

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Žiga Kotnik Klovar

**Primerjava različne strojne opreme
za navigacijo mobilnih robotov**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Mentor: izr. prof. dr. Danijel Skočaj

Ljubljana, 2018

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Zmogljivi mobilni roboti so običajno dragi, zato se za množično poučevanje konceptov mobilne robotike pogosto uporabljajo cenovno bolj dostopne mobilne platforme in senzorji, ki pa zaostajajo v svojih zmogljivostih. V diplomski nalogi primerjajte med seboj natančnost različnih senzorjev za merjenje razdalje ter natančnost dveh robotov, nizkocenovne mobilne platforme TurtleBot ter mobilnega robota Pioneer 3-DX. Primerjajte tudi uspešnost navigacije obeh robotov opremljenih z različnimi merilci razdalje.

Zahvala

*Zahvaljujem se izr. prof. dr. Danijelu Skočaju za pomoč pri izdelavi diplomske naloge.
Zahvaljujem se tudi staršema za podporo pri izdelavi diplomske naloge.*

Žiga Kotnik Klovar, 2018

Kazalo

Povzetek

Abstract

1 Uvod.....	1
2 Strojna in programska oprema.....	3
2. 1 Senzorji.....	3
2. 1. 1 Kinect.....	3
2. 1. 2 Astra Pro.....	4
2. 1. 3 Kinect 2.....	5
2. 1. 4 URG-04LX.....	6
2. 1. 5 Primerjava senzorjev.....	6
2. 2 Roboti.....	7
2. 2. 1 TurtleBot.....	7
2. 2. 2 Pioneer 3-DX.....	9
2. 3 Programska oprema.....	10
2. 3. 1 ROS.....	10
2. 3. 2 Navigacija robota.....	11
3 Eksperimentalna evalvacija.....	13
3. 1 Evalvacija globinskih senzorjev.....	13
3. 1. 1 Postavitev eksperimentov.....	13
3. 1. 2 Eksperimentalni rezultati.....	16
3. 2 Evalvacija premikanja robotov.....	21
3. 2. 1 Vožnja naravnost.....	21
3. 2. 2 Vrtenje na mestu.....	26

3. 3 Evalvacija navigacije robotov.....	27
3. 3. 1 Zgrajeni zemljevidi.....	29
3. 3. 2 Rezultati eksperimentov.....	30
4 Zaključek.....	33
Literatura.....	35

Povzetek

Naslov: Primerjava različne strojne opreme za navigacijo mobilnih robotov

Avtor: Žiga Kotnik Klovar

V diplomskem delu smo preučevali razliko med delovanjem robotov in senzorjev glede na cenovni razred in princip njihovega delovanja. Opremo smo evalvirali v treh različnih fazah. V prvi fazi smo evalvirali delovanje senzorjev, v drugi delovanje robotov in v tretji kombinacijo robotov in senzorjev. Pri eksperimentiranju smo uporabljali senzorje Kinect, Astra Pro, Kinect 2 in URG-04LX ter robota TurtleBot in Pioneer 3-DX. Eksperimente smo opravljili z uporabo robotskega operacijskega sistema ROS, ki olajša programiranje z različnimi roboti in senzorji. Rezultati so pokazali, da je med senzorji najbolje deloval Kinect 2 in med roboti Pioneer 3-DX. Najboljša kombinacija je bila Pioneer 3-DX s senzorjem URG-04LX.

Ključne besede: SLAM, robot, senzor

Abstract

Title: Comparison of different hardware units for mobile robot navigation

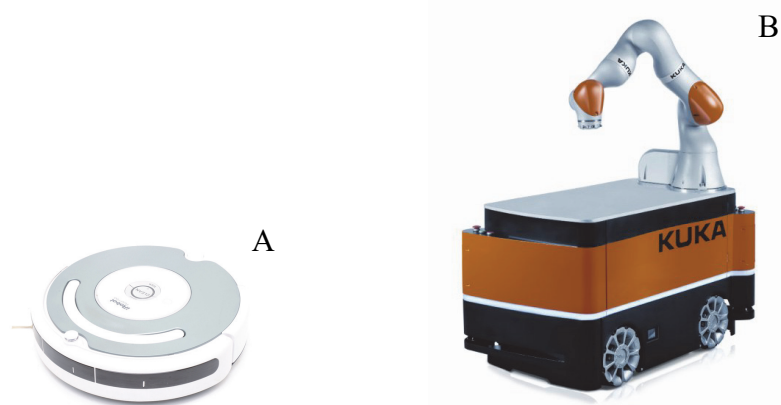
Author: Žiga Kotnik Klovar

We were interested in difference of operation between robots and sensors that differ in price and technology they use. We tested them in three different phases. In the first phase we tested operation of sensors, in the second phase we tested operation of robots and in the last phase we tested combination of robots and sensors. In tests we used four different sensors: Kinect, Astra Pro, Kinect 2 and URG-04LX and we used two different robots: Roomba and Pioneer 3-DX. During tests we used Robot Operating System (ROS). Kinect 2 showed the best performance among the sensors. Pioneer 3-DX was the best performing robot and best combination was Pioneer 3-DX with URG-04LX.

Keywords: SLAM, robot, sensor

1 Uvod

Danes se roboti vedno več uporabljajo. Od gospodinjskih opravil, kot so sesanje, do sestavljanja avtomobilov v tovarnah. Po mestnih cestah je že mogoče videti samovozeče avtomobile. Na trgu se pojavlja vedno več robotov, ki opravljajo mnogo različnih nalog. Istočasno sodijo tudi v različne cenovne razrede, kar se pozna pri njihovem izpolnjevanju nalog. Na sliki 1.1 lahko vidimo primer cenejšega robota Roomba, ki se uporablja za sesanje notranjih prostorov in dražjega industrijskega robota KMR iiwa podjetja KUKA.



Slika 1.1: (a) Robotski sesalnik Roomba [8]. (b) KMR iiwa [7].

Zaradi robotov, ki so sposobni reševati vedno bolj zapletene probleme, si je težko predstavljati, da še ne dolgo nazaj nismo znali z roboti rešiti najenostavnejših nalog, kot so na primer: Kje se nahajam?, Kaj opazujem? in podobno. V šestdesetih letih prejšnjega stoletja so se univerze, ki so razvijale umetno inteligenco, prvič srečale s tem problemom. Pri prvih poskusih so verjeli, da je možno posnemati človeški vid tako, da se robotu doda kamera in bo robot lahko opisal, kaj gleda. Izkazalo se je, da ta postopek ni tako preprost in v naslednjih desetletjih so raziskovalci razvijali različne metode in algoritme za detekcijo in opisovanje predmetov, ki jih ima kamera v vidnem polju [16].

Eden izmed problemov, na katerega naletimo pri programiranju robotov, je hkratna lokalizacija in kartiranje oz. *simultaneous localization and mapping* (SLAM), ki se ukvarja z grajenjem zemljevida in ocenjevanjem stanja robota. Običajno stanje robota opišemo s položajem in usmeritvijo. Poleg tega ga lahko opišemo tudi s trenutno hitrostjo robota,

kalibracijskimi parametri in podatki s senzorjev. Zemljevid nam predstavlja opis okolice, kjer robot deluje. Pri tem opišemo samo tiste stvari, ki so pomembne za pravilno navigacijo robota, npr. ovire ali značilne točke v prostoru. Problem nastane, ker moramo za grajenje zemljevida poznati položaj robota, ki pa ga ne poznamo, če nimamo zemljevida in tako naletimo na ciklični problem. Ta problem je delno rešen z algoritmi SLAM, ki se osredotočajo na delovanje z viri, ki so nam na voljo, namesto na popolno delovanje [2].

V Laboratoriju za umetne vizualne spoznavne sisteme na Fakulteti za računalništvo in informatiko se nahaja več različnih senzorjev in robotov, ki spadajo v različne cenovne razrede. Pri uporabi cenejših senzorjev se pri meritvah razdalje med senzorjem in opazovanim predmetom pojavljajo napake, ki lahko vplivajo na opravljanje eksperimentov. Poleg nenatančnosti senzorjev se lahko problemi pojavijo tudi pri robotih. Lahko se zgodi, da se robot, namesto normalne vožnje, zadrša po tleh. To lahko povzroči težave pri izračunih njegovega položaja.

Cilj naloge je bil pokazati razliko v delovanju opreme višjega cenovnega razreda v primerjavi z opremo nižjega cenovnega razreda in odkriti ali je bolj pomembno, da robot natančno računa odometrijo ali so bolj pomembni dobri senzori, ki vračajo pravilne razdalje do ovir.

V okviru diplomske naloge smo opravljali eksperimente s štirimi različnimi senzori: Kinectom, Astro Pro, Kinectom 2 in URG-04LX. Pri senzorijskih smo ocenili, kako dobro izmerijo razdaljo do predmetov v vidnem polju. Preverili smo delovanje senzorjev pri različni razdaljah in pod različnimi koti od stene.

Naslednji sklop eksperimentov smo opravili z robotoma TurtleBot in Pioneer 3-DX. Z eksperimenti smo želeli pokazati, kako dobro robot oceni svojo prepotovano pot. To smo storili tako, da smo opravili eksperimente pri katerih smo robotoma podali ukaz, naj vozita naravnost in primerjali prevoženo pot s tisto, ki jo je robot izračunal.

Zadnji sklop eksperimentov pa smo opravili s kombinacijo senzorjev in robotov. Pri tem smo evalvirali, kako dobro deluje gradnja zemljevida in umestitev robota nanj. Pri teh eksperimentih smo uporabljali podatke o prepotovani poti, ki jo je robot izračunal, in jih popravljali s podatki, ki smo jih prejeli iz senzorjev. Naloga robota je bila uspešno potovanje čez poligon.

V nadaljevanju naloge smo bolj podrobno opisali delovanje uporabljenih robotov in senzorjev, uporabljene metode pri eksperimentih in rezultate naloge. V drugem poglavju so opisani senzori, roboti in programska oprema, ki smo jo uporabljali pri eksperimentih, v tretjem poglavju so opisani rezultati, v četrtem poglavju pa smo zapisali naše ugotovitve in zaključke, do katerih smo prišli s pomočjo rezultatov eksperimenta.

2 Strojna in programska oprema

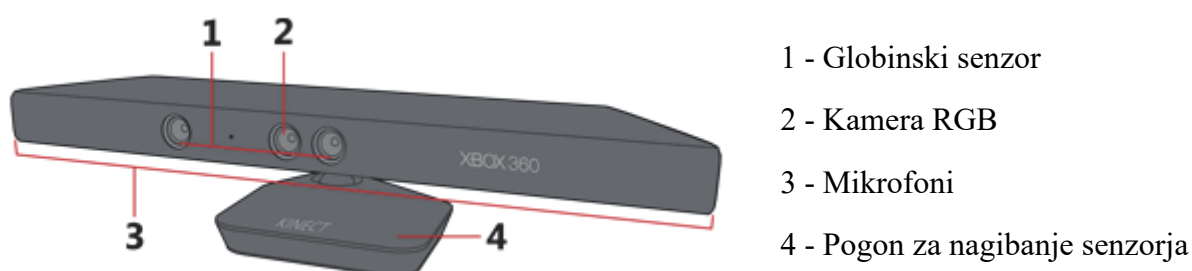
2.1 Senzorji

Pri meritvah smo uporabili več različnih senzorjev in robotov.

Uporabili smo senzorje Kinect, Astra Pro, Kinect 2 in URG-04LX. Zanimalo nas je kako natančno izmerijo razdaljo do predmetov v prostoru. Pri meritvah smo pričakovali, da se bo izkazalo, da so bolj natančni senzorji, ki uporabljajo bolj napredno tehnologijo in so v višjem cenovnem razredu. V to kategorijo spadata Kinect 2 in URG-04LX.

2.1.1 Kinect

Kinect je bil narejen kot pripomoček za igralno konzolo Xbox 360. V prodajo so ga dali novembra 2010. Leta 2012 je Microsoft objavil različico za operacijski sistem Windows skupaj z orodji, ki so razvijalcem omogočila razvoj aplikacij za Kinect v programskih jezikih C++, C# in Visual Basic .Net [17]. Kinect je bilo ob začetku prodaje možno kupiti za 150 dolarjev, z njegovo prodajo so prenehali aprila 2016. Opremljen je s kamero RGB, globinskim senzorjem, mikrofoni in ima pogon za nagibanje senzorja [11]. Na sliki 2.1 je prikazan senzor z vsemi sestavnimi deli.



Slika 2.1: Kinect z označenimi senzorji [11].

Kot igralni pripomoček je Kinect sposoben prepoznati šest ljudi, od tega lahko sledi dvema, pri katerih lahko loči 20 sklepov. Za igro je učinkovit med 1,2 in 3,5 metri, vodoravni zorni kot je 57 stopinj, navpični pa 43 stopinj. Za igranje iger potrebujemo približno 6 m² prostora. Njegova skrajna meja sledenja je med 0,7 in 6 metri. Sposoben je prepoznati tudi kretnje rok,

obrazne izraze in glas. Video, ki ga snema kamera, je v ločljivosti 640 x 480 s 30 okvirji na sekundo oziroma frames per second (FPS). To ločljivost je možno povečati na 1280 x 1024, a se pri tem zmanjša število okvirjev na sekundo. Možno si je ogledati tudi sliko infrardeče kamere, preden se ta pretvori v globinsko mapo. Globinski senzor je učinkovit med 0,7 in 6 metri in zgradi sliko ločljivosti 640 x 480 s 30 FPS.

