POLITECHNIKA WARSZAWSKA

Wydział Mechatroniki

Praca magisterska

Jakub Mikołaj Szlendak

System lokalizacji robota mobilnego w pomieszczeniu zamkniętym na podstawie siły sygnału radiowego

Opiekun pracy:
prof. dr hab. Barbara Siemiątkowska

Konsultant pracy:
mgr inż. Daniel Koguciuk

Warszawa, 2017

Spis treści

Spis rysunków

Rozdział 1

Wstęp

Ostatnie lata przynoszą intensywny rozwój robotyki mobilnej. Roboty znajdują zastosowanie w coraz większej liczbie dziedzin, takich jak transport, przemysł czy wojsko. Niemniej jednak, koszt budowy robota mobilnego pozostaje wysoki, zaś znaczny ułamek kosztu urządzenia stanowią układy sensoryczne.

Z drugiej strony zauważamy równie intensywny rozwój i rozpowszechnienie technologii bezprzewodowych takich jak Bluetooth czy Wi-Fi, które są obecnie dostępnie w niemal każdym obiekcie, zaś koszt zaimplementowania tych technologii na robocie jest relatywnie niewielki. Dlatego coraz częściej pojawiaja się pomysły, aby wykorzystać technologie bezprzewodowe do lokalizacji robotów mobilnych, szczególnie, że lokalizacja satelitarna (GPS, GLONASS) jest niedostępna w pomieszczeniach zamkiętych.

Do najpopularniejszych metod lokalizacji należą m. in:

- lokalizacja w oparciu o wizualne znaczniki i system ich rozpoznawania
- lokalizacja na podstawie stereowizji
- odometria
- lokalizacja na podstawie odległości od znaczników (radiowych, akustycznych itp)
- lokalizacja w oparciu o skaner laserowy

Jedną z najpopularniejszych i najbardziej skutecznych spośród ww. metod jest lokalizacja w oparciu o skaner laserowy. Jednakże koszt takiego skanera wynosi ok. 2500 zł, podczas gdy zestaw znaczników radiowych działających w technologii Bluetooth wraz z kartą Wi-Fi / Bluetooth ze złączem PCI-E kosztuje ok. 200 zł (ceny na rok 2017 wg sklepu Kamami.pl). Ponadto, lokalizacja w oparciu o skaner laserowy posiada szereg wad, z których najpoważniejszą jest tzw. problem porwanego robota. Jeśli robot lokalizowany za pomocą skanera zostanie przeniesiony

w inne miejsce, w którym skan laserowy daje podobny kontur, robot nie będzie w stanie stwierdzić, że został przeniesiony. Jest to szczególnie poważny problem dla robotów pracujących w budynkach posiadających wiele podobnych pomieszczeń.

Biorąc pod uwagę powyższe rozważania, celem niniejszej pracy jest zbadanie przydatności znaczników radiowych wykorzystujących technologię Bluetooth w lokalizacji robota. Praca obejmuje następujące aspekty:

- Analizę teoretyczną i eksperymentalną zależności siły sygnału radiowego w zależności od odległości między odbiornikiem a nadajnikiem,
- Przegląd możliwych technik lokalizacji z wykorzystaniem znaczników radiowych,
- Implementację i analizę eksperymentalną wybranych algorytmów lokalizacji,
- Implementację i analizę eksperymentalną systemu wspomagania lokalizacji laserowej za pomocą znaczników radiowych.

W dalszej części rozdziału pokrótce przedstawiono lokalizację w oparciu o skaner laserowy, jej wady oraz zalety. W rozdziale drugim przedstawiono zarys teoretyczny problemu propagacji fal radiowych w pomieszczeniu, metody obliczania odległości w oparciu o siłę sygnału radiowego oraz dokonano przeglądu najpopularniejszych technologii radiowych pasma ISM. Rozdział trzeci jest poświęcony metodom lokalizacji, jakie są możliwe do zrealizowania za pomocą danych ze znaczników radiowych. W rozdziale czwartym opisano zrealizowany system lokalizacji, natomiast rozdział piąty zawiera wyniki testów i pomiarów systemu. Ostatni rozdział przedstawia interpretację i wnioski płynące z eksperymentów.

1.1 Metody lokalizacji w oparciu o odometrię i skaner laserowy

Sensory odometryczne oraz skanery laserowe (tzw. lidary) to obecnie jedne z najpopularniejszych sensorów wykorzystywanych do lokalizowania robotów mobilnych. Poniżej przedstawiono ich charakterystykę oraz główne wady i zalety.

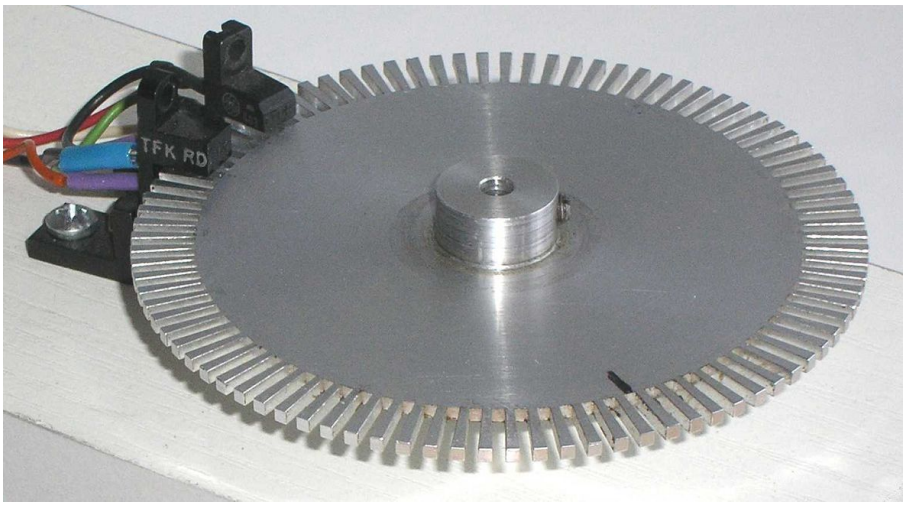
1.1.1 Odometria

Odometria polega na pośrednim pomiarze położenia robota na podstawie jego przemieszczeń. Przemieszczenia są wyznaczane na podstawie pomiaru kątu obrotu kół robota za pomocą enkoderów inkrementalnych. Podstawowym elementem takiego sensora jest tarcza kodowa z podziałką w postaci kresek na przemian odbijających i nieodbijających światła (lub przepuszczających

i nieprzepuszczających, rys. 1.1). Tarcza obraca się razem z napędem robota, zaś podziałka jest oświetlana za pomocą diody podczerwonej. Optyczny układ odczytowy zlicza impulsy świetle które odbiły się od odbijających kresek na tarczy. Znając kinematykę robota, na podstawie zliczonych impulsów można wyznaczyć kąt o jaki obróciły się koła oraz drogę jaką robot przebył. Odometria należy do metod zliczeniowych, tzn. położenie robota jest wyznaczane poprzez zliczanie impulsów. Aby to położenie było wyznaczone dokładnie, muszą być spełnione pewne założenia:

- średnice oraz rozstaw kół są znane,
- koła są współosiowe,
- styk kół z podłożem jest punktowy,
- nie zachodzi poślizg kół.

Niespełnienie powyższych założeń jest podstawowym źródłem błędu pomiaru odometrii, który kumuluje się z czasem. Dlatego odometria nie sprawdza się w lokalizacji na dłuższych dystansach. Natomiast radzi sobie bardzo dobrze w pomiarze przyrostu położenia, wykorzystywanym przez algorytmy lokalizacji oparte na filtrach bayesowskich [**?**]. Jej kolejną zaletą jest niska cena, prosta budowa i powszechna dostępność w większości dostępnych w sprzedaży platfom mobilnych.



Rysunek 1.1: Enkoder inkrementalny wykorzystywany w odometrii

1.1.2 Skaner laserowy

Współcześnie wykorzystywane dalmierze laserowe z funkcją skanowania wykorzystują metodę *time of flight*, polegającą na wysłaniu impulsu świetlnego w danym kierunku i mierzeniu czasu,

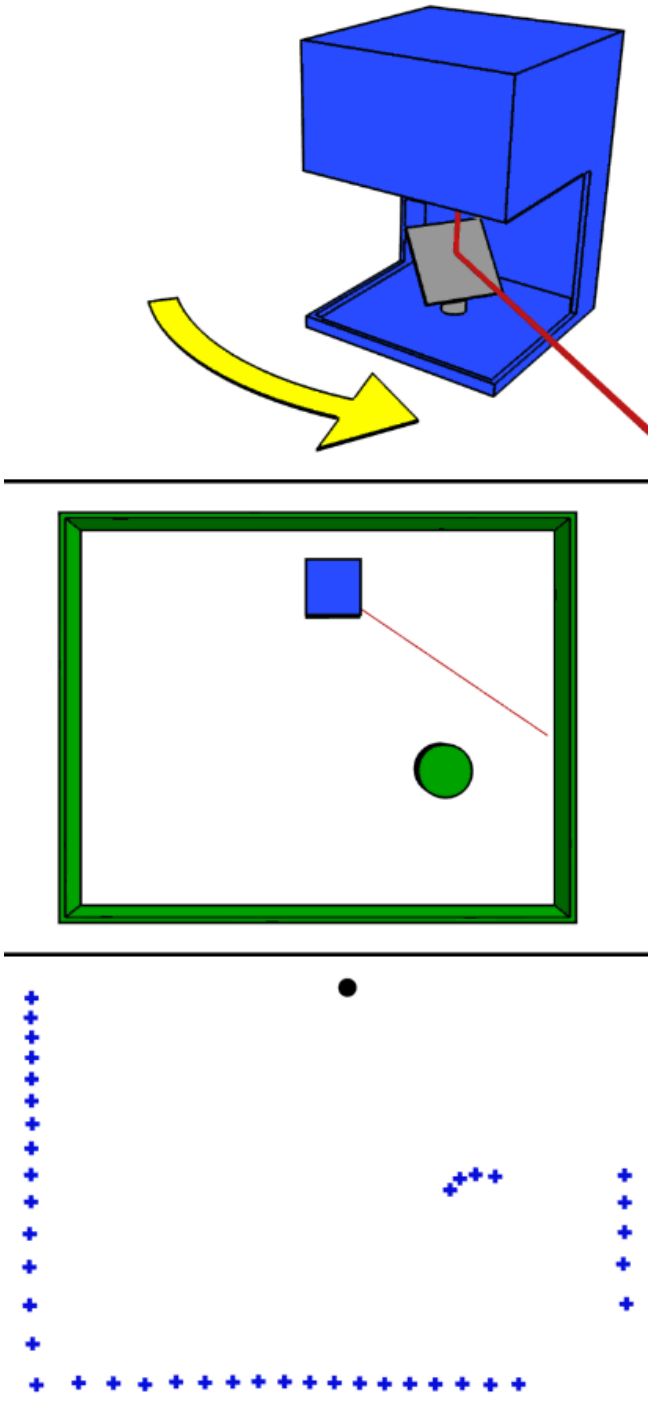
który upływa do powrotu impulsu odbitego od przeszkody. Na podstawie wartości tego czasu, znając prędkość światła, można wyznaczyć odległość od przeszkody. Aby osiągnąć pełny skan pomieszczenia, taki pomiar jest powtarzany w stałym interwale kątowym, np. $\Delta\phi = 0.50°$ w zakresie 180 ° lub większym. Zasadę działania skanera zobrazowano na rys. 1.2. Wynikiem działania skanera jest zbiór par (r_i, ϕ_i), określających odległość od przeszkody i kąt skanowania. Czas wykonania pełnego skanu jest rzędu 10 ms, zaś dokładność pomiaru odległości - rzędu 10 mm.

Skany pomieszczenia w połączeniu z danymi odometrycznymi mogą zostać wykorzystane do zbudowania mapy otoczenia za pomocą algorytmu SLAM (ang. Simultaneouos Localization and Mapping, *Jednoczesna Lokalizacja i Mapowanie*). Mapowanie z użyciem map SLAM polega na dopasowaniu skanu laserowego do konturu mapy i wyliczeniu translacji i rotacji pomiędzy konturem reprezentowanym przez skan laserowy, a konturem mapy, co pozwala na zlokalizowanie robota. Istnieje szereg algorytmów dopasowania skanu do mapy, opisanych szerzej w książce [**?**]. Połączenie danych ze skanera laserowego i odometrii za pomocą filtra cząsteczkowego pozwala na bardzo dobre lokalizowanie robota, zarówno w sytuacji nieznanej mapy (za pomocą SLAM) jak i przy wykorzystaniu już posiadanej mapy.

1.1.3 Problem porwanego robota

Problemem porwanego robota nazywamy sytuację, w której robot lokalizowany za pomocą odometrii i/lub skanera losowego zostanie zabrany i przeniesiony do innego otoczenia. Ponieważ takiemu przemieszczeniu robota nie towarzyszy żaden obrót kół, odometria nie jest w stanie go zarejestrować. Natomiast jeśli robot zostanie przeniesiony np. z pomieszczenia o prostokątnym rozkładzie ścian do innego pomieszczenia o identycznym rozkładzie, również skaner laserowy nie będzie w stanie zdeterminować położenia robota. W wypadku lokalizacji za pomocą filtra cząsteczkowego będzie to skutkowało powstaniem dwóch (lub więcej, jeśli jest więcej podobnych pomieszczeń) skupisk cząsteczek, z których nie można jednoznacznie wybrać właściwego położenia. Podobna niejednoznaczność zachodzi często zaraz po uruchomieniu robota, kiedy jego pozycja w globalnym układzie współrzędnych pozostanie nieznana. W takiej sytuacji, dopóki robot nie wykona przejazdu, podczas którego zostaną zarejestrowane punkty charakterystyczne, nie jest możliwe określenie jego lokalizacji.

Pomocą w takich sytuacjach może okazać się lokalizacja za pomocą znaczników radiowych. Choć jej dokładność jest niższa niż w wypadku skanera laserowego, to zgrubne wskazanie położenia na mapie może rozwiązać niejednoznaczności opisane powyżej i w rezultacie rozwiązać problem porwanego robota.



Rysunek 1.2: Zasada działania skanera laserowego. Od góry: 1. Kierowanie wiązki świetlnej. 2. Skanowane pomieszczenie. 3. Wynikowy zbiór punktów.

Rozdział 2

Zastosowanie znaczników radiowych do mierzenia odległości

Systemy nawigacji satelitarnej takie jak GPS czy GLONASS, jak i systemy lokalizacji za pomocą sieci komórkowej nie znajdują zastosowania w lokalizacji wewnątrz pomieszczeń. Podstawowymi problemami, na jakie natrafiają te systemy, jest brak zasięgu wewnątrz pomieszczeń i związana z tym zbyt mała dokładność. Jednocześnie popularyzacja sieci Wi-Fi i standardu komunikacji bezprzewodowej Bluetooth spowodowała, że urządzenia wykorzystujące te standardy są szeroko dostępne i przystępne cenowo. Sieci Wi-Fi i standard Bluetooth operują paśmie ISM (Industrial, Scientific and Medical, częstotliwości rzędu 2.4 GHz). Dobra dostępność tych technologii skłania do wykorzystania ich także w robotyce.

W niniejszym rozdziale przedstawiono podstawy teoretyczne modelu propagacji fal radiowych w pomieszczeniu, następnie opisano problem modelowania relacji odległość - siła sygnału i możliwe jego rozwiązania. Przedstawiono także najpopularniejsze technologie radiowe pasma ISM, z naciskiem na technologię Bluetooth, wybraną do implementacji rozwiązania.

2.1 Propagacja sygnału radiowego w pomieszczeniu zamkniętym

Do szacowania poziomu sygnału w modelach propagacyjnych wprowadza się parametr zwany współczynnikiem tłumienności trasy. Opisuje on zmniejszenie poziomu sygnału odbieranego w stosunku do mocy sygnału nadawanego w skutek przejścia fal radiowych przez środowisko propagacyjne wprowadzające pewne tłumienie. Transmisję sygnału radiowego opisuje schemat przedstawiony na rys. 2.1. Przez P_{TX} oznaczono moc nadajnika, G_{TX} - zysk anteny nadawczej, G_{RX} - zysk anteny odbiorczej, L - współczynnik tłumienności trasy (ang. *path loss index*),

zaś P_{RX} - moc sygnału odebranego [**?**].

Współczynnik L opisuje własności propagacyjne środowiska, w którym przekazywany jest sygnał. Obejmuje zatem czynniki takie jak tłumienność trasy, obecność ścian budynków lub innych przeszkód terenowych, obecność innych źródeł promieniowania. Związane z wymienionymi czynnikami zjawiska odbicia, dyfrakcji i interferencji sygnału powodują, że zagadnienie wyznaczenia współczynnika tłumienności jest niezwykle trudne. W związku z tą trudnością, najczęściej stosowane są modele empiryczne, a zatem oparte na pomiarach siły sygnału w danym środowisku.

Relację mocy sygnału nadanego i mocy odebranej przez odbiornik można wyrazić następująco:

$$L = P_{TX} - P_{RX} \tag{2.1}$$

Do obliczenia odległości na podstawie siły sygnału RSSI (ang. *Received Signal Strength Indicator*, wskaźnik siły sygnału sygnału odebranego) należy posłużyć się modelem propagacji:

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} + 20\log(\lambda) - 20\log(4\pi) - 10n\log(d) \tag{2.2}$$

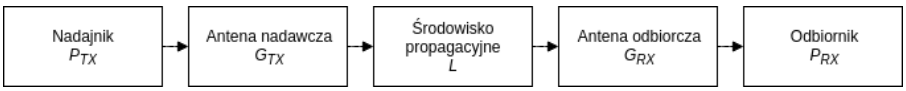
W równaniu 2.2 przez d oznaczono odległość między transmiterem a odbiornikiem, λ oznacza długość fali, zaś współczynnik n określa wpływ przeszkód takich jak ściany. Należy zaznaczyć, że wartości mocy nadajnika P_{TX}, zysków anten G_{RX} i G_{TX}, długości fali λ oraz współczynnika n na ogół nie są znane, podczas gdy moc sygnału odebranego P_{RX} jest zwykle możliwa do określenia, jako że implementacje protokołów radiowych przewidują dołączenie jej wartości do zdekodowanych ramek danych.

Ponadto, wyznaczenie analityczne współczynnika n jest niemożliwe. Zależy on bowiem bardzo silnie od architektury pomieszczenia, położenia nadajnika względem ścian, podłogi i stropu, a także od zakłóceń. Jako podstawowe źródła zakłóceń sygnału można wyróżnić:

- odbiorniki elektryczne dużej mocy, np. kuchenki lub silniki elektryczne,
- silne źródła promieniowania, takie jak kuchenki mikrofalowe,
- inne nadajniki radiowe korzystające z danego pasma, szczególnie bardzo popularne punkty dostępowe sieci Wi-Fi.

Obecność ścian, przedmiotów a także obiektów ruchomych, np. przemieszczających się ludzi, powoduje występowanie odbić i rozproszeń fali radiowej. Pod pewnymi względami może zajść także tzw. efekt falowodowy, wskutek którego fala pokonuje znacznie większą odległość bez tłumienia (dzieje się tak np. w długich korytarzach). Powyższe zjawiska skutkują wzrostem

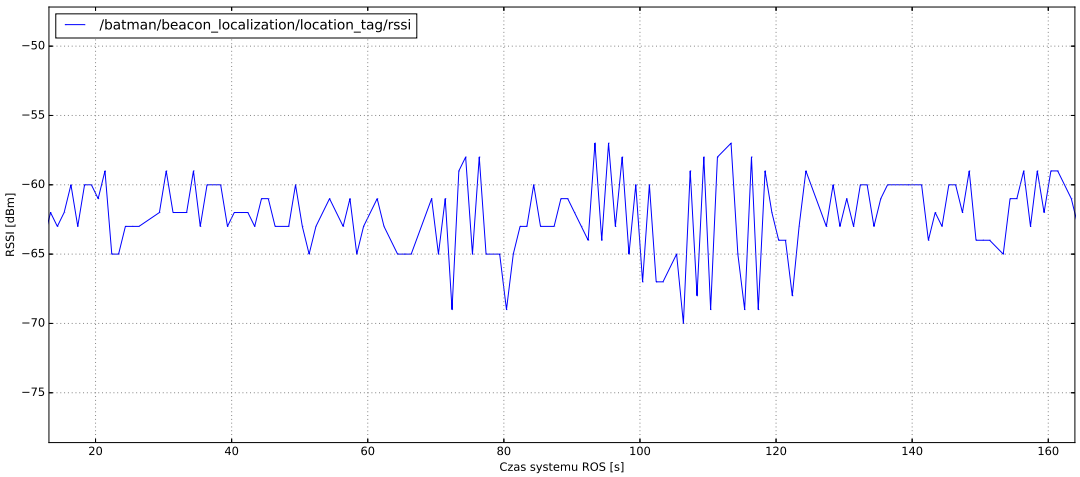
zjawiska wielokrotnego obicia, czyli docierania fali do odbiornika wieloma drogami (np. bezpośrednio z nadajnika i po odbiciu od ściany). To zjawisko jest szczególnie poważną przeszkodą do zastosowania pomiaru siły sygnału RSSI do wyznaczenia odległości.



Rysunek 2.1: Schemat propagacji fal radiowych

2.2 Wyznaczanie odległości na podstawie siły sygnału RSSI

Zgodnie z zależnością 2.2, wraz ze wzrostem odległości nadajnika i odbiornika, wartość RSSI powinna spadać. Jednakże z powodu zjawisk wymienionych w poprzedniej sekcji, ta zależność nie zawsze jest zachowana. Dla odbiornika pozostającego w stałej odległości od nadajnika, wartość RSSI nie pozostaje stała, co obrazują wyniki eksperymentalne (wykres na rys. 2.2).



Rysunek 2.2: Przebieg wartości RSSI dla stacjonarnego odbiornika będącego w odległości 70 cm od nadajnika (znacznik BLE). Pominięto pierwsze 15 sekund pomiaru ze względu na stan nieustalony systemu ROS.

