

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: Automatyka i Robotyka (AIR)
SPECJALNOŚĆ: Robotyka (ARR)

PROJEKT INŻYNIERSKI

Fuzja danych sensorycznych na potrzeby
sterowania robotem klasy micromouse

A sensor data fusion for control a micromouse
type robot

AUTOR:
Marcin Niestrój

PROWADZĄCY PROJEKT:
dr inż. Mariusz Janiak PWr, I-6

OCENA PROJEKTU:

Spis treści

1	Wstęp	3
2	Koncepcja	5
3	Budowa systemu	7
3.1	Filtr Kalmana	7
3.1.1	Opis matematyczny filtru	7
3.1.2	Model robota	8
3.1.3	Algorytm	9
4	Eksperymenty	13
4.1	Eksperymenty symulacyjne	13
4.2	Eksperymenty na robocie	14
5	Podsumowanie	21

Rozdział 1

Wstęp

Konkurencja micromouse [1] [2] polega na jak najszybszym przejeździe robota z narożnika labiryntu (startu) do jego środka (mety). Labirynt składa się z 256 (16x16) kwadratowych pól, każde o wymiarze 180x180mm. Pola labiryntu rozdzielone są ściankami o szerokości 12mm i wysokości 50mm. Środek labiryntu stanowią pola o współrzędnych (8,8), (8,9), (9,8) i (9,9). Maksymalne wymiary robota są ograniczone jedynie przez wymiary labiryntu, w którym przestrzeń między dwoma sąsiednimi, równoległymi ścianami jest równa 168mm. Konkurencja zazwyczaj przebiega w dwóch fazach. W pierwszej fazie robot przeszukuje labirynt w celu znalezienia możliwych ścieżek prowadzących do jego środka. W fazie drugiej robot ma za zadanie przejechać od startu do mety w jak najkrótszym czasie. Zawody wygrywa robot, który osiągnie najmniejszy czas przejazdu od startu do mety. Konkurencja micromouse wymaga od konstruktora połączenia wiedzy wielu dziedzin związanych z robotyką, takich jak: poruszanie się robota w nieznanym środowisku, sprawny algorytm wyszukiwania najszybszej ścieżki, układ sterowania robotem oraz praktyczna realizacja konstrukcji mechanicznej.

Najlepsze na świecie roboty w konkurencji micromouse osiągają prędkości ponad 1m/s podczas skrętów oraz 3.5m/s podczas poruszania się po prostej. Tak szybki przejazd jest możliwy dzięki bardzo dobrej stronie mechanicznej konstrukcji, zaawansowanemu algorytmowi sterowania oraz zależy od jakości informacji o aktualnym stanie robota, jego pozycji w labiryncie.

Algorytmy lokalizacji robotów mobilnych są opisywane w wielu artykułach naukowych. Wszystkie przedstawione w nich metody wykorzystują dane pomiarowe z kilku układów sensorycznych. Najczęściej stosowanymi czujnikami w przypadku robotów mobilnych są enkodery oraz żyroskop. Stosuje się także sensory, dostarczające informacji o pozycji w zewnętrznym układzie odniesienia lub korzystające z charakterystycznych cech otoczenia, dzięki czemu możliwe jest zniwelowanie kumulujących się błędów określania pozycji na bazie estymowanych przemieszczeń.

Podstawowym układem sensorycznym w robotach mobilnych są enkodery zamocowane na kołach lub na osiach mechanicznie z nimi sprzężonymi. Znając wymiary kół oraz odległość między kołami, można obliczyć zależność przemieszczenia liniowego i kąтового od obrotu kół robota. Taka metoda określania pozycji nazywa się odometrią [3]. Źródła błędów odometrii można podzielić na dwie grupy [4]

1. Błędy systematyczne

- Nierówny obwód kół
- Niedokładne ułożenie kół w robocie

- Niepewność rozstawu kół związana z niepunktowym kontaktem koła z podłożem
- Ograniczona rozdzielczość enkoderów

2. Błędy niesystematyczne

- Nierówna powierzchnia po której się porusza robot (wraz z niespodziewanymi obiektami na podłożu)
- Poślizgi wzdłużne
- Poślizgi poprzeczne

Zasadniczym źródłem błędów w ustaleniu pozycji robota jest niepoprawne określenie orientacji robota. Prowadzi ono do systematycznie powiększającego się błędu pozycji w globalnym układzie odniesienia. Informacja o pozycji wyznaczona na podstawie pomiarów z enkoderów jest więc niewystarczająca w większości zastosowań.

Drugim często stosowanym czujnikiem jest żyroskop. Informacja pochodząca z tego czujnika jest wykorzystywana do zmniejszenia błędu orientacji metody odometrii. Wartością mierzoną w żyroskopie jest prędkość kątowna. Interesującą nas orientację robota można obliczyć poprzez całkowanie prędkości obrotowej, co w dyskretnym systemie jest równoważne sumowaniu wartości pomiarowych. Z czujnikiem żyroskopowym nie są związane żadne błędy niesystematyczne. Ważną cechą tych sensorów jest dryf [6], czyli niezerowa wartość mierzona prędkości kątownej w nieruchomym układzie. Prowadzi to do systematycznie rosnącego błędu określenia orientacji robota. Z tego względu dryf żyroskopu należy niwelować poprzez fuzję z innymi danymi sensorycznymi lub poprzez częstą kalibrację w nieruchomym układzie.

Opisane wyżej sensory, enkodery oraz żyroskop, umożliwiają określenie zmieniającej się pozycji robota w zewnętrznym układzie odniesienia, z niewielkimi niepewnościami pomiarowymi. Kumulowanie się błędów wyznaczania pozycji za pomocą tych czujników prowadzi jednak do konieczności zastosowania dodatkowych układów sensorycznych [7]. Dobrym przykładem jest system nawigacji satelitarnej GPS. Dostarcza on bezpośrednio pozycję robota w globalnym układzie odniesienia. Zastosowanie tego systemu ograniczone jest jednak przez słabą dokładność pomiarową i brak możliwości odbioru sygnału w budynkach. Alternatywnym układem sensorycznym są czujniki czerpiące informacje z charakterystycznych cech otoczenia. Są nimi m.in. kamery, dalmierze optyczne i ultradźwiękowe, czujniki dotykowe. W tym projekcie użyte zostaną dalmierze optyczne składające się z diody podczerwieni i fototranzystora. Układ taki ma możliwość dużej częstotliwości pomiaru, natomiast jego dokładność jest wystarczająca przy pomiarze małych odległości robota od ścian labiryntu.

Rozwiązaniem problemów występujących przy określaniu pozycji robota jest zastosowanie fuzji danych sensorycznych. W projekcie tym, zrealizowano to z wykorzystaniem implementacji filtra Kalmana. Rozwiązanie to pozwala na uwzględnienie pomiarów tych samych wielkości fizycznych, mierzonych za pomocą różnych czujników. Enkodery i żyroskop służą do określania przemieszczenia względem pozycji początkowej, natomiast zastosowanie dalmierzy optycznych pozwala na korekcję pozycji i orientacji robota względem ścian. Projekt zrealizowano na istniejącej konstrukcji micromouse Rapid. Jest to robot klasy (2,0) wyposażony w dwa niezależnie napędzane koła za pomocą silników DC.