Globinski senzor je razdeljen na infrardečo kamero (IR) in infrardeči projektor. Infrardeči projektor projicira žarke po sobi v poznanem vzorcu. Kamera IR pregleda sobo in primerja opazovan vzorec z znanimi vzorci. Na podlagi triangulacije izračuna globino, glede na to, pod katerim kotom je zaznal žarek. Triangulacija je postopek, pri katerem ugotovimo lokacijo neke točke, tako da iz dveh znanih točk ustvarimo trikotnik do nje. Da je ta izračun možen, morata biti infrardeči projektor in kamera IR kalibrirana. V primeru Kinecta se kalibracija opravi ob izdelavi vsake naprave [4]. Kalibracija je postopek, pri katerem primerjamo rezultate merilne naprave z znanimi vrednostmi pri istem poizkusu. Tako lahko nastavimo nastavitve senzorja, da nam vrača pravilne vrednosti. Zaradi takšnega delovanja je Kinect sposoben ločiti predmete v okolju, tudi če so vsi enake barve. Po drugi strani pa to pomeni, da ne moremo uporabljati več senzorjev v enem prostoru ali pa na močni sončni svetlobi.

2. 1. 2 *Astra Pro*

Tudi ta senzor je opremljen s kamero IR, kamero RGB in mikrofonom. Astro Pro (slika 2.2) je bilo avgusta 2018 možno kupiti za 150 dolarjev. Na računalnik jo lahko priključimo s priključkom USB 2.0 in ne potrebuje posebnega napajalnika. Podpira operacijske sisteme Linux, Windows in Android. Kamera RGB snema v ločljivosti 1280 x 720 s 30 FPS. Kamera IR deluje na razdalji od 0,6 do 8 metrov, pri čemer je optimalna razdalja od 0,6 do 5 metrov. Globinska slika, ki jo ustvari Astra Pro, je v ločljivosti 640 x 480 s 30 FPS. Vidno polje je 60 stopinj vodoravno in 49.5 stopinj navpično [10]. Globinska slika nastane po enakem postopku kot pri Kinectu.



Slika 2.2: Senzor Astra Pro.

2. 1. 3 Kinect 2

Kinect za Xbox One (slika 2.3) je nadgrajena verzija Kinecta, ki so jo začeli prodajati novembra 2013 in je takrat stal 199 dolarjev. Oktobra naslednjega leta je prišla v prodajo verzija za operacijski sistem Windows, imenovana Kinect 2, ki se je razlikovala samo po tem, da je bil dodan vmesnik z USB 3.0 in napajanjem. Skupaj z verzijo za operacijski sistem Windows je bila izdana tudi nova verzija orodij za razvijalce.

A že po pol leta, aprila 2015, je Microsoft umaknil to verzijo Kinecta s prodajalnih polic in svetoval razvijalcem, naj namesto nje uporabijo verzijo Kinect za Xbox One z adapterjem USB. Zaradi neuspešnosti Kinecta za Xbox One je Microsoft 25. oktobra 2017 objavil prekinitve njegove prodaje in izdelave [11].

Tudi Kinect 2 je bil primarno zasnovan kot igralni pripomoček. Lahko sledi šestim ljudem pod 60 % večjim kotom, če ga primerjamo s Kinectom. Zazna lahko igralčev srčni utrip, težo, pozicijo in orientacijo 25 različnih sklepov, težo na vsaki okončini in igralčev izraz na obrazu.

Kinect 2 uporablja tehnologijo *time-of-flight* in s tem omogoča bolj natančne meritve kot predhodnik. Razdaljo ugotovi glede na to, koliko časa je žarek potoval od senzorja do predmeta in nazaj. Da lahko opravlja meritve, mora biti senzor natančen na 1/10.000.000.000 sekunde. Iz dobljenih meritev Kinect 2 zgradi matriko vrednosti. Vsaka vrednost v matriki predstavlja razdaljo do zaznanega predmeta. Čeprav je še vedno občutljiv na sončno svetlobo, je učinek manjši, kot pri njegovem predhodniku [5].



Slika 2.3: Kinect 2.

2. 1. 4 URG-04LX

Senzor URG-04LX (slika 2.4) je izdelalo Japonsko podjetje Hokuyo, ki se ukvarja z razvojem avtomatizacije. Njihovi izdelki so integrirani v napravah na različnih področjih, kot so npr. tekstilna industrija, transportni sistemi, montažni stroji, polprevodniki in naprave LCD. Cena tega senzorja je 1900 dolarjev [9].

URG-04LX ima dobro natančnost in velik vidni kot, kar omogoča dobro premikanje robotov po neznanem okolju. Senzor je precej majhen. Velik je samo 50 mm x 50 mm in visok 70 mm, kar uporabniku omogoča veliko svobodo pri načrtovanju robota. Za delovanje potrebuje napetost 5 V pri toku 500 mA. Na računalnik ga lahko povežemo preko vmesnika USB ali RS-232C. Senzor lahko deluje pri temperaturah od -10°C do 50°C in do 85 % vlage v zraku. Načrtovan je bil samo za notranjo uporabo, zato ni zagotovljeno, da bo enako dobro deloval v zunanjih prostorih. Meritve lahko opravlja od 0,06 m do 4,1 m pri kotu 240° . Za razliko od ostalih senzorjev, ki smo jih uporabljali, meri samo v eni ravnini. Merjenje razdalje deluje po principu *LiDAR*. Senzor odda svetlobni žarek in meri odboj svetlobe. Razdaljo izračuna glede na pretečen čas in razliko v valovni dolžini. Vrednosti, ki nam jih vrnejo meritve z URG-04LX, predstavljajo le eno ravnino. Zaradi tega je URG-04LX zelo občutljiv na odsevne površine.



Slika 2.4: Senzor URG-04LX.

2. 1. 5 Primerjava senzorjev

V tabeli 1 primerjamo senzorje neposredno med seboj. URG-04LX za razliko od ostalih treh senzorjev meri podatke samo v eni ravnini in ne v prostoru. Čeprav URG-04LX nima možnosti zaznavanja v višino, je njegovo vodoravno vidno polje večje od ostalih senzorjev.

Astra Pro ima z 8 metri dometa najdaljše vidno polje medtem, ko ima URG-04LX s 4,1 metri najkrajši domet.

	Kinect	Astra Pro	Kinect 2	URG-04LX
Domet	0,7 - 6 m	0,6 - 8 m	0,5 - 6 m	0,06 - 4,1 m
Ločljivost	640 x 480	640 x 480	512 x 424	približno 0,36°
Vodoravno vidno polje	57°	60°	70°	120°
Navpično vidno polje	43°	49,5°	60°	/
Princip delovanja	<i>Structured light</i>	<i>Structured light</i>	<i>Time of flight</i>	<i>LiDAR</i>
Napetost	12 V	5 V	12 V	5 V
Tok	1080 mA	370 mA	1250 mA	500 mA
Teža	1274 g	300 g	1451 g	160 g

Tabela 1: Primerjava lastnosti globinskih senzorjev.

2. 2 Roboti

Uporabljali smo predelano Roomba 531, ki se imenuje TurtleBot, in Pioneer 3-DX. Spremembe, ki so bile narejene na Roombi, niso vplivale na hitrost ali natančnost meritev odometrije, ampak so omogočile dodajanje dodatnih senzorjev. Lastnost, ki nas je zanimala, je bila natančnost odometrije, ki nam pove, kolikšno pot je robot prevozil. Ker je Pioneer 3 novejši in dražji, pričakujemo boljše rezultate kot pri TurtleBotu.

Odometrija je uporaba podatkov, ki jih dobimo iz senzorjev za gibanje, za ocenjevanje spreminjanja položaja opazovanega predmeta. V robotiki se uporablja za ocenjevanje položaja robota relativno glede na njegovo začetno pozicijo. Pri robotih se podatki merijo s senzorji, ki merijo obrate koles.

2. 2. 1 TurtleBot

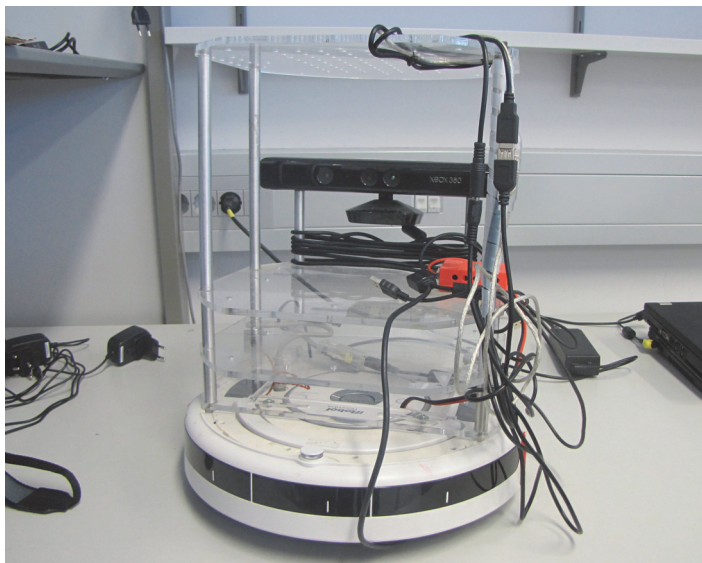
Cenejši od robotov, ki smo jih uporabljali, je bil TurtleBot (slika 2.6), ki je nekoliko spremenjena različica robota Roomba 531 (slika 2.5). To je v osnovi robot, ki je namenjen sesanju hiše. Ima obliko valja s polmerom 17 cm in višino 9,144 cm [8]. Originalna Roomba

ima že vgrajen infrardeč senzor, ki se uporablja za zaznavanje virtualnih zidov, ki jih uporabljamo za omejevanje prostora, v katerem lahko robot deluje. Brez polnjenja lahko deluje okoli tri ure. Po tem času jo je potrebno napolniti, pri čemer lahko sama najde pot do postaje za napajanje.

Različica TurtleBota, ki je bila predelana na fakulteti, je nekoliko drugačna od običajnega TurtleBota. TurtleBot doda Roombi možnost priklopa dodatnih senzorjev in možnost daljšega delovanja brez polnjenja. Pri spremembi se odstrani predel, kjer se nahaja sesalna enota in se zamenja z dodatno baterijo, ki služi za napajanje dodatnih senzorjev in daljši čas delovanja. Izkazalo se je, da lahko pride, zaradi nenadnih pospeškov, do nepravilnega gibanja in se je zato na zadnjem delu robota dodalo kolo za stabilizacijo robota. Robota je možno upravljati s prenosnim računalnikom, ki se nanj priključi s priključkom USB/UART [1].



Slika 2.5: Zgornji del originalnega robota Roomba [8].



Slika 2.6: Roomba po predelavi - TurtleBot.

2. 2. 2 Pioneer 3-DX

Pioneer 3-DX (slika 2.7) je zasnovan za uporabo v notranjih prostorih. Opremljen je z dvema kolesi, ki delujeta z diferencialnim pogonom. Motorji z diferencialnim pogonom imajo možnost, da se vsako kolo vrti različno hitro, kar nam omogoča boljši nadzor nad robotom. S 381 mm v širino in 455 mm v dolžino zasede več prostora kot TurtleBot, a je z 237 mm v višino nižji od njega. Ogrodje robota je narejeno iz aluminija in tehta 9 kg. Sposoben je tovoriti breme do 17 kg. Deluje lahko od 8 do 10 ur v primeru, da nima nobenih dodatkov in uporablja tri baterije.

Baterije, ki jih uporablja, delujejo pri električni napetosti 12 V in imajo kapaciteto 7,2 Ah. Zamenjati jih je možno brez ugašanja robota. Največja hitrost, ki jo lahko doseže naprej ali nazaj je 1,2 m/s, obrača se lahko s hitrostjo 300 stopinj/s. Najvišja ovira, čez katero robot še lahko zapelje, je 2,5 cm, pelje lahko čez 5 centimetrske razpoke in v hrib s 25 % nagibom. Opremljen je tudi z vgrajenim sonarjem in mikrokontrolerjem. Ta ima 32 digitalnih vhodov, 8 digitalnih izhodov, 7 analognih vhodov in 3 serijska vrata [12].



Slika 2.7: Pioneer 3-DX.

K robotu spada tudi programska oprema:

- ARIA - ogrodje za kontroliranje in prejemanje podatkov vseh naprav znamke MobileRobots. Poleg tega vsebuje tudi odprtokodno infrastrukturo in pripomočke, s katerimi lahko napišemo svoje programe za kontroliranje robotov, podporo za internetno povezavo in podporo za programiranje omrežja odjemalec-strežnik.

- MobileSim - odprtokodni simulator za naprave MobileRobots.
- MobileEyes - grafični uporabniški vmesnik za daljinsko upravljanje in nadzor robota.
- Mapper 3-Basic - orodje za ustvarjanje in urejanje zemljevidov za uporabo z ostalo opremo v tem paketu.
- SONARNL - programska oprema za lokalizacijo in navigacijo s sonarjem.

