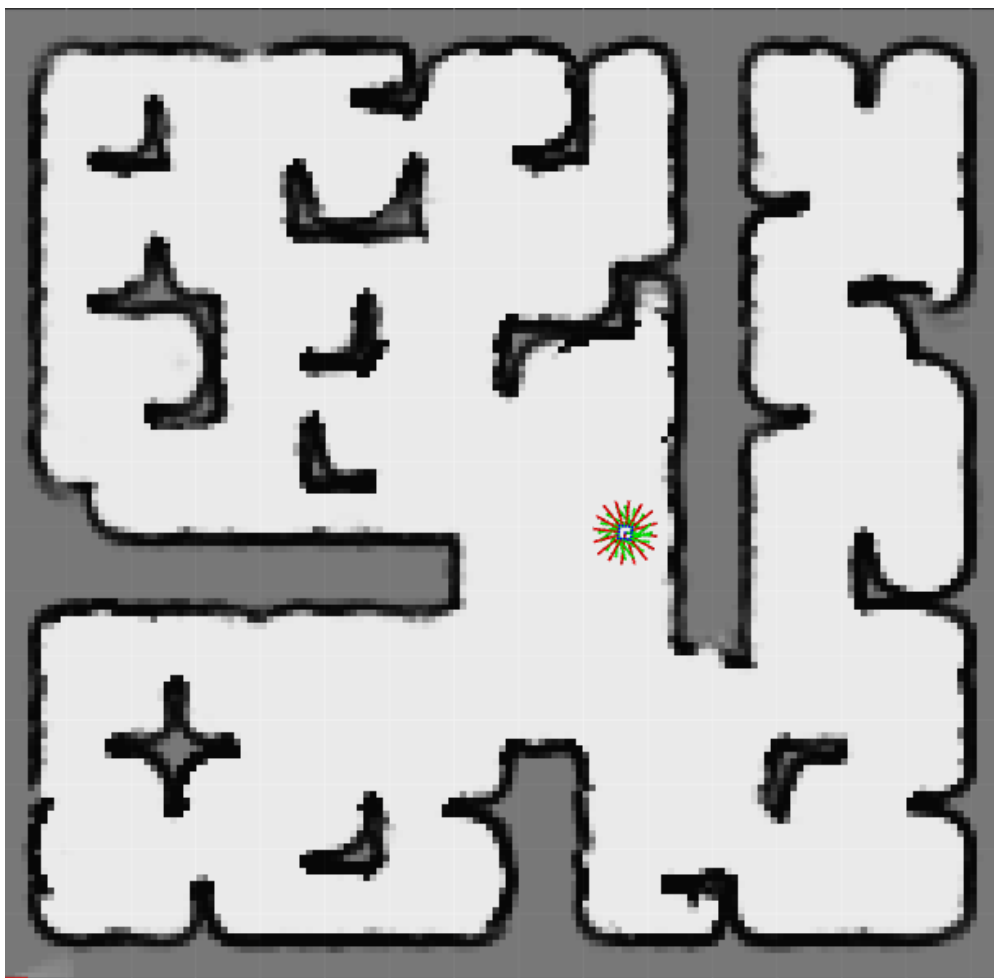
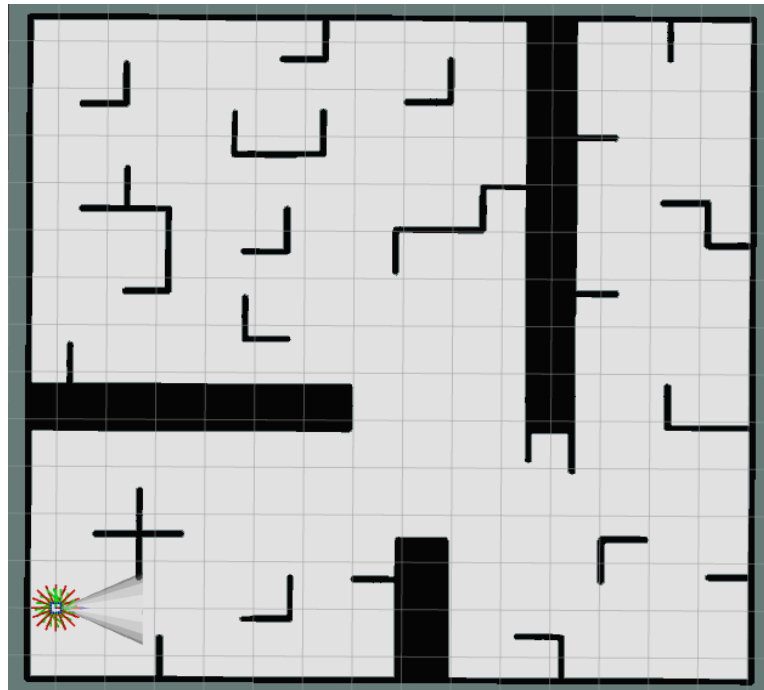