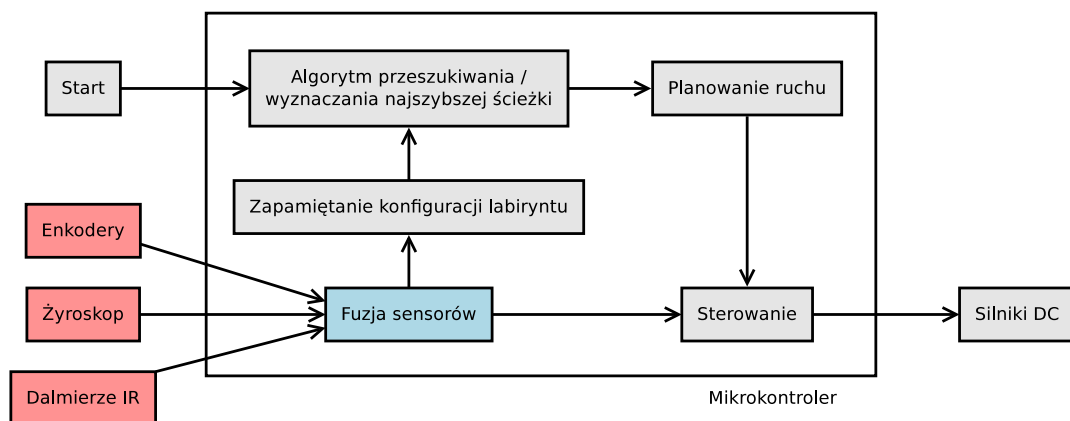
Rozdział 2

Koncepcja

Głównym celem konkurencji micromouse jest dotarcie do środka labiryntu. Aby to zrealizować, robot musi posiadać informację o swojej aktualnej pozycji, rozpoznać w których miejscach są ściany, zapamiętać konfigurację labiryntu oraz wykryć dotarcie do środka labiryntu. Algorytm działania robota można więc podzielić na kilka modułów:

- akwizycja danych sensorycznych,
- określenie pozycji robota na podstawie informacji z czujników (fuzja sensorów),
- algorytm wyznaczania najkrótszej bądź najszybszej ścieżki prowadzącej do środka labiryntu,
- wykrycie i zapamiętanie miejsc w których są ściany,
- planowanie ruchu robota,
- sterowanie pozycją i prędkością,
- odpowiednie wysterowanie silników DC.

Fuzja danych sensorycznych ma kluczowy wpływ na działanie całego robota. Jej głównym zadaniem jest ustalanie pozycji i orientacji, natomiast pośrednio wpływa na jakość sterowania i prawidłowe odczytanie konfiguracji labiryntu. Diagram przedstawiający działanie robota micromouse oraz miejsce fuzyfikacji w algorytmie sterującym robotem przedstawia rysunek 2.1.



Rysunek 2.1 Schemat blokowy robota klasy micromouse

Dane wejściowe filtru to wartości pomiarowe ze wszystkich dostępnych czujników, czyli enkoderów, żyroskopu i dalmierzy optycznych. Danymi wyjściowymi filtru są natomiast: pozycja x i y w kartezjańskim układzie współrzędnych oraz orientacja robota θ . Z uwagi na możliwość wykorzystania informacji z dalmierzy optycznych tylko w charakterystycznych miejscach w labiryncie, głównymi czujnikami odpowiedzialnymi za określenie pozycji robota są enkodery i żyroskop.

Pomiary dostępne z enkoderów magnetycznych są źródłem wiedzy o obrocie kół robota. Na tej podstawie możliwe jest obliczenie przemieszczenia liniowego i kąтового konstrukcji. W projekcie tym wykorzystana zostanie jedynie informacja o przemieszczeniu liniowym. Zdecydowano o tym, ze względu na często występujące poślizgi robota, decydujące o dużej niedokładności w określaniu orientacji konstrukcji.

Wartości pomiarowe z żyroskopu są źródłem informacji o prędkości kątowej. Uzyskanie informacji o orientacji robota względem położenia początkowego jest możliwe poprzez całkowanie mierzonej prędkości obrotowej. Otrzymana wartość orientacji obciążona jest zwiększającym się błędem systematycznym, natomiast zawiera ona bardzo mały współczynnik zawartości szumów pomiarowych. Cechy te świadczą o możliwości przyjęcia pomiarów dostępnych z czujnika żyroskopowego jako rzeczywistej wartości prędkości kątowej robota. Błąd systematyczny określania orientacji, wynikający z dryfu żyroskopu, niwelowany jest natomiast przez wykorzystanie pomiarów z czujników optycznych.

Podczas przejazdu robota micromouse, korekcja pozycji za pomocą dalmierzy możliwa jest tylko w sytuacji gdy robot znajduje się obok ściany lub bezpośrednio przed ścianą. Sytuacje te występują jednak na tyle często, że dalmierze służą jako bardzo dobre źródło informacji do korekcji błędów przemieszczenia liniowego rzędu kilku centymetrów oraz błędów orientacji robota rzędu nawet kilkudziesięciu stopni.

Wyjście filtru zapewniającego fuzję danych sensorycznych w postaci dokładnego położenia robota pozwoli na całkowite rozdzielenie modułu sterowania robota od pozyskiwanych danych sensorycznych. Zapewniona zostanie w ten sposób możliwość osobnego rozwijania algorytmu sterowania na podstawie danych dostępnych przez filtr oraz algorytmu akwizycji danych pomiarowych i ich fuzyfikacji.

Rozdział 3

Budowa systemu

3.1 Filtr Kalmana

Nazwa Filtru Kalmana pochodzi od nazwiska Rudolpha E. Kalmana, który w 1960 roku opublikował słynną pracę, opisującą rekursywne rozwiązanie problemu filtracji dyskretnych sygnałów liniowych. Filtr ten jest typem estymatora działającego w dwóch fazach: predykcji oraz korekcji. Od początku swojego istnienia, filtr Kalmana wykorzystywany był w wielu dziedzinach nauki i techniki. Znalazł on zastosowanie przede wszystkim w układach sensorycznych pojazdów autonomicznych, pojazdach zdalnie naprowadzanych, samolotach, promach kosmicznych, przetwarzaniu sygnałów, a także w ekonomii.

3.1.1 Opis matematyczny filtru

Filtr Kalmana pozwala dokonać fuzji informacji pochodzących z wielu zakłóconych źródeł, dając w wyniku wartości, często niedostępne bezpośrednio w danych pomiarowych. Filtr wykorzystuje wiedzę o systemie i modelu pomiarowym, opisie probabilistycznym niepewności systemu i pomiarów, oraz wszystkich początkowych wartościach stanu.

Model systemu opisuje jak stan systemu zmienia się w czasie pod wpływem sterowań. Jest on wykorzystywany w fazie predykcji filtru. Model systemu określany jest za pomocą liniowego równania (3.1), gdzie x_k oznacza wektor stanu robota w czasie k , macierz A określa zależność stanu systemu w poprzednim kroku do aktualnego stanu, u_{k-1} jest sterowaniem, macierz B określa zależność pomiędzy sterowaniem a aktualnym stanem systemu, natomiast w_{k-1} modeluje szum systemu.

Filtr Kalmana wykorzystuje model pomiaru, do korekcji estymowanego stanu, bazując na dostępnych danych pomiarowych. Równanie (3.2) opisuje proces pomiarowy, w którym H oznacza zależność pomiędzy wartością mierzoną a stanem systemu, natomiast v_k modeluje wektor niepewności pomiarowych.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (3.1)$$

$$z_k = Hx_k + v_k \quad (3.2)$$

Elementy losowe w_k i v_k reprezentowane są przez białe szumy Gaussa, których wartość średnia jest równa 0, natomiast wariancje są równe: Q dla szumu procesu oraz R dla szumu pomiarowego (3.3). Wariancje procesu i pomiaru są parametrami wpływającymi na działanie filtru Kalmana.

$$\begin{aligned} p(w) &\sim N(0, Q) \\ p(v) &\sim N(0, R) \end{aligned} \quad (3.3)$$

Algorytm Dyskretnej Filtracji Kalmana estymuje rzeczywisty stan robota poprzez zastosowanie dwóch faz: predykcji i korekcji. Pierwszym etapem filtracji jest faza predykcji. Na podstawie stanu w poprzednim kroku oraz znanego modelu systemu, estymowane są wartości aktualnego stanu systemu. Równanie (3.4) przedstawia sposób obliczania wektora x stanu robota. W każdym kroku następuje aktualizacja macierzy kowariancji P_k , zgodnie ze wzorem (3.5).