Jak widać z wykresu 2.2, nawet dla nieruchomego odbiornika występują wahania RSSI rzędu 10 dBm. Dlatego konieczne jest zastosowanie filtra wygładzającego. Do wygładzenia wartości RSSI można wykorzystać następujące narzędzia:

- filtr oparty o średnią ruchomą prostą lub ważoną,

- filtr dolnoprzepustowy,
- filtr Kalmana, uwzględniający wiedzę na temat ruchu odbiornika.

Projektując filtr RSSI, należy zwrócić szczególną uwagę na bezwładność, jaką wprowadza on do pomiaru, a która utrudnia lub nawet uniemożliwia pomiar odległości dla odbiornika pozostającego w ruchu. W tym kontekście, szczególnie użyteczny jest filtr Kalmana, który może wykorzystać informację o ruchu odbiornika radiowego pochodzącą np. z systemu lokalizacji robota. Szczegóły dotyczące filtracji sygnału RSSI zostały szerzej omówione w rozdziale 3, gdzie opisano proponowany algorytm lokalizacji robota.

Kolejnym problemem związanym z pomiarem odległości za pomocą siły sygnału RSSI jest wyznaczenie zależności siły sygnału RSSI od odległości $P_{RX}(d)$. Wzór 2.2 zawiera parametry nadajnika i odbiornika oraz medium transmisyjnego, które na ogół nie są znane. Z pomocą przychodzą metody empiryczne. Wykonując serię pomiarów siły sygnału w różnych odległościach od nadajnika i w różnych miejscach pomieszczenia, otrzymuje się zbiór danych który pozwala na wyznaczenie aproksymacji zależności $P_{RX}(d)$. W dalszej części rozdziału opisano podstawowe metody otrzymywania tej aproksymacji.

2.2.1 Aproksymacja za pomocą krzywych wielomianowych

Pierwszym możliwym rozwiązaniem jest dopasowanie krzywej wielomianowej do punktów pomiarowych, proponowane przez autorów artykułu [**?**]. Aproksymowana funkcja ma postać:

$$y_{P_{RX}} = a_0 + a_1 d_x + a_2 d_x^2 + ... + a_n d_x^n \tag{2.3}$$

gdzie $a_0, ..., a_n$ są współczynnikami wielomianu stopnia n, aproksymującego zależność. Mając N punktów pomiarowych $(d_{xi}, yP_{RX}i)$, gdzie $i = 1, 2, ..., N$, błąd aproksymacji e_i wynosi

$$e_i = y_{P_{RX}} - (a_0 + a_1 d_x + a_2 d_x^2 + ... + a_n d_x^n) \tag{2.4}$$

Współczynniki $a_0, ..., a_n$ wyznacza się rozwiązując zadanie minimalizacji błędu średniokwadratowego:

$$S = \sum_{i=1}^{N} e_i^2 \tag{2.5}$$

Wyniki przedstawione w artykule [**?**] wskazują, że najlepsze wyniki dają aproksymacje wielomianami niskiego rzędu $n = 2, n = 3$

2.2.2 Interpolacja

Kolejną metodą aproksymowania zależności $P_{RX}(d)$ proponowaną przez [**?**] jest interpolacja. Pozwala ona na wyeliminowanie niewiadomych składników równania. Rozważmy dwa wybrane pomiary. Oznaczając je indeksami 1 i 2 oraz podstawiając do równania 2.2 otrzymujemy:

$$P_{RX1} = P_{TX} + G_{TX} + G_{RX} + 20\log(\lambda) - 20\log(4\pi) - 10n\log(d_1) \tag{2.6}$$

$$P_{RX2} = P_{TX} + G_{TX} + G_{RX} + 20\log(\lambda) - 20\log(4\pi) - 10n\log(d_2), \tag{2.7}$$

gdzie P_{RX} jest mocą odebranego sygnału, P_{TX} - mocą sygnału nadawanego, G_{RX} i G_{TX} odpowiednio wzmocnieniem anteny odbiorczej i nadawczej, λ - długością fali sygnału radiowego, zaś d - odległością odbiornika od nadajnika.

Odjęcie powyższych równań stronami pozwala wyeliminować niewiadome zależne od sprzętowych aspektów nadajnika i odbiornika, a następnie wyprowadzić zależność interpolowaną postaci:

$$P_{RX} = (P_{RX1} - P_{RX2})\frac{\log d - \log d_2}{\log d_1 - \log d_2} + P_{RX2} \tag{2.8}$$

Wykorzystując dwa punkty pomiarowe i zależność 2.8 można otrzymać aproksymację zależności RSSI od odległości. Punkty do interpolacji należy dobrać eksperymentalnie, aby osiągnąć optymalne wyniki.

2.2.3 Dopasowanie krzywej logarytmicznej lub wykładniczej

Łatwo zauważyć, że w równaniu 2.2 część składników jest stała i zależy od uwarunkowań sprzętowych nadajnika i odbiornika, bądź warunków środowiskowych. Grupując składniki stałe można napisać:

$$P_{RX} = A + B\log(d) \tag{2.9}$$

gdzie:

$$A = P_{TX} + G_{TX} + G_{RX} + 20\log(\lambda) - 20\log(4\pi) \tag{2.10}$$

$$B = -10n \tag{2.11}$$

Wykorzystując narzędzia optymalizacyjne, dostarczane przez pakiety takie jak MATLAB lub bibliotekę SciPy można wykonać dopasowanie krzywej logarytmicznej do punktów pomiarowych, aby otrzymać zależność postaci 2.9 [**?**]. Dokonując odpowiednich przekształceń, można

łatwo wyznaczyć także zależność odwrotną, faktycznie potrzebną w zadaniu lokalizacji. Ma ona postać:

$$d = 10^{\frac{P_{RX}-A}{B}} \tag{2.12}$$

W dalszej części rozdziału opisano popularne technologie radiowe, możliwe do wykorzystania w lokalizacji robota mobilnego.

2.3 Protokół Wi-Fi

Nazwa Wi-Fi (ang. Wireless Fidelity, *Bezprzewodowa wierność*) jest potocznym określeniem zestawu standardów określających budowę bezprzewodowych sieci komputerowych, w szczególności sieci WLAN, czyli bezprzewodowych sieci lokalnych. Wi-Fi wykorzystuje częstotliwości pasma ISM: 2.4 GHz oraz 5.8 GHz. Warstwa fizyczna oraz podwarstwa MAC jest opisywana przez grupę standardów IEEE 802.11. Duża popularność sieci Wi-Fi w budynkach sprawia, że rozwiązanie to dobrze nadaje się do zastosowania w lokalizacji. Istnieje szereg opracowań naukowych wykorzystujących technologię Wi-Fi do lokalizacji wewnątrz pomieszczeń ([**?**]), nie tylko w robotyce, ale także m. in. do lokalizowania osób posiadających telefon komórkowy ([**?**]). Zasilanie sieciowe punktów dostępych pozwala na szybkość transmisji danych znacznie większą niż w innych technologiach radiowych. Jednakże, wadą rozwiązania jest duże zużycie energii, szczególnie po stronie odbiornika który często zasilany jest bateryjnie, oraz koszt urządzeń i ich instalacji większy niż w wypadku opisanego dalej Bluetooth.

2.4 Protokół Bluetooth

Bluetooth jest standardem komunikacji bezprzewodowej, zaprojektowanym do wymiany danych na krótkie dystanse, korzystający z fal krótkich UHF pasma ISM (częstotliwości rzędu 2.4 GHz). Standard Bluetooth jest zarządzany przez organizację Bluetooth Special Interest Group (SIG), będącą zrzeszeniem firm z branż komunikacyjnej, sieciowej, informatycznej i elektronicznej [**?**].

Pasmo ISM (ang. *Industrial, Scientific and Medical*, Przemysłowe, Naukowe i Medyczne) wykorzystywane przez Bluetooth jest nielicencjonowane. Dla zapewnienia odporności na zakłócenia, Bluetooth wykorzystuje technologię rozpraszania widma FHSS (ang. *Frequency Hopping Spread Spectrum*) polegającą na "skakaniu" po częstotliwościach dostępnych w paśmie w kolejnych odstępach czasu. Protokół Bluetooth jest oparty o pakiety, ze strukturą master-slave [**?**].

2.4.1 Bluetooth Low Energy

Bluetooth Low Energy, znane także pod nazwą handlową Bluetooth Smart, jest czwartą wersją specyfikacji standardu Bluetooth. Standard ten został zorientowany na zapewnienie transmisji o minimalnym koszcie energetycznym. Podstawowe cechy BLE to:

- niskie zużycie energii, pozwalające urządzeniom operować przez długi (rzędu roku) czas na jednej baterii guzikowej,
- mały rozmiar i koszt urządzeń,
- kompatybilność z popularnymi urządzeniami obecnymi na rynku (smartfony, tablety, laptopy).

Dla osiągnięcia efektywności energetycznej BLE wykorzystuje odmienny w stosunku do klasycznego Bluetooth model komunikacji. Występują dwa tryby pracy:

- tryb rozgłoszeniowy (ang. *advertising mode*),
- tryb połączenia (ang. *connection mode*).

W trybie rozgłoszeniowym urządzenie wysyła pakiety nieskierowane do żadnego odbiornika w stałym interwale czasowym. Pomiędzy nadawaniem procesor oraz radio urządzenia mogą pozostawać w stanie uśpienia, co pozwala na znaczną oszczędność energii. Pojedynczy pakiet rozgłoszeniowy zawiera 31 bajtów użytecznych danych i może być odebrany przez dowolne urządzenie w trybie nasłuchu (ang. *scan mode*). Nasłuchujące odbiorniki, po przechwyceniu pakietu rozgłoszeniowego i tym samym odkryciu urządzenia nadawczego, mogą nawiązać z nim połączenie. Tryb połączenia, podobnie jak w klasycznym Bluetooth, jest nawiązywany pomiędzy dwoma urządzeniami i służy do wymiany danych pomiędzy nimi. Przebywając w trybie rozgłoszeniowym, średnie zużycie prądu przez urządzenie Bluetooth może pozostawać na poziomie μA, co pozwala osiągnąć wspomniane wcześniej długie czasy pracy na baterii. Ma to szczególne znaczenie w robotyce mobilnej, gdzie roboty na ogół pracują na zasilaniu akumulatorowym.

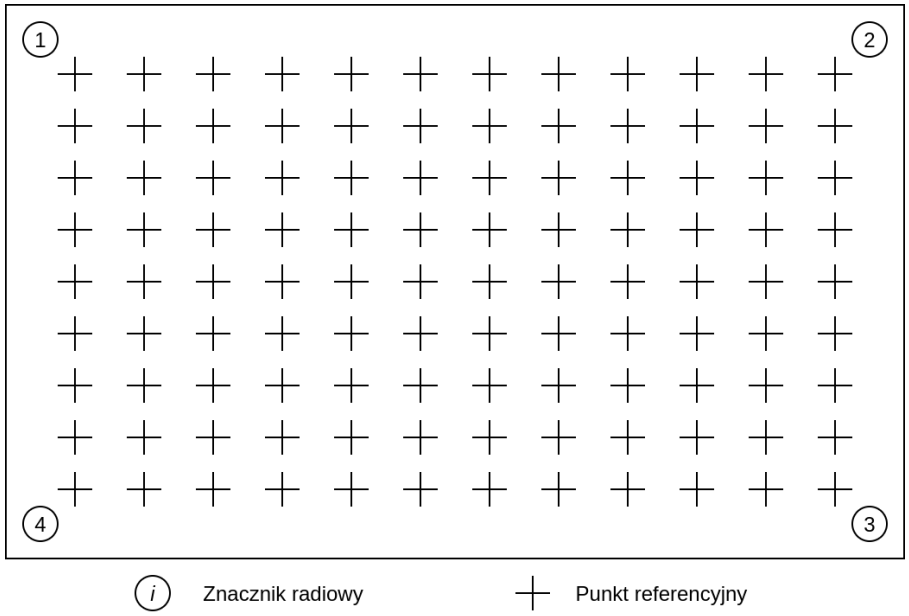
Rozdział 3

Metody lokalizacji na podstawie znaczników radiowych

Znaczniki radiowe, rozmieszczone w ustalonych miejscach środowiska pracy robota, pozwalają na podjęcie zadania lokalizacji na kilka różnych sposobów. W niniejszym rozdziale przedstawiono podstawowe metody lokalizacji.

3.1 Fingerprinting

Metoda fingerprintingu (ang.*fingerprint* - odcisk palca) opiera się na zbiorze punktów referencyjnych. Punkty referencyjne zbiera się w fazie off-line, zapisując w bazie danych położenie danego punktu w przestrzeni oraz referencyjny wektor RSSI postaci $P_{ref} = [P_{ref\ 1} \cdots P_{ref\ N}]$, gdzie N jest liczbą znaczników w środowisku, zaś $P_{ref\ i}$ oznacza siłę sygnału RSSI z i-tego znacznika [**?**]. Przykładowe rozmieszczenie znaczników i punktów referencyjnych pokazuje schemat na rys. 3.1.



Rysunek 3.1: Rozmieszczenie znaczników radiowych i punktów referencyjnych w metodzie fingerprintingu. Znaczniki radiowe oznaczono kółkami z cyfrą, punkty referencyjne

W fazie on-line lokalizowany odbiornik rejestruje wektor siły sygnału $P = [P_{RX\ 1} \cdots P_{RX\ n}]$ dla danego położenia, następnie obliczana jest odległość wektora P od wektorów referencyjnych P_{ref} według metryki euklidesowej [**?**]:

$$D(Z, Z_i) = \sqrt{\sum_{j=1}^{N} (P_{ref\ i\ j} - P_{RX\ j})^2}. \tag{3.1}$$

Z odległością 3.1 jako kryterium, można wyszukać K najbliższych sąsiadów wektora P_{RX} w zbiorze punktów referencyjnych P_{ref}. Publikacja [**?**] proponuje następujące podejścia:

1. $K = 1$ - za wynik lokalizacji przyjmowany jest najbliższy punkt referencyjny,
2. $K > 1$ - za wynik lokalizacji przyjmowany jest środek ciężkości najbliższych K punktów referencyjnych.

Wyszukiwanie sąsiadów można prowadzić na pełnym zbiorze punktów referencyjnych lub tylko na jego części, ograniczonej np. do pewnego promienia wokół poprzedniej pozycji. Pozwala to na przyspieszenie wyszukiwania [**?**]. Jakość lokalizacji można dodatkowo poprawić wykorzystując metody probabilistyczne takie jak filtr Bayesa lub cząsteczkowy [**?**].

Niewątpliwą zaletą lokalizacji w oparciu o fingerprinting jest to, iż uwzględnia ona nieizotropowy rozkład siły sygnału RSSI w zależności od położenia odbiornika. Ponadto nie jest konieczne wyznaczanie modelu propagacji radiowej do przeliczania RSSI na odległość. Jeśli

siatka punktów referencyjnych jest odpowiednio gęsta, fingerprinting jest w stanie zapewnić zadowalające rezultaty. Z drugiej strony, lokalizacja tą metodą wymaga przeprowadzenia dość złożonej kalibracji w fazie off-line (budowanie zbioru punktów referencyjnych), ponadto konieczne jest zapewnienie metody wydajnego przeszukiwania zbioru referencyjnego, zaś jeśli zbiór ten zawiera mało punktów, jakość lokalizacji może być niezadowalająca.

3.2 Trilateracja

Trilateracja, obok triangulacji, jest jedną z metod geometrycznych wyznaczania położenia obiektu za pomocą pomiaru kąta lub odległości względem ustalonego węzła. Triangulacja opiera się na pomiarze kątów, natomiast trilateracja - odległości. Dzięki temu możliwe jest zastosowanie jej w lokalizacji w oparciu o pomiar siły sygnału RSSI, związanej z odległością.

Zadanie trilateracji polega na wyznaczeniu położenia lokalizowanego obiektu, znając odległości tego obiektu od 3 (przypadek dwuwymiarowy) lub 4 (przypadek trójwymiarowy) punktów referencyjnych o znanym położeniu. W dalszej części pracy rozważany będzie tylko przypadek dwuwymiarowy. Oznaczmy i-ty punkt referencyjny przez $O_i(x_i, y_i)$. Odległość punktu referencyjnego od lokalizowanego obiektu znajdującego się w punkcie $P(x, y)$ wynosi R_i (rys. 3.2). O ile punkty O_i nie są współliniowe, punkt P znajduje się na przecięciu okręgów (O_1, R_1), (O_2, R_2), (O_3, R_3). Dlatego rozwiązanie zadania trilateracji można zapisać następująco [**?**]:

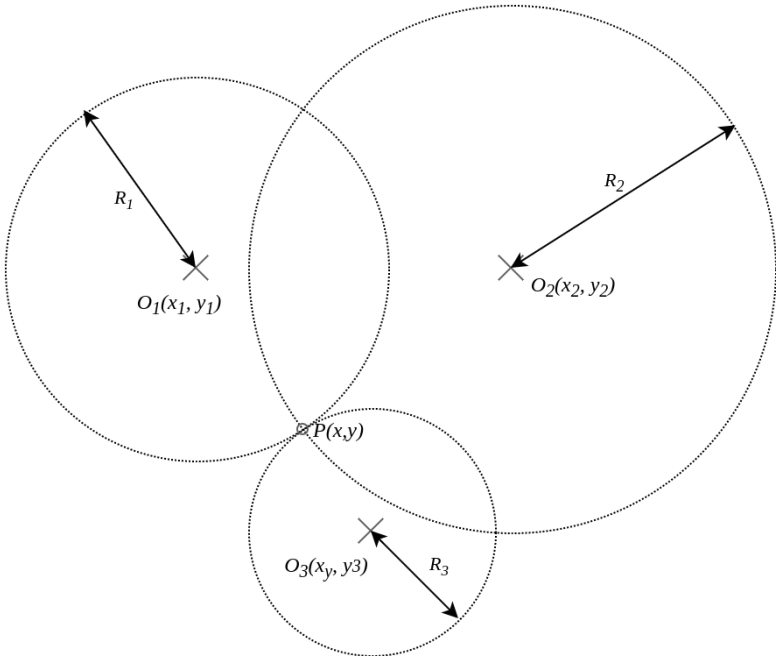
$$(x - x_1)^2 + (y - y_1)^2 = R_1^2 \tag{3.2}$$

$$(x - x_2)^2 + (y - y_2)^2 = R_2^2 \tag{3.3}$$

$$(x - x_3)^2 + (y - y_3)^2 = R_3^2 \tag{3.4}$$

Jest to układ równań nieliniowych. Ponieważ liczba punktów referencyjnych może być większa od 3, możemy mieć do czynienia z układem nadokreślonym. Do rozwiązania takiego układu można wykorzystać metodę macierzy pseudo-odwrotnych [**?**]. Generalnie, analityczne rozwiązanie takiego układu równań przedstawia znaczne trudności lub jest niemożliwe [**?**], natomiast rozwązanie numeryczne jest obarczone znacznym błędem [**?**]. Ponadto, w realistycznym przypadku, okręgi nie będą przecinać się w jednym punkcie. Możliwe jest, że będzie więcej punktów przecięcia (rys. 3.3) lub okręgi nie będą się przecinać (rys. 3.4). W obu tych przypadkach powyższy układ równań nie będzie miał rozwiązań.

W celu osiągnięcia dobrego rozwiązania numerycznego zadania trilateracji, można posłużyć się algorytmem iteracyjnym [**?**]. Algorytm wymaga, żeby znane były położenia co najmniej 3 punktów referencyjnych (x_i, y_i), oraz zmierzone odległości lokalizowanego obiektu od tych



Rysunek 3.2: Trilateracja - przypadek idealny

punktów R_i, oraz początkowa estymata rozwiązania (x_e, y_e). W każdej iteracji obliczana jest różnica między zmierzoną, a estymowaną odległością obiektu od każdego punktu referencyjnego:

$$f_i = |d_i - \sqrt{(x_i - x_e)^2 + (y_i - y_e)^2}| \tag{3.5}$$

Oznaczając przez $\vec{R}_k$ estymatę rozwiązania w k-tej iteracji:

$$\vec{R}_k = \begin{pmatrix} x_e \\ y_e \end{pmatrix} \tag{3.6}$$

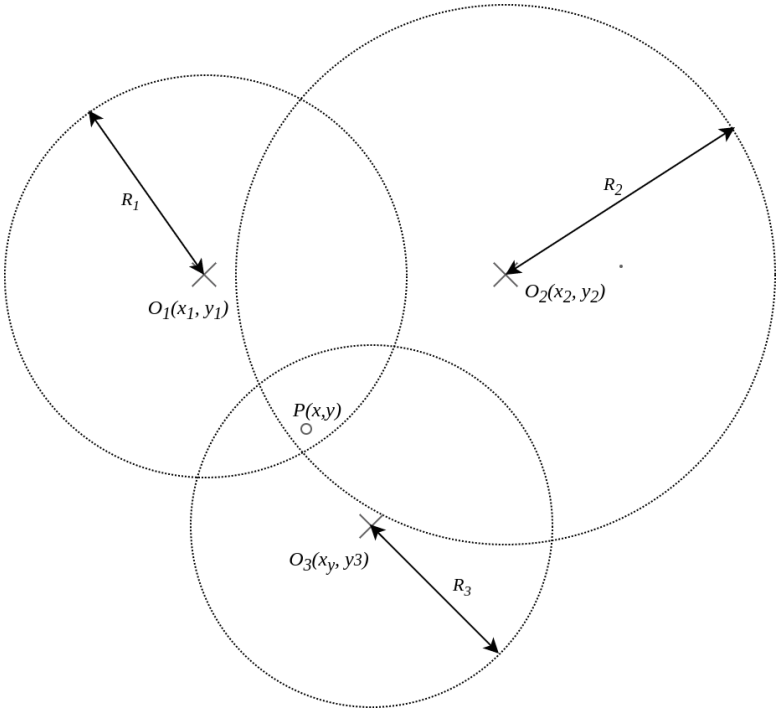
i posługując się metodą Newtona [**?**] możemy przedstawić równanie następnej iteracji:

$$\vec{R}_{k+1} = \vec{R}_k - (\mathbf{J}_k^T \mathbf{J}_k)^{-1} \mathbf{J}_k^T \vec{f}_k \tag{3.7}$$

gdzie:

$$\vec{f}_k = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}, \tag{3.8}$$

zaś przez $\mathbf{J}_k$ oznaczamy macierz Jakobiego funkcji f:



Rysunek 3.3: Trilateracja - przypadek rzeczywisty - okręgi nakładają się

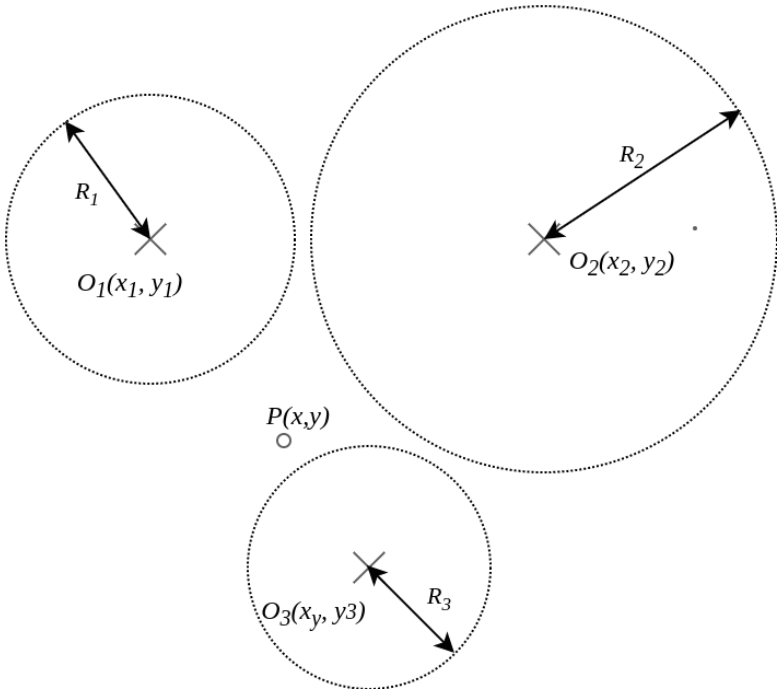
$$\mathbf{J}_k = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \\ \vdots & \vdots \\ \frac{\partial f_n}{\partial x} & \frac{\partial f_n}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{(x_1-x_e)}{\sqrt{(x_i-x_e)^2+(y_i-y_e)^2}} & \frac{(y_1-y_e)}{\sqrt{(x_i-x_e)^2+(y_i-y_e)^2}} \\ \frac{(x_2-x_e)}{\sqrt{(x_i-x_e)^2+(y_i-y_e)^2}} & \frac{(y_2-y_e)}{\sqrt{(x_i-x_e)^2+(y_i-y_e)^2}} \\ \vdots & \vdots \\ \frac{(x_3-x_e)}{\sqrt{(x_i-x_e)^2+(y_i-y_e)^2}} & \frac{(y_n-y_e)}{\sqrt{(x_i-x_e)^2+(y_i-y_e)^2}} \end{pmatrix}. \tag{3.9}$$

Aby wyznaczyć rozwiązanie zadania trilateracji, należy wykonywać kolejne iteracje metody Newtona zgodnie z równaniem 3.7. Jako estymatę początkową można przyjąć np. rozwiązanie zlinearyzowanego układu równań 3.2 - 3.4.

