

Wydział Elektroniki i Technik
Informacyjnych
Politechnika Warszawska

Pracownia Dyplomowa Inżynierska 1
sprawozdanie

Autonomiczna Nawigacja Manipulatora
Dwuręcznego z Wielokierunkową Bazą
Mobilną

Marcin Skrzypkowski

opiekun pracy:
dr inż. Wojciech Szynkiewicz

Warszawa, 21 stycznia 2020

Spis treści

1	Cel pracy	3
2	Przegląd Literatury	4
2.1	Wykorzystanie czujników laserowych w budowaniu mapy	4
2.2	Wykorzystanie kamery Kinect w budowaniu mapy	4
2.3	Wykorzystanie wielu czujników w autonomicznej lokalizacji robota	5
2.4	Planery lokalne w bazach holonomicznych	5
3	Struktura systemu	6
4	Opis Zadania	8
4.1	Budowanie Mapy	8
4.2	Lokalizacja	8
4.3	Holonomiczność	9
5	Wstępne Badania	10
5.1	Testy holonomiczności bazy	10
5.2	Budowanie mapy	11
5.3	lokalizacja	12
6	Plan Działania	16
	References	17

skupić się bardziej na podejściu ruchowym niż zadaniowym

globalna nawigacja: -

lokalny planner: -poprawa holonomiczności

tworzenie mapy i lokalizacja: -wykorzystanie kinecta i lidarów do tworzenia

mapy -połączenie map z lidarów - map_merger -wykorzystanie lidarów, odometrii, imu do bieżącej lokalizacji

wrzucić linki do pakietów, które będą wykorzystywane do navigation stack?

opisać działanie navigation stack?

1 Cel pracy

Możliwość poruszania się robota jest wymaganiem wielu współczesnych aplikacji (Z. Riaz A. Pervez, 2010). Niniejsza praca skupia się na implementacji algorytmów autonomicznej lokalizacji, planowania ruchu robota Velma na bazie wielokierunkowej oraz wykonywania zadanych trajektorii. Ponieważ problem planowania części manipulacyjnej jest rozpatrywany w pracy Grzegorza Fijałkowskiego, nie jest on tu poruszany. Elementy przedstawione w następnych rozdziałach dotyczą ruchu w przestrzeni zamkniętej, można je podzielić na dwa główne aspekty.

Pierwszym z nich jest lokalizacja. W mojej pracy inżynierskiej zostaną przedstawione prace mające na celu wykorzystanie obydwu czujników laserowych do jednoczesnej budowy dwuwymiarowej mapy zajętości oraz wykorzystanie kamery Kinect w dodawaniu do niej elementów znajdujących się powyżej zasięgu pozostałych czujników.

Dodatkowym problemem lokalizacji, który zostanie poruszony, jest badanie jej dokładności w zależności od wykorzystanych czujników. Badania mają odpowiedzieć na pytanie, które czujniki najlepiej w tym celu wykorzystywać.

Drugim aspektem jest poruszanie robotem holonomicznym. W obecnej chwili baza jest w stanie poruszać się we wszystkich kierunkach, a moim zadaniem jest przeprowadzenie automatyzacji działań potrzebnych do osiągnięcia tego efektu przez dodanie odpowiedniego algorytmu planowania.

Punkty przedstawione powyżej są rozwinięciem istniejącej platformy, przygotowanej przez studentów Politechniki Warszawskiej. Celem dalszego rozwoju jest stworzenie platformy spełniającej współczesne standardy autonomicznych robotów zarówno mobilnych, jak i manipulacyjnych, ale, jak zostało wspomniane powyżej, moja praca jest skupiona wokół mobilnych aspektów platformy.

2 Przegląd Literatury

2.1 Wykorzystanie czujników laserowych w budowaniu mapy

Wykorzystanie czujników laserowych jest popularnym sposobem na stworzenie dokładnej mapy otoczenia. W użyciu spotkać można skanery zarówno dwu- jak i trójwymiarowe. Ten drugi rodzaj mógłby rozwiązać problem opisany w sekcji 1 polegający na wykrywaniu przeszkód znajdujących się powyżej bazy robota. Jednak czujniki trójwymiarowe są drogie, dlatego w artykule (Rejas, Sanchez, Glez-de Rivera, Prieto, & Garrido, 2015) podana jest propozycja wykorzystania skanera dwuwymiarowego w celu otrzymania przestrzennej chmury punktów. Jednak w przypadku platformy Velmy z bazą wielokierunkową zastosowanie takiej metody jest na chwilę obecną niemożliwe, a ewentualna modyfikacja trudna do wykonania.