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (3.4)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (3.5)$$

Drugim etapem w Filtracji Kalmana jest faza korekcji, w której uaktualniamy wektor stanu na podstawie wykonanych pomiarów. Pierwszym krokiem jest obliczenie wzmocnienia Kalmana (3.6), które oznacza w pewnym sensie jak bardzo pomiar wpłynie na estymowaną wartość stanu. Obliczenie wartości wzmocnienia K wykonywane jest z uwzględnieniem wariancji modelu pomiarowego R . Kolejnym działaniem jest skorygowanie nowego wektora stanu, na podstawie różnicy pomiędzy wartością pomiarową oraz wartością estymowaną w fazie predykcji (3.7). Ostatnim elementem filtracji jest uaktualnienie macierzy kowariancji P_k (3.8).

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (3.6)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (3.7)$$

$$P_k = (I - K_k H)P_k^- \quad (3.8)$$

3.1.2 Model robota

Układ równań (3.9) przedstawia opis przemieszczenia liniowego zastosowanego w filtrze Kalmana, w którym v_k oznacza prędkość robota w chwili k , natomiast a_k oznacza przyspieszenie robota w chwili k .

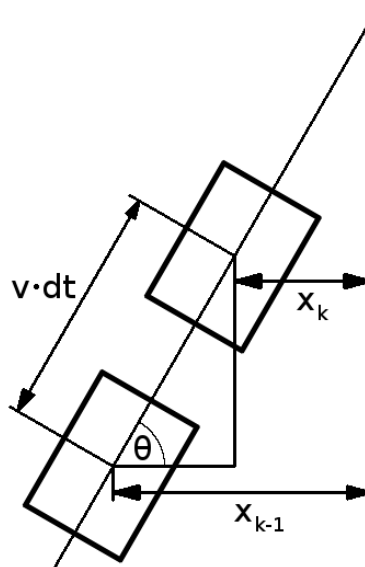
$$\begin{cases} v_k = v_{k-1} + a_{k-1} \cdot dt \\ a_k = a_{k-1} \end{cases} \quad (3.9)$$

Równania (3.10) opisują estymowaną orientację i prędkość kątową, gdzie θ_k oznacza orientację, ω_k prędkość kątową, natomiast ω_{gyro} jest prędkością kątową odczytaną z żyroskopu.

$$\begin{cases} \theta_k = \theta_{k-1} + \omega_k \cdot dt \\ \omega_k = \omega_{gyro} \end{cases} \quad (3.10)$$

Orientację robota można wyznaczyć także na podstawie informacji o odległości robota od ściany. Zależność ta przedstawiona jest na rysunku 3.1. Relację między orientacją a zmianą odległości od ściany w jednostce czasu opisuje więc równanie (3.11).

$$\begin{cases} \theta_k = \arccos\left(\frac{x_k - x_{k-1}}{v_k \cdot dt}\right), \text{ przy poruszaniu się robota w kierunku równoległym z osią OY} \\ \theta_k = \arcsin\left(\frac{y_k - y_{k-1}}{v_k \cdot dt}\right), \text{ przy poruszaniu się robota w kierunku równoległym z osią OX} \end{cases} \quad (3.11)$$



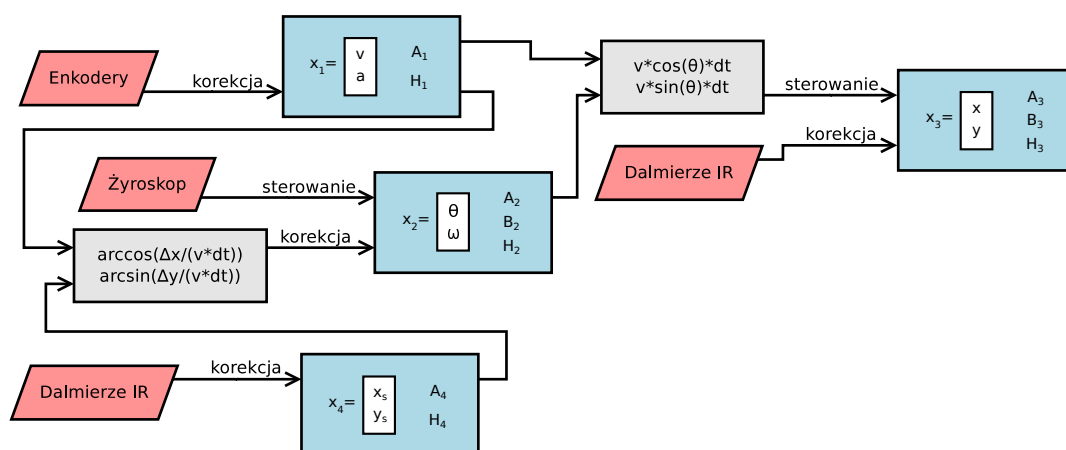
Rysunek 3.1 Sposób wyznaczania orientacji robota na podstawie odległości od ściany

Układ równań (3.12) przedstawia model matematyczny określający pozycję x i y robota w globalnym układzie odniesienia, gdzie x_k oznacza współrzędną x w chwili k , natomiast y_k współrzędną y w chwili k .

$$\begin{cases} x_k = x_{k-1} + v_k \cdot \cos(\theta_k) \cdot dt \\ y_k = y_{k-1} + v_k \cdot \sin(\theta_k) \cdot dt \end{cases} \quad (3.12)$$

3.1.3 Algorytm

Na rysunku 3.2 przedstawiono schemat ideowy algorytmu fuzji danych sensorycznych. Czerwonym kolorem oznaczono wejścia filtra, natomiast niebieskie bloki stanowią wyjścia. Cały algorytm podzielono na cztery moduły, które są od siebie wzajemnie zależne. Dane sensoryczne uwzględniane w fazie predykcji filtra w postaci sterowania u są oznaczone na diagramie jako *sterowanie*. Informacje z czujników, które są brane pod uwagę w fazie korekcji opisane są na schemacie jako *korekcja*.



Rysunek 3.2 Diagram przedstawiający algorytm fuzji danych sensorycznych

Pierwszą częścią algorytmu jest filtracja prędkości liniowej. Jej celem jest dostarczenie

informacji o prędkości dla pozostałych modułów algorytmu. Zdecydowano się na wprowadzenie dodatkowego parametru dla wektora stanu, którym jest przyspieszenie liniowe. Działanie to ma na celu poprawienie własności dynamicznych filtru, szybszego reagowania na zachodzące zmiany. Równania (3.13) przedstawiają obliczanie nowych wartości wektora stanu x_1 .

$$\left\{ \begin{array}{l} \overbrace{\begin{bmatrix} v_k \\ a_k \end{bmatrix}}^{x_{1,k}^-} = \overbrace{\begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}}^{A_1} \cdot \overbrace{\begin{bmatrix} v_{k-1} \\ a_{k-1} \end{bmatrix}}^{x_{1,k-1}^-} \\ \overbrace{\begin{bmatrix} v_k \\ a_k \end{bmatrix}}^{x_{1,k}} = \overbrace{\begin{bmatrix} v_k \\ a_k \end{bmatrix}}^{x_{1,k}^-} + K_{1(2 \times 2)} \cdot \left(\underbrace{\begin{bmatrix} v_{enc} \end{bmatrix}}_{z_{1,k}} - \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{H_1} \cdot \underbrace{\begin{bmatrix} v_k \\ a_k \end{bmatrix}}_{x_{1,k}^-} \right) \end{array} \right. \quad (3.13)$$