W porównaniu do fingerprintingu, trilateracja nie wymaga złożonej fazy off-line. Z drugiej strony, jeśli do pomiaru odległości wykorzystywana jest siła sygnału RSSI, pomiar jest obarczony wszystkimi błędami opisanymi w rozdziale 2, w szczególności nie ma możliwości, aby uwzględnić nieizotropowy rozkład siły sygnału w pomieszczeniu.

3.3 Filtr cząsteczkowy

Filtr cząsteczkowy, należący do metod Monte Carlo, jest bardzo popularnym narzędziem w robotyce mobilnej. Jest to nieparametryczny, quasi-bayesowski filtr probabilistyczny, pozwalający na reprezentowanie stanu robota w postaci funkcji gęstości prawdopodobieństwa, radzący



Rysunek 3.4: Trilateracja - przypadek rzeczywisty - okręgi wogóle się nie przecinają

sobie z nie-gaussowskimi i nieliniowymi modelami pomiaru. Poniżej przedstawiono podstawy teorii filtrów cząsteczkowych oraz możliwe zastosowanie w lokalizacji na podstawie odległości od znaczników radiowych.

W filtrze cząsteczkowym stan robota jest traktowany jako zmienna losowa $x \in X$. Dla robota poruszającego się na płaszczyźnie lokalizacja opisywana jest przez trzy współrzędne: $x = (x, y, \theta)$, gdzie x, y są współrzędnymi pojazdu w układzie współrzędnych związanym z środowiskiem pracy, zaś θ jest orientacją robota w tym układzie. Rozkład gęstości prawdopodobieństwa położenia robota $p(x_k|z_k)$ w chwili k jest reprezentowany przez dyskretny zbiór N_p cząsteczek, z których każda reprezentuje hipotezę na temat położenia robota x_i oraz przypisaną jej wagę w_i, opisującą stopień potwierdzenia tej hipotezy [**?**], [**?**]:

$$S_k = \{x_i, w_i\}, i = 0, 1, \cdots, N_p, 0 \leqslant w_i \leqslant 1, \tag{3.10}$$

Algorytm filtra cząsteczkowego składa się z kilku etapów:

1. *Faza predykcji.* Obliczany jest rozkład gęstości a posteriori w chwili k: $p(x_k|x_{k-1}, u_{k-1})$ na podstawie rozkładu gęstości prawdopodobieństwa położenia robota w chwili poprzedniej $p(x_{k-1}|z_{k-1})$ oraz wektora sterowań u_{k-1}. Uwzględniany jest model ruchu robota -najczęściej wykorzystywane są sensory odometryczne. Akcja sterowania u_{k-1} jest aplikowana dla każdej cząsteczki ze zbioru S_{k-1}. W wyniku powyższej operacji otrzymujemy

uaktualniony zbiór cząsteczek:

$$S'_k = \{x'_i, w'_i\}, i = 0, 1, \cdots, N_p. \tag{3.11}$$

2. *Pomiar.* Na podstawie pomiaru każdej cząsteczce ze zbioru S'_k przypisywana jest nowa waga w_i. Wagi są wyznaczane na podstawie rozkładu gęstości prawdopodobieństwa pomiaru: $p(z_k|x_k)$. Wynikiem jest nowy zbiór S_k, którego wagi są normalizowane:

$$w_i = \frac{w_i}{\sum_{j=1}^{N_p} w_j} \tag{3.12}$$

Wagi cząsteczek w nowym zbiorze S_k estymują rozkład gęstości prawdopodobieństwa położenia robota $p(x_k|z_k)$.

3. *Resampling.* Po kilku iteracjach wg. punktów 1 i 2, większość cząsteczek miała by pomijalnie małe wagi, przez co estymacja gęstości prawdopodobieństwa $p(x_k|z_k)$ byłaby mało dokładna (estymowana małą ilością cząsteczek). Stąd konieczność dokonania resamplingu cząsteczek (ang. ponowne próbkowanie). Cząsteczki o największych wartościach wag w_i są powielane w zbiorze, natomiast cząsteczki o małych wagach są usuwane ze zbioru [**?**]. Resampling jest zagadnieniem złożonym i istnieje kilka różnych algorytmów do wykonania tego zadania [**?**], [**?**].

4. *Wyznaczenie położenia robota.* Na podstawie zbioru cząsteczek S_k należy wyznaczyć położenie robota, będące wynikiem działania filtra. Jako położenie robota można przyjąć położenie reprezentowane przez cząsteczkę o największej wadze, lub średnią ważoną cząsteczek (lub średnią ważoną cząsteczek w pewnym otoczeniu cząsteczki o największej wadze) [**?**]:

$$x_k = \sum_{j=0}^{N_p} w_j \cdot x_j \tag{3.13}$$

3.3.1 Filtr cząsteczkowy dla robota wyposażonego w sensor odometryczny i system pomiaru odległości od znaczników

Rozważmy robota o napędzie różnicowym, wyposażonego w sensory odometryczne oraz system wyznaczania odległości od znaczników znajdujących się w ustalonych miejscach środowiska pracy robota, np. system działający w oparciu o znaczniki radiowe. Stan robota jest opisywany przez wektor x postaci:

$$x_k = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \tag{3.14}$$

.

Model ruchu

Ruch robota, rozumiany jako względna różnica dwóch położeń $x_{k-1} = (x\ y\ \theta)^T$ oraz $x_k = (x'\ y'\ \theta')^T$, może być zamodelowany jako złożenie trzech ruchów prostych: obrotu o kąt δ_{rot1}, przemieszczenia prostoliniowego o długość δ_{trans} oraz obrotu o kąt δ_{rot2}. Mając dane dwa wektory położenia robota, x_{k-1} oraz x_k, parametry tych ruchów prostych można wyznaczyć następująco [**?**]:

$$\delta_{rot1} = \text{atan2}(y' - y, x' - x) - \theta, \tag{3.15}$$

$$\delta_{trans} = \sqrt{(x - x')^2 + (y - y')^2}, \tag{3.16}$$

$$\delta_{rot2} = \theta' - \theta - \delta_{rot1}. \tag{3.17}$$

Aby uwzględnić błąd wyznaczenia przemieszczenia, należy uwzględnić szum pomiarowy ε_b o rozkładzie normalnym, zerowej wartości średniej i wariancji b:

$$\hat{\delta}_{rot1} = \delta_{rot1} - \varepsilon_{\alpha_1|\delta_{rot1}|+\alpha_2|\delta_{trans}|}, \tag{3.18}$$

$$\hat{\delta}_{trans} = \delta_{trans} - \varepsilon_{\alpha_3|\delta_{trans}|+\alpha_4|\delta_{rot1}+\delta_{rot2}|}, \tag{3.19}$$

$$\hat{\delta}_{rot2} = \delta_{rot2} - \varepsilon_{\alpha_1|\delta_{rot2}|+\alpha_2|\delta_{trans}|}. \tag{3.20}$$

Parametry $\alpha_1, \cdots, \alpha_4$ są zależne od budowy robota.

Ostatecznie, równanie uaktualnienia ze stanu x_{k-1} do stanu x_k wygląda następująco [**?**]:

$$x_k = \begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \hat{\delta}_{trans}\cos(\theta + \hat{\delta}_{rot1}) \\ \hat{\delta}_{trans}\sin(\theta + \hat{\delta}_{rot1}) \\ \theta + \hat{\delta}_{rot1}\hat{\delta}_{rot2} \end{pmatrix} \tag{3.21}$$

To równanie jest używane do predykcji stanu robota w kroku 1 algorytmu filtra cząsteczowego.

Model pomiaru

Zadaniem modelu pomiaru w lokalizacji Monte Carlo jest estymowanie rozkładu gęstości prawdopodobieństwa położenia robota na podstawie mapy otoczenia i pomiarów. Zakładamy, że w otoczeniu robota znajdują się znaczniki (ang. *landmarks*) które można jednoznacznie rozróżnić, zaś ich położenie na mapie jest dokładnie znane. Znacznik taki jest opisywany przez

trójkę (x, y, b), gdzie x, y to położenie znacznika w środowisku pracy robota, zaś b jest unikalną wartością dyskretną, identyfikującą znacznik. Mapa środowiska robota jest wtedy wektorem [**?**]:

$$M = (m_1, ..., m_n) = \left(\begin{pmatrix} x_1 \\ y_1 \\ b_1 \end{pmatrix}, \cdots, \begin{pmatrix} x_n \\ y_n \\ b_n \end{pmatrix} \right) \tag{3.22}$$

Obserwowany znacznik jest nazywany cechą (ang. *feature*) i jest reprezentowany przez parę (r, b), gdzie r jest zmierzoną odległością od znacznika, zaś b jego identyfikatorem. W danej chwili czasu k pomiar opisywany przez wektor obserwowanych cech [**?**]:

$$f(z) = (f_1, ..., f_n) = \left(\begin{pmatrix} r_1 \\ b_1 \end{pmatrix}, \begin{pmatrix} r_2 \\ b_2 \end{pmatrix}, \cdots \right) \tag{3.23}$$

Każda z cech odnosi się do konkretnego znacznika. Wprowadźmy funkcję dopasowującą, która na podstawie cechy i mapy przypisuje cechę do znacznika: $c(f_i, M)$. W rozpatrywanym przypadku, funkcja dopasowująca odnajduje znacznik na podstawie jego identyfikatora b. Następnie wyznaczmy odległość (euklidesową) pomiędzy hipoteczną pozycją robota $\hat{p}$ a zaobserwowanym znacznikiem m:

$$\hat{r}_i = \sqrt{(m_x - \hat{p}_x)^2 + (m_y - \hat{p}_y)^2} \tag{3.24}$$

oraz różnicę między zmierzoną odległością od znacznika, a odległością wynikłą z hipotetycznej pozycji $\hat{p}$ (równanie 3.24):

$$\Delta r_i = r_i - \hat{r}_i \tag{3.25}$$

Do zastosowania w filtrze cząsteczowym potrzebny jest rozkład prawdopodobieństwa wystąpienia wektora pomiarów $f(z)$ dla danej hipotetycznej pozycji $\hat{p}$ i mapy M. Wynika on ze złożenia rozkładów prawdopodobieństwa zaobserwowania poszczególnych cech. Ponieważ zaobserwowanie poszczególnych cech to zdarzenia niezależne, można napisać:

$$p(f(z)|\hat{p}, M) = \prod_i p(f_i|\hat{p}, M) \tag{3.26}$$

Rozkład prawdopodobieństwa danej cechy jest złożeniem prawdopodobieństw zaobserwowania znacznika w danej odległości i o danym identyfikatorze. Po raz kolejny słuszne jest założenie niezależności tych obserwacji. Ponadto, można założyć że system pomiarowy wyznacza identyfikator znacznika w sposób pewny: $p(b|\hat{p}, M) = 1$. To prowadzi do ostatecznego równania na wiarygodność pomiaru:

$$p(f_i|\hat{p}, M) = p(r_i|\hat{p}, M) \tag{3.27}$$

Wiarygodność pomiaru odległości można zamodelować np. rozkładem normalnym o odchyleniu standardowym σ :

$$p(r_i|\hat{p}, M) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-\Delta r_i}{2\sigma^2}\right) \tag{3.28}$$

Tak wyznaczoną wiarygodność można wykorzystać w kroku 2 filtra cząsteczkowego do uaktualnienia wag cząsteczek zgodnie z wynikiem pomiaru.

Rozdział 4

Projekt systemu lokalizacji robota

W niniejszym rozdziale przedstawiono projekt systemu lokalizacji robota, opartego na znacznikach radiowych. Jego zadaniem jest akwizycja i filtrowanie danych ze znaczników radiowych, konwersja wartości siły sygnału RSSI na odległość, oraz rozwiązywanie zadania lokalizacji na podstawie zebranych danych.

System akwizycji danych został zaprojektowany w taki sposób, aby możliwe było eksperymentalne porównanie różnych metod lokalizacji.

4.1 Platforma sprzętowa

4.1.1 Znacznik radiowy

Protokół

Znacznik radiowy do wykorzystania w systemie lokalizacji robota winien spełniać szereg wymagań:

- *Mały rozmiar* - urządzenie powinno być zwarte i małych rozmiarów, aby możliwe było łatwe umieszczenie go w miejscu pracy robota
- *Zasilanie bateryjne* - aby zapewnić swobodę rozmieszczenia, urządzenie powinno być zasilane. Ponadto, aby zapewnić bezobsługowość systemu, czas pracy na baterii powinien wynosić co najmniej 1 rok
- *Niski koszt jednostkowy* - system lokalizacji wymaga co najmniej 3 znaczników aby był użyteczny, stąd istotny jest koszt jednostkowy znacznika

Do zbioru technologii, jakie pozwalają wypełnić powyższe wymagania, należą m. in. Wi-Fi (por. rozdział 2.3), ZigBee, Bluetooth wraz z niskoenergetyczną odmianą Bluetooth Low Energy

(por. rozdział 2.4). Wszystkie wymienione technologie korzystają z nielicencjonowanego pasma ISM. Technologia ZigBee jest jednak mało popularna, a co za tym idzie - droższa. Technologia Wi-Fi zapewnia duży zasięg, jednakże za cenę wysokiego zużycia energii.

Protokół Bluetooth w wersji powyżej 4.0, znany pod nazwą Bluetooth Low Energy lub Bluetooth Smart, zapewnia zasięg rzędu 20 metrów przy zachowaniu bardzo niskiego zużycia energii. Istotną zaletą jest to, że układy radiowe BLE są obecne w większości współczesnych smartfonów i laptopów, zaś koszt jednostkowy podstawowego urządzenia BLE wynosi ok 30 zł. Ponadto, istnieją już opracowania dotyczące lokalizacji za pomocą znaczników wykorzystujących Wi-Fi [**?**] oraz klasyczny Bluetooth [**?**]. Wybranie BLE jako protokołu do implementacji pozwala na eksperymentalne sprawdzenie przydatności tego protokołu do lokalizacji, przy jednoczesnym wypełnieniu postawionych wymagań.

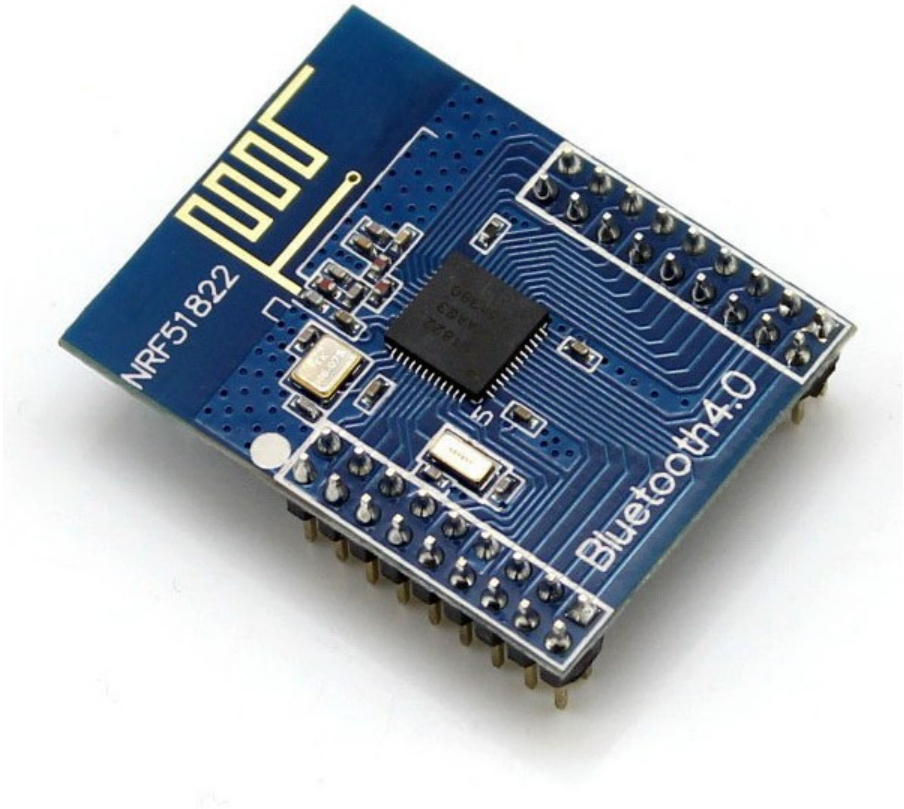
Moduł BLE

Do implementacji znacznika wybrano mikrokontroler Nordic Semiconductor nRF51. Jest on oparty o 32-bitowy rdzeń ARM Cortex M0+. W ramach pojedynczego układu scalonego, oprócz procesora, zintegrowano radio 2,4 GHz oraz szereg peryferiów, takich jak liczniki, zegary, interfejsy komunikacyjne czy sprzętowy układ szyfrujący wg algorytmu AES128. Procesor jest taktowany z częstotliwością 16 MHz, ponadto dysponuje oscylatorem o częstotliwości 32 kHz do precyzyjnego mierzenia czasu. Wykorzystany w projekcie wariant mikrokontrolera posiada 32 kB pamięci RAM oraz 256 kB pamięci flash, wykorzystywanej do przetrzymywania programu oraz do dyspozycji aplikacji.

Podstawową zaletą mikrokontrolera nRF51 jest bardzo niski pobór prądu. W trakcie pracy radia pobiera on ok. 8 mA, natomiast w stanie uśpienia zużycie prądu spada do 2,6 μA. Ponieważ czas pracy radia jest o rząd wielkości krótszy od czasu, który procesor spędza w uśpieniu, średnie zużycie prądu wynosi ok 5μA i waha się w zależności od skonfigurowanego interwału rozgłaszania [**?**].

Jako że niniejsza praca skupia się na zaprojektowaniu oprogramowania i przeprowadzeniu eksperymentów, a nie budowaniu platformy sprzętowej, do projektu wykorzystano gotowy moduł z procesorem oferowany przez sklep Botland (rys. 4.1). Posiada on, oprócz procesora nRF51822, wszystkie elementy do rozwoju aplikacji Bluetooth, tj. 2 oscylatory kwarcowe, antenę, układ BALUN (BALancer-UNbalancer, konwerter sygnału dla anteny) oraz wyprowadzenie wszystkich nóżek procesora na złącza typu goldpin. Moduł posiada interfejs SWO do programowania i debugowania. Po dołączeniu zasilania (np. koszyczka na baterię CR2032) moduł działa jako samodzielny znacznik (beacon) BLE. Koszt takiego modułu wynosi ok. 30 zł.

Do programowania modułu wykorzystano programator J-Link znajdujący się na płytce roz-



Rysunek 4.1: Moduł BLE z procesorem nRF51

wojowej nRF52 Developement Kit.

Oprogramowanie znacznika radiowego

W implementacji oprogramowania znacznika zrealizowano następujące funkcjonalności:

- praca w trybie rozgłoszeniowym z interwałem 100 ms,
- zawartość pakietu rozgłoszeniowego pozwalająca odróżnić beacon systemu od ewentualnych innych urządzeń BLE w okolicy,
- zawartość pakietu rozgłoszeniowego pozwalająca rozróżnić poszczególne beacony wchodzące w skład systemu,
- możliwość konfiguracji zawartości pakietu rozgłoszeniowego za pomocą aplikacji mobilnej.

Należy zaznaczyć, że zasadnicza funkcjonalność znacznika tj. przekazywanie siły sygnału RSSI jest dostępna dla każdej konfiguracji rozgłaszania, ponieważ jest elementem wbudowanym w stos Bluetooth.