Czujniki dwuwymiarowe cieszą się rosnącą popularnością, czego dowodem jest duża liczba istniejących algorytmów wykorzystywanych do wykrywania linii w uzyskanych skanach. Również konfiguracja dwóch czujników obecna na przedstawionej w mojej pracy platformie jest względnie popularnym rozwiązaniem, zapewniającym widoczność w pełnych 360 stopniach (Nguyen, Martinelli, Tomatis, & Siegwart, 2005).

2.2 Wykorzystanie kamery Kinect w budowaniu mapy

W przypadkach robotów działających w środowiskach zewnętrznych popularne jest wykorzystanie modułów GPS, kamer RGB oraz czujników laserowych do budowy mapy terenu (Shalal, Low, McCarthy, & Hancock, 2015). Wykorzystanie kamery Kinect w mapowaniu przestrzeni na świeżym powietrzu jest związane z dużymi trudnościami ze względu na charakterystykę działania czuj-

nika. Jednak niektóre badania wskazują na wyższość takiego rozwiązania nad wykorzystaniem stereowizji w konkretnych przypadkach (Hernandez-Aceituno, Arnay, Toledo, & Acosta, 2016).

Popularne również jest wykorzystanie samego czujnika Kinect do stworzenia trójwymiarowej mapy pomieszczeń, posiada on biblioteki dostępne na wolnych licencjach i umożliwia szybkie uzyskanie mapy głębi. Tworzenie map dwuwymiarowych z mapy głębi tej kamery nie jest nowym pomysłem i wykazuje poprawę w tworzeniu mapy w porównaniu z wykorzystaniem samych czujników dwuwymiarowych potencjalnie niewykrywających niektórych przeszkód, dając gwarancję ich wykrycia, z zastrzeżeniem przedmiotów przezroczystych (Kamarudin et al., 2013). Niektóre badania dowodzą, że wykorzystanie kamery firmy Microsoft przy pomocy systemu ROS i pakietu GMapping skutkuje lepszymi rezultatami niż budowanie mapy z nieprzefiltrowanych skanów czujników laserowych (Omara & Sahari, 2015). W przypadku robota Velmy na bazie dookólnej nie przewiduję rezygnacji z czujników laserowych na rzecz pozostawienia jedynie kamery Kinect, ze względu na wielkość środowiska, w którym baza będzie w przyszłości operować. Sodontkowo jak wspomniano wcześniej czujniki te dają pełne 360 stopni pokrycia w przestrzeni dwuwymiarowej, podczas gdy kamera posiada jedynie 57x43 (Arjayay, 2020).

2.3 Wykorzystanie wielu czujników w autonomicznej lokalizacji robota

2.4 Planery lokalne w bazach holonomicznych

Ze względu na mniejszy stopień złożoności problemu, w wielu przypadkach roboty wykorzystujące odometrię oraz czujniki laserowe budowane są na bazach różnicowych.

3 Struktura systemu

W celu dokładnego sformułowania zadań, które należy wykonać do osiągnięcia celu postawionego w sekcji 1, potrzebne jest przedstawienie systemu, na którym będę pracować. Poniżej opisana jest część związana z symulacją. Mimo że docelowo wszystkie rozwiązania mają zostać przeniesione na rzeczywisty sprzęt, implementacja, wszelkie testy i badania w mojej pracy zostaną przeprowadzone w środowisku symulacji. Cała platforma została oparta na popularnym systemie ROS (Robotics, 2019), który udostępnia liczne narzędzia wspomagające programowanie robotów.

Początkowo wielokierunkowa baza mobilna i manipulator były oddzielnymi bytami, ale dzięki wysiłkom Jakuba Sikory (Sikora, 2020) stanowią całość. Robot został stworzony w symulatorze Gazebo (Foundation, 2019), który umożliwia nie tylko przetestowanie fizycznych właściwości robota jak prędkości i reakcja na algorytmy sterowania, ale też zbudowanie własnego środowiska do testowania wszystkich aspektów robota, jak na przykład tworzenie trójwymiarowej mapy otoczenia z pomocą kamery Kinect (jeden ze sposobów jest przedstawiony tutaj (sauravagarwal, 2016)). Poza Kinectem do wykorzystania w platformie są czujniki laserowe, jednostka inercyjna oraz standardowe kamery RGB. Czujnik laserowy firmy Sick przeznaczony do skanowania w dwóch wymiarach (prawdopodobnie NAV2xx?) (sick, 2019) o zasięgu kątowym 270 stopni oraz odległości skanowania do 50m. Baza jezdna posiada cztery koła szwedzkie umożliwiające poruszanie się w dowolnym kierunku oraz obrót w miejscu.