2. 3 Programska oprema

2. 3. 1 ROS

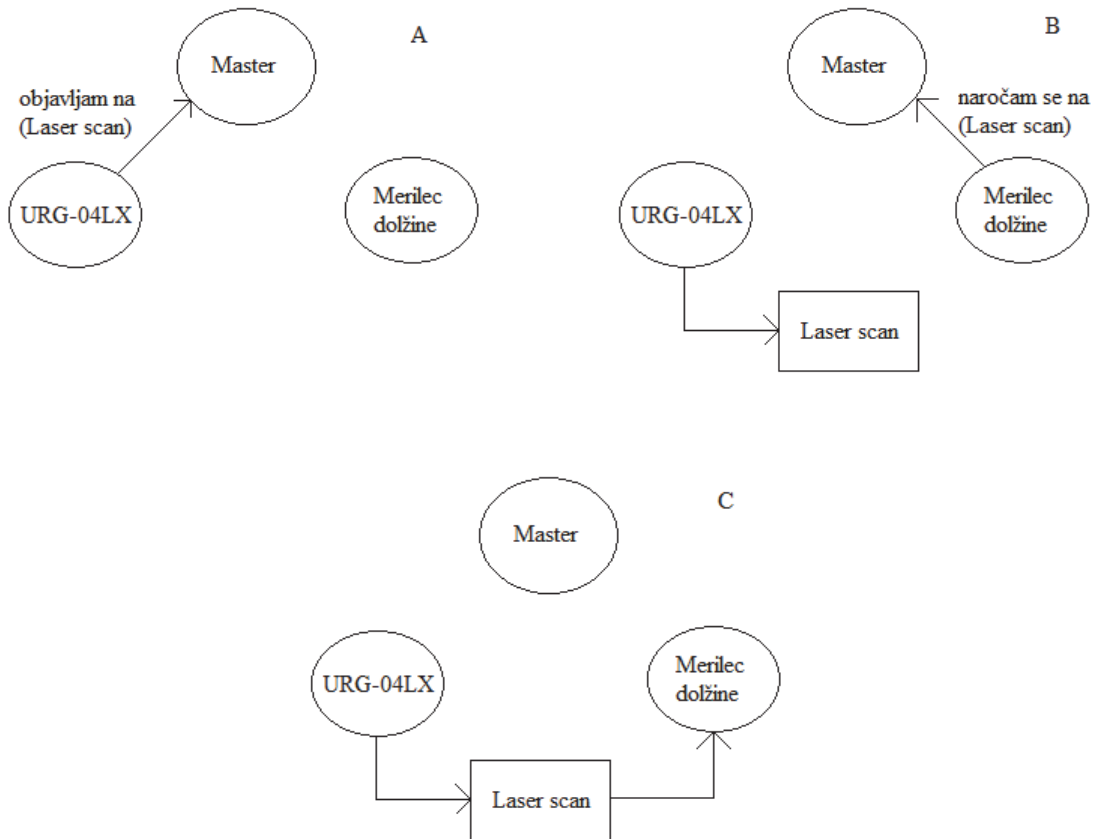
Pri meritvah smo uporabljali *Robot Operating System* (ROS), ki je prilagodljivo okolje za pisanje programske opreme za robote. To je zbirka orodij, knjižnic in konvencij, ki poskuša poenostaviti delo pri pisanju programov za čim več različnih robotov [13].

ROS deluje po principu pošiljanja sporočil *izdajatelj/naročnik* (*publisher/subscriber*). Programi, ki delujejo v okviru ROSa, so predstavljeni kot *vozlišča* (*nodes*). Vsako *vozlišče* ima lahko več *izdajateljev* (*publisher*) in *naročnikov* (*subscriber*). *Izdajatelj* pošlje sporočilo na določeno *temo* (*topic*) in ne *naročniku* neposredno. Pri tem mu je vseeno, ali kateri *naročnik* gleda to *temo*. Podobno *naročniki* gledajo *teme* za sporočila, pri čemer jim je vseeno ali obstaja kakšen *izdajatelj*, ki na tisto *temo* pošilja sporočila. Takšen sistem omogoča, da niso vsa *vozlišča* na istem računalniku in nam omogoča preprosto komunikacijo med računalniki. Eno *vozlišče* ima lahko več *izdajateljev* in *naročnikov*. Poleg *vozlišč* je potrebno še *glavno vozlišče* (*master*), ki poskrbi, da se lahko *vozlišča* povežejo na določeno *temo* [14]. Na sliki 2.8 je prikazan primer komunikacije med *vozlišči* in *glavnim vozliščem*.

Pri poskusih z roboti smo uporabljali različico ROS Kintetic, ki je bila izdana 23. 5. 2016 in je bila razvita za Ubuntu 16.04. Z nekaj popravki jo je možno uporabljati tudi na drugih operacijskih sistemih. Ob namestitvi ROS Kintetic se naložijo knjižnice in orodja za uporabo TurtleBota, kar omogoča uporabo Roombe in Kinecta z ROSom. Poleg tega smo potrebovali še knjižnice za uporabo Kinect 2, Astra Pro, URG-04LX in Pioneer 3-DX.

- Za Kinect 2 smo uporabili knjižnice *iai-kinect2*, katerih avtor je Thiemo Wiedemeyer in so javno dostopne na https://github.com/code-iai/iai_kinect2 [6].
- Za Astro Pro smo uporabili knjižnice *ros_astra_camera*, katerih avtor je Tim Liu in so dostopne na https://github.com/orbbec/ros_astra_camera.

- Za URG-04LX smo uporabili knjižnice `urg_node`, katerih avtorja sta Chad Rockey in Mike O'Driscoll in so dostopne na https://github.com/ros-drivers/urg_node.
- Za Pioneer 3-DX smo uporabili knjižnice ROSARIA, katerih avtor je Srećko Jurić-Kavelj in so dostopne na <https://github.com/amor-ros-pkg/rosaria>.



Slika 2.8: (a) Vozlišče URG-04LX komunicira z glavnim vozliščem in mu pove, da objavlja na temo Laser scan. (b) Vozlišče Merilec dolžine komunicira z glavnim vozliščem in mu pove, da se naroča na temo Laser scan, med tem vozlišče URG-4LX že objavlja na to temo. (c) Vozlišče URG-04LX objavlja na temo Laser scan in vozlišče Merilec dolžine bere iz nje.

2.3.2 Navigacija robota

Za navigacijo robota smo uporabili pakete, ki so že bili implementirani v ROSu.

Za gradnjo zemljevida smo uporabili metodo SLAM, ki je implementirana v paketu `slam_gmapping`. Ta paket uporablja za grajenje mape filtre delcev *Rao-Blackwell*. Glavna

ideja te metode je oceniti posteriorno verjetnost $p(x_{1:t} | z_{1:t}, u_{0:t})$ za možne poti robota. Ocena je izračunana na podlagi podatkov senzorjev $z_{1:t}$ in podatkov odometrije $u_{0:t}$.

S posteriorno verjetnostjo o možnih poteh robota lahko izračunamo posteriorno verjetnost zemljevida in možne poti robota:

$$p(x_{1:t}, m | z_{1:t}, u_{1:t-1}) = p(m | x_{1:t}, z_{1:t}) * p(x_{1:t} | z_{1:t}, u_{0:t})$$

Posteriorno verjetnost $p(m | x_{1:t}, z_{1:t})$ zemljevida je možno rešiti analitično, če imamo podatke o $z_{1:t}$ in $x_{1:t}$. Za oceno posteriorne vrednosti $p(x_{1:t} | z_{1:t}, u_{0:t})$ mapiranje *Rao-Blackwell* uporabi filtriranje delcev, kjer je vsak zemljevid povezan z vsemi vzorci. Vsak zemljevid je zgrajen na podlagi podatkov $z_{1:t}$ iz senzorja in poti robota $x_{1:t}$ [3].

Za lokalizacijo robota smo uporabljali prilagodljivo lokalizacijo *Monte Carlo* oziroma *Adaptive Monte Carlo Localization (AMCL)*, ki je implementirana v paketu *navigation*.

Lokalizacija *Monte Carlo* je postopek lokalizacije, pri katerem predvidevamo položaj robota glede na zbirko N naključno izbranih in uteženih točk. Vsaka točka predstavlja položaj robota, pri čemer ima podano še verjetnost p , da je ta položaj pravilen. Ob premiku robota se vsaka nova točka generira iz naključno izbranih točk prejšnje zbirke, pri čemer imajo večjo verjetnost izbire, če imajo večjo vrednost p . Pri tem postopku se upošteva samo premik robota, zato se skozi čas povečuje negotovost postavitve robota. Pri tem pomagajo meritve senzorjev, saj lahko z njimi pravilno utežimo točke, s katerimi ocenjujemo položaj. Ob vsakem novem generiranju točk jih imamo enako število. To pa ni potrebno, zato se je razvila prilagodljiva lokalizacija *Monte Carlo*. Ko poskušamo robota lokalizirati prvič, potrebujemo veliko več točk, kot v kasnejših korakih, saj na začetku ne vemo o poziciji robota nič. *AMCL* združi generiranje novih točk na podlagi premika robota in uteževanje točk na podlagi meritev senzorja v en sam korak. Tako se lahko ta postopek ustavi, če med generiranjem novih točk seštevka verjetnosti p generiranih točk prekorači prag t . Ko prekoračimo prag, vemo da imamo dovolj točk, da lahko natančno ocenimo položaj robota in tako ni potrebno nadaljnje generiranje točk [2].

Določanje poti robota na zemljevidu je implementirano v paketu *navigation*. Najprej se zgradi zemljevid, na katerem se določijo območja, kjer robot ne sme voziti. Za določanje poti, po kateri mora robot voziti, se izvede *Dijkstrov* algoritem, pri čemer predvideva, da je robot okrogel. Algoritem preveri vse možne postavitve robota med začetno in končno točko ter na podlagi tega najde pot, po kateri mora robot voziti. Na podlagi izbrane poti začne robot prejemati ukaze za premikanje [15].

3 Eksperimentalna evalvacija

Meritve, ki smo jih opravljali, smo razdelili na tri različne kategorije:

1. Meritve s senzorji.
2. Meritve z roboti.
3. Meritve z roboti in senzorji.

Meritve smo razdelili na ta način zato, da smo lahko ugotovili natančnost posameznih senzorjev in robotov ter tako lahko predvidevali, katera kombinacija bo pokazala najbolj natančne rezultate.

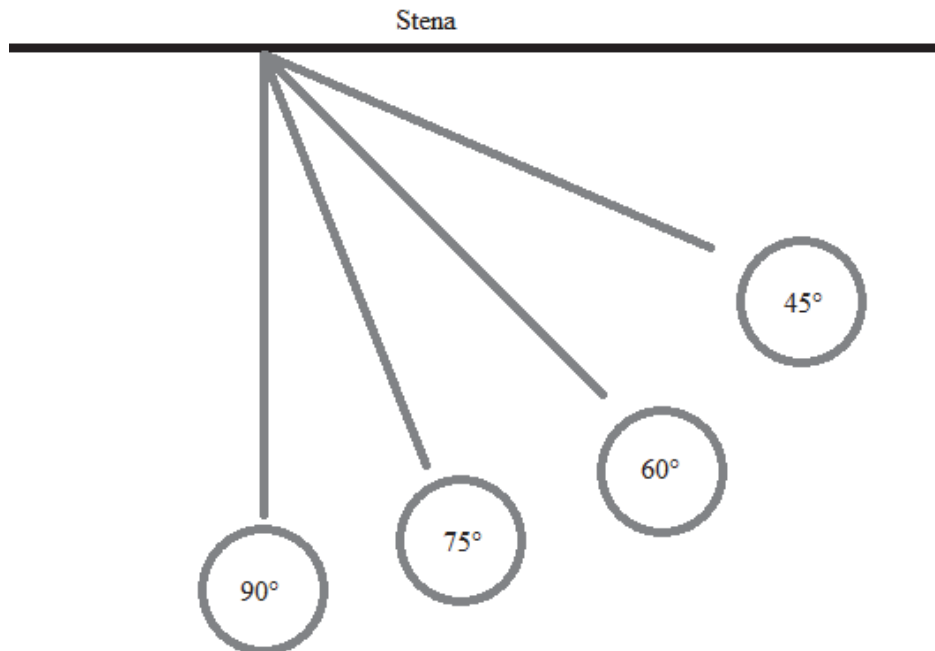
3.1 Evalvacija globinskih senzorjev

3.1.1 Postavitev eksperimentov

Pri senzorjih smo najprej opravili meritve, pri katerih smo primerjali delovanje senzorjev pri različnih razdaljah in kotih od stene. Eksperimente smo opravili pri 100, 150, 200, 250 in 300 centimetrov od stene in pod koti 90° , 75° , 60° , 45° in 30° . Postavitev pod različnimi koti je vidna na sliki 3.1. Čeprav Kinect, Astra Pro in Kinect 2 lahko opravljajo meritve v prostoru, smo meritve omejili na eno ravnino, ker je tako lažje primerjati rezultate eksperimentov. Poleg tega je URG-04LX, ki smo ga uporabljali, zmožen meritve opravljati samo v eni ravnini.