Tabela 4.1 opisuje strukturę pakietu rozgłoszeniowego znacznika. Specyfikacja protokołu BLE przewiduje, że w ramce rozgłoszeniowej znajdą się pola reprezentujące poszczególne informacje [**?**]. Pole składa się z 2 + N bajtów: pierwszy bajt zawiera długość pola, drugi określa typ informacji, pozostałe zawierają daną informację. Łączny rozmiar wszystkich pól nie może przekroczyć 31 bajtów. Należy także nadmienić, że do długości pola wlicza się bajt typu. Specyfikacja przewiduje typy takie jak nazwa, klasa urządzenia, lista dostępnych serwistów itp.

W ramce znacznika wykorzystano następujące typy (por. tab. 4.1):

Flagi - pole to jest obowiązkowe w każdej ramce. Zawiera jednobajtowe pole bitowe z flagami określającymi m.in. widoczność urządzenia.

Nazwa - pole to zawiera nazwę urządzenia, wyświetlaną przez urządzenia nasłuchujące rozgłaszania (np. smartfon). Nazwa urządzenia to „locationTAG"

Zawartość producenta - pole to jest przeznaczone do dowolnego wykorzystania przez producenta urządzenia. Zawarto w nim główną informacyjną treść ramki.

Pole "Zawartość producenta" zawiera następujące informacje (por. tab. 4.1):

Kod producenta - obowiązkowy kod producenta. Dla producentów niezarejestrowanych w Bluetooth SIG jego wartość wynosi FFFFh

Identyfikator grupy - 4-bajtowa liczba całkowita (unsigned int32), służąca do odróżniania grup beaconów (np. zebranych w jednym pomieszczeniu)

Identyfikator znacznika - 6-bajtowy identyfikator indywidualny dla znacznika. Jest on równy adresowi fizycznemu MAC znacznika.

4.1.2 Robot

W projekcie wykorzystano robota Pioneer 3-AT, będącego na wyposażeniu Laboratorium Robotyki Mobilnej Instututu Automatyki i Robotyki (rys 4.2). Jest to czterokołowy robot o napędzie różnicowym, zaprojektowany jako platforma edukacyjna z szerokimi możliwościami dalszej rozbudowy. Robot w laboratorium został wyposażony w komputer jednopłytkowy Nvidia Jetson TK1, charakteryzujący się dużą mocą obliczeniową i wsparciem przetwarzania równoległego dzięki technologii CUDA. Komputer działa pod kontrolą systemu operacyjnego Ubuntu Linux i wykorzystuje platformę ROS.

Komputer jest połączony z niskopoziomowymi sterownikami sensorów oraz aktuatorów robota za pomocą interfejsu szeregowego RS-232. Ponadto robot posiada kartę Intel Wireless-AC

Tabela 4.1: Struktura pakietu rozgłoszeniowego

Ramka rozgłoszeniowa [31B]											
Flagi [3B]			Nazwa [13B]			Zawartość [15B]					
Długość [1B]	*Typ* [1B]	Flagi [1B]	*Długość* [1B]	*Typ* [1B]	Nazwa [11B]	*Długość* [1B]	*Typ* [1B]	Kod producenta [2B]	Identyfikator grupy [4B]	Identyfikator znacznika [6B]	Wyrównanie [1B]

7260, zapewniającą łączność Wi-Fi oraz Bluetooth, w tym także Bluetooth Low Energy. Układ sensoryczny robota obejmuje enkodery kół, sonar, kamerę USB oraz skaner laserowy Hokuyo URG-04LX-UG01. Do teleoperacyjnego sterowania robotem wykorzystywany jest kontroler z konsoli do gier Xbox [**?**].

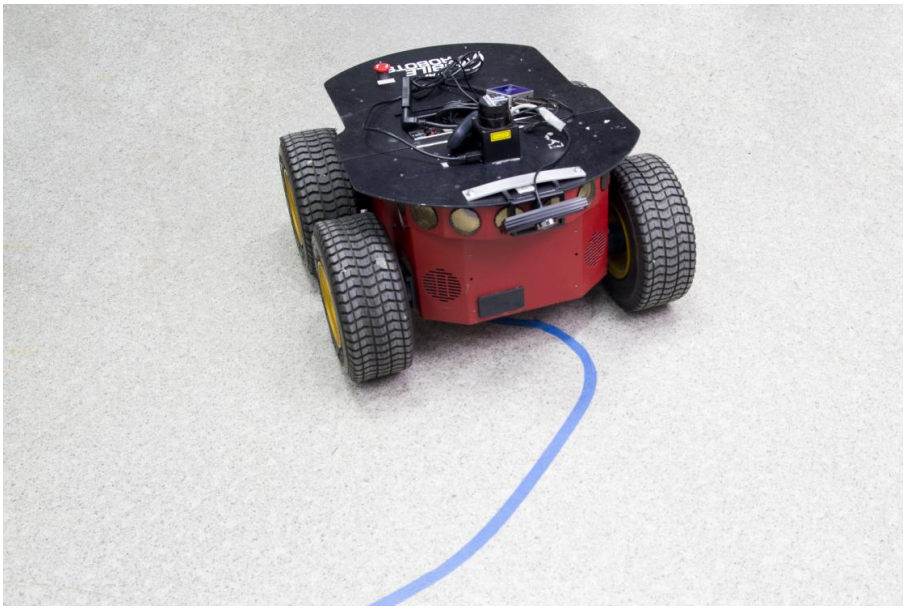
4.2 Oprogramowanie

Ponieważ robot Pioneer, jak wspomniano powyżej, działa pod kontrolą systemu ROS, zdecydowano zaimplementować oprogramowanie nasłuchujące jako aplikacje ROS.

ROS umożliwia pisanie aplikacji w trzech językach programowania [**?**] [**?**]:

- C++
- Python
- JavaScript (Node.js)

Język C++ pozwala na napisanie programu bardzo wydajnego obliczeniowo, ponieważ jest to język kompilowany. Jednakże przygotowanie programu i jego zmiany wymagają relatywnie dużego nakładu pracy. Język Python, jako język interpretowany, znacznie gorzej radzi sobie w zadaniach obliczeniowych, jednak programy są łatwiejsze do napisania i utrzymania ze względu na prostą składnię, wysokopoziomowość i bogaty zasób bibliotek. Język JavaScript podobnie jak Python, ma przeciętną wydajność obliczeniową. Jednak biblioteki ROS w języku JavaScript zostały zaimplementowane stosunkowo niedawno i można się spodziewać, że zbiór bibliotek jest niekompletny lub niestabilny.



Rysunek 4.2: Robot Pioneer 3-AT [**?**]

Ostatecznie, do implementacji aplikacji nasłuchującej wybrano język Python. Wydajność obliczeniowa nie jest problemem, ponieważ aplikacja nie wykonuje żadnych złożonych obliczeń na dużych zbiorach danych. Natomiast elastyczność języka Python pozwoliła na zaprojektowanie aplikacji w taki sposób, aby możliwe było łatwe jej modyfikowanie i rozszerzanie, przy relatywnie niewielkim nakładzie pracy.

4.2.1 Gromadzenie danych ze znaczników

Gromadzenie danych ze znaczników radiowych jest zadaniem aplikacji nasłuchującej. Nasłuch sensu stricto jest realizowane przez sterowniki stosu Bluetooth systemu operacyjnego. Jednakże zastosowanie danych ze znaczników do lokalizacji wymaga postawienia pewnych dodatkowych założeń:

- *Nasłuch ciągły* - większość interfejsów do sterowników Bluetooth udostępnia możliwość skanowania w poszukiwaniu rozgłaszających beaconów przez pewien określony czas, po czym skanowanie jest przerywane. Tymczasem system lokalizacji wymaga, aby skanowanie przeprowadzać w sposób ciągły. Optymalnie, powinno być możliwe przechwycenie każdego pakietu rozgłoszeniowego.

- *Filtrowanie urządzeń* - w środowisku pracy systemu potencjalnie mogą się znaleźć inne urządzenia BLE. Aplikacja nasłuchująca powinna je odfiltrować.

- *Interfejsy do dalszego przetwarzania* - zebrane w postaci pakietów rozgłoszeniowych dane

powinny być udostępnione do dalszego przetwarzania poprzez odpowiedni interfejs.

Integracja z API Bluetooth

Ponieważ oprogramowanie ROS działa w systemie operacyjnym Linux, konieczne jest zintegrowanie aplikacji ROS ze sterownikiem Bluetooth. W większości dystrybucji systemu Linux, Bluetooth działa pod kontrolą stosu BlueZ, będącego oficjalną implementacją stosu Bluetooth dla systemów Linux.

Do integracji aplikacji w języku Python z API Bluetooth konieczne jest zastosowanie dodatkowej biblioteki. Ponieważ implementacja takiej biblioteki jest zadaniem złożonym i wykraczającym poza zakres niniejszej pracy, wykorzystano bibliotekę bluepy autorstwa Iana Harvey'a. Biblioteka ta stanowi interfejs API BlueZ w języku C do języka Python [**?**].

Architektura aplikacji nasłuchującej

Tor przetwarzania aplikacji rozłożony jest na dwa wątki wymieniające dane za pomocą chronionej struktury danych.

API biblioteki BlueZ jest zaprojektowane jako synchroniczne. Oznacza to, że funkcja odpowiedzialna za wykonanie skanowania BLE zwraca dopiero po zakończeniu skanowania. Dlatego, aby zapewnić responsywność programu, skanowanie jest uruchamiane w oddzielnym wątku. Funkcja jest wywoływana w pętli, wykonując skanowanie o zadanej długości. Pętla może być przerwana przez odpowiednią funkcję i wtedy wątek kończy pracę. Każde znalezione podczas skanowania urządzenie jest dodawane do kontenera. Kontener jest oparty na mapie (w języku Python nazywanej słownikiem, ang. *dictionary*), w której kluczem jest adres fizyczny MAC beacona, zaś wartością - obiekt przechowujący informacje odebrane z rozgłaszania. Kontener został wzbogacony o dwa dodatkowe mechanizmy:

- *Usuwanie starych wpisów* - wpisy, reprezentujące pakiety rozgłoszeniowe, starsze niż zadana wartość mogą zostać usunięte z kontenera.
- *Ochrona dostępu* - dla zapewnienia bezpiecznej synchronizacji między wątkami, każdy dostęp do danych w kontenerze (odczyt, pobranie całej zawartości, dodanie nowego wpisu) jest chroniony za pomocą obiektu *Lock*, zapewniającego atomowość dostępu.

W wątku głównym programu prowadzone jest publikowanie wiadomości zawierających dane rozgłoszeniowe. W tym celu aplikacja wykorzystuje obiekt opakowujący klasę *Publisher* biblioteki ROS [**?**]. Jego rolą jest odfiltrowanie urządzeń nienależących do systemu oraz posiadających nieprawidłowy identyfikator grupy (por. 4.1.1).

Ustalono, że wszystkie aplikacje systemu będą działać w przestrzeni nazw składającej się z nazwy robota oraz członu *beacon_localization*. Aplikacja nasłuchująca działa pod nazwą *beacon_listener*. Znalezione urządzenia są publikowane do tematu o nazwie *location_tag*.

4.2.2 Filtracja i konwersja danych RSSI na odległość

Jak wspomniano w rozdziale 2, wartość siły sygnału RSSI podlega wahaniom nawet w wypadku, gdy nadajnik i odbiornik są nieruchome względem siebie. Ponadto, wahania wartości są dość duże, rzędu 5 dBm. Takie wahanie przekładałoby się bezpośrednio na wahanie przeliczonej odległości i znacznie osłabiało jakość lokalizacji. Dlatego konieczne jest wprowadzenie filtracji wartości siły sygnału, takiej, aby wahania stanu stacjonarnego zostały wygładzone. Jednocześnie bezwładność wprowadzana przez filtr nie może być zbyt duża, ponieważ odbiłoby się to negatywnie na lokalizacji podczas ruchu robota.

Filtracja i konwersja na odległość jest realizowana przez oddzielną aplikację ROS, noszącą nazwę *rssi2distance*. Aplikacja ta subskrybuje temat *location_tag*, zaś przefiltrowane i skonwertowane dane publikuje w tematach *distances/*<nazwa filtra> . Ten sam wynik działania aplikacji jest dostępny również za pomocą usługi ROS działającej pod tą samą nazwą. Dane wynikowe są publikowane w postaci tablicy, zawierającej wszystkie znaczniki radiowe będące aktualnie w zasięgu, łącznie z przefiltrowaną wartością RSSI, przeliczoną odległością oraz pozycją znacznika w układzie współrzędnych mapy pomieszczenia.

Filtracja

Zaimplementowano następujące filtry (kursywą podano nazwy filtrów używane do nazywania wynikowych tematów i usług):

1. *recent* - brak filtra. Filtr zwraca najnowszą znaną wartość RSSI. Jest to najprostsza strategia, używania głównie do testów podczas rozwoju oprogramowania.

2. *moving_average* - średnia krocząca. Filtr zwraca średnią kroczącą prostą z 5 ostatnich pomiarów:
$$\bar{P}_k = \frac{P_k + P_{k-1} + \cdots + P_{k-4}}{5} \tag{4.1}$$

3. *probabilistic* - prosty filtr probabilistyczny. Opiera się on na założeniu, że zmiana odległości od znacznika ze stałą prędkością będzie skutkować stałą zmianą wartości RSSI. Działanie filtra można podzielić na fazy estymacji i predykcji [**?**]. Równania estymacji:
$$\hat{P}_{estk} = \hat{P}_{predk} + a(P_k - \hat{P}_{predk}) \tag{4.2}$$

$$\hat{V}_{estk} = \hat{V}_{predk} + \frac{b}{T_s}(P_k - \hat{P}_{predk}) \tag{4.3}$$

Równania predykcji:

$$\hat{P}_{pred\{k+1\}} = \hat{P}_{estk} + \hat{V}_{estk}T_s \tag{4.4}$$

$$\hat{V}_{pred\{k+1\}} = \hat{V}_{estk} \tag{4.5}$$

W powyższych równaniach $\hat{P}_{estk}$ jest wynikiem działania filtra - wygładzoną wartością RSSI, V_{estk} i V_{predk} to wartość estymowanego tempa zmiany RSSI, a i B są stałymi strojącymi filtr, zaś T_s jest okresem próbkowania RSSI.

Konwersja RSSI na odległość

Model propagacji, służący do przeliczania RSSI na odległość, również został zaimplementowany na podstawie wzorca Strategii. Model realizuje równanie 2.12.

Mapa znaczników

Mapa znaczników jest publikowana przez dodatkową aplikację o nazwie *beacon_tf_publisher*. Pobiera ona konfigurację mapy w postaci koordynatów i odpowiadających im identyfikatorów znacznika z pliku YAML, następnie publikuje w postaci transformacji z układu współrzędnych mapy do układów poszczególnych znaczników, wykorzystując bibliotekę *tf*.

4.2.3 Trilateracja

Dysponując oprogramowaniem do zbierania danych ze znaczników i konwersji siły sygnału RSSI na odległość, przystąpiono do projektowania właściwych aplikacji lokalizujących. Pierwsza z nich wykorzystuje technikę trilateracji. Strukturę aplikacji rozwiązującej zadanie trilateracji oparto na pętli wykonującej się z określonym interwałem. W każdej iteracji wykonywane jest zapytanie do usługi ROS udostępnianej przez aplikację *rssi2distance*. W odpowiedzi otrzymywana jest lista znaczników znajdujących się w zasięgu, wraz z przefiltrowaną wartością RSSI, przeliczoną wartością odległości i pozycją znacznika w układzie mapy. Możliwy jest wybór metody filtracji za pomocą odpowiedniego parametru na serwerze parametrów ROS.

Jeśli w danej iteracji w zasięgu znajdowało się co najmniej trzy znaczniki, wykonywane jest rozwiązanie zadania trilateracji. Dostępne są dwie, bardzo zbliżone metody trilateracji:

1. *metoda Newtona* - opisana w rozdziale 3.2
2. *metoda L-BFGS (Limited-Memory Broyden–Fletcher–Goldfarb–Shanno* - jest to metoda minimalizacji należąca do rodziny metod quasi-Newtonowskich, implementowana przez bilbiotekę *scipy*

Metoda trilateracji również może być wybierana za pomocą serwera parametrów.

Wynik obliczeń jest konwertowany na wiadomość typu *geometry_msgs/Pose*), reprezentującą pozycję i orientację obiektu w przestrzeni trójwymiarowej. Jest to typowa konwencja w systemie ROS, nawet jeśli lokalizacja ogranicza się do dwóch wymiarów. Wynik jest publikowany w temacie *beacon_localization/bl_pose* oraz jako transformacja z układu mapy do układu robota.

4.3 Filtr cząsteczkowy

Ponieważ algorytm trilateracji wykorzystuje tylko dane ze znaczników, jest wrażliwy na zakłócenia wynikłe z ich działania, np. rozrzut wartości RSSI w stanie ustalonym. Dlatego następnym krokiem było zaprojektowanie aplikacji lokalizującej robota w oparciu nie tylko o dane ze znaczników, ale także z sensorów odometrycznych. W celu zintegrowania tych danych wykorzystano filtr cząsteczkowy, co do teorii opisany w rozdziale 3.3. Filtr cząsteczkowy zaimplementowano w oparciu o implementację autorstwa Dakoty Nelsona dostępną na licencji MIT. Źródłowa implementacja dotyczyła filtra cząsteczkowego wykorzystującego skaner laserowy oraz sensor odometryczny. Aplikacja subskrybuje do tematu *distances/*<nazwa filtra>, gdzie publikowane są tablice znajdujących się w zasięgu znaczników, wraz z ich pozycjami na mapie. Aby ograniczyć obciążenie komputera obliczeniami, algorytm filtra jest wykonywany tylko jeśli przemieszczenie kątowe lub liniowe robota wynikłe z pomiaru odometrycznego przekroczyło pewien konfigurowalny próg. Jako metodę resamplingu wykorzystano algorytm SRR (*Sequential Random Resampling*, ang. sekwencyjny losowy resampling) [**?**].

Rozdział 5

Badania eksperymentalne

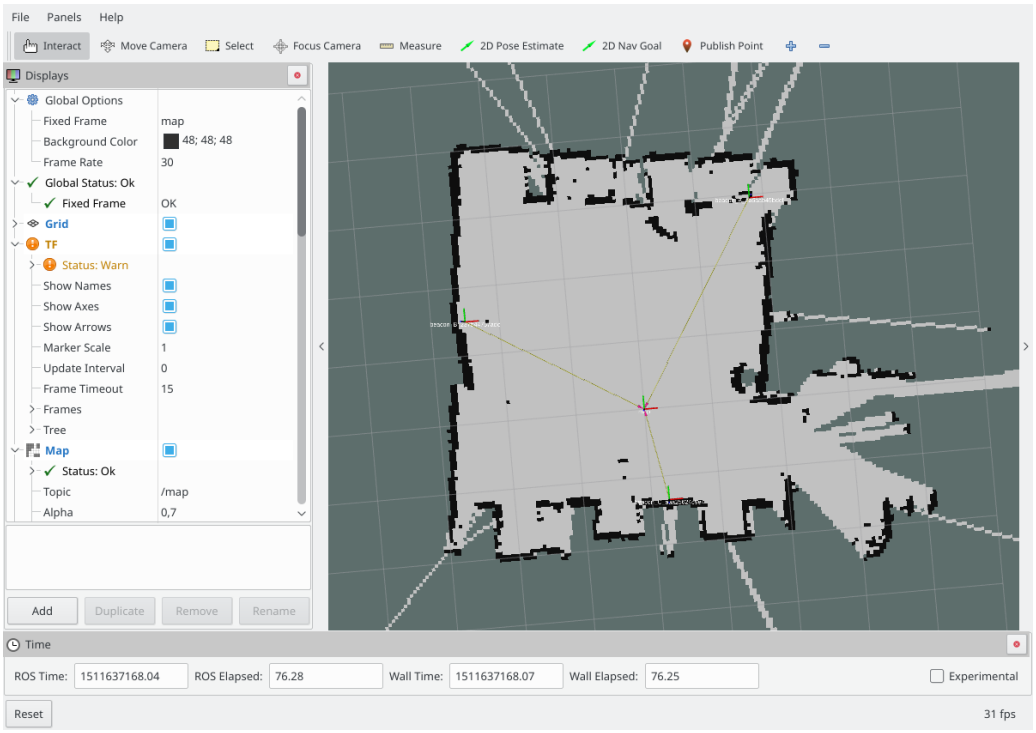
5.1 Wykorzystane narzędzia

Do rozwoju i testowania oprogramowania wykorzystano przede wszystkim narzędzia dostępne w systemie ROS. Przygotowano jednak także kilka dodatkowych narzędzi.

RViz

Program RViz (*ROS Visualization*) jest należącym do systemu ROS narzędziem wizualizacji danych. Posiada szereg integracji, pozwalających m. in. na:

- wyświetlanie mapy pomieszczenia wyznaczonej metodą SLAM lub inną,
- wyświetlanie układów współrzędnych i ich wzajemnych relacji (integracja z biblioteką *tf*),
- subskrypcję i wizualizację niektórych typów wiadomośći dotyczących geometrii robota, m. in. *Pose*, *PoseWithCovariance*,
- wizualizację danych ze skanera laserowego,
- interakcję z robotem np. poprzez zadawanie celu nawigacji.



Rysunek 5.1: Okno programu RViz

Gmapping

Gmapping jest narzędziem należącym do pakietu OpenSLAM, służącym do budowania dwuwymiarowych map pomieszczeń za pomocą SLAM (Simultaneous Localization And Mapping, ang. *Jednoczesna Lokalizacja i Mapowanie*). Program wykorzystuje pomiar ze skanera laserowego oraz odometrię.

Rosbag

Program Rosbag pozwala na zapisywanie do pliku danych publikowanych w wybranych tematach wraz z ich znacznikami czasowymi, celem późniejszego odtworzenia. Narzędzie to pozwoliło na nagranie szeregu testowych przejazdów robota, celem późniejszego zbudowania mapy pomieszczenia i wykonania pomiarów. Zaawansowane funkcjonalności manipulowania zegarem systemu ROS pozwalają na integrowanie danych odtwarzanych za pomocą Rosbag z danymi generowanymi w czasie rzeczywistym.