W celu przetestowania obecnie istniejących algorytmów i porównania ich do zaimplementowanych przeze mnie wykorzystane zostaną dwa środowiska, przedstawione na rysunkach poniżej

Obecnie w systemie planowania zaimplementowana jest prosta wersja systemu nawigacji wykorzystująca koncepcję połączenia planowania globalnej ścieżki

oraz lokalnego planera z pakietu *teb_local_planner*. Nie spełnia on wymagań nawigacji wielokierunkowej, jej implementacja jest jednym z zadań, które zamierzam wykonać.

Do celów lokalizacji w obecnym momencie wykorzystywana jest jedynie odometria bazy, obliczana na podstawie prędkości. Istnieje również możliwość przekazania dokładnej pozycji z symulatora Gazebo.

4 Opis Zadania

W sekcji 1 przedstawione zostały trzy główne problemy, które moja praca inżynierska ma na celu rozwiązać.

4.1 Budowanie Mapy

W celu poprawnej lokalizacji konieczna jest uprzednio zbudowana mapa, w tym przypadku dwuwymiarowa mapa zajętości. Jak już wcześniej zostało wspomniane, obecnie budowanie mapy wykorzystuje jeden czujnik laserowy, dowolny z obydwu dostępnych. Docelowy system będzie w stanie wykorzystać je jednocześnie przyspieszając tworzenie mapy zajętości. Wykorzystanie jedynie czujników w bazie w tak wysokim robocie powoduje duże ryzyko pominięcia przeszkód znajdujących się poza ich zasięgiem, jak przykładowo stół na cienkich nogach. Taka przeszkoda pojawi się na mapie zajętości w postaci czterech drobnych plam, a więc istnieje ryzyko, że robot uderzy w blat stołu nie wiedząc, że ten się tam znajduje.

W celu rozwiązania tego problemu planowana jest integracja kamery Kinect do opisanego przed chwilą systemu. Trójwymiarowa reprezentacja przestrzeni z kamery zostanie rzutowana na mapę dwuwymiarową przygotowaną przez skanery. Do przebadania zostanie problem blokowania niektórych pozycji docelowych bazy, przykładowo tak przygotowana mapa zajętości uniemożliwi bazie wjechanie częściowo pod blat, a taka operacja zwiększyła by zasięg manipulatorów nad powierzchnią stołu.

4.2 Lokalizacja

Obecnie lokalizacja robota odbywa się jedynie z pomocą odometrii, liczonej na podstawie prędkości bazy jezdnej, lub poprzez pobieranie wiadomości bezpośrednio z symulatora Gazebo. Ponieważ dokładność pierwszej metody jest niska

(Sikora, 2020), a druga nie ma prawa bytu jeżeli celem jest testowanie lokalizacji, należy wykorzystać dostępne czujniki. Przetestowane zostaną pojedynczy czujnik laserowy, kamera Kinect oraz jednostka inercyjna w wielu konfiguracjach, aby sprawdzić, czy najlepsze wyniki daje jeden konkretny rodzaj czujnika, czy dwa lub trzy rodzaje połączone ze sobą.

4.3 Holonomiczność

Obecna implementacja planera lokalnego nie wspiera ruchu bazy we wszystkich kierunkach, po zadaniu pozycji planer globalny poprawnie wyznacza ścieżkę, lecz planer lokalny steruje robotem jak bazą różnicową, nie wykorzystując większej ilości stopni swobody bazy. Docelowo należy dostroić dostarczony już planer lokalny aby wykorzystywał pełnię możliwości platformy, lub, jeśli ten sposób zawiedzie, znaleźć i zaimplementować inny, który zapewni wymaganą specyfikację.

5 Wstępne Badania

5.1 Testy holonomiczności bazy

Aby móc prowadzić prace nad platformą holonomiczną należy najpierw upewnić się, że jej sterownik działa prawidłowo. W celu weryfikacji przetestowano symulator z pomocą skryptu ustawiającego prędkości w taki sposób, aby robot jechał po okręgu. Na temat */cmd_vel* były publikowane odpowiednie prędkości w osiach x oraz y bazy jezdnej. Ilustracja poniższa pokazuje, że sterownik działa bez zarzutu i sterując wspomnianymi prędkościami można prowadzić bazę w dowolnym kierunku.