MARKOVIĆ, LOVRO AUTOMATIKA 0036479998	SVEUČILIŠTE U ZAGREBU, FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA ZAVOD ZA AUTOMATIKU I RAČUNALNO INŽENJERSTVO	8.1.2018.
	Mobilna robotika	
	IZVJEŠTAJ S LABORATORIJSKIH VJEŽBI	

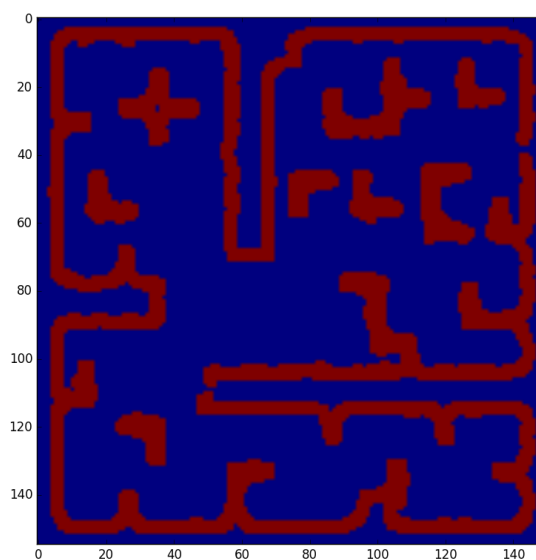
1. VJEŽBA – Izgradnja mrežaste karte zauzetosti prostora primjenom Bayesovog pravila



Slika 1: Izgrađena mapa uvjetnih vjerojatnosti $P(S = O \mid R1:k)$, gdje je bijela boja vjerojatnost 0, a crna 1



Slika 2: Izvorna mapa prostora



Slika 3: Polje zauzetosti sa skaliranim vrijednostima na 0, ako je vjerojatnost toga polja manje od 0.5, inače 1

Uvod

U ovoj vježbi bilo je potrebno naći uvjetne vjerojatnosti svakog pojedinog polja mape $P(S = O \mid R_{1:K})$. To je napravljeno iterativnim postupkom korištenjem jednadžbe modela sonara koja vraća vjerojatnost zauzeća polja uz uvjet jednog mjerenja, tj. $P(S = O \mid R_k)$.

Komentari i problemi

Broj sonara nije mijenjan, iako će program raditi nešto brže ukoliko isključimo neke sonare, ali pokrivenost trenutnog okolnog prostora biti će manja, tj. nepotpuna.

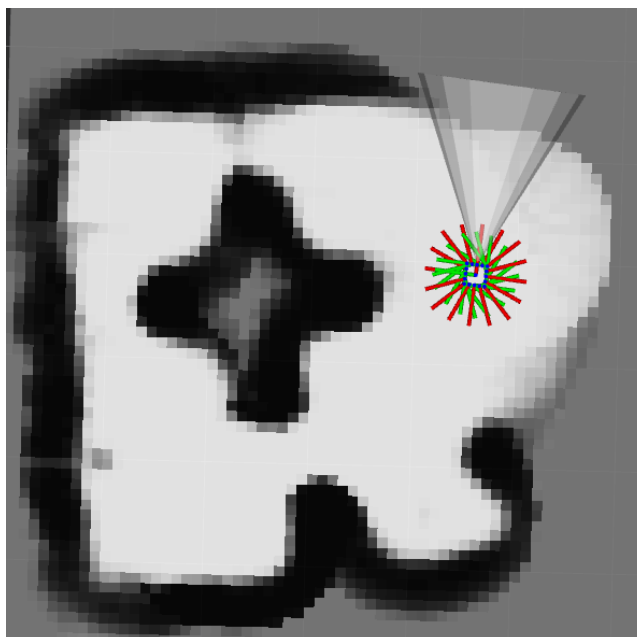
Najteži dio vježbe bio je ispravno izračunati poziciju i kuteve sonara u odnosu na dio mape na kojem računamo vjerojatnosti.