Dodatkowe skrypty

W ramach testów systemu przygotowano także szereg dodatkowych skryptów w języku Python:

- skrypt do zbierania danych RSSI do pliku tekstowego, celem dalszej obróbki,

- skrypt do dopasowania modelu 2.12 do danych pomiarowych.

5.2 Zmienność wartości siły sygnału RSSI

5.2.1 Metodyka badań

Przedmiotem niniejszego eksperymentu było zbadanie przebiegu wartości RSSI w czasie dla sytuacji, w której nadajnik i odbiornik są nieruchome. Dodatkowo, wyznaczono histogramy wartości RSSI celem aproksymowania dystrybucji prawdopobieństwa wartości RSSI.

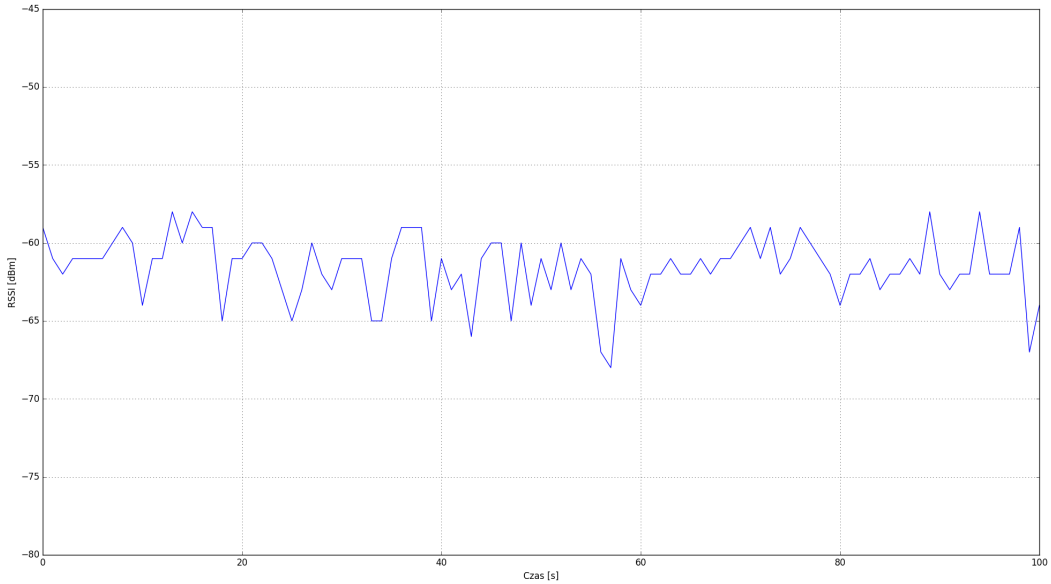
Wykorzystano znacznik opisany w rozdziale 4.1.1 oraz odbiornik Wi-Fi/Bluetooth Intel 7260, zamontowany w laptopie Lenovo Y50 działającego pod kontrolą systemu Linux Mint 18.

Testy przeprowadzono w pomieszczeniu o wymiarach 1.5 x 20 m (korytarz).Pomieszczenie to charakteryzował wysoki poziom zakłóceń radiowych w paśmie ISM (liczne punkty dostępowe Wi-Fi i odbiorniki elektryczne). Znacznik radiowy i odbiornik znajdowały się na wysokości 70 cm nad podłogą. Przeprowadzono pomiar dla odległości 1 m i 5 m. Pomiar polegał na zebraniu 100 wartości siły sygnału RSSI. Tak zebrane dane obrazowano w postaci wykresu czasowego i histogramu.

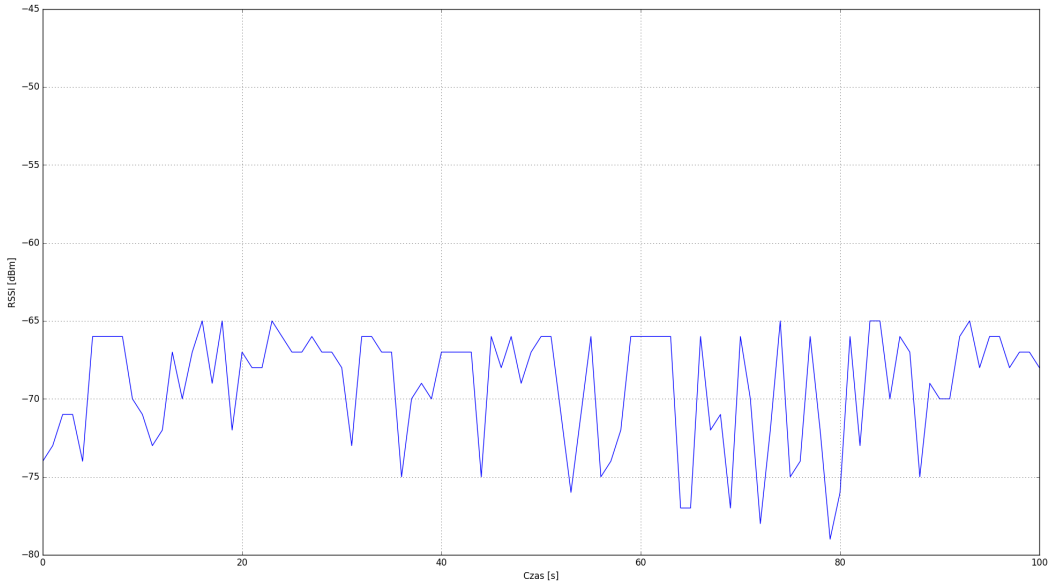
5.2.2 Wyniki

Z wykresów 5.2 i 5.3 widać, że siła sygnału RSSI waha się znacznie nawet dla stanu ustalonego. Amplituda wahań dla odległości 1 m jest średnio rzędu 7 dBm. Dla odległości 5 m jest znacznie większa, rzędu 7 m. Stanowi to poważną przeszkodę dla pomiaru odległości za pomocą RSSI - kumulują się bowiem dwa negatywne czynniki: po pierwsze zależność RSSI od odległości jest w zakresie powyżej 2 m niemal płaska, po drugie - występują duże wahania wartości RSSI, które w tym zakresie będą się przekładać na znaczne wahania odległości.

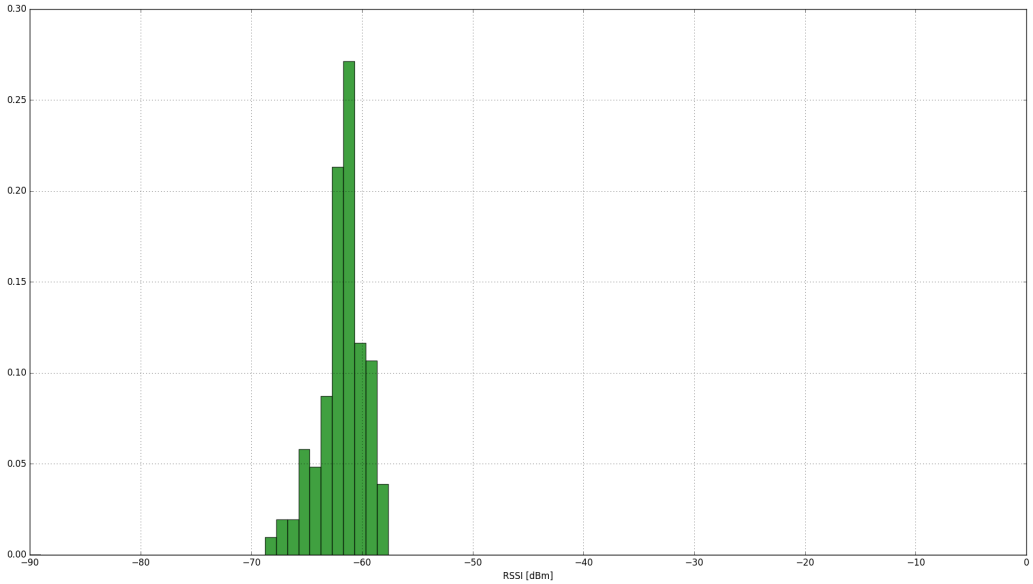
Wykresy 5.4 i 5.5 zawierają histogramy wygenerowane z danych pomiarowych. Dla odległości 1 m rozkład wyników pomiarów jest zbliżony do Gaussowskiego, jednak dla 5 m obserwujemy wyraźną asymetrię w kierunku niższych wartości siły sygnału. Jest to wynik wymienionych w rozdziale 2 zakłóceń, szczególnie wielokrotnych odbić od ścian pomieszczenia. Także szerokość rozkładu jest większa niż w wypadku odległości 1 m.



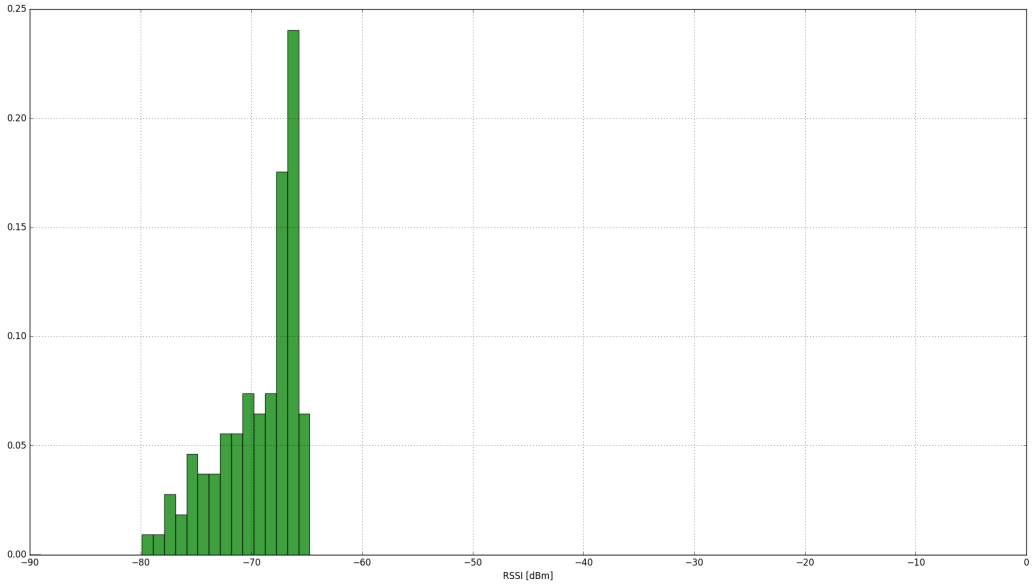
Rysunek 5.2: Przebieg wartości siły sygnału RSSI w czasie dla odległości odbiornika od nadajnika równej 1 m



Rysunek 5.3: Przebieg wartości siły sygnału RSSI w czasie dla odległości odbiornika od nadajnika równej 1 m



Rysunek 5.4: Histogram siły sygnału RSSI w czasie dla odległości odbiornika od nadajnika równej 1 m



Rysunek 5.5: Histogram siły sygnału RSSI w czasie dla odległości odbiornika od nadajnika równej 1 m

5.3 Zależność siły sygnału RSSI od odległości

5.3.1 Metodyka badań

Celem niniejszego eksperymentu jest zweryfikowanie modelu przedstawionego w rozdziale 2. Środowisko testowe było identyczne jak w punkcie 5.2. W każdym punkcie pomiarowym zebrano 30 próbek RSSI, które następnie uśredniono i wyznaczono odchylenie standardowe. Następnie wykonano dopasowanie krzywych logarytmicznych według metody interpolacji oraz dopasowania krzywej (por. rozdział 2.2).

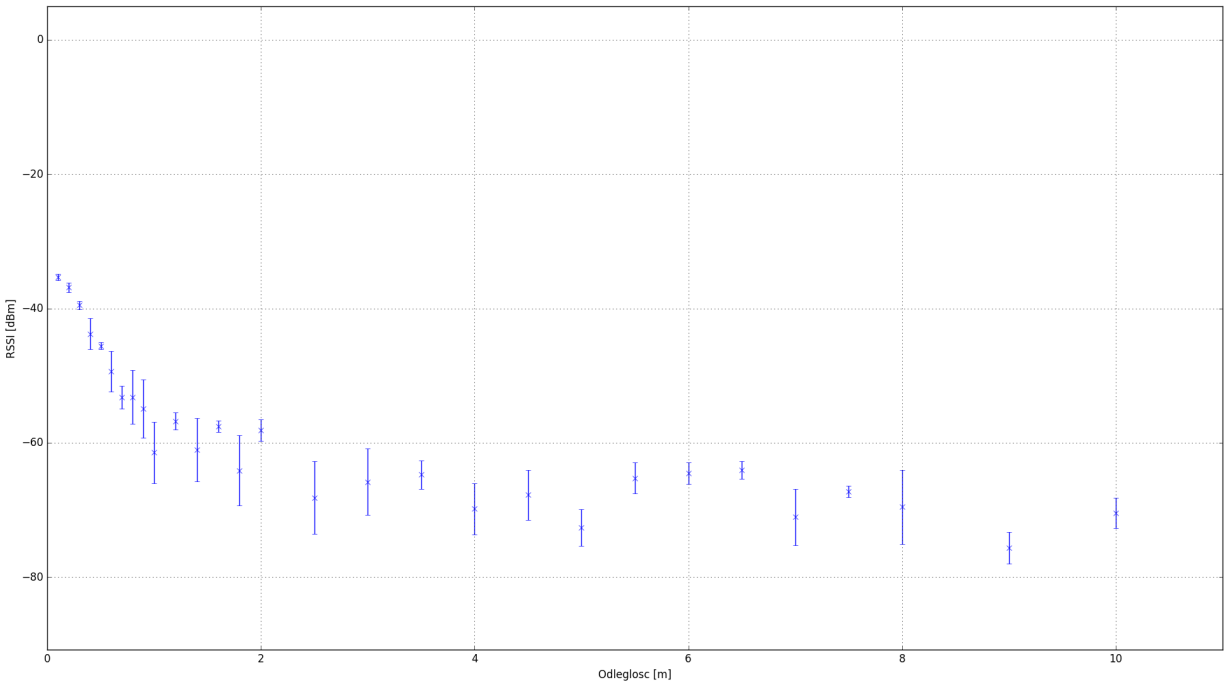
5.3.2 Wyniki

Na wykresie 5.6 widać logarytmiczną zależność pomiędzy odległością, a wartością siły sygnału RSSI, zgodną z oczekiwaniami teoretycznymi opisanymi w 2.2. Jednocześnie wraz ze wzrostem odległości od nadajnika, wzrasta odchylenie standardowe pomiarów, co oznacza że wahania wartości RSSI były większe.

W odległości 3.5m oraz 5 m obserwujemy niewielki wzrost siły sygnału RSSI, co może wskazywać, że w tych miejscach środowiska testowego zachodziły warunki powodujące lokalne wzmocnienie fali. Należy zwrócić uwagę, iż w zakresie odległości powyżej 2 m wartość RSSI pozostaje na stałym poziomie, wahając się w granicach 10 dBm. Z tego powodu jakość pomiaru odległości za pomocą RSSI w tym zakresie będzie bardzo słaba.

Wykres 5.7 ukazuje dopasowane do wyników pomiaru krzywe: metodą interpolacji (2.8) i poprzez dopasowanie krzywej 2.12 za pomocą funkcji *curve_fit* z bilbioteki *scipy* dla języka Python. Należy zwrócić uwage, iż w porównaniu z wykresem 5.6 zamienione zostały osie układu współrzędnych, tj. wykres 5.7 przedstawia problem z perspektywy przeliczania RSSI na odległość.

Funkcja *curve_fit* uwzględnia wszystkie punkty pomiarowe wraz z ich niepewnościami, dlatego zapewnia lepsze dopasowanie od interpolacji, biorącej pod uwagę tylko dwa punkty pomiarowe. Jednakże, aby osiągnąć zadowalające dopasowanie, trzeba albo ręcznie wybrać dwa optymalne punkty do dopasowania interpolacyjnego, albo usunąć punkty zbyt odstające od pozostałych dla metody dopasowania krzywej.



Rysunek 5.6: Wykres wartości siły sygnału RSSI w zależności odległości

5.4 Porównanie metod filtracji siły sygnału RSSI

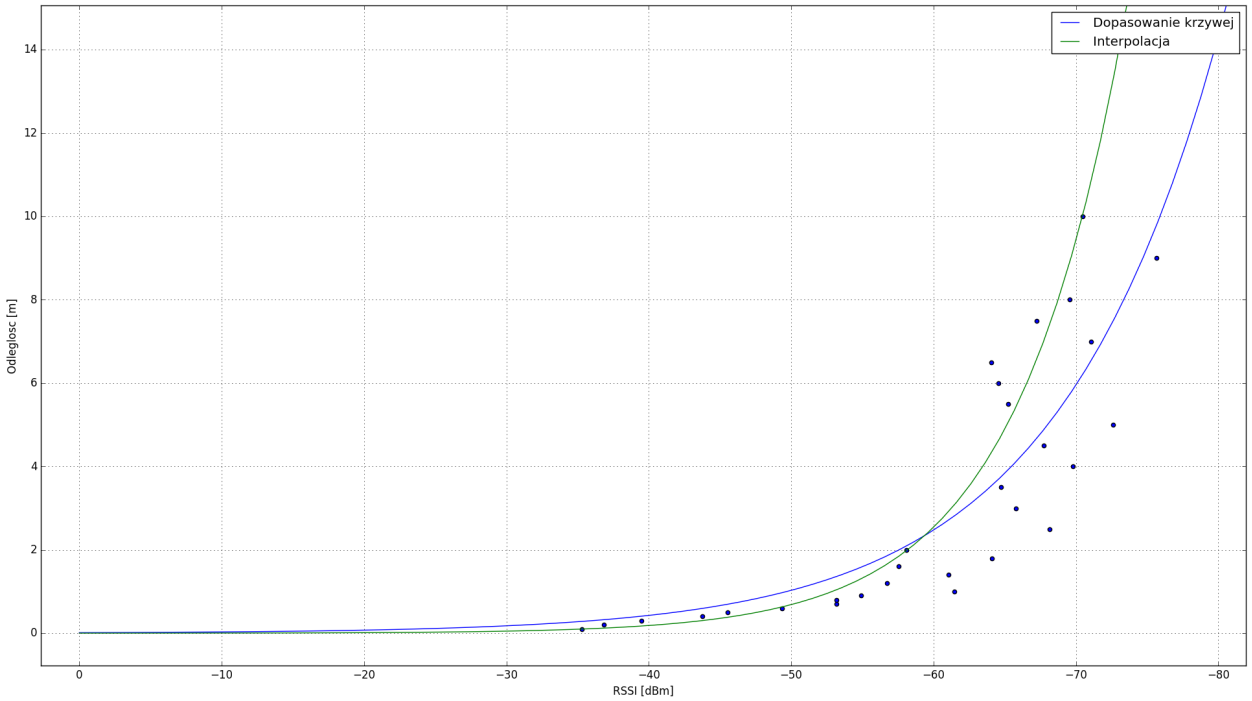
5.4.1 Metodyka badań

Eksperyment miał na celu porównanie skuteczności metod filtracji sygnału RSSI opisanych w 2.2. Pożądanym wynikiem jest dobre wygładzenie stacjonarnych wahań RSSI przy zachowaniu małej bezwładności, tj. szybkiej reakcji na rzeczywistą zmianę siły sygnału RSSI, wynikłą ze zmiany odległości od znacznika.

W ramach eksperymentu posłużono się zbiorem danych nagranym za pomocą programu rosbag. Obejmował on pomiary RSSI zbierane z interwałem 1 sekundy, dla następującej sekwencji: przez 40 sekund znacznik pozostaje w odległości 1 m od odbiornika, następnie zostaje przeniesiony na odległość 2 m, po kolejnych 20 sekundach zostaje przeniesiony z powrotem na odległość 1 m gdzie pozostaje przez 40 sekund.

5.4.2 Wyniki

Filtr średniej ruchomej (por. 4.2.2) wprowadza znaczną bezwładność do układu (por. rys 5.8), przy czym dla większej liczby kroków średniej $N = 10$ bezwładność nie jest znacząco większa.



Rysunek 5.7: Wykres dopasowania krzywych potęgowych do wyników pomiaru

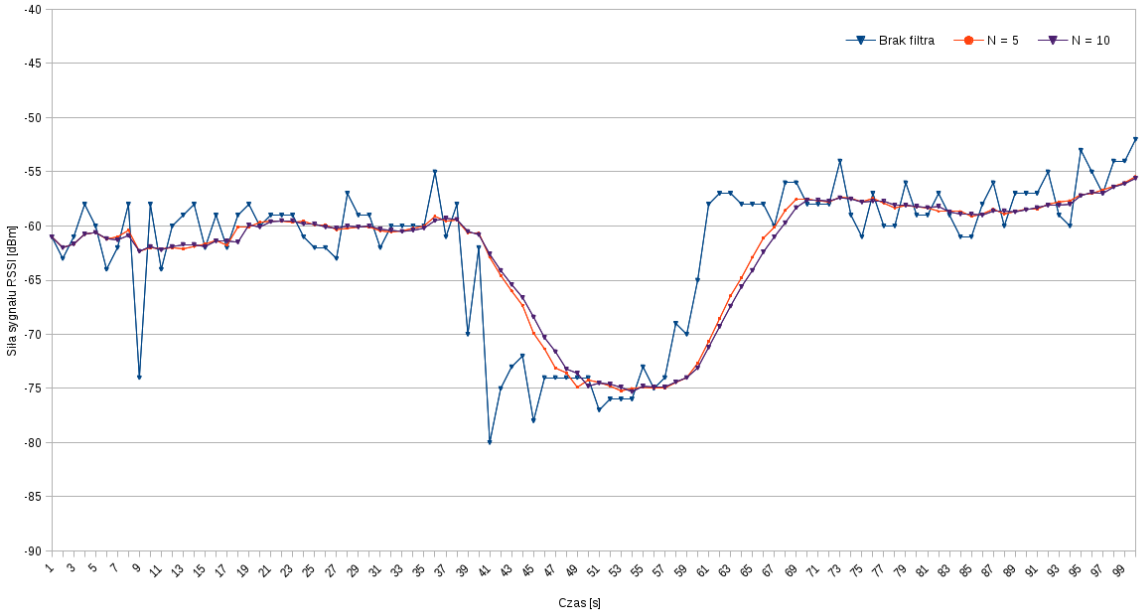
Filtr zapewnia dobre wygładzenie dla nieruchomego odbiornika. W wypadku zmiany odległości, filtr potrzebuje 10 sekund aby dość do stanu ustalonego, co dyskwalifikuje jego zastosowanie w lokalizacji dynamicznej.