Vsak eksperiment smo opravljali 10 sekund. V tem času so senzorji opravili med 100 in 200 meritev odvisno od senzorja in hitrosti računalnika, na katerega je senzor priključen. Astra Pro in Kinect sta opravljala meritve hitreje kot URG-04LX in Kinect 2. Pred začetkom eksperimenta smo senzor postavili na prej omenjene razdalje in ustrezen kot. Ko smo bili prepričani, da je senzor pravilno postavljen, smo začeli opravljati meritve. Med vsakim eksperimentom je računalnik zapisoval meritve v datoteko za kasnejšo obdelavo. Pri senzorjih Kinect, Kinect 2 in Astra Pro je vsaka meritev vračala 40 vrednosti, pri URG-04LX pa 20 vrednosti. Število shranjenih vrednosti, ki predstavljajo razdaljo med senzorjem in steno, je odvisno od ločljivosti senzorja. V tem primeru ima URG-04LX manjšo ločljivost kot ostali trije senzorji. Ker pa so bile te vrednosti izmerjene pod drugačnimi koti (slika 3.2), smo jih

morali po opravljenem eksperimentu projicirati na premico, ki potuje pravokotno na senzor. Brez tega koraka ne bi mogli izračunati povprečne vrednosti in standardnega odklona meritev.

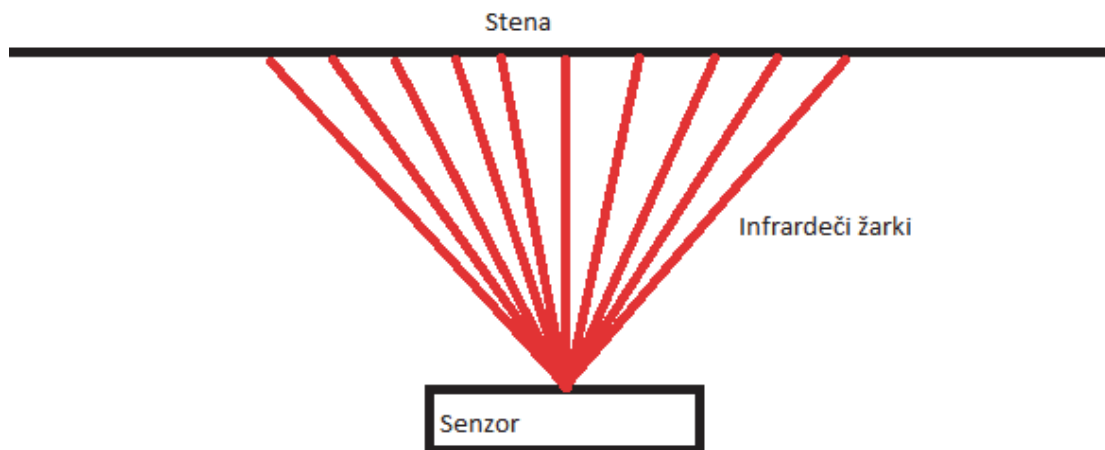


Slika 3.1: Prikaz postavitve senzorjev pod različnimi koti.

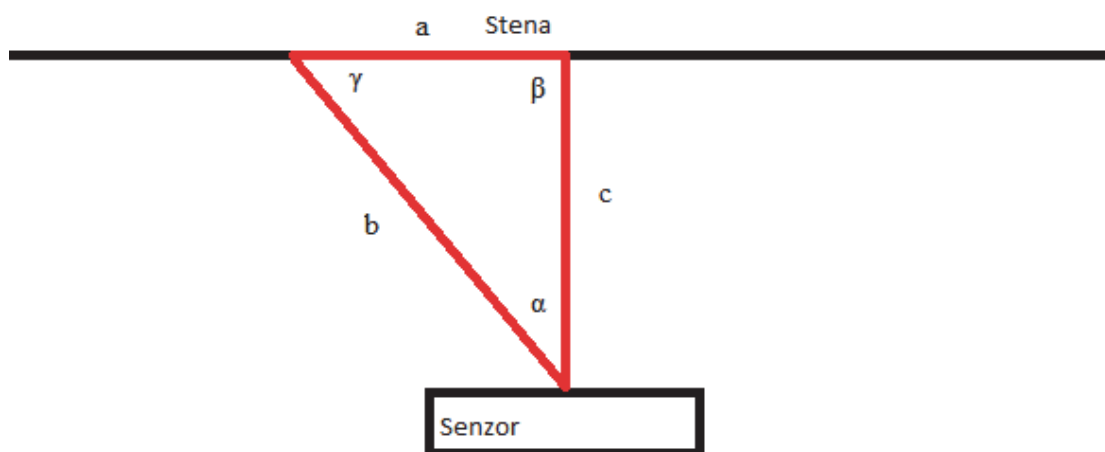
Ob meritvi smo poznali kot, pod katerim je žarek izstopil iz sensorja (kot α na sliki 3.3), razdaljo, ki jo je senzor dobil od žarka (stranica b na sliki 3.3) in kot, pod katerim je bil senzor usmerjen na steno (kot β na sliki 3.3). S temi podatki je možno izračunati vse stranice trikotnika, pri čemer ena izmed njih leži na premici, ki potuje pravokotno na senzor (stranica c na sliki 3.3). Po končanih eksperimentih smo izračunali standardni odklon in povprečno vrednost vseh meritev v istem eksperimentu in jih primerjali med seboj.

Za izračun projekcije smo uporabili sinusni izrek:

$$\frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{c}{\sin \gamma} = 2 * R$$



Slika 3.2: Skica potovanja žarkov med senzorjem in steno.



Slika 3.3: Prikaz uporabljenega trikotnika za izračun razdalje med steno in senzorjem.

Evalvirali smo tudi, kako dobro se senzorji obnesejo na različni svetlobi. To smo poskusili pri treh različni pogojih:

1. Pri dnevni svetlobi, ki prihaja skozi okno.
2. Pri dnevni svetlobi, ki prihaja skozi okno in pri prižganih lučeh.
3. V zatemnjeni sobi z ugasnjenimi lučmi.

Kljub temu, da bi naj bili vsi senzorji občutljivi na svetlobo, ni bilo vidnih razlik med rezultati posameznih eksperimentov.

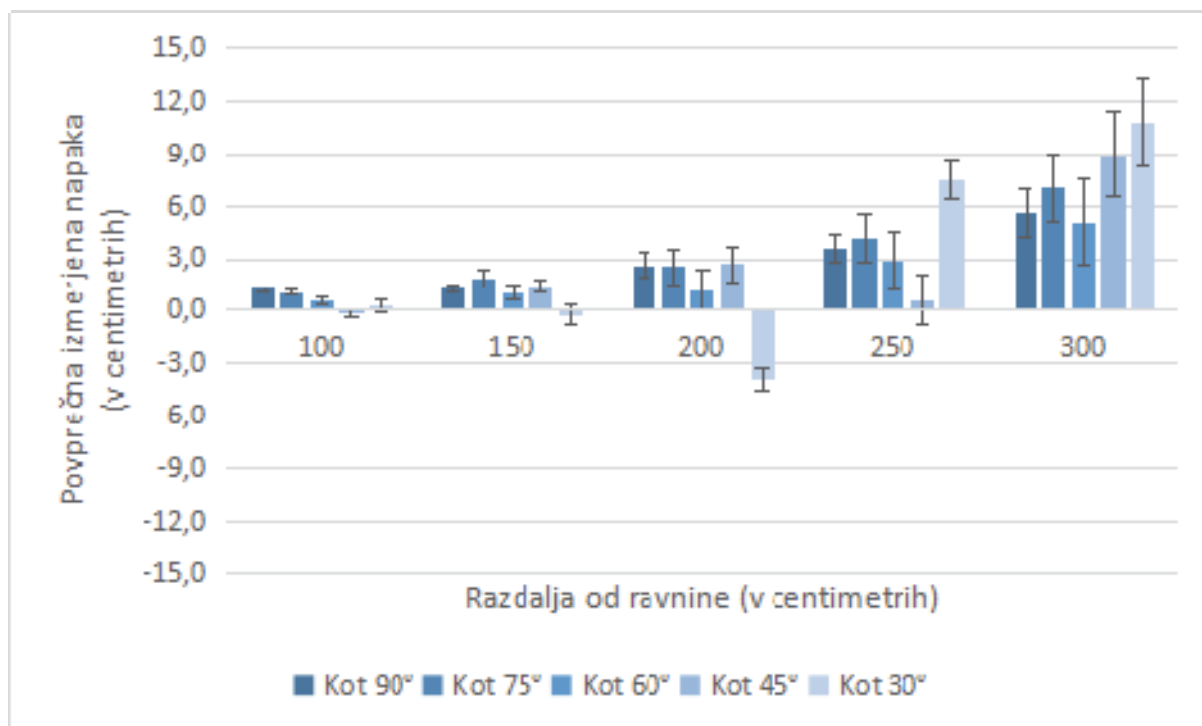
Poleg različne osvetljenosti smo preizkusili tudi delovanje senzorjev na ogledalu. Tudi tukaj so vsi senzorji vračali podobne rezultate. Razdalja, ki so jo izmerili, je bila enaka razdalji od sensorja do ogledala in od ogledala do predmeta, ki je bil odsevan v ogledalu. Poleg tega smo opazili, da je URG-04LX veliko bolj občutljiv na površine, ki odbijajo sončno svetlobo. Zaradi tega smo morali poskrbeti, da se površina, na kateri smo opravljali eksperimente, ni bleščala.

3. 1. 2 Eksperimentalni rezultati

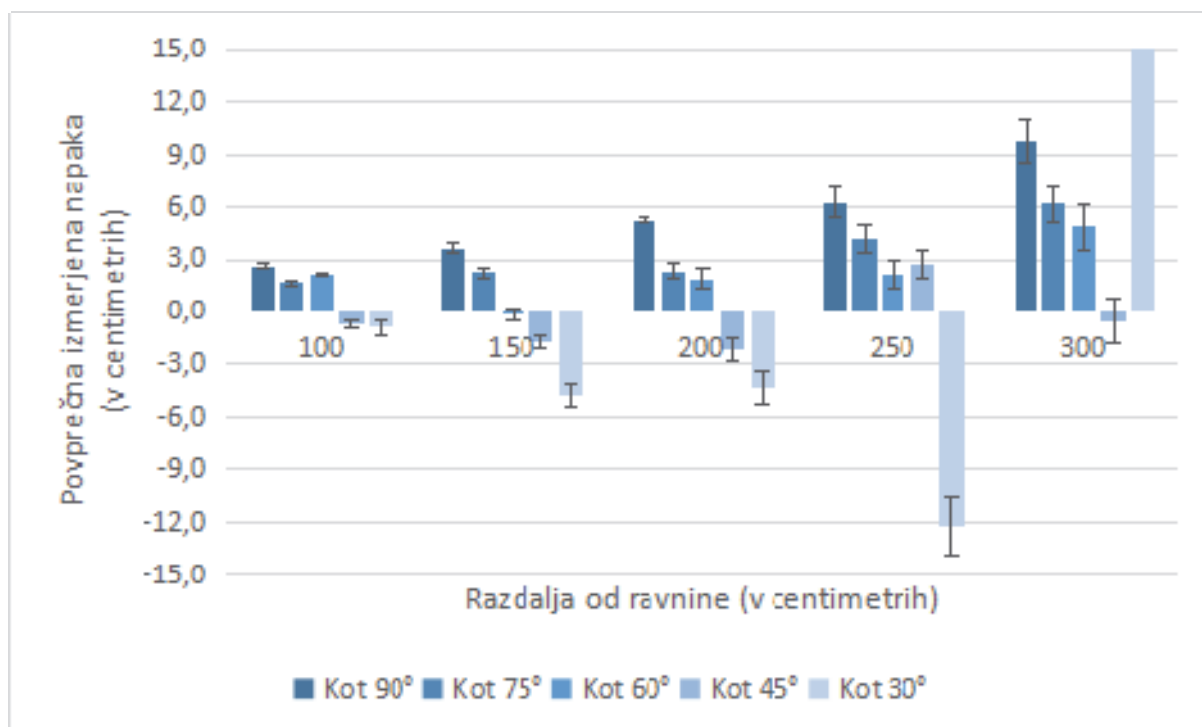
Na slikah v tem podpoglavju predstavljamo rezultate eksperimentov v obliki stolpčnih grafov s prikazom standardnega odklona, pri čemer stolpci prikazujejo povprečno izmerjeno napako.

Izkazalo se je, da Kinect dobro meri razdaljo, še posebej pri razdaljah pod dvema metroma. Iz grafov na sliki 3.4 je razvidno, da pri meritvah pod kotom 90° nastane najmanjša napaka. Pri kotih 90° je največja povprečna napaka 5,59 cm, ki se je pojavila pri eksperimentu na razdalji 300 cm. Pri razdalji 100 in 150 cm pod tem kotom je bila povprečna napaka 1,2 cm. Istočasno se vidi, da je standardni odklon meritev pod tem kotom v glavnem manjši, kot pri ostalih kotih. To pomeni, da so meritve bolj natančne. Pri vseh razdaljah, razen pri 300 cm, je standardni odklon manjši od enega centimetra. Pri kotih 75° , 60° in 45° se standardni odklon povečuje z vsako povečano razdaljo približno enako hitro, čeprav se napake v izmerjeni razdalji ne opazi. To nam pove, da je, kljub dovolj natančnemu rezultatu, v meritvah veliko nihanj. Pri meritvah pod kotom 30° so bili rezultati najmanj natančni in tudi standardni odklon je bil daleč največji pri 100 cm. Kljub temu pa je standardni odklon naraščal počasneje, kot pri ostalih kotih in je bil pri razdaljah 200 in 250 cm manjši kot pri kotih 75° , 60° in 45° .