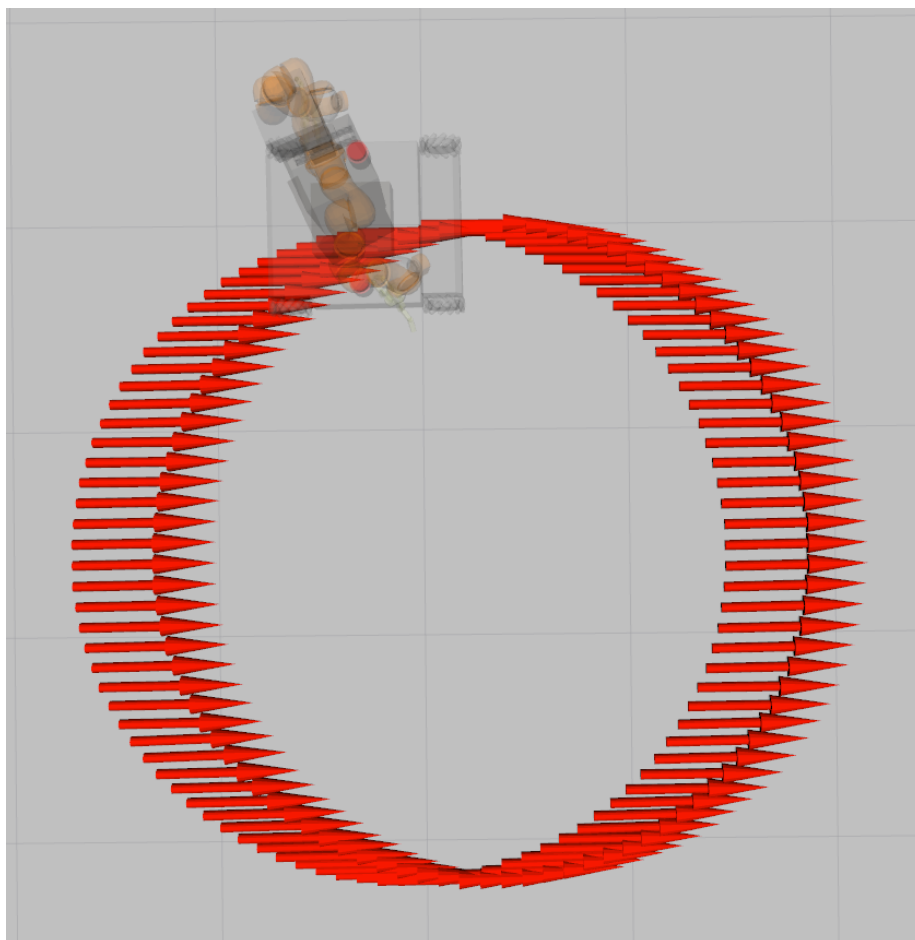
Kierunek strzał oznacza, że orientacja bazy podczas ruchu nie ulegała zmianie, natomiast wszystkie punkty układają się w pełen okrąg.

Następnym testem, który został przeprowadzony, był test planera globalnego i lokalnego udostępnionych razem z platformą. Ze względu na nacisk położony na holonomiczny ruch bazy poświęciłem więcej uwagi algorytmowi planowania lokalnego, plan globalny był jedynie weryfikowany przez sprawdzenie, czy została wytyczona ścieżka. Zadaniem robota było przejechanie z punktu o współrzędnych $[1.0, 2.0]$ do współrzędnych $[2.0, 0.0]$.

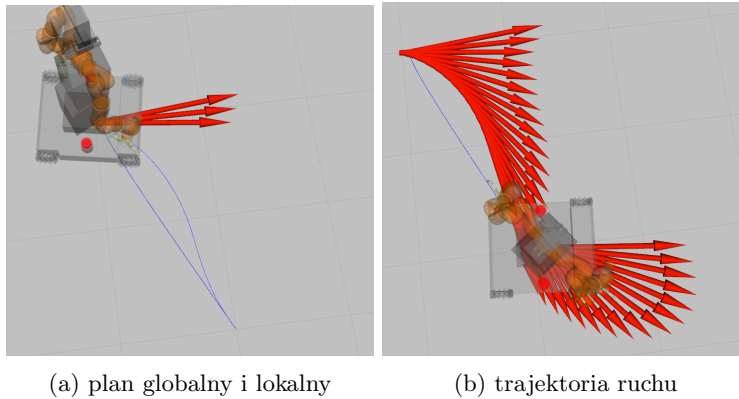
Na pierwszym rysunku z 1 pokazano, jak plan globalny prawidłowo wyznaczył najkrótszą ścieżkę do punktu docelowego. Natomiast ścieżka algorytmu lokalnego pokazuje, że robot jest traktowany jako baza różnicowa, co ostatecznie dowodzi trajektoria ruchu widoczna na drugiej ilustracji.