Filtr probablistyczny jest strojony za pomocą dwóch parametrów: a - opisujący jak istotna jest wartość poprzedniego pomiaru w estymacji bieżącego, oraz b - opisujący jak istotny jest przyrost wartości RSSI. Innymi słowy, parametr a opisuje wygładzanie filtra, zaś b opisuje czułość na zmiany RSSI. Zbyt niska wartość a powoduje dużą bezwładność filtra, z kolei wartość $a = 1$ powoduje że wartość estymowana jest wartości zmierzonej. Zbyt duża wartość b spowoduje oscylacje, a nawet utratę stabilności filtra.

W eksperymencie porównano dwa zestawy nastaw filtra (rys. 5.9):

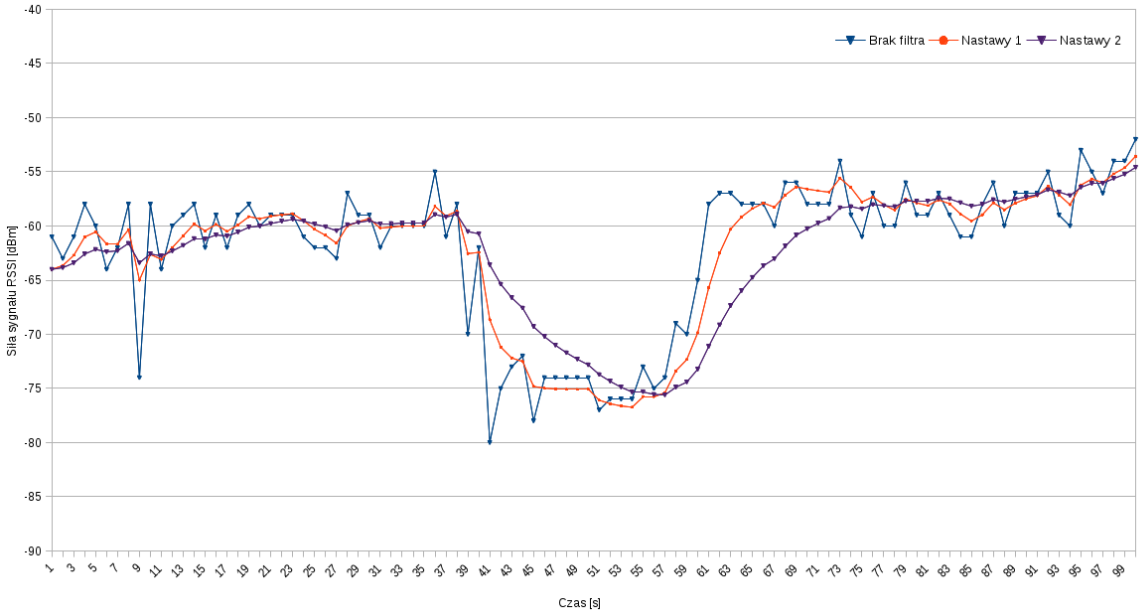
- **Nastawy 1** $a = 0.35$ $b = 0.02$
- **Nastawy 2** $a = 0.15$ $b = 0.005$

Nastawy 2 zapewniają dobre wygładzenie w stanie ustalonym oraz bardzo dużą bezwładność. Zwiększanie parametru b spowoduje jedynie zaistnienie oscylacji, bez poprawy reakcji filtra.

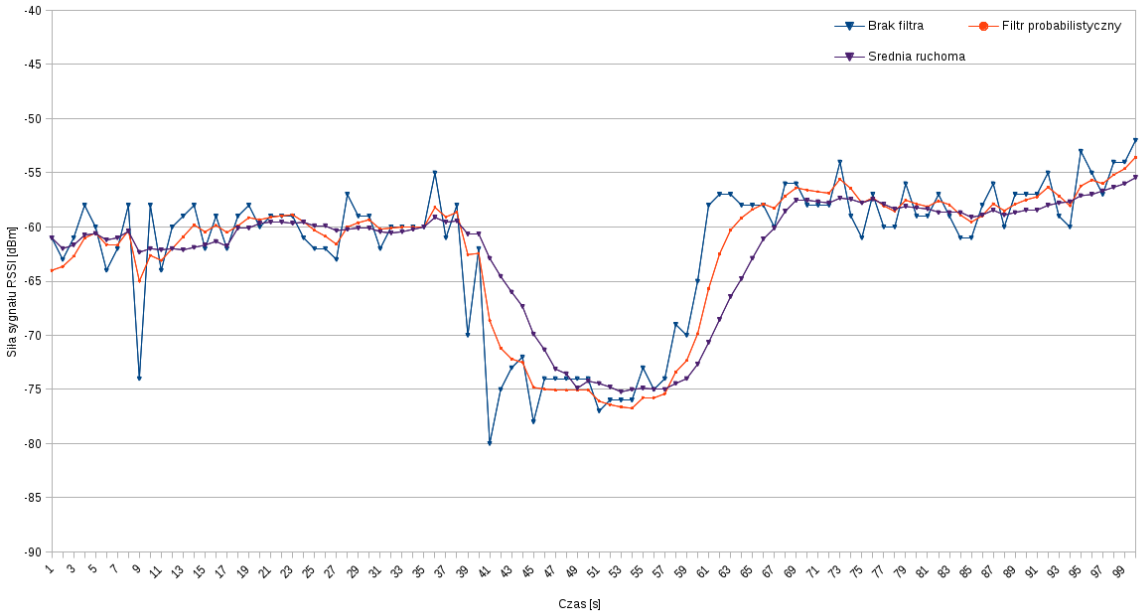


Rysunek 5.8: Filtr średniej ruchomej

Z kolei nastawy 1 stanowią dobry kompromis pomiędzy wygładzeniem a reakcją na zmiany RSSI. Filtr reaguje na zmianę odległości w ciągu około 4 sekund, zaś amplituda wahań wartości w stanie ustalonym jest rzędu 3 dBm. Na wykresie 5.10 przedstawiono przebieg RSSI bez filtracji, z filtracją średnią ruchomą dla $N = 5$ oraz filtracją probabilistyczną wg. nastaw 1. Filtr średniej ruchomej ma znacznie większą bezwładność od filtra probabilistycznego. Dlatego do testowania lokalizacji robota wykorzystano właśnie filtr probabilistyczny, jako lepiej radzący sobie w sytuacjach dynamiczych.



Rysunek 5.9: Filtr probabilistyczny



Rysunek 5.10: Porównanie metod filtracji

5.5 Porównanie metod lokalizacji

W niniejszej sekcji opisano przebieg i wyniki badania zasadniczej części niniejszej pracy - algorytmu lokalizacji. W celu wykonania badań, konieczne było zbudowanie mapy środowiska pracy

robota oraz skalibrowanie znaczników, tj. wyznaczenie parametrów modelu 2.12.

5.5.1 Metodyka badań

Badania metod lokalizacji przeprowadzono na opisanym w rozdziale 4.1 robocie Pioneer 3-AT, w pomieszczeniu Laboratorium Robotyki Mobilnej Wydziału Mechatroniki PW. Do zbudowania mapy pomieszczenia, potrzebnej do pomiarów, wykorzystano program Gmapping. Budowanie mapy polegało na wykonaniu 3 przejazdów dookoła pomieszczenia. W trakcie przejazdów unikano gwałtownych przyspieszeń, hamowań i skrętów robota, w celu ograniczenia uślizgu kół, który wprowadza błąd do mapowania SLAM. Tak wyznaczoną mapę zapisano w pliku graficznym PNG. Jeden piksel obrazka reprezentował 5 cm przestrzeni. Następnie zlokalizowano na mapie umiejscowienie znaczników, korzystając z punktów charakterystycznych.

W celu skalibrowania znaczników zebrano szereg punktów pomiarowych w różnych odległościach od znacznika. Aby wyeliminować wpływ wahań RSSI na wyznaczony model, każdy pomiar składał się z 20 uśrednionionych wartości RSSI. Następnie wyznaczono model w oparciu o interpolację i dopasowanie funkcji wykładniczej.

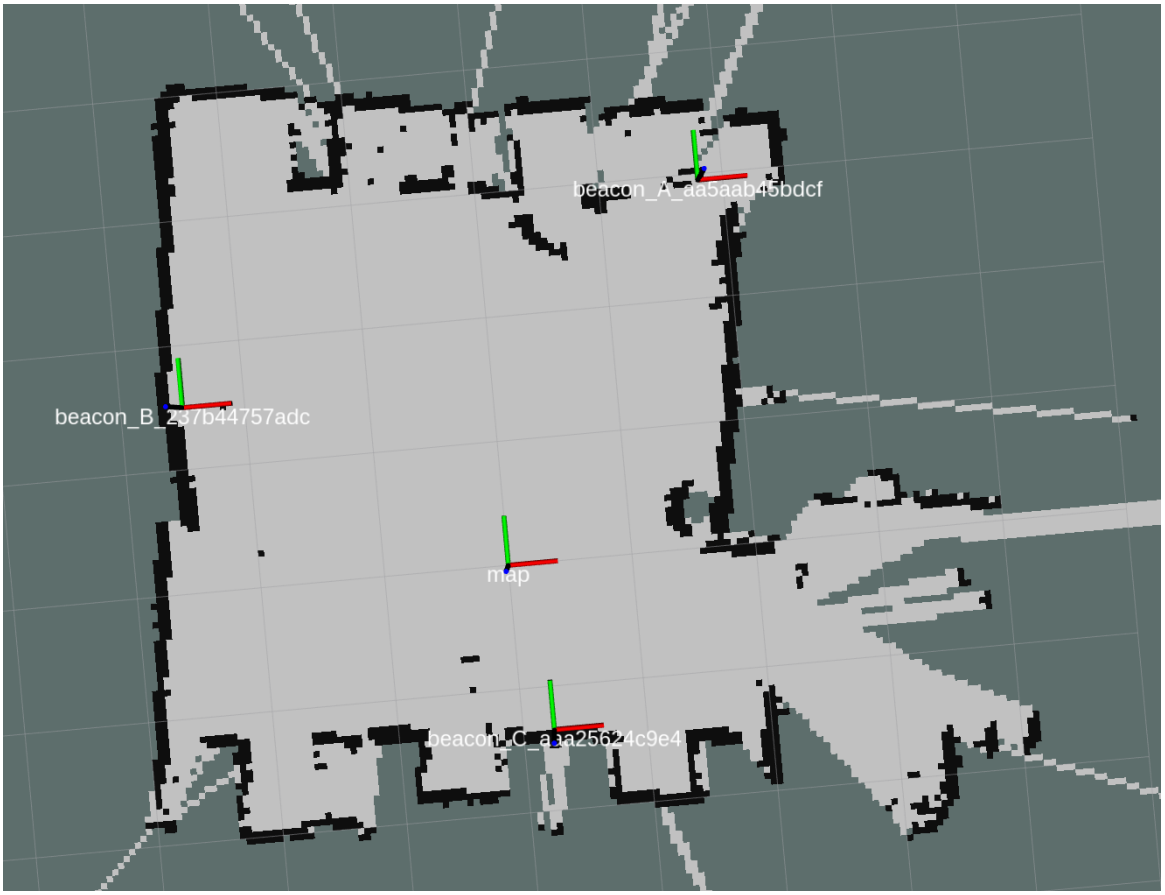
Do badania metod lokalizacji jako odniesienie wykorzystano lokalizację AMCL. Po wykonaniu kilku przejazdów robota po pomieszczeniu oparta o filtr cząsteczkowy lokalizacja AMCL zapewnia dokładność zbliżoną do rozdzielczości mapy, co najmniej o rząd wielkości lepszą niż dokładność lokalizacji w oparciu o znaczniki. Aby zapewnić rzetelne porównanie metod lokalizacji, posłużono się danymi nagranymi programem Rosbag, zatem obydwie metody zostały użyte na identycznych danych testowych.

W obydwu metodach porównano wyniki lokalizacji obliczone z odległości przefiltrowanych przez poszczególne filtry (por. 5.4).

5.5.2 Wyniki

Mapa środowiska

Rysunek 5.11 przedstawia wyznaczoną metodą SLAM mapę laboratorium. Punkty w kolorze szarym oznaczają obszar wolny, zaś w kolorze czarnym - zajęty. Pojedyncze czarne punkty i szare linie stanowią wynik zakłóceń SLAM i nie mają negatywnego wpływu na lokalizację. Obszar w prawym dolnym rogu mapy obrazuje otwarte drzwi do sąsiedniego pomieszczenia, które nie było objęte mapowaniem. Na mapie zobrazowane są pozycje początków układów współrzędnych: mapy oraz 3 znaczników.



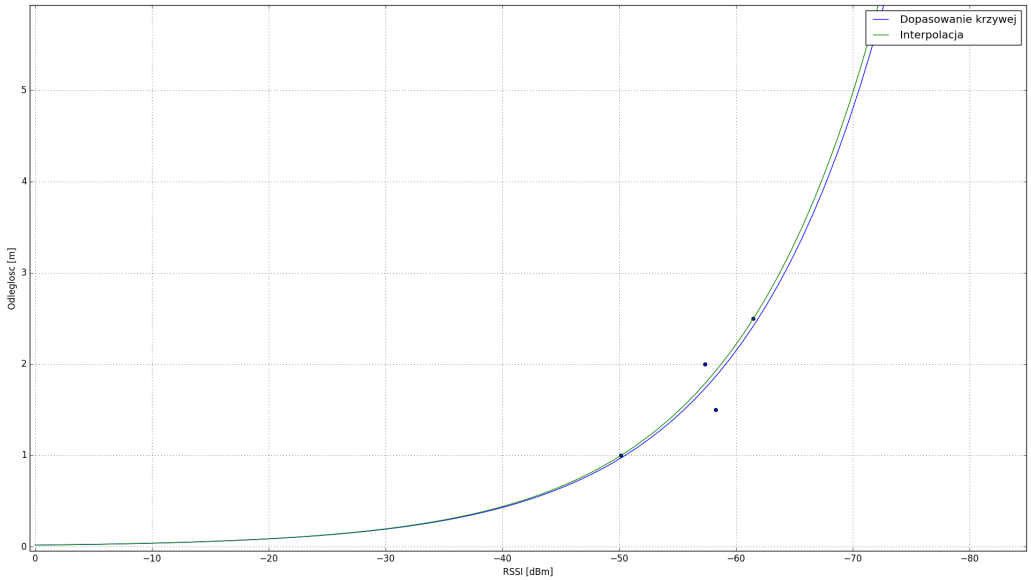
Rysunek 5.11: Mapa środowiska wyznaczona metodą SLAM za pomocą programu GMapping z zaznaczonymi znacznikami radiowymi.

Kalibracja znaczników

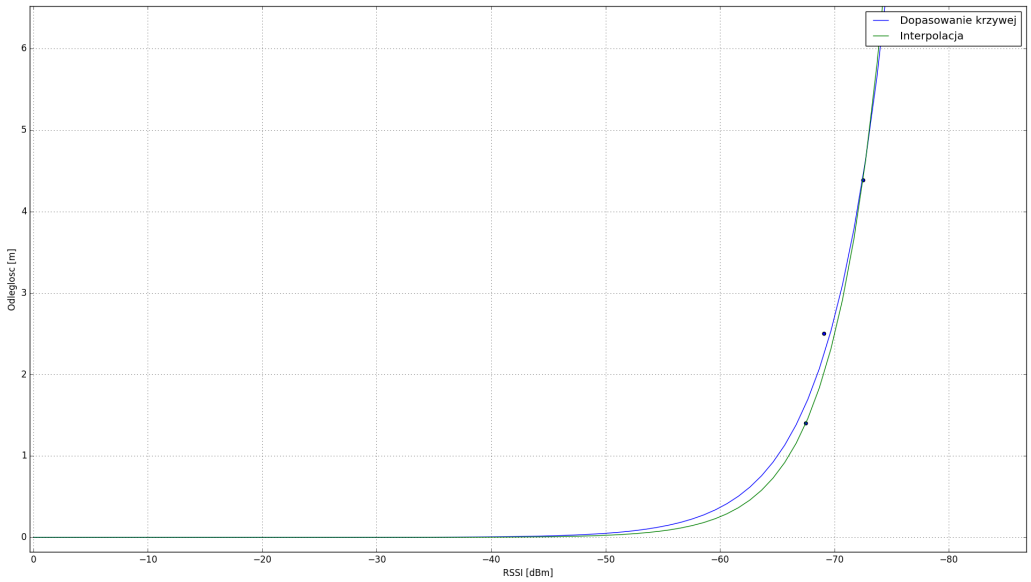
Tabela 5.1 przedstawia wyniki kalibracji znaczników. Z wyników pomiaru usunięto punkty znacząco odbiegające od oczekiwanego przebiegu. Dopasowane krzywe wraz z punktami pomiarowymi wykreślono na rys. 5.12, 5.13 oraz 5.14.

Tabela 5.1: Wyniki kalibracji

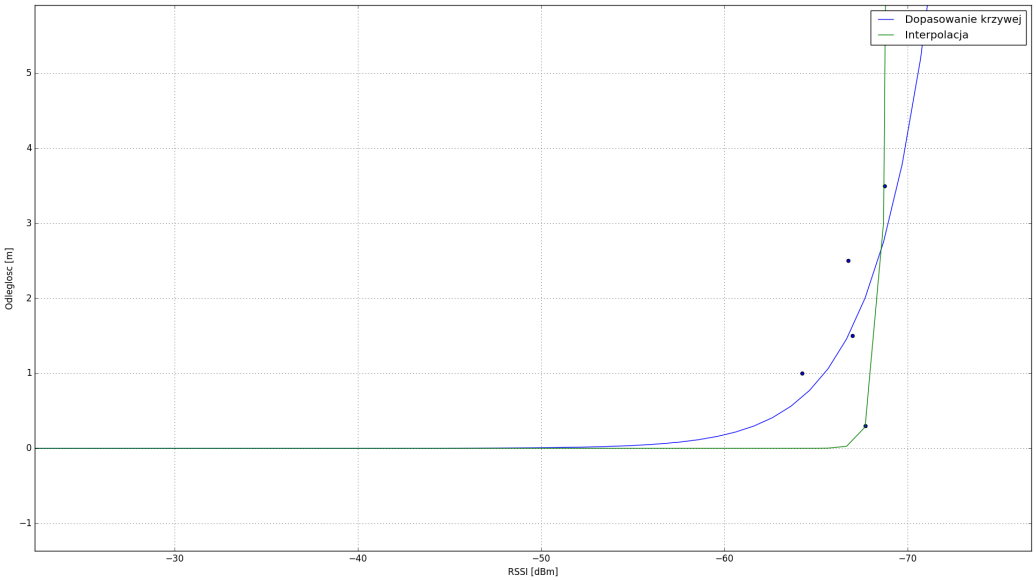
Znacznik	**Interpolacja**	**Dopasowanie krzywej potęgowej**
A	$a = 50,515,\ b = 28,631$	$a = 50,167,\ b = 28,480$
B	$a = 65,036,\ b = 11,547$	$a = 66,028,\ b = 10,074$
C	$a = 65,468,\ b = 7,319$	$a = 68,215,\ b = 0,984$



Rysunek 5.12: Wyznaczanie modelu znacznika A



Rysunek 5.13: Wyznaczanie modelu znacznika B



Rysunek 5.14: Wyznaczanie modelu znacznika C

Trilateracja

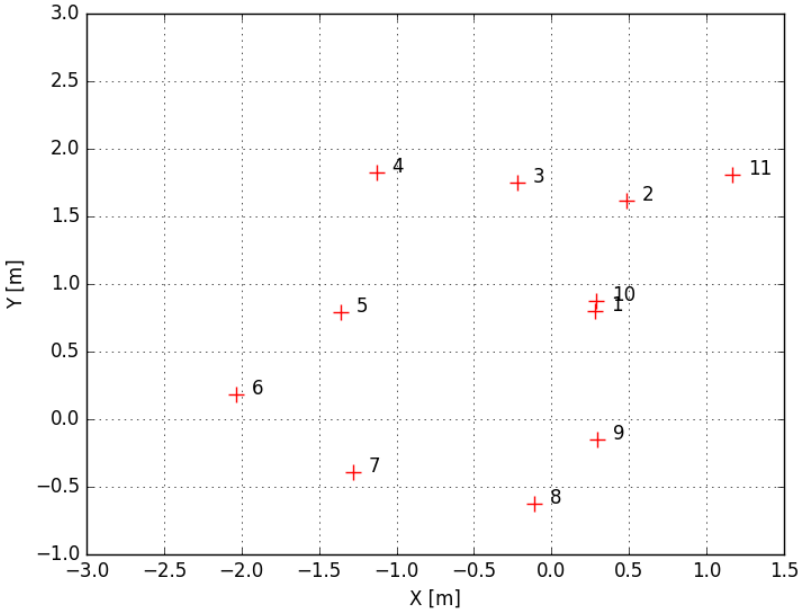
Pierwsze próby lokalizacji za pomocą trilateracji przeprowadzono wykorzystując model wyznaczony w poprzednim punkcie metodą dopasowania krzywej. Jednakże, jak pokazuje rys. 5.16, nastawy te powodują bardzo duży rozrzut wyników pomiaru - większość wyznaczonych położeń robota wypada poza mapą i nie ma związku z rzeczywistym jego położeniem. Rys. 5.16 obrazuje ten problem dla jednego punktu pomiarowego, jednakże podobnie duży rozrzut występuje również dla pozostałych punktów. Należy zwrócić uwagę na inną skalę osi wykresu 5.16 w porównaniu do wykresów 5.17 - 5.27. Dyskusję możliwych przyczyn takiego zachowania przedstawiono w rozdziale 6.