Rezultati meritev za senzor Astra Pro so prikazani na sliki 3.5. Senzor meri razdalje natančno pri vseh kotih razen pri 30° in 90° . Pri tem je pri kotu 90° standardni odklon nizek do vključno razdalje 200 cm. Od 100 do 200 cm je standardni odklon manj kot 0,3 cm. Pri teh razdaljah povprečna razdalja narašča od 2,6 cm pri 100 cm do 5,18 cm pri 200 cm. Standardni odklon narašča hitreje pri kotih 75° , 60° in 45° med 100 in 200 cm, kot pri kotu 90° , a se ta izenači pri razdalji 250 cm. Pri kotu 30° se, v primerjavi z ostalimi koti, pokažejo bistvene razlike v izmerjeni dolžin. Med 100 in 250 cm je izmerjena razdalja bistveno manjša od dejanske postavitve sensorja od stene. Pri razdalji 300 cm pa je ta vrednost večja od postavitve sensorja od stene. Največja napaka 41,5 cm je bila izmerjena na razdalji 300 cm pod kotom 30° .



Slika 3.4: Povprečna izmerjena napaka eksperimentov s Kinectom.



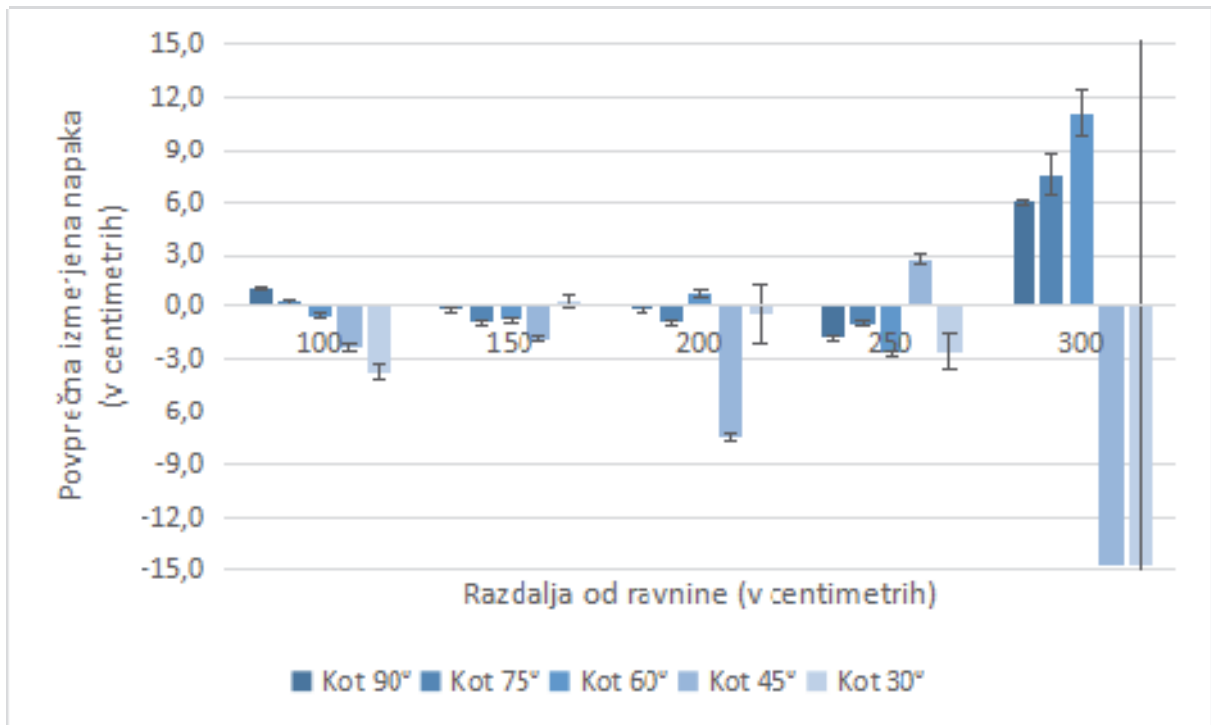
Slika 3.5: Povprečna izmerjena napaka eksperimentov z Astro pro.

Rezultati Kinecta 2 so prikazani na sliki 3.6. Senzor je vračal točne rezultate pri vseh preverjenih kotih z nekaj manjšimi napakami pri določenih eksperimentih. Težave so se pojavile pri kotu 30° , kjer se je standardni odklon močno povečal pri razdalji 200, 250 in 300 cm. Pri razdalji 300 cm se je celo zgodilo, da je bilo odstopanje tako veliko, da se podatkov ne vidi v celoti. Pri kotu 30° na razdalji 300 cm je bila izmerjena napaka -180,6 cm in pri kotu 45° je bila izmerjena napaka -40,1 cm. Napaka je ostala kljub temu, da smo eksperiment večkrat ponovili. Pri tem smo vsakič poskrbeli, da je bil senzor pravilno postavljen. Pri istem kotu je bil, pri razdaljah 100 in 250 cm, povprečni rezultat nekoliko manjši od dejanske postavitve senzorja. Napaka v meritvah se je pojavila tudi pri kotu 45° na razdalji 200 cm, kljub temu, da ni bilo razlike v standardnem odklonu pri tej razdalji v primerjavi z ostalimi meritvami.

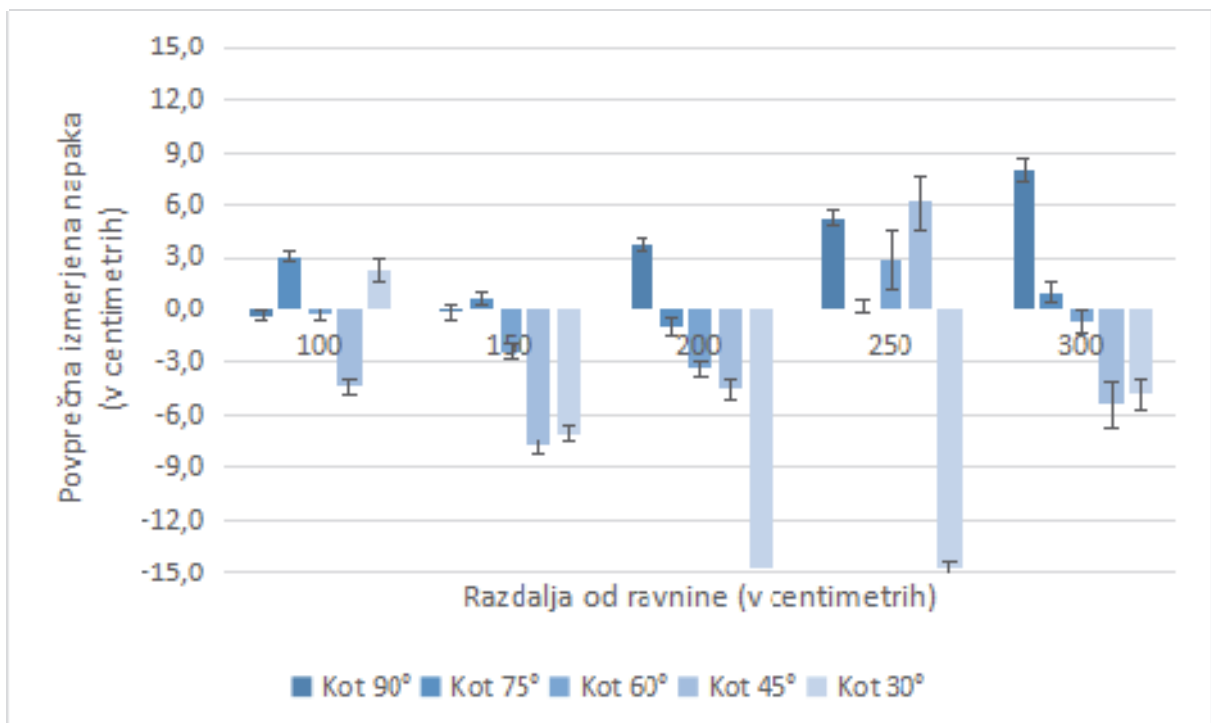
Rezultati URG-04LX so prikazani na sliki 3.7. Če smo pri Kinectu pričakovali nenatančne rezultate in dobili natančne rezultate, smo pri URG-04LX pričakovali točne rezultate, dobili pa smo, za njegov cenovni razred, netočne rezultate. Kljub temu je pri večini eksperimentalnih primerov standardni odklon manj kot en centimeter. Samo eksperimenti pri kotih 60° in 45° na razdalji 250 cm in eksperiment pri kotu 60° na razdalji 300 cm so pokazali skok nad enim centimetrom v standardnem odklonu. Največje napake v meritvah so se pojavljale pri kotu 30° pri razdaljah med 150 in 250 cm. Poleg tega so se napake pojavile tudi pri kotu 45° pri razdalji 100 in 150 cm. Največji napaki v meritvah povprečne vrednosti sta bili 1,7 oziroma 1,6 cm na razdaljah 200 in 250 cm pri kotu 30° . Najbolj točne meritve so bile pri kotu 75° , kjer je od 150 do 250 cm izmerjena povprečna napaka manj kot en centimeter.

Na sliki 3.8 je prikazana primerjava meritev vseh senzorjev pri kotu 90° . Na razdalji do 200 cm se je izkazalo, da sta Kinect 2 in URG-04LX najbolj točna senzorja. Pri tem ima Kinect 2 manjši standardni odklon. Pri 200 cm in več so meritve Kinecta 2 najbolj točne. Astra Pro se je pri tem kotu obnesla najslabše z netočnimi meritvami in visokim standardnim odklonom na razdaljah pri 250 in 300 cm.

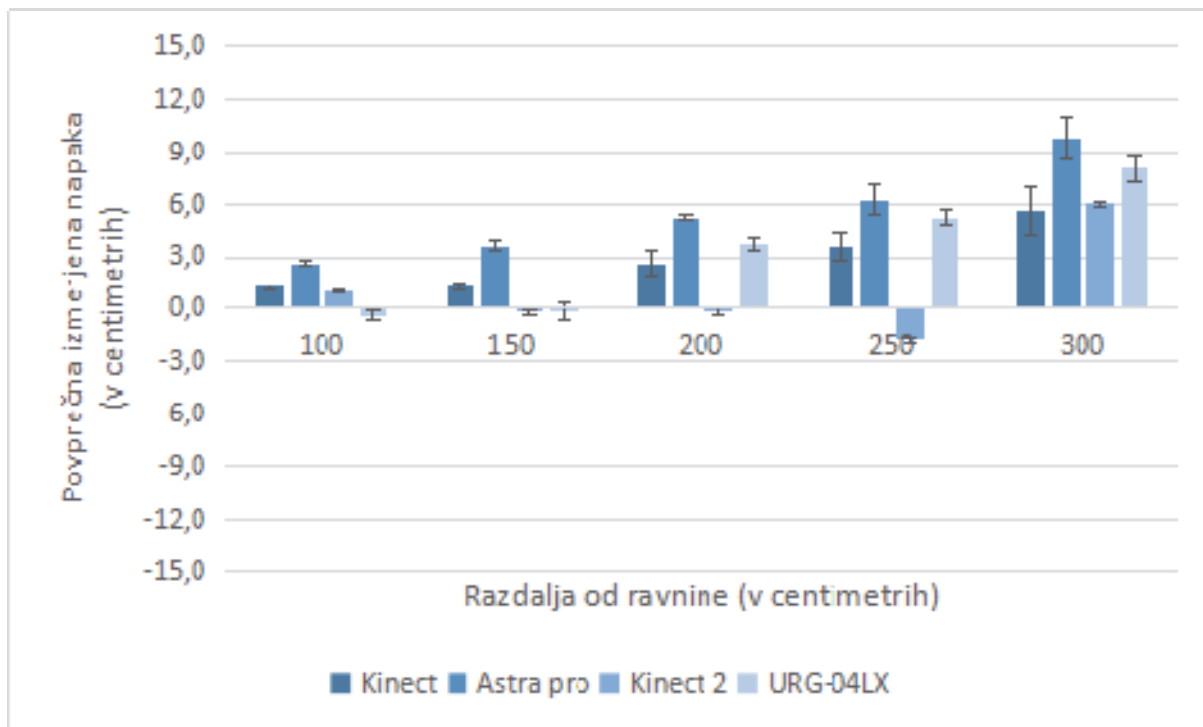
Na sliki 3.9 je prikazana primerjava vseh senzorjev pri kotu 60° . Do razdalje 200 cm se najbolj obnese senzor Kinect 2. Pri 200 cm se pojavi napaka v meritvah pri vseh senzorjih razen pri Kinectu 2. URG-04LX pri vseh razdaljah, razen pri 250 cm, nameri manj, kot je njegova postavitev od ravnine. Poleg tega se Kinectu pri tej razdalji močno poveča standardni odklon. Pri razdalji 250 in 300 cm je Kinect 2 manj natančen, pri čemer se mu standardni odklon močno poveča pri 300 cm.