5.2 Budowanie mapy

W obecnej chwili do budowy mapy można wykorzystywać jedynie pojedynczy czujnik laserowy znajdujący się po lewej lub po prawej stronie bazy jezdnej. Jest również zaimplementowana projekcja mapy zajętości generowanej przez



Rysunek 1: ruch po okręgu ze stałą orientacją względem świata



Rysunek 2: The same cup of coffee. Two times.

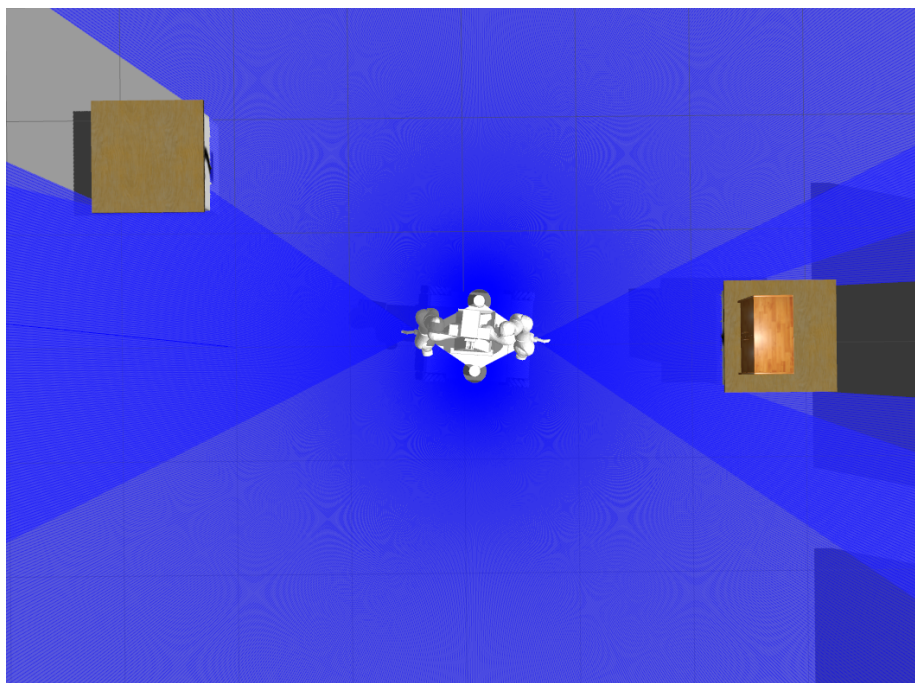
kamerę Kinect na płaszczyznę dwuwymiarową, jednak nie została jeszcze ona zintegrowana z modułem odpowiedzialnym za tworzenie map statycznych, na chwilę pisania tego rozdziału budowanie mapy opiera się tylko na pojedynczym skanerze laserowym. Na ilustracjach 2 pokazano, jak na mapę lokalną został rzutowany fragment wystającego blatu stolika.

Na rysunku 3 przedstawiono pojedynczy skan laserowy wykonany w nieznanym środowisku przed poruszeniem bazą robota.