Przyjęte wartości wariancji procesu i pomiaru zostały przedstawione w równaniu (3.14).

$$\begin{aligned} Q_1 &= \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \\ R_1 &= 200 \end{aligned} \quad (3.14)$$

Celem kolejnego modułu w algorytmie filtracji jest estymowanie orientacji i prędkości kątowej robota. Wartościami pomiarowymi mającymi wpływ na ten fragment algorytmu są: prędkość kątowa ω_{gyro} z czujnika żyroskopowego oraz orientacja określona na podstawie zmiany odległości od ścian labiryntu ($x_{s,k}$ oraz $y_{s,k}$). Prędkość kątowa ω_{gyro} uzyskana z żyroskopu uwzględniania jest już w fazie predykcji i służy jako sterowanie u . Powodem tego jest niski poziom szumów tego czujnika oraz rosnący błąd systematyczny, którym jest dryf. Błąd ten uwzględnia się w procesie pomiarowym, korygując wartość orientacji robota za pomocą zależności (3.11). Podczas poruszania się robota w kierunku równoległym do osi OY, postać układu równań opisujących obliczanie nowych wartości wektora stanu x_2 jest przedstawiona na (3.15a). W przypadku poruszania się w kierunku równoległym do osi OX postać równań opisuje układ (3.15b).

$$\left\{ \begin{array}{l} \overbrace{\begin{bmatrix} \theta_k \\ \omega_k \end{bmatrix}}^{x_{2,k}^-} = \overbrace{\begin{bmatrix} 1 & dt \\ 0 & 0 \end{bmatrix}}^{A_2} \cdot \overbrace{\begin{bmatrix} \theta_{k-1} \\ \omega_{k-1} \end{bmatrix}}^{x_{2,k-1}^-} + \overbrace{\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}}^{B_2} \cdot \overbrace{\begin{bmatrix} 0 \\ \omega_{gyro} \end{bmatrix}}^{u_{2,k-1}} \\ \overbrace{\begin{bmatrix} \theta_k \\ \omega_k \end{bmatrix}}^{x_{2,k}} = \overbrace{\begin{bmatrix} \theta_k \\ \omega_k \end{bmatrix}}^{x_{2,k}^-} + K_{2(2 \times 2)} \cdot \left(\underbrace{\left[\arccos \left(\frac{x_{s,k} - x_{s,k-1}}{v_k \cdot dt} \right) \right]}_{z_{2,k}} - \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{H_2} \cdot \underbrace{\begin{bmatrix} \theta_k \\ \omega_k \end{bmatrix}}_{x_{2,k}^-} \right) \end{array} \right. \quad (3.15a)$$

$$\left\{ \begin{array}{l} \underbrace{\begin{bmatrix} \theta_k \\ \omega_k \end{bmatrix}}_{x_{2,k}} = \underbrace{\begin{bmatrix} 1 & dt \\ 0 & 0 \end{bmatrix}}_{A_2} \cdot \underbrace{\begin{bmatrix} \theta_{k-1} \\ \omega_{k-1} \end{bmatrix}}_{x_{2,k-1}} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}}_{B_2} \cdot \underbrace{\begin{bmatrix} 0 \\ \omega_{gyro} \end{bmatrix}}_{u_{2,k-1}} \\ \underbrace{\begin{bmatrix} \theta_k \\ \omega_k \end{bmatrix}}_{x_{2,k}} = \underbrace{\begin{bmatrix} \theta_k \\ \omega_k \end{bmatrix}}_{x_{2,k}^-} + K_{2(2x2)} \cdot \left(\underbrace{\begin{bmatrix} \arcsin\left(\frac{y_{s,k} - y_{s,k-1}}{v_k \cdot dt}\right) \end{bmatrix}}_{z_{2,k}} - \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{H_2} \cdot \underbrace{\begin{bmatrix} \theta_k \\ \omega_k \end{bmatrix}}_{x_{2,k}^-} \right) \end{array} \right. \quad (3.15b)$$

Przyjęte wartości wariancji Q i R zostały zaprezentowane w równaniu (3.16).

$$Q_1 = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \quad (3.16)$$

$$R_1 = 100000$$

Określenie współrzędnych położenia robota wymaga znajomości przebytej drogi oraz orientacji robota. Informacje te są dostępne poprzez wyżej opisane moduły. Obliczane wartości współrzędnych położenia, zgodnie ze wzorem (3.12), stanowią wejście do filtru w postaci sterowania u . Pomiary z dalmierzy optycznych są uwzględniane natomiast w fazie korekcji filtru Kalmana. Postać równań opisujących obliczanie nowych wartości wektora stanu x_3 została przedstawiona na (3.17a) oraz (3.17b), odpowiednio dla każdego kierunku ruchu robota, gdzie x_{pom} oraz y_{pom} to odpowiednio wartość współrzędnej x oraz wartość współrzędnej y , dostępna z pomiaru czujnika odległości.

$$\left\{ \begin{array}{l} \underbrace{\begin{bmatrix} x_k \\ y_k \end{bmatrix}}_{x_{3,k}} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{A_3} \cdot \underbrace{\begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix}}_{x_{3,k-1}} + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{B_3} \cdot \underbrace{\begin{bmatrix} v_k \cdot \cos(\theta_k) \cdot dt \\ v_k \cdot \sin(\theta_l) \cdot dt \end{bmatrix}}_{u_{3,k-1}} \\ \underbrace{\begin{bmatrix} x_k \\ y_k \end{bmatrix}}_{x_{3,k}} = \underbrace{\begin{bmatrix} x_k \\ y_k \end{bmatrix}}_{x_{3,k}^-} + K_{3(2x2)} \cdot \left(\underbrace{\begin{bmatrix} x_{pom} \end{bmatrix}}_{z_{3,k}} - \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{H_3} \cdot \underbrace{\begin{bmatrix} x_k \\ y_k \end{bmatrix}}_{x_{3,k}^-} \right) \end{array} \right. \quad (3.17a)$$

$$\left\{ \begin{array}{l} \underbrace{\begin{bmatrix} x_k \\ y_k \end{bmatrix}}_{x_{3,k}} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{A_3} \cdot \underbrace{\begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix}}_{x_{3,k-1}} + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{B_3} \cdot \underbrace{\begin{bmatrix} v_k \cdot \cos(\theta_k) \cdot dt \\ v_k \cdot \sin(\theta_l) \cdot dt \end{bmatrix}}_{u_{3,k-1}} \\ \underbrace{\begin{bmatrix} x_k \\ y_k \end{bmatrix}}_{x_{3,k}} = \underbrace{\begin{bmatrix} x_k \\ y_k \end{bmatrix}}_{x_{3,k}^-} + K_{3(2x2)} \cdot \left(\underbrace{\begin{bmatrix} y_{pom} \end{bmatrix}}_{z_{3,k}} - \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{H_3} \cdot \underbrace{\begin{bmatrix} x_k \\ y_k \end{bmatrix}}_{x_{3,k}^-} \right) \end{array} \right. \quad (3.17b)$$

Równanie (3.18) przedstawia przyjęte wartości wariancji procesu Q oraz pomiaru R .