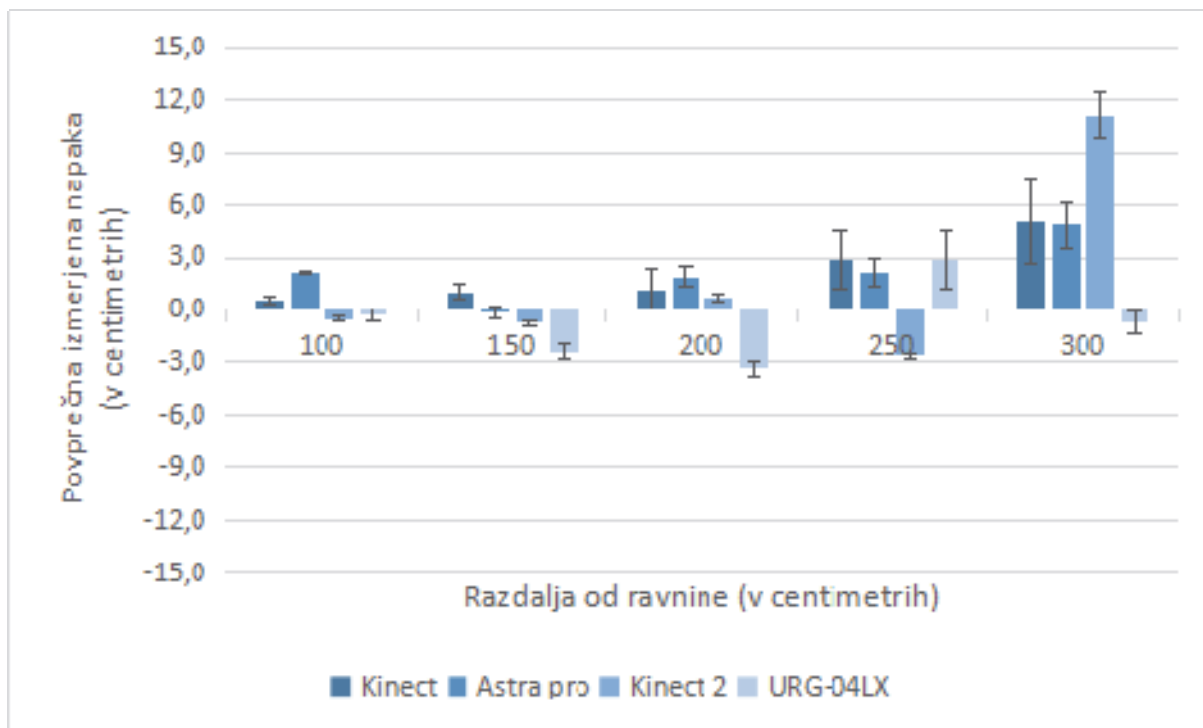
Slika 3.6: Povprečna izmerjena napaka eksperimentov s Kinectom 2.



Slika 3.7: Povprečna izmerjena napaka eksperimentov z URG-04LX.



Slika 3.8: Povprečna izmerjena napaka senzorjev pri kotu 90°.



Slika 3.9: Povprečna izmerjena napaka senzorjev pri kotu 60°.

Eksperimenti so pokazali, da je najbolj natančen senzor Kinect 2. Kljub presenetljivi natančnosti Kinecta pri povprečni vrednosti meritev, ima ta največji standardni odklon, kar pomeni, da lahko hitro pride do napak v meritvah, če senzor ni na miru ali ne more opravljati meritev dovolj dolgo. Astra Pro in URG-04LX sta se v glavnem obnesla slabše kot Kinect 2. To je bilo pričakovano za senzor Astra Pro, saj Kinect 2 uporablja bolj napredno tehnologijo za merjenje globine. To pa ne drži za URG-04LX. Presenetilo nas je, da je URG-04LX vračal slabše rezultate kot Kinect 2, čeprav je URG-04LX veliko dražji.

3. 2 Evalvacija premikanja robotov

Robota, s katerima smo opravljali eksperimente, sta predelana Roomba oziroma TurtleBot in Pioneer 3-DX. Z njima smo opravljali dva različna tipa eksperimentov:

1. Vožnja naravnost.
2. Vrtenje na mestu.

Pri eksperimentih smo uporabljali program, s katerim smo robotu podali ukaz, naj vozi z določeno hitrostjo naprej ali pa se obrača, dokler ne doseže zadanega cilja. Pri eksperimentu, kjer je robot vozil naravnost, smo preizkusili tri različne hitrosti: 0,1, 0,25, in 1 m/s. Preden je robot začel vožnjo smo njegov začetni položaj označili na tleh, da smo kasneje lahko prepotovano pot izmerili. Robot je prevoženo pot izračunal iz podatkov, ki jih je dobil iz odometrije. Vozil je še nekaj časa po tem, ko je dosegel cilj, ker se ni mogel takoj ustaviti. Robot je je posredoval dva podatka, ki jih je izračunal iz odometrije. Prvi je bila razdalja od začetka poti do cilja, ko je program prenehal pošiljati ukaze, drugi pa razdalja od začetka poti do dejanske točke zaustavitve robota.

3. 2. 1 Vožnja naravnost

Rezultati TurtleBota v primerjavi s Pioneerjem 3-DX so bili manj zanesljivi (pri enakih pogojih je vračal različne rezultate). Poleg tega se je pojavila še težava, da TurtleBot ni bil sposoben voziti popolnoma naravnost. Takrat ko je vozil s hitrostjo 0,1 in 0,25 m/s, je zavijal počasi v desno. Zavoj se je opazil šele, ko smo opravljali eksperimente pri dveh in treh metrih. Iz meritev, ki smo jih opravili, predvidevamo, da na vsak meter zavije 4° v desno. Še večji zavoj se je pojavil pri hitrosti 1 m/s, a v drugo smer. Zavoj je bil očiten že pri eksperimentih dolžine pol metra. Takoj, ko je robot prejel ukaz za premik, je poskušal speljati s podano hitrostjo, namesto da bi počasi pospeševal do nje. Z eksperimentiranjem smo ugotovili, da desno kolo pospeši do končne hitrosti hitreje kot levo, kar je povzročilo zavijanje. A tudi ko je

bila končna hitrost dosežena, je desno kolo še vedno delovalo hitreje in je robot počasi zavijal celo pot. Ta težava se pri Pioneerju 3-DX ni pojavila.

V tabelah 2 in 3 predstavlja prva vrstica (v tabelah 2 in 3: *dosežen cilj*) vsakega eksperimenta vrednost odometrije, ki je bila zabeležena, ko je robot dosegel zadani cilj. V drugi vrstici (v tabelah 2 in 3: *program*) je zabeležena vrednost odometrije, ko se je robot ustavil. V tretji vrstici (v tabelah 2 in 3: *izmerjeno*) je zabeležena vrednost, ki smo jo izmerili z metrom od začetnega položaja robota do končnega položaja robota. Pri merjenju meritev z metrom smo bili na milimeter natančni.

Pri rezultatih je bolj pomembno, da sta si čim bolj podobni vrednosti, ki jo je vrnil program, ko se robot ustavi (druga vrstica pri vsakem eksperimentu) in razdalja, ki smo jo izmerili z metrom (tretja vrstica pri vsakem eksperimentu). Iz rezultatov je razvidno, da je Pioneer 3-DX deloval bolj dosledno kot TurtleBot. Zanimivo je, da je bil Pioneer manj dosleden pri eksperimentih, kjer se je premikal s hitrostjo 0,1 m/s. Iz tabele lahko opazimo, da Pioneer pri hitrosti 1 m/s nadaljuje pot še kakšen meter od točke, kjer bi se moral ustaviti. To se zgodi, ker, za razliko od TurtleBota, počasi pospešuje in zavira, kar pomeni, da bo peljal dlje. To ni velik problem, saj se takemu obnašanju lahko prilagodimo. Ker je možno napako predvideti, lahko prilagodimo ukaze, da opravi tako dolgo pot, kot želimo.

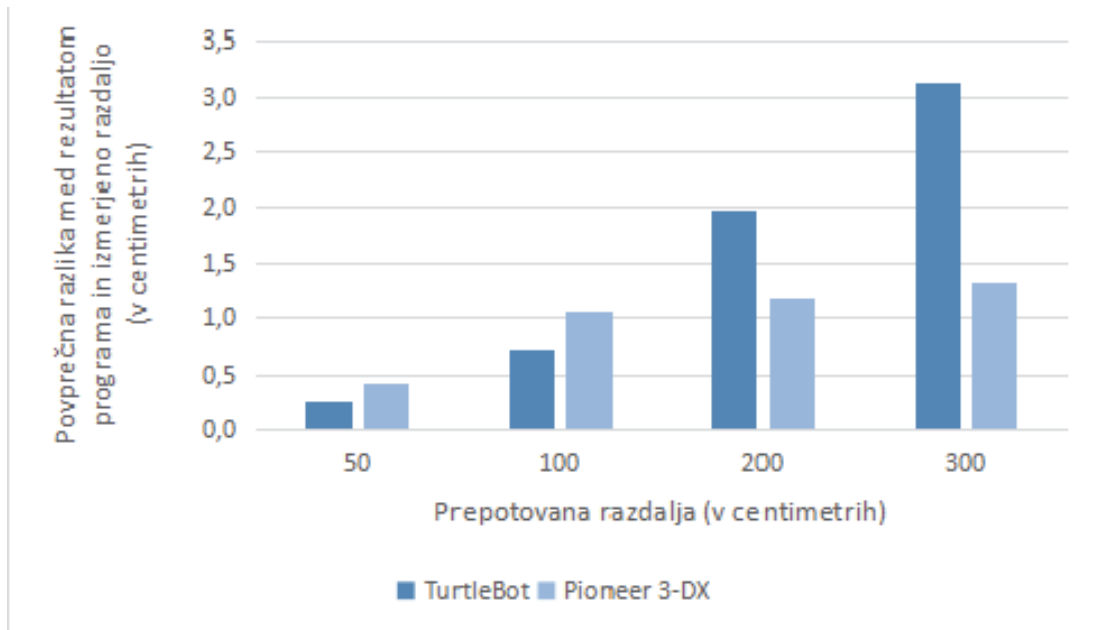
Na slikah 3.10, 3.11 in 3.12 je predstavljena povprečna razlika med prepotovano potjo, ki jo je nameril robot, in izmerjeno potjo pri različnih hitrostih. Pri vseh eksperimentih se je pokazalo, da je TurtleBot natančnejši pri krajših razdaljah, do enega metra. Pri večjih razdaljah pa je natančnejši Pioneer 3-DX. Pri tem je pri krajših razdaljah razlika med njima manjša, kot pri večjih razdaljah.

TurtleBot	hitrost (m/s)	dolžina (cm)	prvi eks. (cm)	drugi eks. (cm)	tretji eks. (cm)
dosežen cilj	0,1	50	51,2	50,08	50,1
program			54,7	53,5	53,4
izmerjeno			54,3	53,2	53,5
dosežen cilj	0,25	50	50,7	50,6	50,7
program			58,9	58,5	59,8
izmerjeno			58,9	58,7	60,0
dosežen cilj	1	50	50,5	54,1	50,4
program			62,1	66,2	61,5
izmerjeno			62,5	66,5	61,8
dosežen cilj	0,1	100	100,4	100,4	100,0
program			103,7	103,7	103,3
izmerjeno			102,6	102,7	102,2
dosežen cilj	0,25	100	100,0	101,6	102,0
program			107,4	109,8	109,9
izmerjeno			107,1	109,2	109,0
dosežen cilj	1	100	100,3	100,9	100,2
program			111,6	111,6	111,5
izmerjeno			111,6	111,3	111,3
dosežen cilj	0,1	200	200,7	200,4	200,4
program			204,2	204,4	203,7
izmerjeno			201,2	201,6	200,9
dosežen cilj	0,25	200	200,7	203,1	202,7
program			208,1	210,5	210,0
izmerjeno			206,0	207,8	207,9
dosežen cilj	1	200	202,9	202,2	202,4
program			210,6	210,3	211,7
izmerjeno			209,5	208,9	211,2
dosežen cilj	0,1	300	300,7	301,2	300,2
program			304,4	304,4	303,5
izmerjeno			299,7	299,8	299,0
dosežen cilj	0,25	300	300,3	302,5	300,1
program			308,2	309,9	307,5
izmerjeno			304,0	305,5	303,5
dosežen cilj	1	300	302,9	302,0	301,0
program			306,9	305,2	305,3
izmerjeno			307,8	305,7	305,6

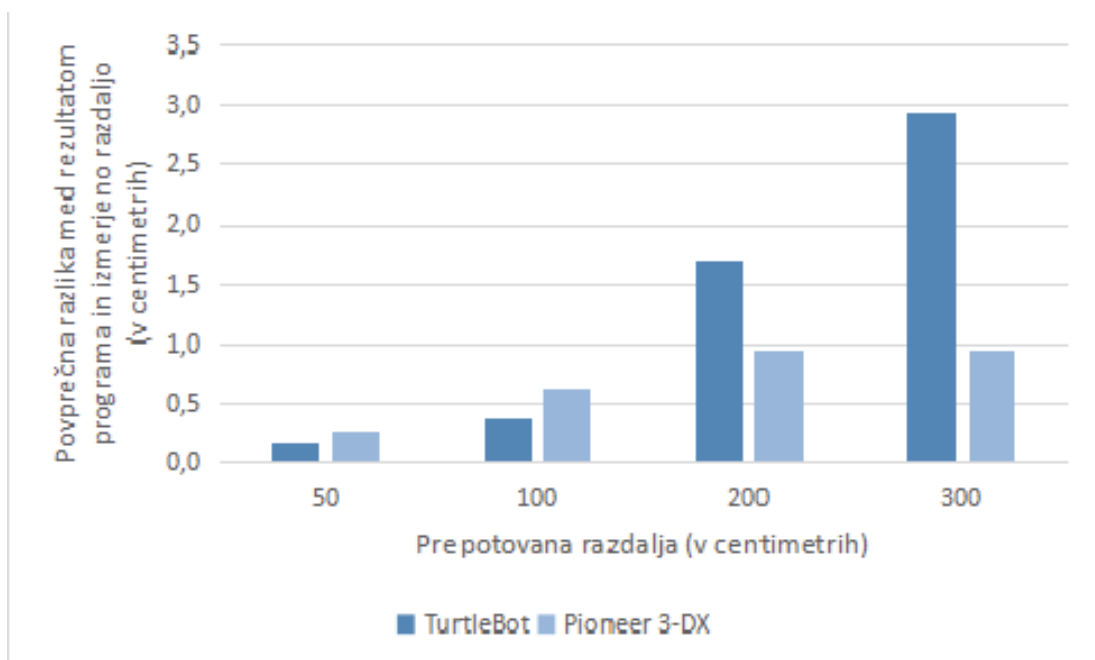
Tabela 2: Rezultati eksperimentov s TurtleBotom.