Cjelokupni algoritam dosta se ubrzao nakon što se prilikom izračunavanja vjerojatnosti polja pogođenih pojedinim sonarom automatski zanemarena ona polja izvan trenutnog vidokruga robota, tj. njihova vjerojatnost bit će izvorno postavljena na 0.5 za trenutni sonar.

Karta prikazana na Slici 1. sagrađena je korištenjem mjerenja nezašumljene odometrije. Ukoliko se koristi odometrija sa dodanim šumom i robot zapne u nekom trenutku, pojavit će se napravnosti na karti, npr. odmak od detektniranih i stvarnih prepreka, nakošene prepreke itd.

Kako bi program ispravno radio potrebno je polagano voziti robota uokolo, te ponekad zastajati / okretati robota kako bi se povećala vjerojatnost mjestima na kojima postoje prepreke odnosno smanjila tamo gdje one nisu.

Algoritam također često zaobli rubove prepreka te dugo vremena treba da se popune uske rupe i prostori. Takav problem se može riješiti sporijim prolaskom kroz navedena područja. Povećanjem parametra Δ_r dobije se puno deblje područje prepreka. To je pola širine dijela na kojem sonar poljima pridodjeljuje najveću vjerojatnost. Utjecaj tog parametra može se vidjeti na Slici 4. Njegovim smanjivanjem, osjenčani rub prepreke postaje tanji.



Slika 4: Δ_r povećan 3 puta na 0.3 m

Povećavanjem maksimalne vrijednosti uvjetne vjerojatnosti senzora modela (p_O) i istovremenim smanjivanjem minimalne vrijednosti senzora modela (p_E) značajno se povećava brzina icrtavanja mape, odnosno polja će ili vrlo brzo pobijeliti ili vrlo brzo pocrniti. Takav pristup, iako je brži, nije toliko siguran ukoliko postoji značajan šum u sustavu.

2. VJEŽBA – Lokalizacija mobilnog robota primjenom Kalmanovog filtra i sonara

Uvod

U ovoj vježbi bilo je potrebno odrediti položaj i orijentaciju robota korištenjem proširenog kalmanovog filtra. Za *a-priori* izračun stanja sustava uzeta je jednadžba koja opisuje kinematički model robota, a za korekciju korištena je jednadžba simulirane udaljenosti mjerenja sonara od njegove najbliže prepreke i trenutnog centra robota.

Prikaz rezultata bit će proveden koristeći izvornu mapu za detekciju prepreka te mapu snimljenu u prvoj laboratorijskoj vježbi.

Komentari i problemi

Glavni problem ovog lokalizacijskog algoritma je veliko vrijeme računanja pojedine iteracije. Razlog tome je dio koda u kojem se računa simulacija i-tog sonara, tj. traže se koordinate najbliže prepreke koju vidi i-ti sonar. Taj problem djelomično je riješen filtriranjem mape tako da se na kraju dobiju samo pozicije prepreka, te automatskim odbacivanjem svih prepreka koje su izvan dosega sonara.

Javlja se problem i kod estimacije kuta zakreta robota što se osobito vidi prilikom korištenja snimljene mape prostora te naglih zakreta robota. Jedan od uzroka tome je korekcija mjerenjem koje ne uzima u obzir kut zakreta robota (jer se računa samo udaljenost). Djelomično rješenje problema dobiveno je smanjivanjem maksimalne linearne i kutne brzine robota, nakon čega su dobiveni prihvatljivi odzivi.