$$Q_3 = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \quad (3.18)$$

$$R_3 = 200000$$

Obliczanie orientacji robota względem ścian wymaga znajomości zmiany odległości między robotem i ścianą oraz drogi przebytej przez robota. Bezpośrednia informacja z czujników nie nadaje się do tego celu z powodu zbyt dużych wartości szumów pomiarowych. Celem ostatniego modułu jest filtracja odległości od ściany, na potrzeby korekcji orientacji robota. Układy równań (3.19a) oraz (3.19b) przedstawiają obliczanie nowych wartości wektora stanu x_3 , odpowiednio dla każdego kierunku ruchu.

$$\left\{ \begin{array}{l} \overbrace{\begin{bmatrix} x_{s,k} \\ y_{s,k} \end{bmatrix}}^{x_{4,k}^-} = \overbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}^{A_4} \cdot \overbrace{\begin{bmatrix} x_{s,k-1} \\ y_{s,k-1} \end{bmatrix}}^{x_{4,k-1}} \\ \underbrace{\begin{bmatrix} x_{s,k} \\ y_{s,k} \end{bmatrix}}_{x_{4,k}} = \underbrace{\begin{bmatrix} x_{s,k} \\ y_{s,k} \end{bmatrix}}_{x_{4,k}^-} + K_{4(2 \times 2)} \cdot \left(\underbrace{\begin{bmatrix} x_{pom} \end{bmatrix}}_{z_{4,k}} - \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{H_4} \cdot \underbrace{\begin{bmatrix} x_{s,k} \\ y_{s,k} \end{bmatrix}}_{x_{4,k}^-} \right) \end{array} \right. \quad (3.19a)$$

$$\left\{ \begin{array}{l} \overbrace{\begin{bmatrix} x_{s,k} \\ y_{s,k} \end{bmatrix}}^{x_{4,k}^-} = \overbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}^{A_4} \cdot \overbrace{\begin{bmatrix} x_{s,k-1} \\ y_{s,k-1} \end{bmatrix}}^{x_{4,k-1}} \\ \underbrace{\begin{bmatrix} x_{s,k} \\ y_{s,k} \end{bmatrix}}_{x_{4,k}} = \underbrace{\begin{bmatrix} x_{s,k} \\ y_{s,k} \end{bmatrix}}_{x_{4,k}^-} + K_{4(2 \times 2)} \cdot \left(\underbrace{\begin{bmatrix} y_{pom} \end{bmatrix}}_{z_{4,k}} - \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{H_4} \cdot \underbrace{\begin{bmatrix} x_{s,k} \\ y_{s,k} \end{bmatrix}}_{x_{4,k}^-} \right) \end{array} \right. \quad (3.19b)$$

Przyjęte wartości wariancji procesu i pomiaru dla tej części algorytmu zostały zaprezentowane w równaniu (3.20).

$$\begin{array}{l} Q_4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ R_4 = 1000 \end{array} \quad (3.20)$$

Algorytm fuzji danych sensorycznych dla poszczególnych modułów przebiega zgodnie z równaniami (3.4-3.8).

Rozdział 4

Eksperymenty

4.1 Eksperymenty symulacyjne

Do przeprowadzenia pierwszych badań filtru wykorzystano środowisko Matlab. O wyborze zdecydowała łatwość implementacji równań macierzowych oraz wbudowane funkcje do generowania wykresów.

Symulacje rozpoczęto od niezależnej części algorytmu, jaką jest filtracja prędkości liniowej. Na wejście filtru podano sumę wygenerowanej prędkości (dla ruchu po linii prostej) oraz szumu białego o zerowej wartości średniej. Wynik filtracji Kalmana przedstawiono na rysunku 4.1. Parametry filtru zostały dobrane eksperymentalnie. Można zauważyć, że zwiększenie wariancji pomiaru R oraz zmniejszenie wariancji procesu Q , widoczne na rysunku 4.1(d), prowadzi do bardziej gładkich przebiegów na wyjściu filtru, niż ma to miejsce na rysunku 4.1(c).

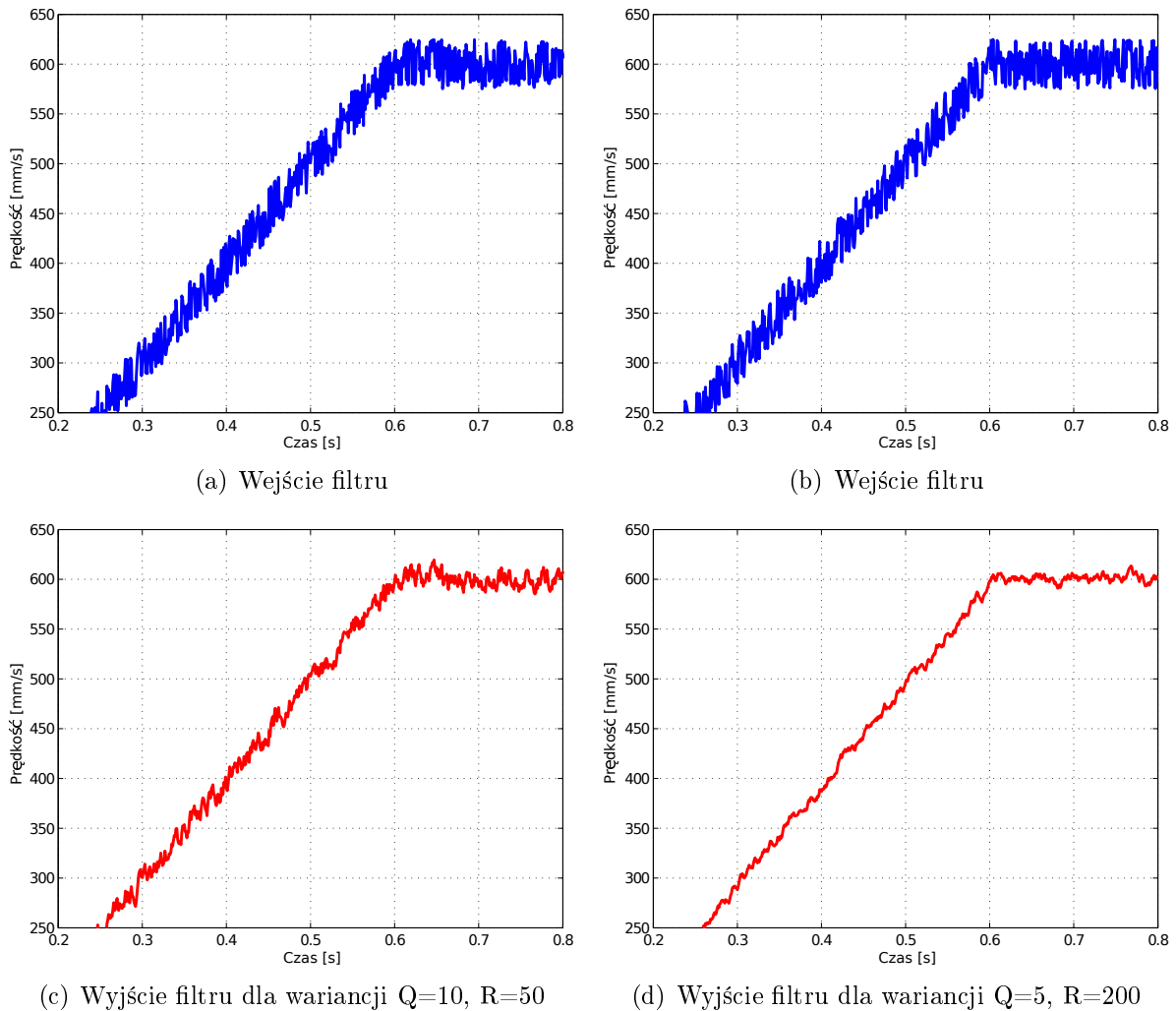
Kolejnym krokiem była symulacja wszystkich negatywnych sytuacji które mogą mieć miejsce podczas przejazdu robota w labiryncie, dla całego algorytmu filtracji. W tym celu wygenerowano zaszumione wartości pomiarowe. Głównym celem filtru w przeprowadzonym eksperymencie było utrzymanie prawidłowych wartości orientacji robota θ oraz jego pozycji y względem środka toru jazdy. Przyjęto następujące wartości pomiarowe:

- prędkość kątowna równa $\omega = 0.25^\circ/s$ (symulacja dryfu żyroskopu),
- odległość od środka toru jazdy w postaci szumu białego o wartościach z przedziału $(-1, 1)mm$ (oznaczająca zaszumione pomiary z dalmierzy optycznych),
- prędkość liniowa symulująca rozpędzanie robota do $60cm/s$ i jego zatrzymanie.