Pioneer 3-DX	hitrost (m/s)	dolžina (cm)	prvi eks. (cm)	drugi eks. (cm)	tretji eks. (cm)
dosežen cilj	0,1	50	50,9	50,0	50,9
program			54,6	53,8	54,7
izmerjeno			54,9	54,3	55,2
dosežen cilj	0,25	50	51,5	51,5	51,5
program			67,2	67,3	67,2
izmerjeno			67,4	67,6	67,5
dosežen cilj	1	50	52,9	52,9	52,9
program			131,0	131,0	130,0
izmerjeno			131,2	131,5	131,4
dosežen cilj	0,1	100	100,7	100,7	100,7
program			104,5	104,5	104,5
izmerjeno			105,6	105,5	105,5
dosežen cilj	0,25	100	101,4	101,4	101,4
program			117,2	117,1	117,1
izmerjeno			117,7	118,0	117,6
dosežen cilj	1	100	107,0	106,9	106,9
program			217,1	217,1	217,1
izmerjeno			217,7	218,0	217,9
dosežen cilj	0,1	200	200,3	200,3	200,3
program			204,0	203,0	204,1
izmerjeno			205,5	204,0	205,1
dosežen cilj	0,25	200	201,1	201,1	201,2
program			216,9	216,9	216,9
izmerjeno			217,9	217,8	217,8
dosežen cilj	1	200	204,3	204,3	204,2
program			314,4	314,5	314,4
izmerjeno			315,5	315,0	315,2
dosežen cilj	0,1	300	300,8	300,7	300,9
program			304,5	303,5	304,5
izmerjeno			305,8	304,8	305,9
dosežen cilj	0,25	300	300,9	300,8	300,8
program			316,6	316,6	316,6
izmerjeno			317,5	317,5	317,5
dosežen cilj	1	300	301,7	301,7	301,6
program			411,8	411,8	411,9
izmerjeno			412,3	412,4	412,8

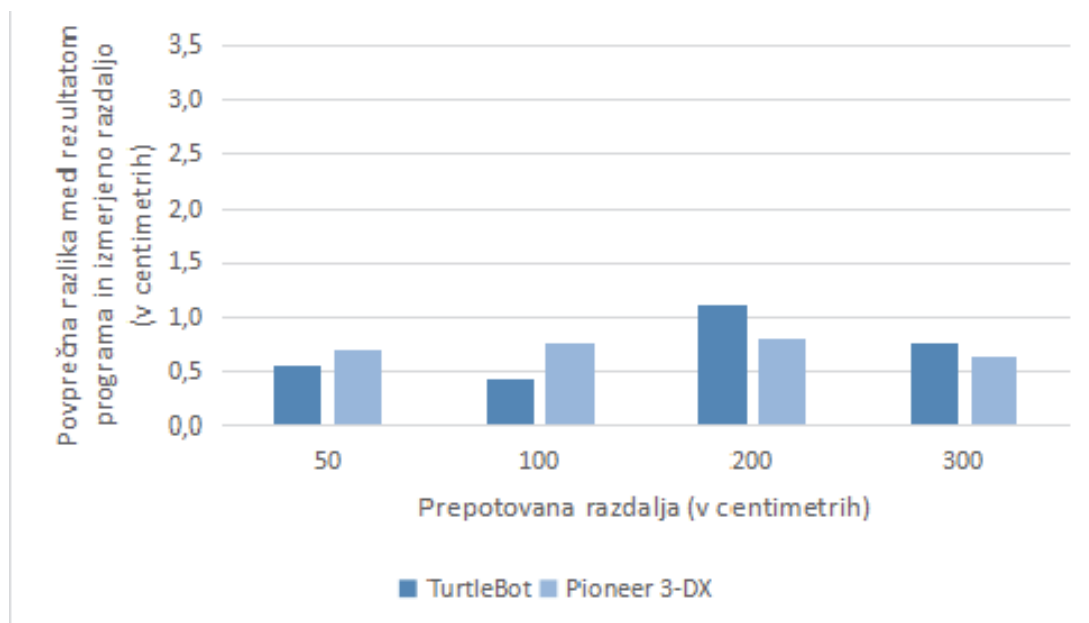
Tabela 3: Rezultati eksperimentov z robotom Pioneer 3-DX.



Slika 3.10: Razlika med rezultatom programa in izmerjeno razdaljo pri hitrosti 0,1 m/s.



Slika 3.11: Razlika med rezultatom programa in izmerjeno razdaljo pri hitrosti 0,25 m/s.



Slika 3.12: Razlika med rezultatom programa in izmerjeno razdaljo pri hitrosti 1 m/s.

3. 2. 2 Vrtenje na mestu

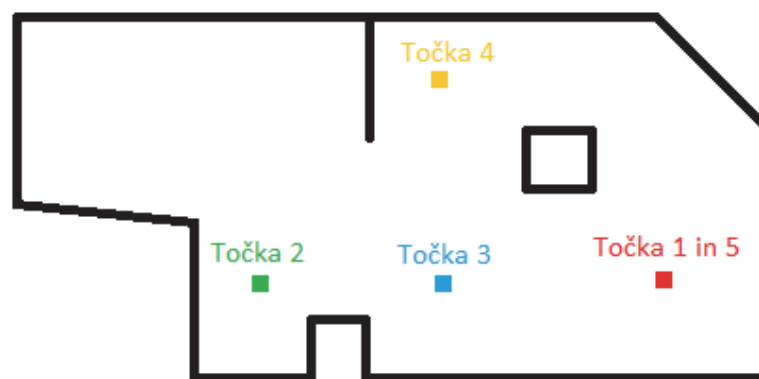
Rezultati v tabeli 4 so prikazani na enak način, kot v tabelah 2 in 3. Vsi rezultati so zapisani v stopinjah. Podatki v vrstici *izmerjeno* so bili, tako kot prej, izmerjeni ročno. Ker smo tukaj opazovali kako se robot vrti, je bil podatek izmerjen s kotomerom. Pri tem smo bili na stopinjo natančni. Pri vrtenju smo, poleg hitrosti, ki smo jo zabeležili, eksperimentirali tudi z manjšimi hitrostmi. Pri njih se je pojavil problem, da je TurtleBot drsel po tleh, namesto da bi se mu kolesa normalno vrtela in zaradi tega rezultatov nismo zabeležili. Vsi rezultati eksperimentov v tabeli so podani v stopinjah. Povprečna razlika med izmerjenim kotom in kotom, ki ga je vračal program, je bila pri TurtleBotu 3,8 cm in pri Pioneerju 3-DX 0,4 cm. To nam pove, da je bil Pioneer 3-DX tudi pri tem eksperimentu bolj natančen od TurtleBota.

Vrtenje	Kot	Hitrost (°/s)	Eks. 1	Eks. 2	Eks. 3	Eks. 4	Eks. 5	Eks. 6	Eks. 7
TurtleBot									
dosežen cilj	90°	22,5	91,1	90,3	90,8	91,5	93,1	91,3	92,1
program			99,0	99,5	98,9	99,0	101,6	98,9	99,5
izmerjeno			97	92	96	95	97	97	96
Pioneer 3-DX									
dosežen cilj	90°	22,5	91,3	91,4	90,2	91,3	91,3	91,4	91,3
program			98,6	98,7	97,6	98,6	98,6	98,7	98,6
izmerjeno			99	99	98	99	98	99	99

Tabela 4: Rezultati eksperimentov pri vrtenju na mestu.

3.3 Evalvacija navigacije robotov

Pri teh meritvah smo eksperimentirali s tremi kombinacijami robotov in senzorjev: TurtleBot s Kinectom in URG-04LX ter Pioneer 3-DX z URG-04LX. Za vsako kombinacijo smo zgradili zemljevid, ki smo ga v eksperimentih uporabili za umestitev robota v svet in določanje ciljev. Ker smo uporabili robote in senzorje skupaj, smo lahko popravljali napake, ki so se pojavljale pri odometriji, s podatki, ki smo jih prejeli od senzorjev. Na zemljevidu smo določili točke, do katerih se mora robot premakniti. Za vsako kombinacijo smo izbrali pet točk na približno enakih mestih. Na sliki 3.13 vidimo postavitev točk.



Slika 3.13: Postavitev točk, ki jih je robot moral doseči.

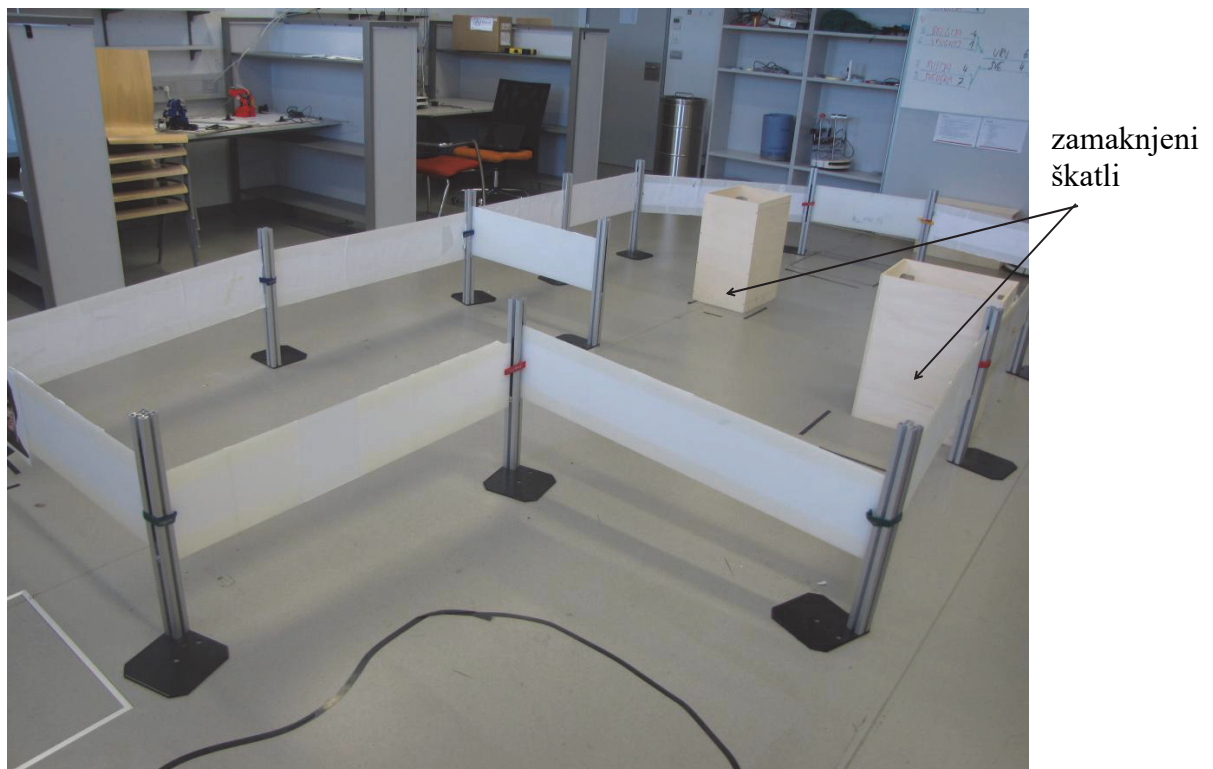
Pred vsakim eksperimentom smo morali robota lokalizirati. Lokalizacija je postopek, pri katerem robota umestimo na podan zemljevid. Po lokalizaciji smo pognali program, ki je robotu pošiljal ukaze, kam mora peljati. Eksperimente smo opravljali na treh različnih poligonih, a vedno z istim zemljevidom. Ti poligoni so bili:

- Zemljevid in poligon sta enaka. (slika 3.14)
- Škatle na poligonu so predstavljene za nekaj decimetrov. (slika 3.15)
- Škatle na poligonu so odstranjene. (slika 3.16)

Pri tem nas je zanimalo, kako se bo robot obnašal na poligonu, ki ne ustreza zemljevidu, ki ga uporablja za lokalizacijo.