Matrica nesigurnosti P u inicijalnom koraku postavljena je kao jedinična matrica pomnožena sa konstantom (100). Nakon nekoliko pokretanja smanjena je nesigurnost parametara x i y dok je nesigurnost kuta zakreta ostala je jednaka. Sa takvim inicijalnim vrijednostima dobivena je bolja estimacija.

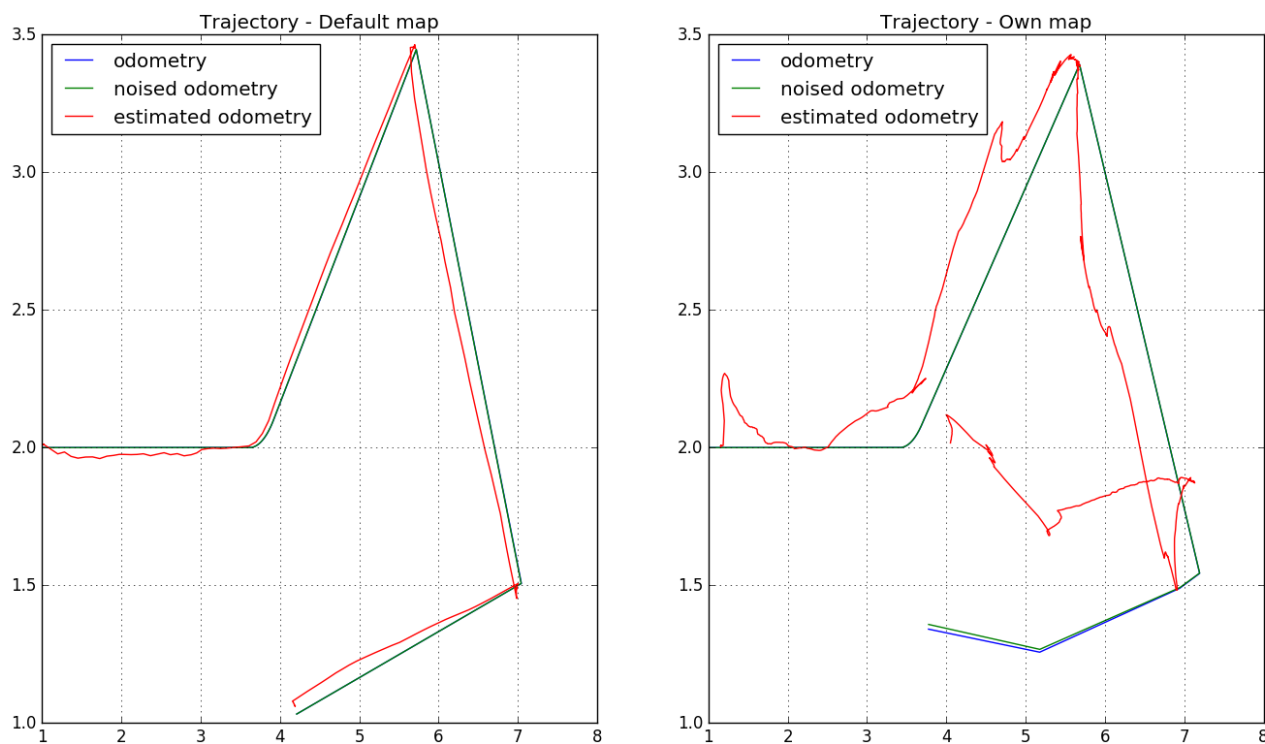
Kao inicijalna vrijednost estimacije u početnom koraku uzeta su očitavanja sa zašumljene odometrije.