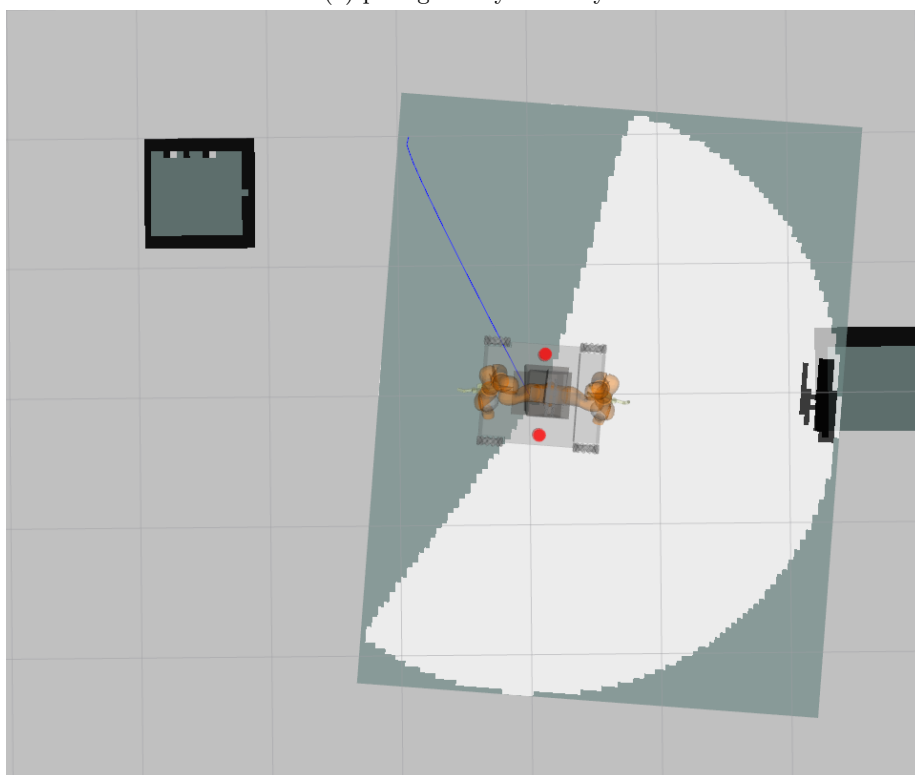
Podczas próby przetestowania modułu *gmapping* wykorzystującego pojedynczy skaner laserowy do zbudowania statycznej mapy zajętości okazało się, że w systemie znajdują się błędy uniemożliwiające budowę mapy. Na rysunku 4 ukazane zostało drzewo transformacji, na którym widać, że pierwotnym układem współrzędnych jest *odom*, z którego następnie jest przeliczana transformacja na układ *world*, a z tego równolegle na *torso_base* oraz równolegle *map*. Jest to błędna struktura, poprawna posiada *world* jako pierwotny układ współrzędnych, od niego przechodzi transformacja do układu *map*, stąd do *odom*, a wreszcie do *torso_base*, bez rozgałęzień.

5.3 lokalizacja

Ta część badań jest ściśle zależna od budowania mapy oraz poprawności drzewa transformacji. Dopóki ten element wykazuje błędne działanie, lokalizacja robota z pomocą odometrii oraz laserów i kamery Kinect, ale też wysyłanie pozycji innych przedmiotów w jego otoczeniu jest niemożliwa. Dlatego ta część badań zostaje wstrzymana do czasu usunięcia błędów w drzewie transformacji. Na rysunku ... ukazano próbę jednoczesnej lokalizacji i budowania mapy robota przy opisanym stanie systemu. Robot jechał ze stałą prędkością 0.4 w kierunku x bazy i przejechał w rzeczywistości około 6 metrów.