Na wartość wyjściową orientacji θ podano dwa skoki jednostkowe wynoszące 5° (rysunek 4.3, punkt A) oraz -10° (rysunek 4.3, punkt C). Dodatkowo wprowadzono błędną wartość współrzędnej y na początku ruchu robota wynoszącą $20mm$ (rysunek 4.2, punkt A). Skokowe zmiany wartości oznaczają wszelkie nieprzewidywane zmiany wartości pomiarowych względem rzeczywistych, a w szczególności poślizgi kół oraz duże wartości dryfu żyroskopu.

W eksperymencie symulowano korzystanie z czujników odległości tylko w wybranych momentach, tak jak ma to miejsce podczas przejazdu robota micromouse. Pomiary przy pomocy dalmierzy były wykonywane w przedziale $x \in (300, 400) \cup (700, 800)$.

Wyniki symulacji zostały przedstawione na rysunku 4.2 oraz rysunku 4.3. Można zauważyć, że zarówno wartość błędu współrzędnej y oraz orientacji θ znacznie się zmniejsza. W końcowej fazie korzystania z pomiarów odległości błąd pozycji robota względem ścian



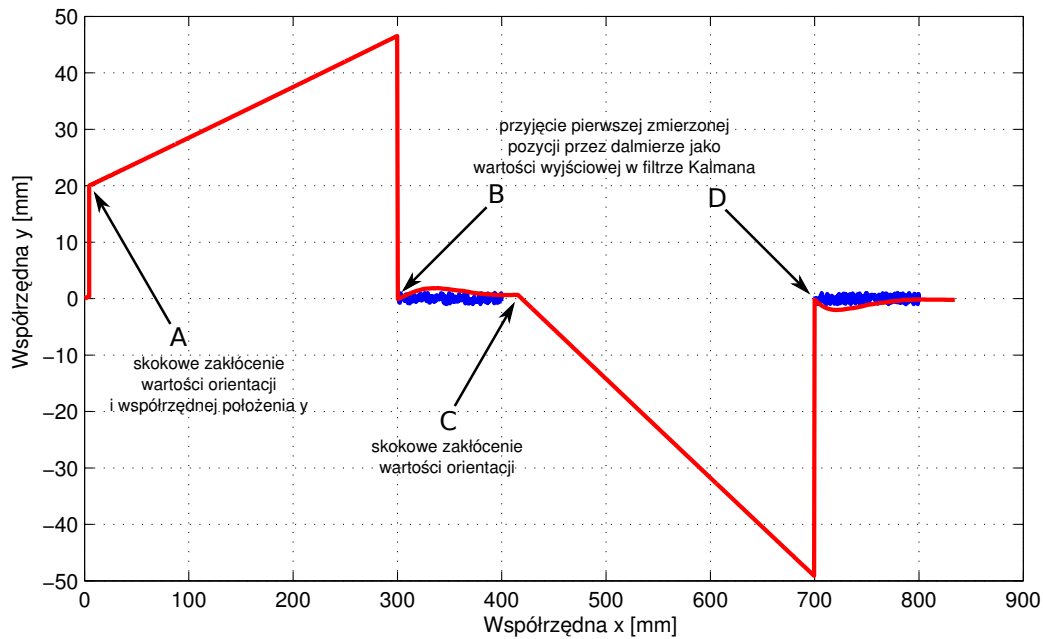
Rysunek 4.1 Symulacja jazdy robota po linii prostej - filtracja prędkości liniowej

zmniejsza się do wartości około 1mm , natomiast błąd orientacji zawiera się w przedziale $\pm 1^\circ$. Podane wartości można ocenić jako bardzo małe, nie mające znaczącego wpływu dla algorytmu sterowania robotem.

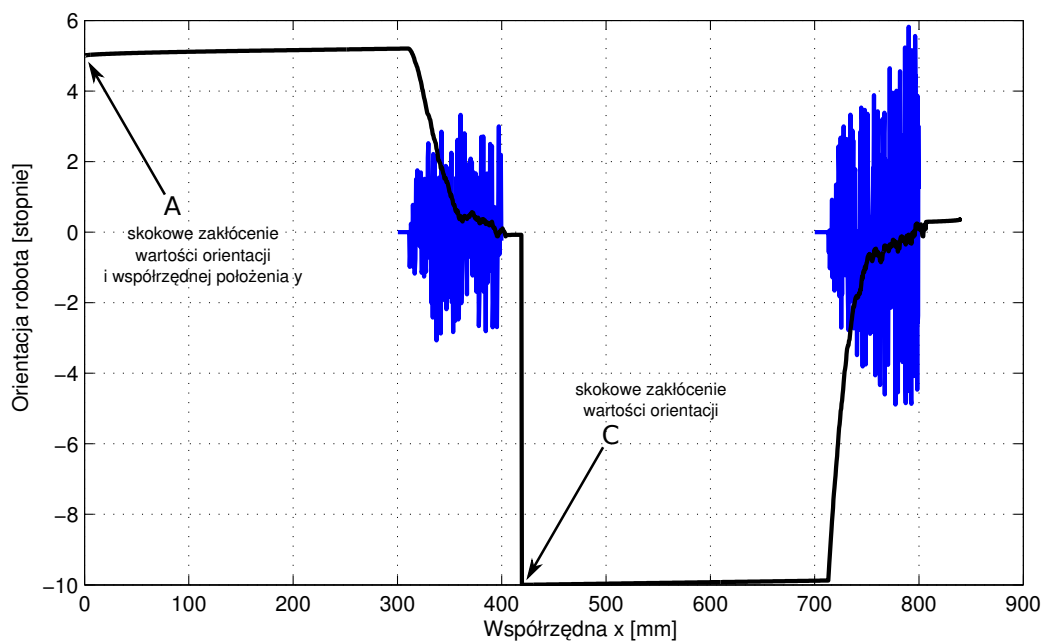
4.2 Eksperymenty na robocie

Na potrzeby przeprowadzenia eksperymentów z udziałem robota, zaimplementowano algorytm fuzji sygnałów w języku C. Podobnie jak symulacje, eksperymenty na robocie rozpoczęto od dostosowania parametrów dla filtracji prędkości liniowej. Wykres prędkości został przedstawiony na rysunku 4.4. Dzięki gładkim przebiegom prędkości liniowej, można znacznie łatwiej dobrać parametry regulatora PD oraz poprawić jego działanie. Zwiększono w tym celu wartość wariancji pomiaru do $R = 500$. Wyniki symulacji zaprezentowano na rysunku 4.5.