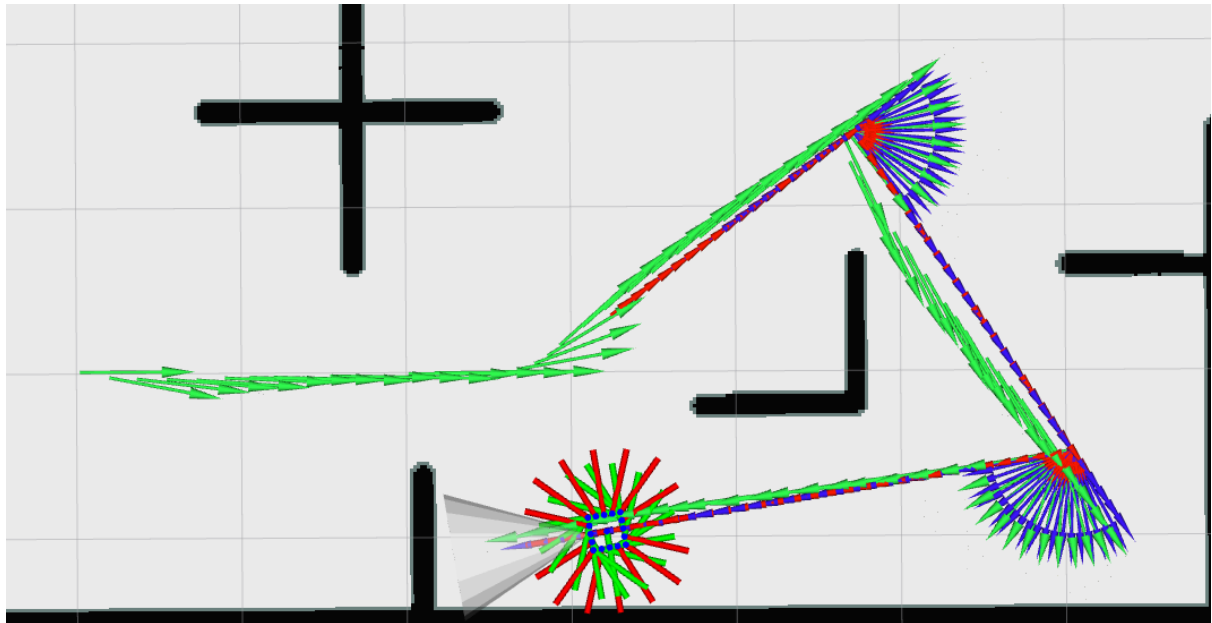
U nastavku će biti prikazane pogreške i trajektorije robota korištenjem izvorne mape i mape iz prvog zadatka za pronalazak prepreka. Srednja kvadratna pogreška izračunata je kao trag matrice P u k -tom trenutku.



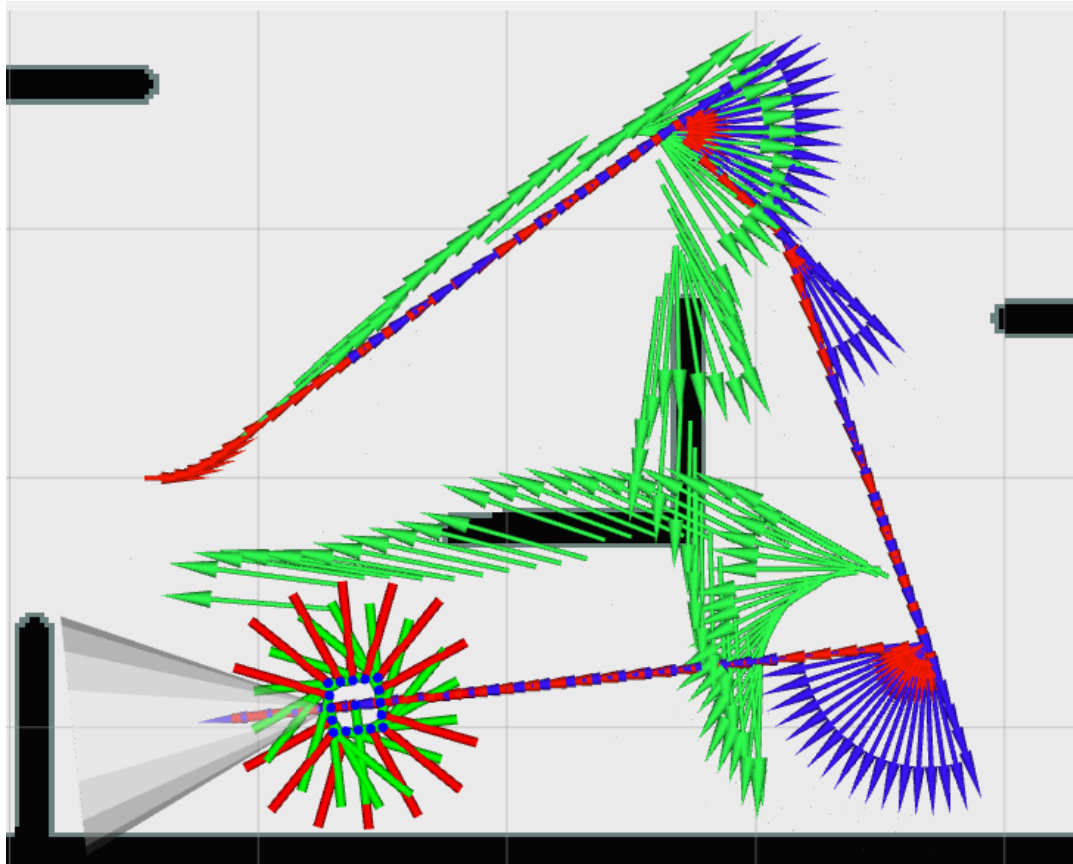
Slika 5: Usporedba srednje kvadratne pogreške uz korištenje izvorne i vlastite mape.