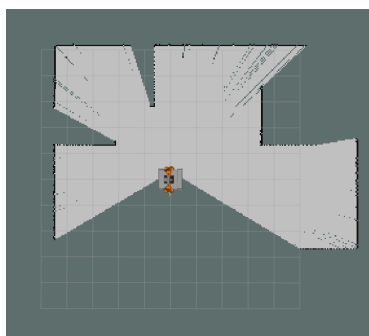


(a) plan globalny i lokalny

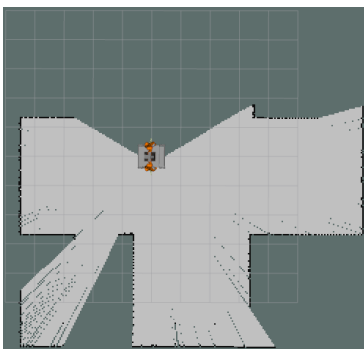


(b) trajektoria ruchu

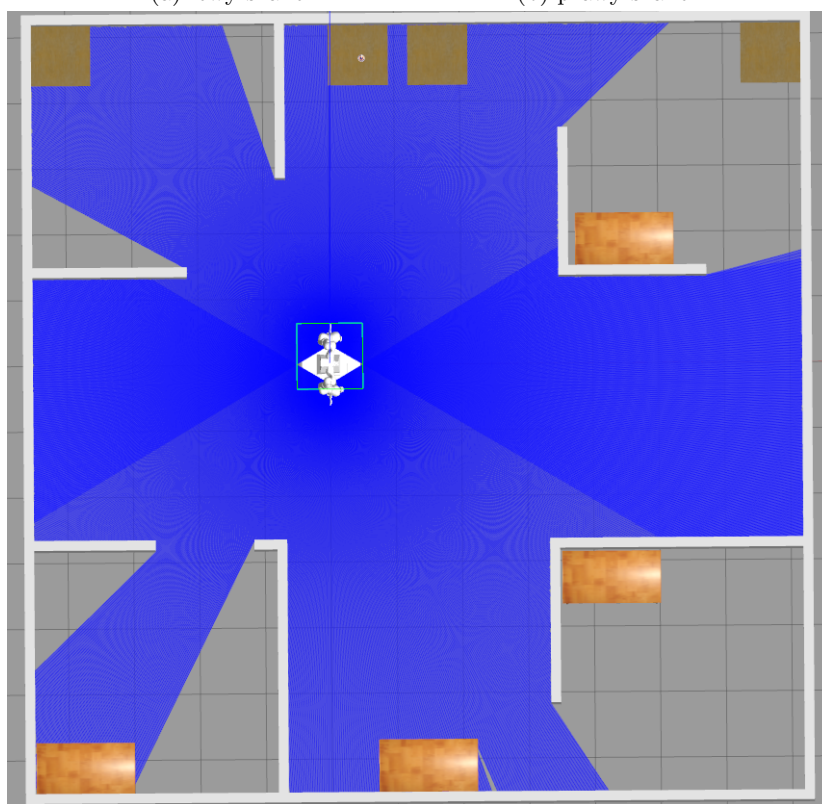
Rysunek 3: rzutowanie skanu kamery Kinect