Po pomyślnie zakończonych eksperymentach przejazdu robota po linii prostej, kolejnym krokiem były przejazdy robota w labiryncie. Na rysunku 4.6 przedstawiono konfigurację labiryntu przygotowaną na potrzeby testów. Niebieską linią oznaczono pozycję zarejestrowaną przez robota podczas jednego z przejazdów. Celem eksperymentów było zbadanie zachowania robota podczas przeszukiwania labiryntu. Utrudnieniem w poru-



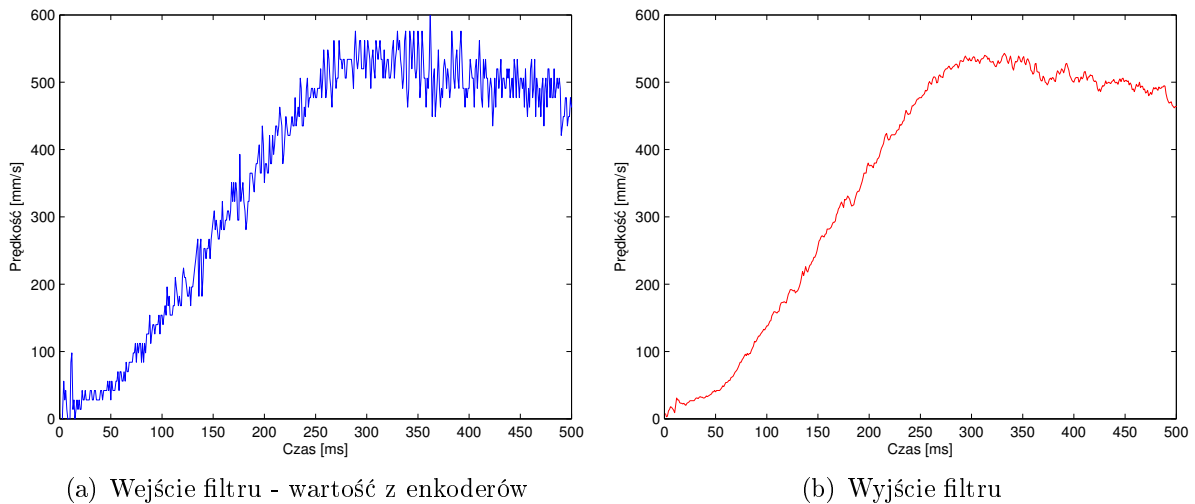
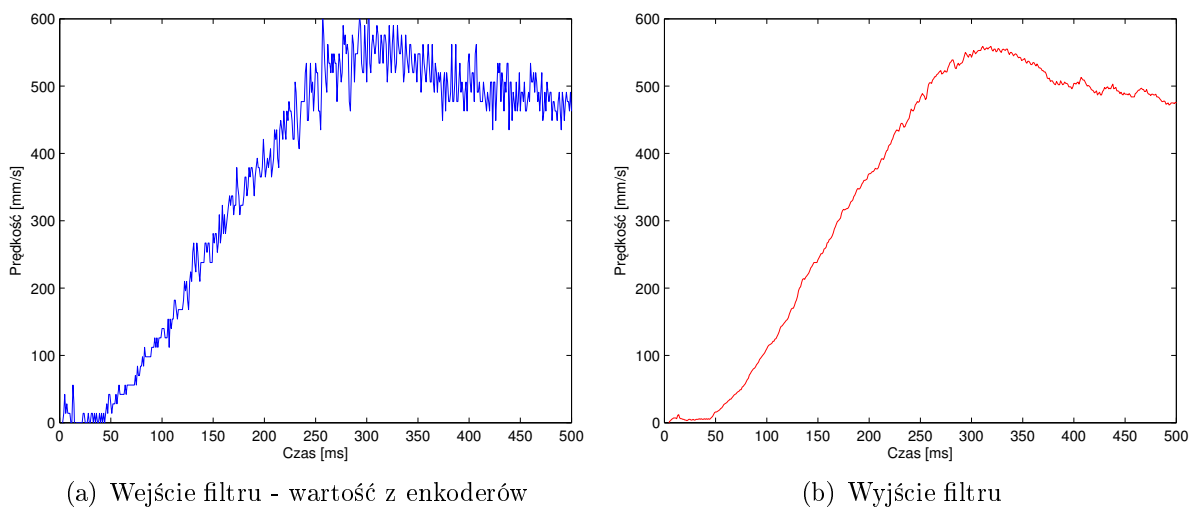
Rysunek 4.2 Symulacja jazdy robota po linii prostej - współrzędne położenia (czerwony - wyjście filtra, niebieski - pomiar z czujników odległości)



Rysunek 4.3 Symulacja jazdy robota po linii prostej - orientacja (czarny - wyjście filtra, niebieski - wejście)

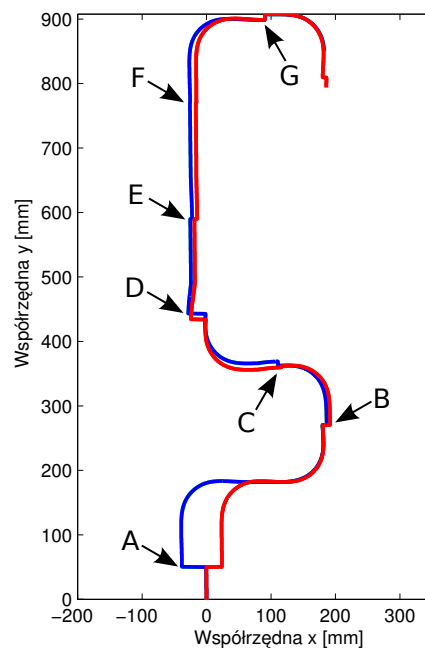
szaniu się robota w nieznanym labiryncie jest konieczność korygowania pozycji robota względem ścian na mniejszym odcinku drogi, niż ma to miejsce podczas szybkich przejazdów. W trakcie przejazdów testowych robot wyrównywał do pojedynczej ściany na odcinku 10mm.

Tor jazdy zarejestrowany przez robota, dla różnych początkowych błędów pozycji,

Rysunek 4.4 Filtracja prędkości liniowej - dla wariacji pomiaru $R=200$ Rysunek 4.5 Filtracja prędkości liniowej - dla wariacji pomiaru $R=500$

został przedstawiony na rysunku 4.7. Przejazdy robota były wykonywane z prędkością liniową równą $v = 50\text{cm/s}$, natomiast skręcanie odbywało się z prędkością kątową wynoszącą $500^\circ/\text{s}$. Dodatkowym negatywnym czynnikiem, wpływającym na jazdę robota, były poślizgi. Związane to było z niskim współczynnikiem przyczepności między kołami a podłożem. Na rysunku 4.7 zaznaczono strzałkami miejsca rozpoczęcia korekcji odległości i orientacji względem ścian labiryntu. Wyniki testów świadczą o bardzo dobrym korygowaniu pozycji robota przez czujniki odległości. Podczas każdego przejazdu robot wracał na właściwą trajektorię. Skokowe zmiany wartości współrzędnych widoczne w miejscach rozpoczęcia wyrównywania do ścian oznaczają przyjęcie pierwszej zmierzonej pozycji przez dalmierze jako wartości wyjściowej w filtrze Kalmana. W wyniku tego czas potrzebny na prawidłową korekcję pozycji robota jest znacznie krótszy.