Slika 6: Usporedba putanje robota uz korištenje izvorne i vlastite mape.





Slika 7: Prikaz odometrije(Crveno), zašumljene odometrije(Plavo) i estimirane odometrije(Zeleno) za izvornu kartu



Slika 8: Prikaz odometrije(Crveno), zašumljene odometrije(Plavo) i estimirane odometrije(Zeleno) za vlastitu kartu

Estimirana trajektorija dobivena vlastitom kartom očekivano je lošija od trajektorije dobivene izvornom kartom. Kako bi dobili bolju trajektoriju bilo bi potrebno pažljivije iscrtavati mapu kod rubova kako bi dobivena mapa bila što sličnija izvornoj. Cjelokupni algoritam bi se mogao još više ubrzati implementacijom efikasnijeg algoritma za pronalazak najbližih prepreka sonara budući da je to računski najzahtjevnija operacija trenutno

VJEŽBA 3 – Planiranje gibanja mobilnog robota

Uvod

U ovoj vježbi bilo je potrebno isplanirati putanju pomoću A* algoritma te napraviti kontroler koji će pomicati robota po svim točkama potanje.

Rezultati će biti prikazani pomoću videa sa snimljenom mapom i zašumljenom odometrijom, te sa estimiranom odometrijom.

Komentari i problemi

Jedan od problema u ovoj vježbi jest što predloženi kontroler ne radi potpuno ispravno za ovaj problem. Problem se javlja kada upravljačke veličine i linearne i kutne brzine poprimu velike vrijednosti. Tada će robot vrlo često zaobići željenu ciljnu poziciju. Kao rješenje tog problema dodan je uvjet da ukoliko je upravljačka veličina kutne brzine jako velika tada će se vršiti samo rotacija oko z osi, a za linearnu brzinu uzimat će se nula. Na taj način robot će biti sposoban prelaziti točke kako god one bile postavljene.

Predloženi put A* algoritma često bude postavljen vrlo blizu zapreka na mapi što može rezultirati zaglavljenjem robota. U tu svrhu mapa je jednom na početku izvedbe algoritma zadebljana za 2 polja oko svake pronađene prepreke. Koristeći takvu mapu robot se neće moći više zabijati u prepreke.

Jedno od mogućih poboljšanja ovog algoritma bilo bi ubrzanje kretanja robota. To bi se moglo napraviti pronalaskom boljih parametara regulatora ili implementacijom boljeg upravljačkog algoritma. Ciljne udaljenosti u ovom algoritmu često će biti vrlo male što znači da sama linearna brzina robota nikada neće moći biti osobito velika zbog samog upravljačkog algoritma. Preskakanjem nekih točaka robot se može ubrzati, ali onda riskiramo zabijanje u prepreke.

Kao rezultat ove vježbe bit će dodani videi koji prikazuju putanje robota.