Slika 3.14: Fotografija poligona s škatlami, ki so postavljene enako kot na zemljevidu.



Slika 3.15: Fotografija poligona z zamaknjenimi škatlami.

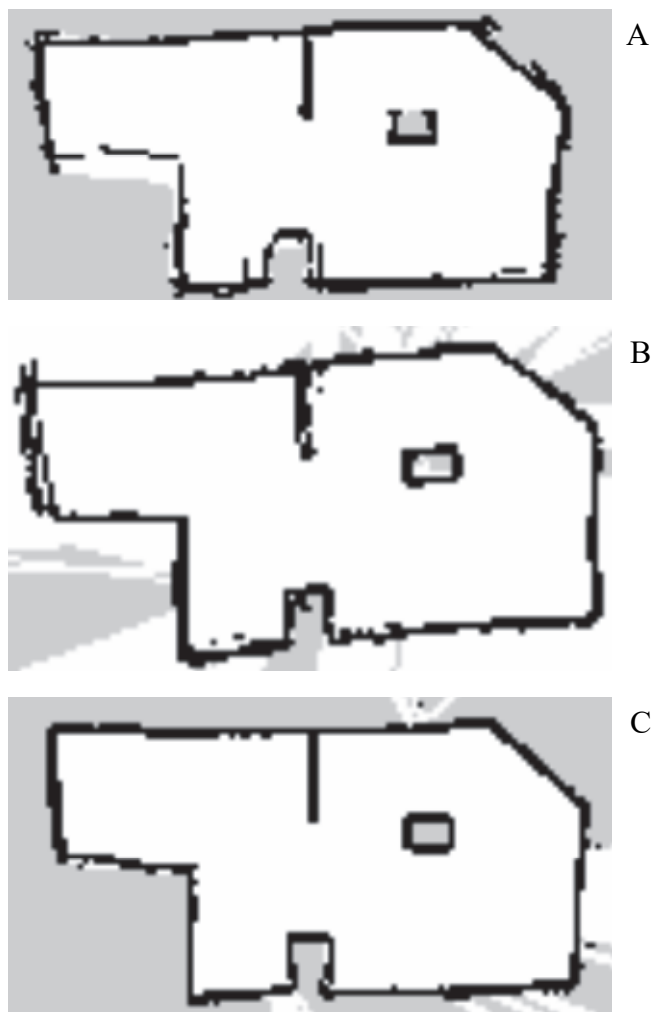


Slika 3.16: Fotografija poligona brez škatel.

3. 3. 1 Zgrajeni zemljevidi

Zemljevide smo gradili za vsako kombinacijo robota in senzorja posebej. Že pri gradnji zemljevidov so bile razvidne razlike v njihovem delovanju.

Iz zemljevidov se vidi, da je Pioneer 3-DX z URG-04LX ustvaril bolj ostro sliko, kar nam omogoča lažjo lokalizacijo in navigacijo. Na sliki 3.17 (a) so vidne luknje v stenah, kar povzroča težave pri lokalizaciji robota med njegovim delovanjem. Če primerjamo sliki 3.17 (b) in 3.17 (c), vidimo, da ima, čeprav sta bila oba zemljevida narejena s podatki, ki so prišli iz URG-04LX, slika 3.17 (c) bolj ostre robove. Do tega pride, ker program ustvari zemljevid iz podatkov, ki jih dobi iz senzorja (dolžina do ovir) in robota (odometrija, ki mu pove, za koliko se je robot premaknil). In ker ima Pioneer 3-DX bolj natančno odometrijo, je zemljevid bolj natančen, čeprav je podatek o razdalji prišel iz istega senzorja.



Slika 3.17: (a) Zemljevid narejen s kombinacijo TurtleBota in Kinecta.
(b) Zemljevid narejen s kombinacijo TurtleBota in URG-04LX.
(c) Zemljevid narejen s kombinacijo Pioneerja 3-DX in URG-04LX.

3.3.2 Rezultati eksperimentov

Pri eksperimentih smo izbrali pet točk. Začetna in končna točka sta se nahajali na isti poziciji, kar je olajšalo analizo rezultatov. V tabelah 5 in 6 je v prvi vrstici pri vsakem eksperimentu zapisan podatek, ki ga je vrnil program ob končani vožnji na podlagi odometrije robota (v tabelah 5 in 6: *odometrija*). V vsaki drugi vrstici je zapisan podatek meritve, ki smo ga izmerili z metrom (v tabelah 5 in 6: *meritev*). Pri meritvi, ki smo jo opravljali z metrom, smo bili natančni do enega milimetra. Podatki v tabelah predstavljajo odmik robota od končne točke.

V tabeli 5 so prikazani rezultati eksperimentov, kjer sta se zemljevid in poligon ujemala. Iz eksperimentov je razvidno, da je imel TurtleBot približno enake napake, ne glede na to ali je bil uporabljen senzor URG-04LX ali Kinect. Zaradi tega domnevamo, da je bolj pomembna dobra odometrija, kot pa dober senzor. V primeru, da odometrija ne meri pravilno prepotovane poti, začenja program dobivati podatke o bližini ovir, ki se ne ujemajo s pričakovano pozicijo. Če se to zgodi, ni nujno, da bo program pravilno lokaliziral robota in zaradi tega pride do napak v končni poziciji.

Pioneer 3-DX ni dosegel četrte točke. Predvidevamo, da se je to zgodilo, ker program pri računanju možne poti robota predpostavi, da robot zasede nekoliko več prostora, kot ga dejansko. Tako robot ni bil sposoben najti poti do te točke. Predvidevamo, da iz podobnega razloga Pioneer 3-DX ni bil sposoben končati poti po poligonu s prestavljenimi škatlami. Ker zemljevid ni bil točen glede na dejansko postavitev poligona, je bilo težje najti natančen položaj robota. To je povzročilo, da program ni mogel najti poti, po kateri robot zagotovo ne bi naletel na oviro.

Pravilni zemljevid	TurtleBot in Kinect	TurtleBot in URG-04LX	Pioneer 3-DX in URG-04LX
Eksperiment 1			
odometrija (centimetri)	39,5	5,6	3,9
meritev (centimetri)	19,5	16,0	4,0
Eksperiment 2			
odometrija (centimetri)	14,2	31,1	17,4
meritev (centimetri)	10,5	25,1	17,0
Eksperiment 3			
odometrija (centimetri)	34,5	26,3	4,9
meritev (centimetri)	8,2	17,1	2,5
Eksperiment 4			
odometrija (centimetri)	24,9	21,5	1,8
meritev (centimetri)	15,1	18,8	2,1

Tabela 5: Rezultati eksperimentov, ki kažejo razdaljo med začetno in končno postavitevjo, na poligonu, ki se ujema z zemljevidom.

Podatki v tabeli 6 so predstavljeni na enak način kot v tabeli 5. Pri eksperimentih z zamaknjenimi škatlami se je spet pokazalo, da ni bistvene razlike pri rezultatih TurtleBota, ne glede na to, kateri senzor uporablja. Zadnjega sklopa eksperimentov (brez škatel na poligonu), ki smo jih nameravali narediti, nismo mogli, saj se je izkazalo, da program ni mogel najti poti, ki bi jo lahko poslal robotom. To se je zgodilo, ker program robota ni mogel pravilno

lokalizirati in zaradi tega ni mogel najti poti. Lokalizacija ni uspela, ker se zemljevid ni ujemal s poligonom.

Zamaknjene ovire	TurtleBot - Kinect	TurtleBot - URG-04LX
Eksperiment 1		
odometrija (centimetri)	46,9	21,5
meritev (centimetri)	7,3	9,3
Eksperiment 2		
odometrija (centimetri)	29,8	25,7
meritev (centimetri)	13,6	21,0
Eksperiment 3		
odometrija (centimetri)	39,8	16,9
meritev (centimetri)	37,5	19,6
Eksperiment 4		
odometrija (centimetri)	30,1	18,8
meritev (centimetri)	23,2	20,7

Tabela 6: Rezultati eksperimentov, ki kažejo razdaljo med začetno in končno postavitvijo, na poligonu z zamaknjenimi ovirami.

4 Zaključek

Cilj diplomske naloge je bil pokazati razlike v delovanju senzorjev, robotov in sodelovanja med njimi pri gradnji zemljevida in umeščanju robota v svet. Za doseg ciljev smo eksperimentirali posebej s senzorji in roboti ter na koncu še s kombinacijo robotov in senzorjev. Pri senzorjih nas je najbolj zanimalo, kako dobro delujejo pri različnih razdaljah in kotih od opazovanega predmeta. Pri robotih nas je zanimalo, kako dobro robot računa odometrijo. Pri kombinaciji robotov in senzorjev smo opazovali, kakšna je razlika med načrtovanim in dejanskim koncem poti robota.

Kot smo pričakovali, se je izkazalo, da je najboljša kombinacija robota Pioneer 3-DX in senzorja URG-04LX. S to kombinacijo dobimo najbolj natančne rezultate o prepotovani poti robota. Edini problem, ki se pojavi je ta, da je Pioneer 3-DX večji od TurtleBota in ima zaradi tega probleme z manevriranjem po manjših prostorih.

Ena izmed možnosti za izboljšavo delovanja Pioneerja 3-DX bi bila nastavitev parametrov programa, ki računa pot po kateri robot potuje. Druga možnost je, da bi napisali nov program, ki ne bi predvideval, da je robot okrogle oblike, kot je to počel uporabljeni program iz paketa navigation. Druga možnost bi bila verjetno veliko bolj učinkovita, a tudi časovno bolj zahtevna.

Literatura

- [1] Luka Čehovin Zajc, Anže Rezelj in Danijel Skočaj. Teaching intelligent robotics with a low-cost mobile robot platform, maj 2015
- [2] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99), juli 1999.
- [3] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pages 2432–2437, April 2005.
- [4] K. Khoshelham. Accuracy analysis of kinect depth data, avgust 2011.
- [5] Jeffrey Meisner, 2. oktober 2018. Collaboration, expertise produce enhanced sensing in Xbox One. [Internet] Dostopno na: https://blogs.technet.microsoft.com/microsoft_blog/2013/10/02/collaboration-expertise-produce-enhanced-sensing-in-xbox-one/ [Dostopano: 10. avgust 2018].
- [6] T. Wiedemeyer, "IAI Kinect2," Dostopno na: https://github.com/code-iai/iai_kinect2, Institute for Artificial Intelligence, University Bremen, 2014 – 2015, [Dostopano junij 2018].
- [7] BVV, 10. januar 2016. KUKA Roboter CEE GmbH, organizační složka. [Internet] Dostopano na: <https://www.bvv.cz/en/msv/news/kuka-roboter-cee-gmbh-organizational-unit/> [Dostopano: 1. september 2018]
- [8] Good gear guide. Roomba 530 specifications. [Internet] Dostopno na: https://www.goodgearguide.com.au/review/irobot/roomba_530/359265/specs/ [Dostopano: 12. avgust 2018].
- [9] Hokuyo, 2014. Distance Data Output/URG-04LX. [Internet] Dostopno na: <https://www.hokuyo-aut.jp/search/single.php?serial=165> [Dostopano: 12. avgust 2018].
- [10] Orbbec. Orbbec Astra Pro. [Internet] Dostopno na: <https://orbbec3d.com/product-astra-pro/> [Dostopano: 10. avgust 2018].

- [11] Microsoft. Kinect sensor for Xbox 360 components. [Internet] Dostopno na: <https://support.xbox.com/en-CA/xbox-360/accessories/kinect-sensor-components> [Dostopano: 10. avgust 2018].
- [12] Mobilerobots. Pioneer 3-DX. [Internet] Dostopno na: <http://www.mobilerobots.com/Libraries/Downloads/Pioneer3DX-P3DX-RevA.sflb.ashx> [Dostopano: 12. avgust 2018]
- [13] ROS. About ROS. [Internet] Dostopno na: <http://www.ros.org/about-ros/> [Dostopano: 12. avgust 2018]
- [14] ROS wiki. Concepts. [Internet] (Nazadnje spremenjeno: 21. juni 2014) Dostopno na: <http://wiki.ros.org/ROS/Concepts> [Dostopano: 12. avgust 2018].
- [15] ROS wiki. Navigation. [Internet] (Nazadnje spremenjeno: 10. marec 2017) Dostopno na: <http://wiki.ros.org/navigation> [Dostopano: 26. avgust 2018].
- [16] Wikipedia. Computer Vision. [Internet] (Nazadnje spremenjeno: 10. avgust 2018) Dostopno na: https://en.wikipedia.org/wiki/Computer_vision [Dostopano: 10. avgust 2018].
- [17] Wikipedia. Kinect. [Internet] (Nazadnje spremenjeno: 26. julij 2018) Dostopno na: <https://en.wikipedia.org/wiki/Kinect> [Dostopano: 10. avgust 2018].