Kolejny eksperyment polegał na przejeździe robota z wyłączoną, w algorytmie sterowania robotem, korekcją pozycji względem ścian. Zarejestrowaną przez robota ścieżkę zaprezentowano na rysunku 4.8. Rezultatem testu jest przejazd robota w stałej odległości od ścian labiryntu, z prawidłowo utrzymywaną orientacją robota. Możliwość takiego sterowania robotem jest bardzo ważna, ponieważ umożliwia elastyczne dobranie parametrów



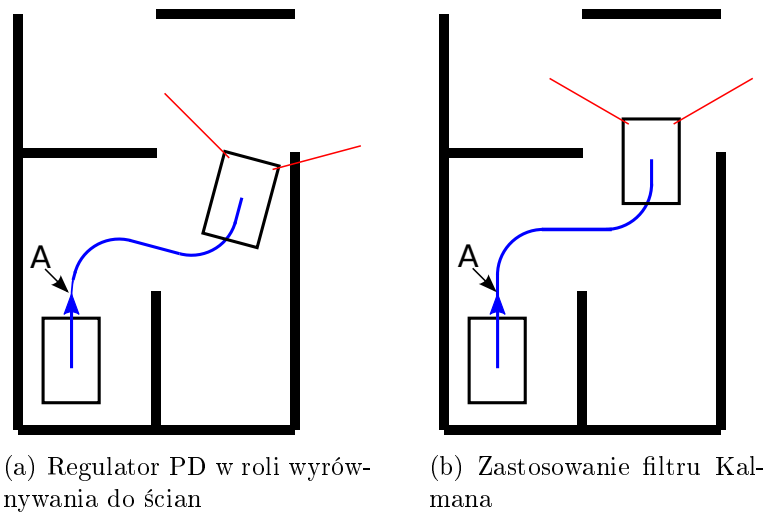
Rysunek 4.8 Tor jazdy robota podczas przeszukiwania labiryntu z prędkością 50cm/s - wyłączone równanie odległości do ścian

stego regulatora PD oraz robota korzystającego z algorytmu fuzji danych sensorycznych zostało przedstawione na rysunku 4.9. W punkcie A rozpoczyna się korekcja pozycji robota względem ścian labiryntu. Wykorzystanie regulatora PD wiąże się ze zmianą orientacji robota pod wpływem nieprawidłowej odległości od ściany. Dla dużego początkowego błędu pozycji efektem działania algorytmu jest znaczne powiększenie się błędu orientacji. Zbyt mały dystans przeznaczony na wyrównanie do ścian wiąże się z nieprawidłową orientacją w końcowej fazie korekcji. Bezpośredni wpływ odczytów z dalmierzy na zmianę orientacji prowadzi więc w wielu sytuacjach do nieprawidłowego wykrycia ścian labiryntu, co zaprezentowano na rysunku 4.9(a). Konsekwencją tego jest błędne wyznaczenie trasy przez algorytm najkrótszej ścieżki oraz utrata prawidłowej pozycji robota w labiryncie. Zastosowanie fuzji danych sensorycznych pozwala rozróżnić błędy orientacji od błędów odległości, wynikiem czego jest prawidłowy przejazd robota (rysunek 4.9(b)). Pokazano w ten sposób przewagę wykorzystania filtra Kalmana w ustalaniu pozycji robota.

Ważną cechą algorytmów stosowanych w robotach micromouse jest ich czas wykonywania. W tym celu przeprowadzono badania czasu obliczania filtra Kalmana. Wyniki zostały przedstawione w tabeli 4.1. Maksymalny czas obliczeń nie przekroczył $30\mu s$. Biorąc pod uwagę fakt, że obliczenia są wykonywane w przerwaniu cyklicznym o okresie $T = 1ms$, algorytm filtracji zajmuje maksymalnie 3% czasu wykonywania całego programu w robocie. Wykorzystywany mikrokontroler oparty jest o architekturę *ARM* z rdzeniem *Cortex-M4*, taktowany jest zegarem 168MHz oraz posiada sprzętową jednostkę do obliczeń zmiennoprzecinkowych *FPU*.

Tabela. 4.1 Czas wykonywania algorytmu

Pomiar odległości z dalmierzy	Włączony	Wyłączony
Średni czas wykonywania	$28\mu s$	$14\mu s$



Rysunek 4.9 Porównanie zachowań robota

Rozdział 5

Podsumowanie

Celem pracy było skonstruowanie algorytmu zapewniającego fuzję danych sensorycznych w robocie klasy micromouse. Użyto w tym celu podstawowej wersji filtru Kalmana, dla systemów liniowych. Dla każdego czujnika wyprowadzono zależność pozycji robota od wartości pomiarowej. Na podstawie filtru powstał algorytm korzystający z wszystkich dostępnych w robocie sensorów, dający w wyniku współrzędne x i y położenia robota oraz jego orientację θ . Do szybszej implementacji algorytmu pomocne okazało się przeprowadzenie symulacji. Dzięki temu możliwe było także dokładne zbadanie wpływu parametrów poszczególnych części algorytmu oraz jakości danych pomiarowych na wartości wyjściowe filtru. Przedmiotem badań była również odporność algorytmu na skokowe zmiany wartości wyjściowych filtru. Po przeprowadzeniu eksperymentów można stwierdzić, że zastosowanie algorytmu fuzyfikacji daje o wiele lepsze wyniki w porównaniu do bezpośredniego korzystania z danych pomiarowych. Ważną cechą zastosowania fuzyji sygnałów w robocie micromouse jest rozdzielenie błędu pozycji i orientacji. Podejście takie daje również większe możliwości sterowania robotem. Zadanie algorytmu sterowania jest w tym wypadku ograniczone do korzystania jedynie z informacji o aktualnej pozycji. Możliwe jest wykorzystanie do tego celu bardziej zaawansowanych algorytmów oraz osobnego rozwijania tej części oprogramowania. Lepsze działanie robota wykorzystującego fuzję danych sensorycznych poparte jest również stabilniejszą jazdą w labiryncie oraz znacznie zmniejszonym zagrożeniem odczytania błędnej konfiguracji labiryntu. W efekcie robot może osiągnąć mniejsze czasy przejazdu, co jest celem konkurencji micromouse.

Lepsze działanie algorytmu filtracji można zapewnić poprzez dodanie dodatkowych wartości pomiarowych. W projekcie tym nie zostały użyte informacje dostępne z czujników skierowanych na przód robota. Na podstawie odczytów wartości odległości z tych sensorów można określić zarówno odległość od ściany jak i orientację robota. Dodatkowo przydatną informacją byłby pomiar miejsca wystąpienia przejścia ściana-puste pole i korekcja przebytego przez robota dystansu. Obecnie algorytm filtracji przewiduje korzystanie z żyroskopu o stałym dryfie wartości mierzonej. Konieczna jest w tym wypadku częsta kalibracja w nieruchomym układzie. Działanie robota można by więc udoskonalić poprzez estymowanie dryfu żyroskopu. Warto byłoby również porównać działanie algorytmu z filtrami przeznaczonymi specjalnie dla systemów nieliniowych, czyli rozszerzonym filtrem Kalmana (EKF) oraz bezśladowym filtrem Kalmana (UKF).

Bibliografia

- [1] Strona domowa konstruktora Ng Beng Kiat,
<https://sites.google.com/site/ngbengkiat/Downhome>
- [2] Strona poświęcona robotom klasy micromouse, <http://micromouseonline.com>
- [3] J. Borenstein, H.R. Everett, L. Feng, D. Wehe, "Mobile Robot Positioning - Sensors and Techniques", Invited paper for the Journal of Robotic Systems, Special Issue on Mobile Robots, 1997
- [4] J. Borenstein, L. Feng, "Gyrodometry: A New Method for Combining Data from Gyros and Odometry in Mobile Robots", Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, 1996
- [5] R. Negenborn, "Robot Localization and Kalman Filters. On finding your position in a noisy world", Utrecht University, 2003
- [6] H. Chung, L. Ojeda, J. Borenstein, "Accurate Mobile Robot Dead-reckoning With a Precision-calibrated Fiber Optic Gyroscope", IEEE Transactions on Robotics and Automation, 2001
- [7] E. Kiriy, M. Buehler, "Three-state Extended Kalman Filter for Mobile Robot Localization", CIM Technical Report, 2002