Dlatego też podjęto próbę wyznaczenia właściwego modelu ręcznie, metodą prób i błędów. Bazując na modelach wyznacznonych automatycznie, wykorzystując narzędzie Rosbag oraz narzędzia do wykreślania zależności siły sygnału od odległości, dobrano model zapewniający wyniki o mniejszym rozrzucie:

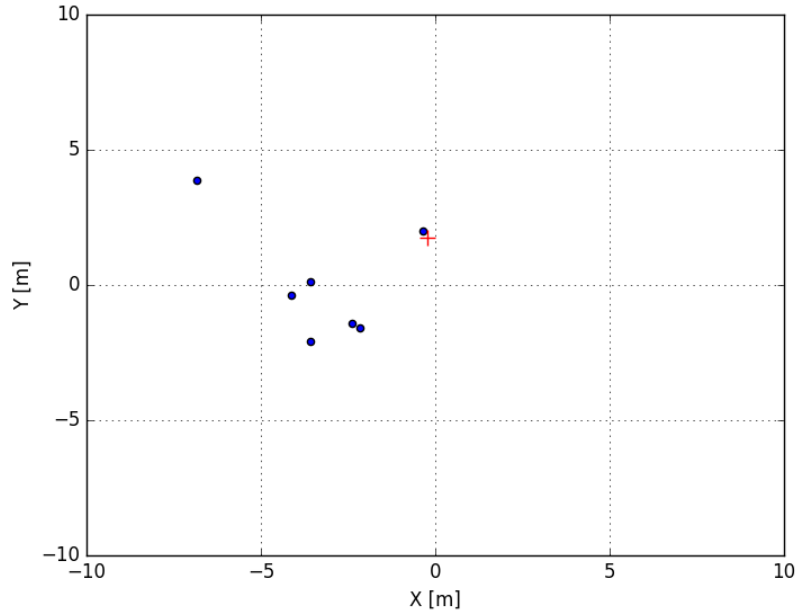
$$a = 50, b = 50 \tag{5.1}$$

Wyniki z wykorzystaniem modelu wyznaczonego ręcznie pokazują rys. 5.17 - 5.27. Dla porównania, rysunki te, z wyjątkiem rys. 5.16 posiadają jednakową skalę. Rysunek 5.16 ma zmienioną skalę aby ukazać problem rozrzutu wyników.

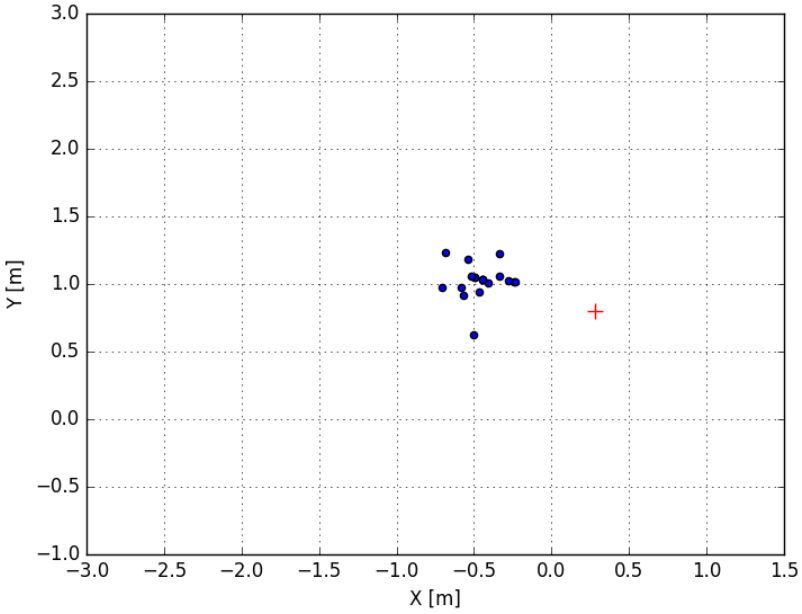
Na rysunkach, czerwony krzyżyk oznacza punkt odniesienia, niebieskie kropki symbolizują wyniki trilateracji. Na rysunku 5.15 przedstawiono położenie kolejnych punktów odniesienia.



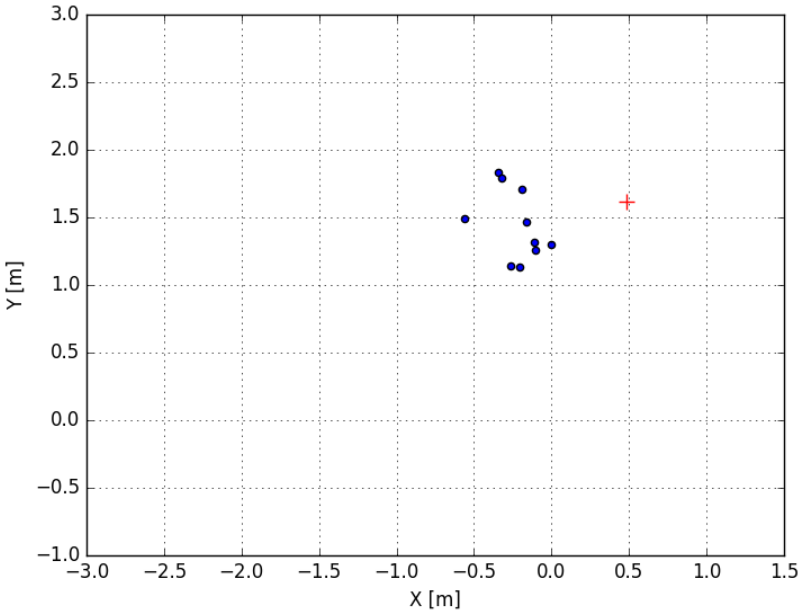
Rysunek 5.15: Położenie punktów pomiarowych.



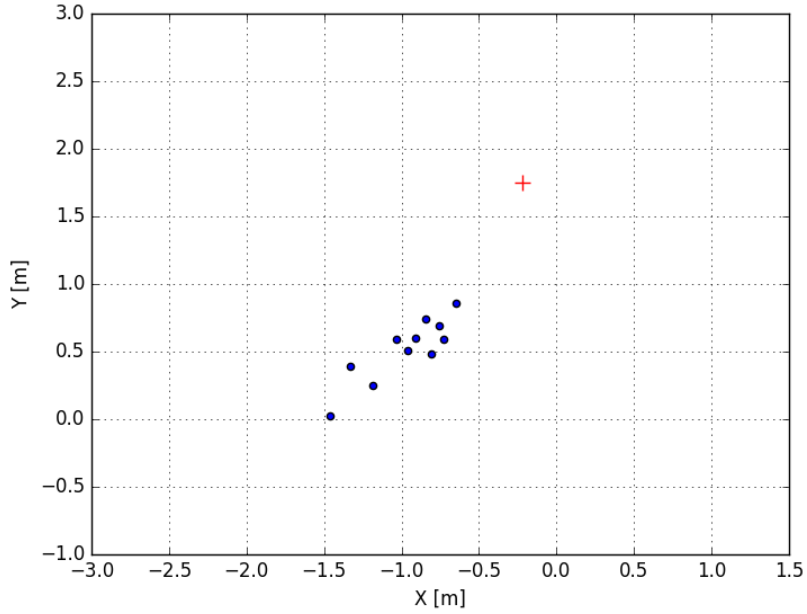
Rysunek 5.16: Wyniki trilateracji dla modelu obliczonego przez dopasowanie krzywej



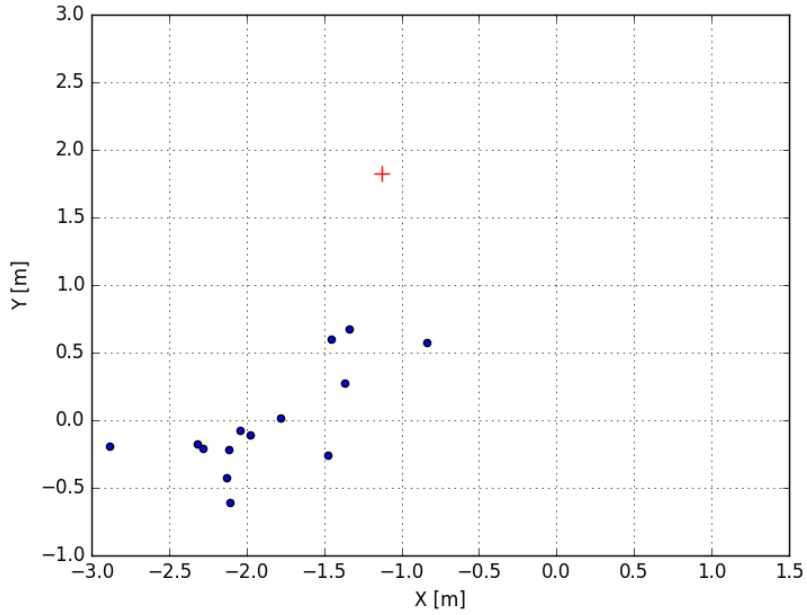
Rysunek 5.17: Wyniki trilateracji dla ręcznie dobranego modelu - punkt 1



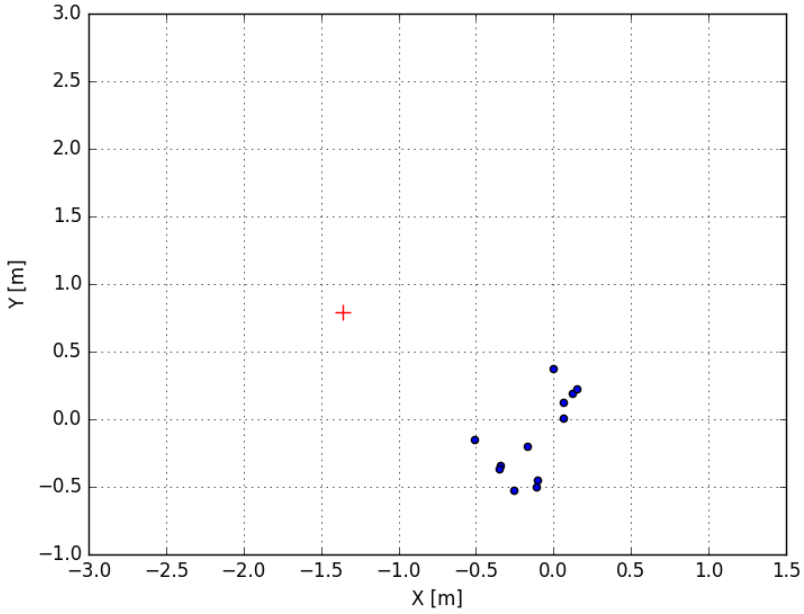
Rysunek 5.18: Wyniki trilateracji dla ręcznie dobranego modelu - punkt 2



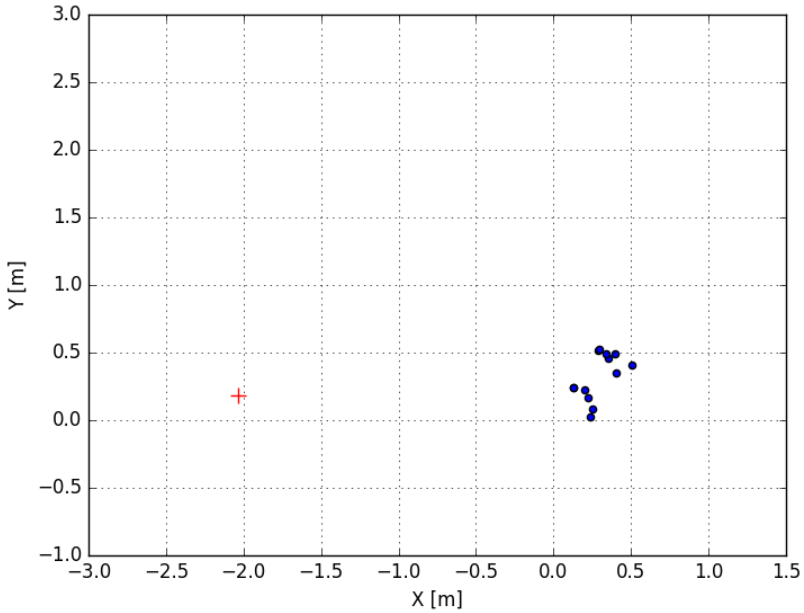
Rysunek 5.19: Wyniki trilateracji dla ręcznie dobranego modelu - punkt 3



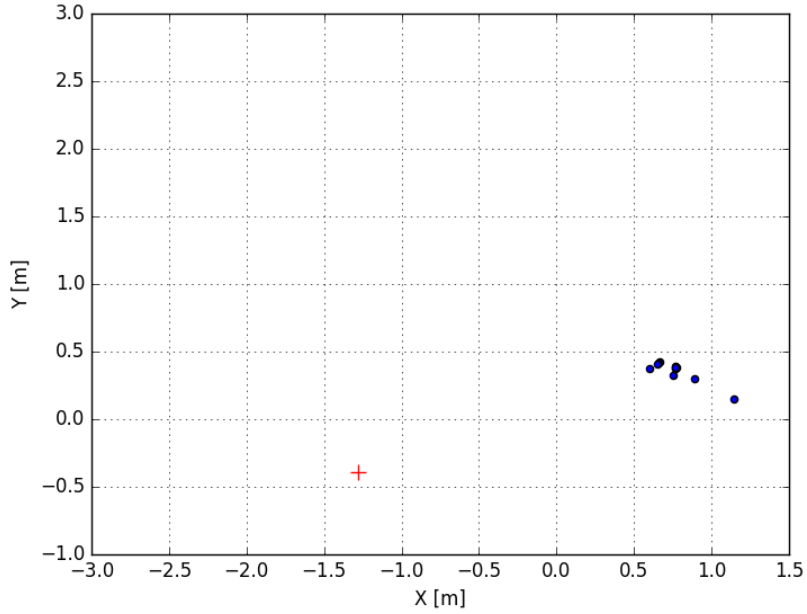
Rysunek 5.20: Wyniki trilateracji dla ręcznie dobranego modelu - punkt 4



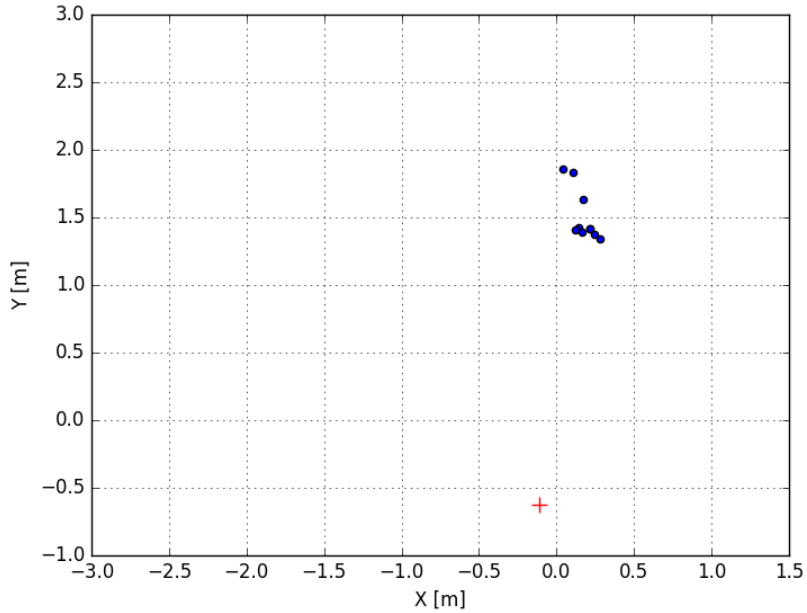
Rysunek 5.21: Wyniki trilateracji dla ręcznie dobranego modelu - punkt 5



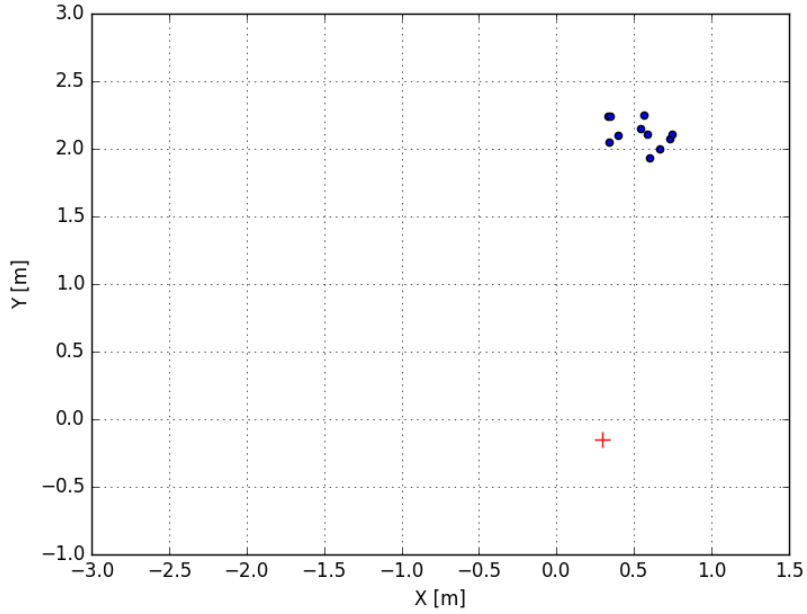
Rysunek 5.22: Wyniki trilateracji dla ręcznie dobranego modelu - punkt 6



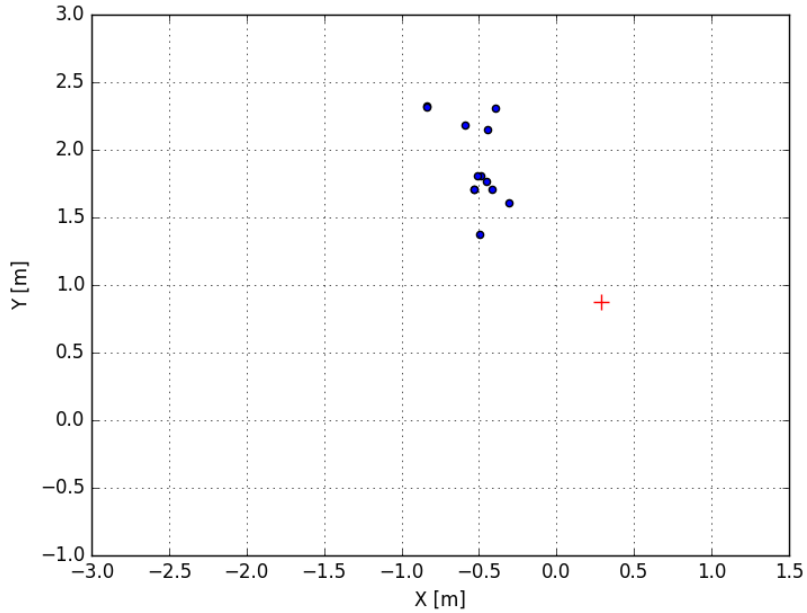
Rysunek 5.23: Wyniki trilateracji dla ręcznie dobranego modelu - punkt 7



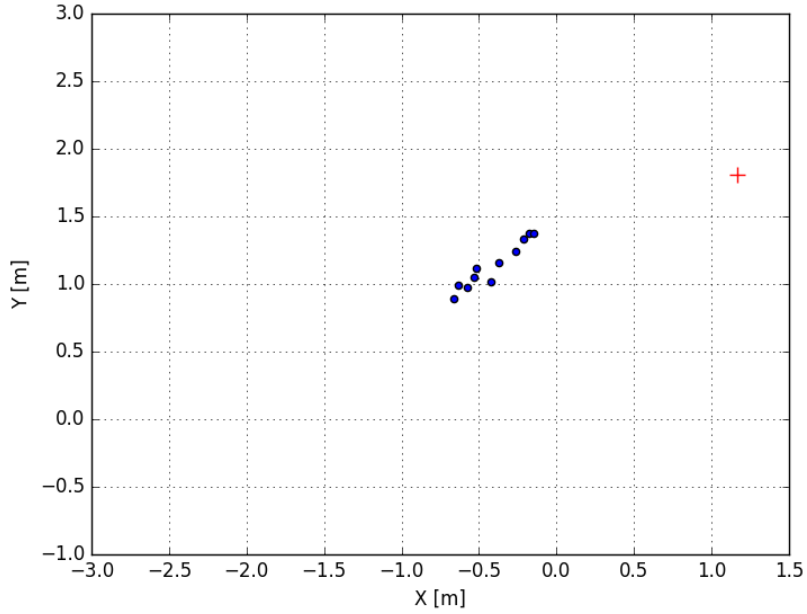
Rysunek 5.24: Wyniki trilateracji dla ręcznie dobranego modelu - punkt 8



Rysunek 5.25: Wyniki trilateracji dla ręcznie dobranego modelu - punkt 9



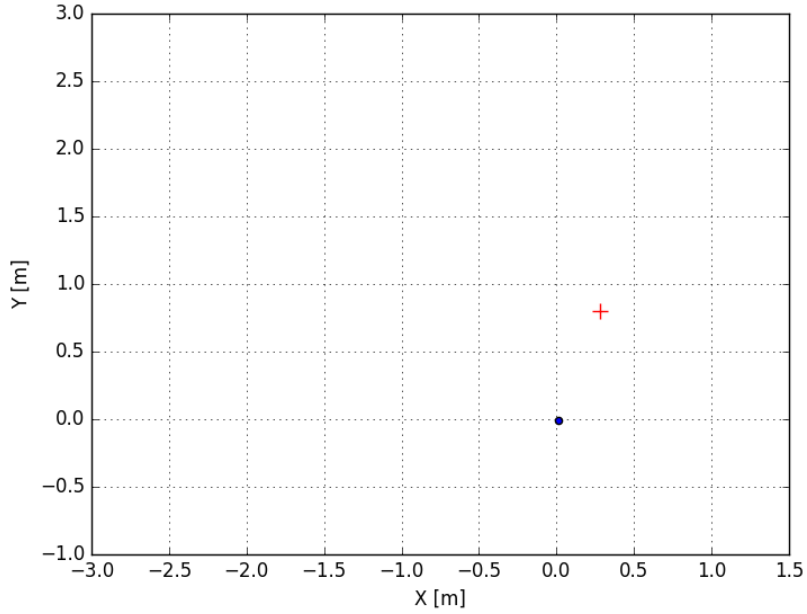
Rysunek 5.26: Wyniki trilateracji dla ręcznie dobranego modelu - punkt 10