(a) lewy skaner

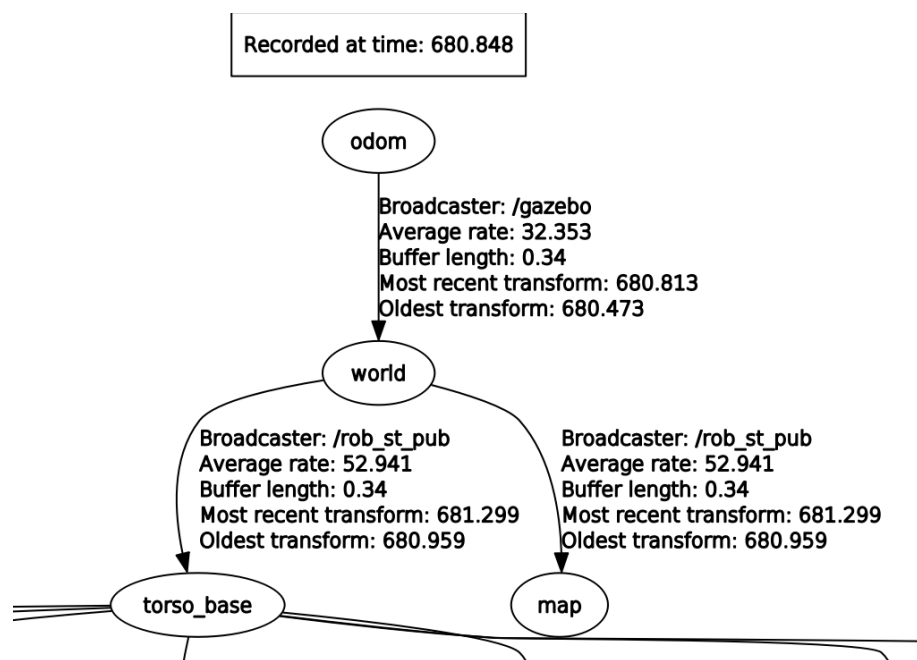


(b) prawy skaner

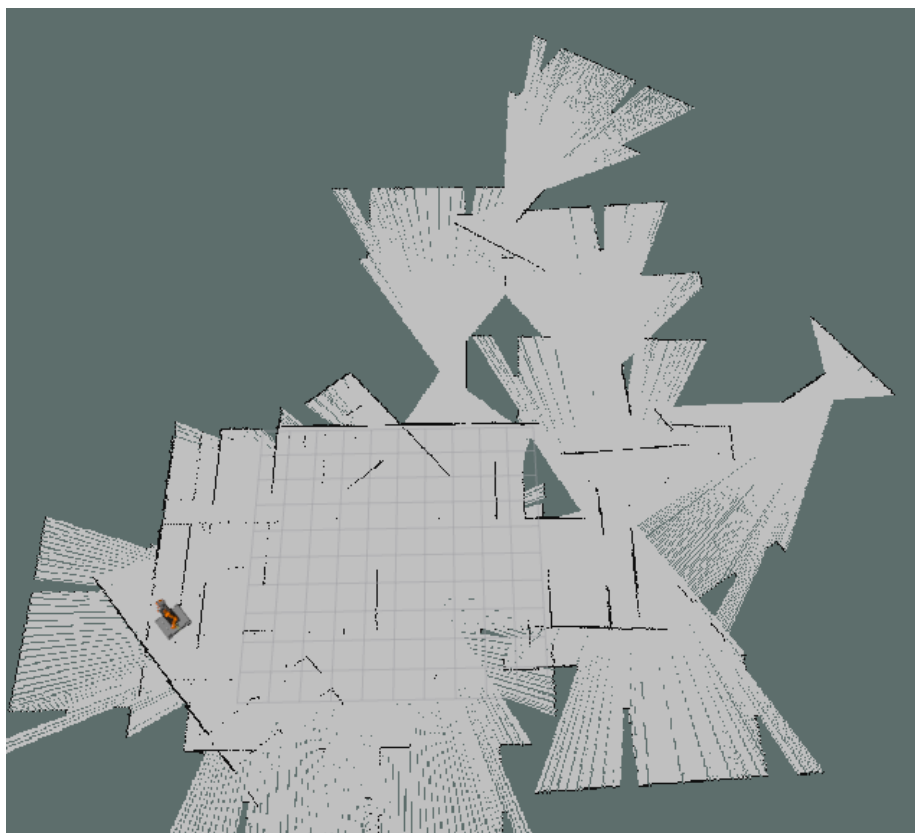


(c) pokrycie obydwu skanerów

Rysunek 4: pokrycie przestrzeni



Rysunek 5: błędne drzewo transformacji



Rysunek 6: błąd w dynamicznym tworzeniu mapy statycznej i w lokalizacji

6 Plan Działania

plan

Bibliografia dla mnie:

References

- Arjayay, M. M. J. B. (2020). *Kinect*. Retrieved 2020-01-18, from <https://en.wikipedia.org/wiki/Kinect>
- Foundation, O. S. R. (2019). *Gazebo*. Retrieved 2020-01-18, from <http://gazebo-sim.org/>
- Hernandez-Aceituno, J., Arnay, R., Toledo, J., & Acosta, L. (2016). Using kinect on an autonomous vehicle for outdoors obstacle detection. *IEEE Sensors Journal*, 16(10), 3603–3610.
- Kamarudin, K., Mamduh, S. M., Shakaff, A. Y. M., Saad, S. M., Zakaria, A., Abdullah, A. H., & Kamarudin, L. M. (2013). Method to convert kinect's 3d depth data to a 2d map for indoor slam. In *2013 IEEE 9th international colloquium on signal processing and its applications* (pp. 247–251).
- Nguyen, V., Martinelli, A., Tomatis, N., & Siegwart, R. (2005). A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics. In *2005 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1929–1934).
- Omara, H. I. M. A., & Sahari, K. S. M. (2015). Indoor mapping using kinect and ros. In *2015 international symposium on agents, multi-agent systems and robotics (isamsr)* (pp. 110–116).
- Rejas, J.-I., Sanchez, A., Glez-de Rivera, G., Prieto, M., & Garrido, J. (2015). Environment mapping using a 3d laser scanner for unmanned ground vehicles. *Microprocessors and Microsystems*, 39(8), 939–949.
- Robotics, O. (2019). *Robot operating system*. Retrieved 2020-01-18, from <https://www.ros.org/about-ros/>

- saaravagarwal. (2016). *How to setup kinect with ros and rgbd slam*. Retrieved 2020-01-18, from <https://www.sauravag.com/2016/10/how-to-setup-kinect-with-ros-and-rgbd-slam/>
- Shalal, N., Low, T., McCarthy, C., & Hancock, N. (2015). Orchard mapping and mobile robot localisation using on-board camera and laser scanner data fusion—part b: Mapping and localisation. *Computers and Electronics in Agriculture*, 119, 267–278.
- sick. (2019). *2d lidar sensors nav2xx*. Retrieved 2020-01-18, from <https://www.sick.com/us/en/detection-and-ranging-solutions/2d-lidar-sensors/nav2xx/c/g3>
- Sikora, J. (2020). *Integracja części manipulacyjnej i bazy mobilnej robota Velma na potrzeby symulacji* (Bachelor’s thesis). WEiTl.
- Z. Riaz A. Pervez, J. I., M. Ahmer. (2010, nov). *A fully autonomous indoor mobile robot using slam*. Retrieved from <https://ieeexplore.ieee.org/document/5625691>