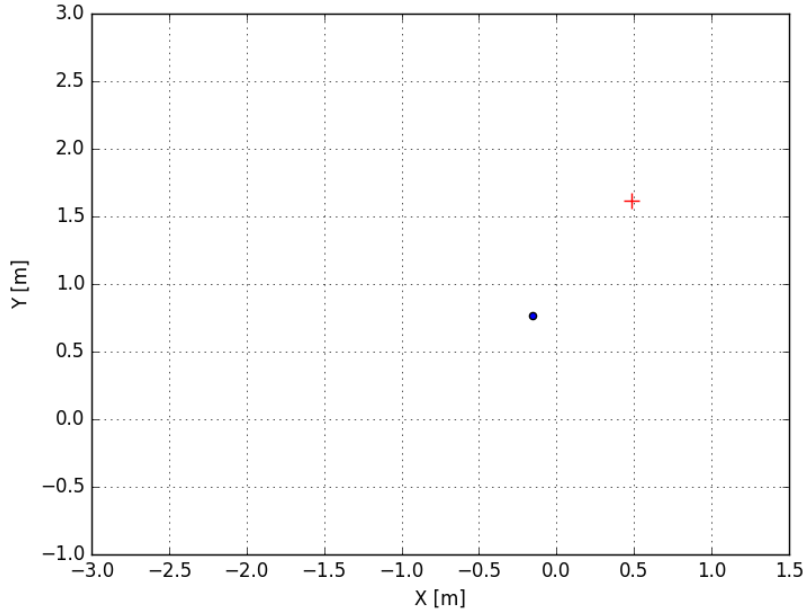
Rysunek 5.27: Wyniki trilateracji dla ręcznie dobranego modelu - punkt 11

Filtr cząsteczkowy

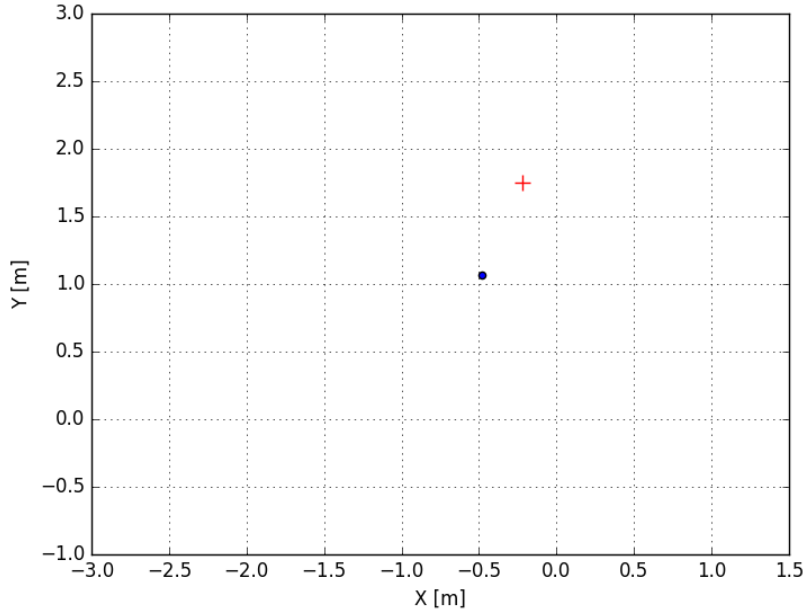
Jak wspomniano wcześniej, badanie filtra cząsteczkowego przeprowadzono na tym samym zbiorze danych co badania metody trilateracji. Podobnie jak w wypadku trilateracji, wykorzystano model wyznacznony ręcznie. Wyniki lokalizacji w poszczególnych punktach ukazują wykresy 5.28 - 5.38. Można zauważyć, że na wykresach zaznaczono miejszą ilość punktów wynikowych niż w wypadku trilateracji. Ta różnica wynika z faktu, że wynik trilateracji jest publikowany w stałym interwale (1 sekunda), natomiast filtr cząsteczkowy jest przeliczany tylko wtedy, kiedy odległość liniowa lub kątowa pokonana przez robota i zmierzona za pomocą odometrii przekroczy pewien próg. Jest to związane z optymalizacją wydajności programu. Dlatego też, kiedy robot jest nieruchomy, nowe punkty nie są publikowane - bowiem byłyby identyczne jak poprzednie.



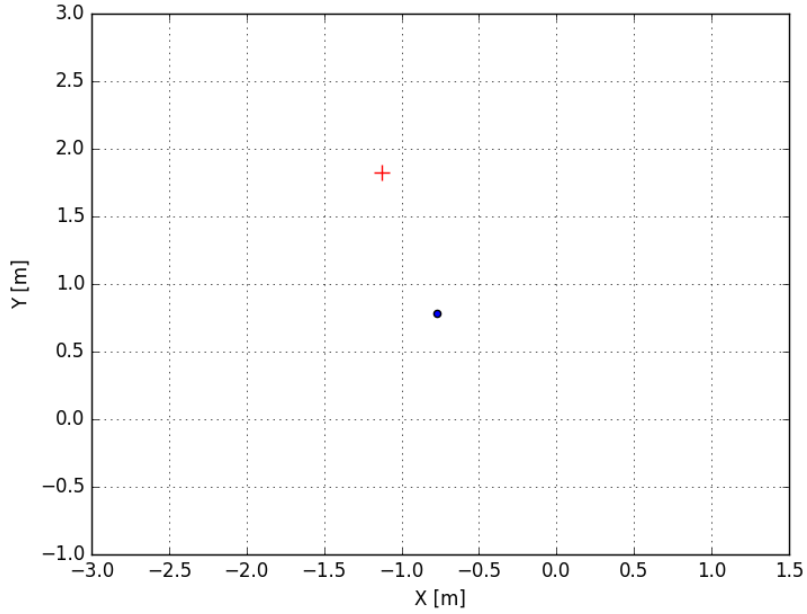
Rysunek 5.28: Wyniki lokalizacji za pomocą filtra cząsteczkowego dla ręcznie dobranego modelu - punkt 1



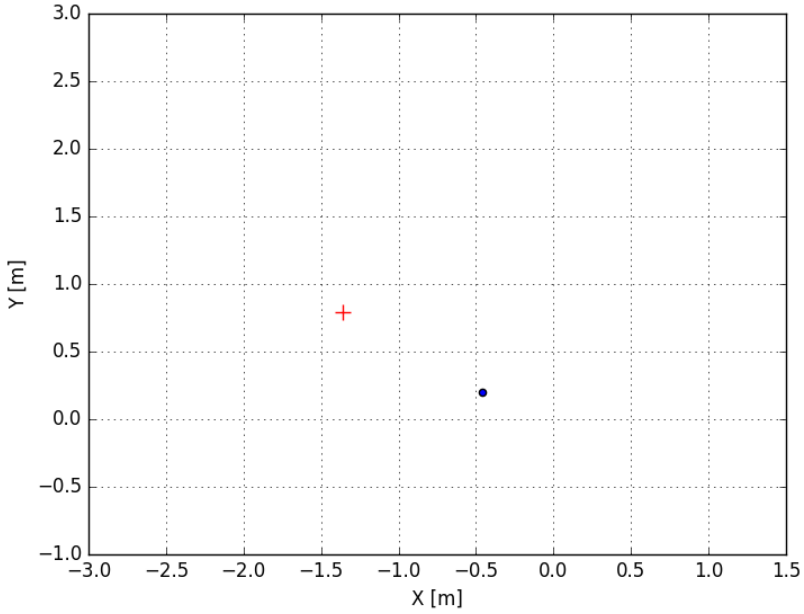
Rysunek 5.29: Wyniki lokalizacji za pomocą filtra cząsteczkowego dla ręcznie dobranego modelu - punkt 2



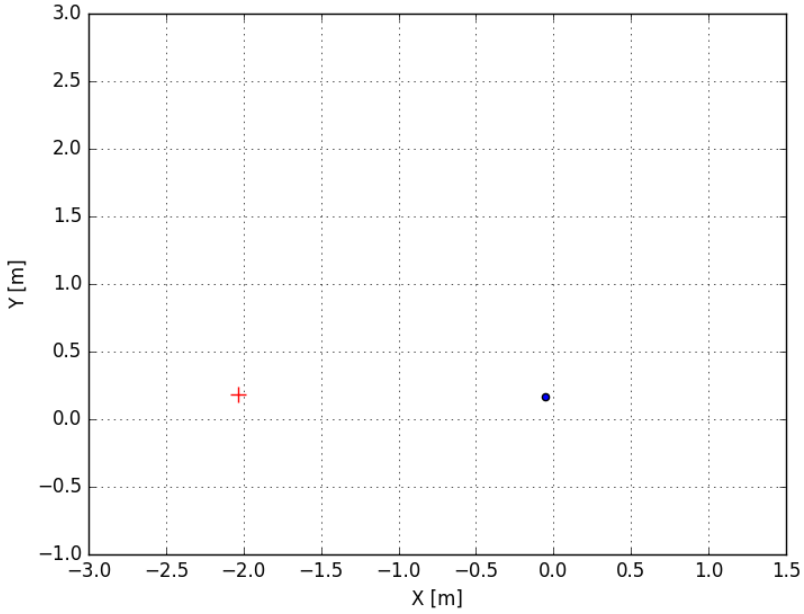
Rysunek 5.30: Wyniki lokalizacji za pomocą filtra cząsteczkowego dla ręcznie dobranego modelu - punkt 3



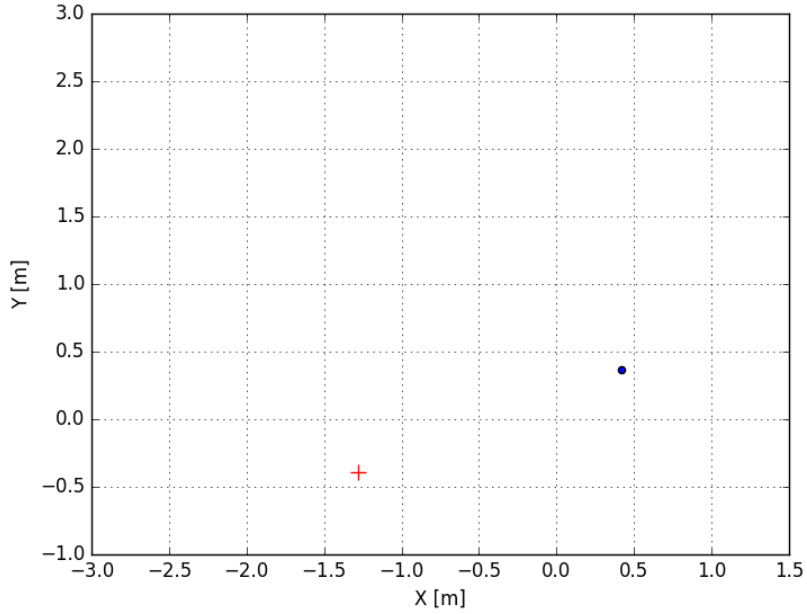
Rysunek 5.31: Wyniki lokalizacji za pomocą filtra cząsteczkowego dla ręcznie dobranego modelu - punkt 4



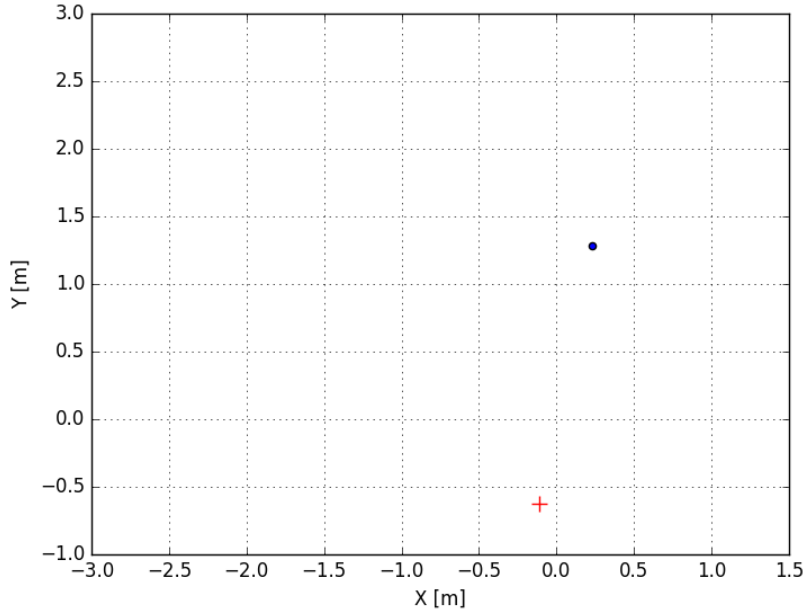
Rysunek 5.32: Wyniki lokalizacji za pomocą filtra cząsteczkowego dla ręcznie dobranego modelu - punkt 5



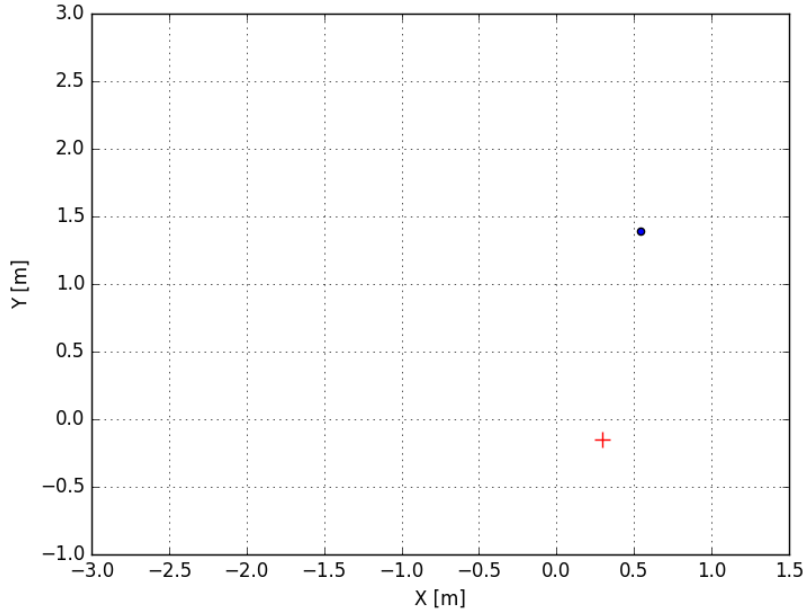
Rysunek 5.33: Wyniki lokalizacji za pomocą filtra cząsteczkowego dla ręcznie dobranego modelu - punkt 6



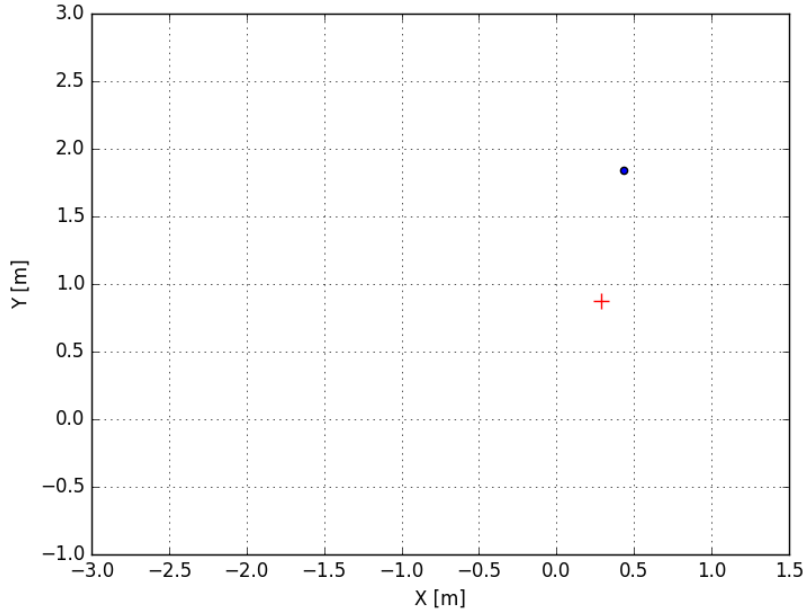
Rysunek 5.34: Wyniki lokalizacji za pomocą filtra cząsteczkowego dla ręcznie dobranego modelu - punkt 7



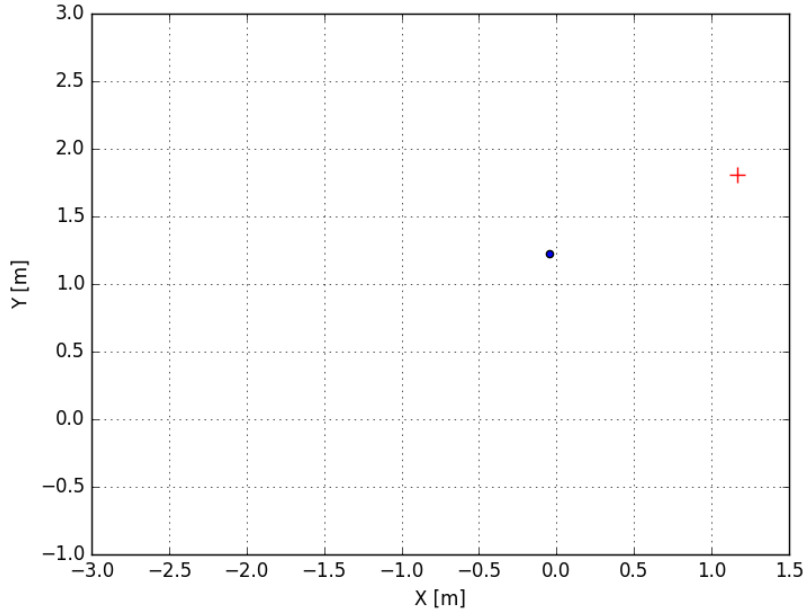
Rysunek 5.35: Wyniki lokalizacji za pomocą filtra cząsteczkowego dla ręcznie dobranego modelu - punkt 8



Rysunek 5.36: Wyniki lokalizacji za pomocą filtra cząsteczkowego dla ręcznie dobranego modelu - punkt 9



Rysunek 5.37: Wyniki lokalizacji za pomocą filtra cząsteczkowego dla ręcznie dobranego modelu - punkt 10



Rysunek 5.38: Wyniki lokalizacji za pomocą filtra cząsteczkowego dla ręcznie dobranego modelu - punkt 11

5.6 Rozwiązanie problemu porwanego robota za pomocą lokalizacji BLE i AMCL

Rozdział 6

Wnioski

Bibliografia

[1] Aswin N Raghavan, Harini Ananthapadmanaban, Manimaran S Sivamurugan, and Balaraman Ravindran. Accurate mobile robot localization in indoor environments using bluetooth. *ANRH Ananthapadmanaban*, 2010.

[2] Bing-Fei Wu, Cheng-Lung Jen, and Kuei-Chung Chang. Neural fuzzy based indoor localization by kalman filtering with propagation channel modeling. *IEEE International Conference on Systems, Man and Cybernetics*, 2007.

[3] Adel THALJAOUI, Thierry VAL, Nejah NASRI, and Damien BRULIN. Ble localization using rssi measurements and iringla. *University of Toulouse*, 2007.

[4] Eduardo Navarro, Benjamin Peuker, and Michael Quan. Wi-fi localization using rssi fingerprinting. *California Polytechnic State University*, 2009.

[5] Evennou and Marx. Advanced integration of wifi and inertial navigation systems for indoor mobile positioning. *Division RD, TECH/IDEA, France Telecom*, 2006.

[6] Willy Herman and William S. Murphy Jr. Determination of a position in three dimensions using trilateration and approximate distances. *Decision Sciences*, 1995.

[7] Aswin N Raghavan, Harini Ananthapadmanaban, Manimaran S Sivamurugan, and Balaraman Ravindran. Accurate mobile robot localization in indoor environments using bluetooth. *Indian Institute of Technology Madras*, 2005.

[8] Erin-Ee-Lin Lau, Boon-Giin Lee, Seung-Chul Lee, and Wan-Young Chung. Enhanced rssi-based high accurancy real-time user location tracking system for indoor and outdoor environments. *International journal on smart sensing and intelligent systems, vol. 1 no. 2*, 2008.

[9] Katarzyna Brzozowska-Rup and Antoni Leon Dawidowicz. Metoda filtru cząsteczkowego. *Matematyka Stosowana 10/2009*, 2009.

[10] Łukasz Chechliński and Daniel Koguciuk. A comprehensive approach to teaching mobile robotics. *Journal of Automation, Mobile Robotics and Intelligent Systems*, 2017.

[11] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. Stanford University, 2000.

[12] Roger R Labbe Jr. *Kalman and Bayesian filters in Python*. 2016.

[13] Barbara Siemiątkowska. *Nawigacja robotów mobilnych - preskrypt*. Instytut Automatyki i Robotyki, Wydział Mechatroniki Politechniki Warszawskiej, 2011.

[14] Jonas Schwertfeger. *Tutorial on a Probabilistic Measurement Model based on Landmark Range and Bearing Informationt*. Brown University Department of Computer Science, 2007.

[15] Jason M. O'Kane. *A Gentle Introduction to ROS*. University of South Carolina, 2013.

[16] Aaron Martinez and Enrique Fernández. *Learning ROS for Robotics Programming*. Packt Publishing, 2013.

[17] Dokumentacja online pakietu ROS. `wiki.ros.org`. Dostęp: 2017-07-11.

[18] Dokumentacja sdk mikrokontrolerów rodziny nrf51. `https://infocenter.nordicsemi.com/index.jsp`. Dostęp: 2017-07-11.

[19] Specyfikacja standardu Bluetooth. `https://www.bluetooth.com/specifications/bluetooth-core-specification`. Dostęp: 2017-07-11.

[20] Dokumentacja biblioteki PyBlueZ. `https://github.com/karulis/pybluez/wiki`. Dostęp: 2017-07-11.

[21] Dokumentacja biblioteki SciPy. `https://docs.scipy.org/`. Dostęp: 2017-07-11.

Wykaz skrótów

AES	Advanced Encryption Standard
API	Application Programming Interface
BLE	Bluetooth Low Energy
FHSS	Frequency Hopping Spread Spectrum
GATT	Generic Attibute
GCC	GNU Compiler Collection
ISM	Industrial, Scientific, Medical (pasmo częstotliwości)
MAC	Media Access Control
RAM	Random Access Memory
ROS	Robot Operating System
RSSI	Radio Signal Strength Indicator
SDK	Software Development Kit
UHF	Ultra High Frequency