

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

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

PROJEKT INŻYNIERSKI

System lokalizacji robota klasy Micromouse

Micromouse mobile robot localisation system

AUTOR:
Mateusz Nowakowski

PROWADZĄCY PRACĘ:
dr inż. Robert Muszyński, I-6

OCENA PRACY:

Spis treści

1	Wstęp	1
1.1	Cel projektu i zadania robota	1
2	Charakterystyka robota klasy Micromouse	3
2.1	Charakterystyka robota wykorzystanego do badań	3
2.2	Wymagania stawiane robotowi	3
2.2.1	Określanie orientacji robota	5
2.2.2	Określanie przejechanej drogi	5
2.2.3	Pomiar odległości oraz wykrywanie ścian labiryntu	6
2.3	Sterowanie robotem klasy Micromouse	6
2.3.1	Obrót o zadany kąt	6
2.3.2	Ruch po prostej o zadaną odległość	7
3	Narzędzia	8
3.1	Czujniki	8
3.1.1	Żyroskop L3GD20	8
3.1.2	Enkoder magnetyczny AS5040	8
3.1.3	Czujnik optyczny na bazie fotodiody i fototranzystora	8
3.2	Filtracja i fuzja pomiarów	10
3.2.1	Filtracja pomiarów pochodzących z czujnika odległości	10
3.2.2	Filtracja pomiarów pochodzących z żyroskopu	10
3.2.3	Fuzja sygnałów przy pomiarze orientacji robota	11
3.2.4	Fuzja sygnałów podczas pomiarze przemieszczenia	11
3.3	Metoda lokalizacji	12
3.4	Dwuwarstwowy interfejs algorytmu	12
3.4.1	Lokalizacja pola labiryntu, w którym znajduje się robot dokładnością do pola labiryntu	12
3.4.2	Lokalizacja robota względem środka danej komórki	12
4	Konstrukcja robota	13
4.1	Elementy składowe robota	13
4.2	Oprogramowanie	16
4.2.1	Aktualizacja pomiarów	16
4.2.2	Eliminacja poślizgów - profiler prędkości	16
4.2.3	Kolejka poleceń	16
4.2.4	Algorytm sterowania przemieszczeniem i orientacją robota	16
5	Badania	21
5.1	Sposób dokonywania pomiarów przemieszczenia i orientacji robota	21
5.2	Porównanie jakości sterowania orientacją robota przy wykonywaniu obrotu w miejscu	21
6	Wnioski	25
	Bibliografia	25

Rozdział 1

Wstęp

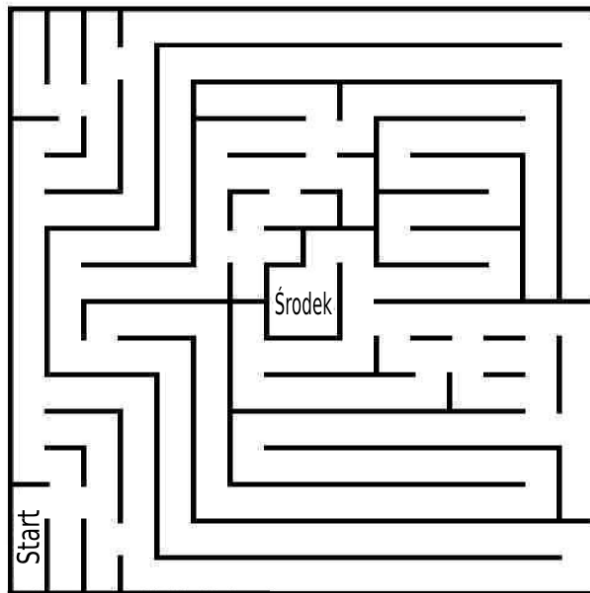
1.1 Cel projektu i zadania robota

Robot klasy Micromouse to robot mobilny przeznaczony do poruszania się w labiryncie. Labirynt jest kwadratem złożonym z 256 pól, z których każde ma wymiary 18x18cm [1]. Celem konkurencji Micromouse jest pokonanie w jak najkrótszym czasie drogi od początku do środka labiryntu. Poprzez środek labiryntu rozumiemy pole utworzone z 4 pól labiryntu z jednym wjazdem, początek labiryntu wybierany jest arbitralnie. Schemat przykładowego labiryntu przedstawiono na rysunku 1.1.

W naturalny sposób rozmiar robota ograniczony jest przez rozmiar pojedynczego pola labiryntu. Waga robota nie jest ograniczana. Robot musi być w pełni autonomiczny i nie może kontaktować się z żadnymi zewnętrznymi urządzeniami.

W konkurencji oceniane są dwa przejazdy. W pierwszym z nich robot jedzie wykorzystując zaimplementowany algorytm bez żadnej wiedzy o labiryncie i stara się dojechać do jego środka. Po dojechaniu tam, w drugim z przejazdów robot może wykorzystywać informacje o labiryncie uzyskane podczas pierwszego przejazdu. Zadaniem robota ponownie jest dojechanie do środka labiryntu w jak najkrótszym czasie.

Celem projektu jest zaprojektowanie, implementacja i zbadanie własności algorytmów pozwalających na określenie pozycji robota klasy Micromouse w labiryncie. Optymalną pod względem minimalizacji prawdopodobieństwa kolizji strategią poruszania się po labiryncie wydaje się być przemieszczanie się robota pomiędzy środkami pól labiryntu. W celu realizacji takiej strategii poruszania zdecydowano się na wyróżnienie dwóch poziomów określania pozycji: lokalnego i globalnego. Pierwszy z wymienionych poziomów ma na celu określanie pozycji i orientacji robota wewnątrz komórki labiryntu, w której aktualnie znajduje się robot oraz akwizycję danych o kształcie labiryntu. Informacje te wykorzystywane będą do takiej nawigacji robotem aby przy odwiedzeniu kolejnych komórek labiryntu robot znajdował się jak najbliżej ich środka. Informacje pochodzące z poziomu globalnego wykorzystywane będą zaś do określenia komórki labiryntu w której znajduje się robot. Dane o aktualnej pozycji robota, kształcie labiryntu oraz położeniu jego środka wykorzystane będą do wyznaczenia ścieżki do środka labiryntu. W celu realizacji projektu niezbędnym jest posiadanie robota klasy Micromouse, z pomocą którego przetestowane oraz porównane będą mogły zostać algorytmy lokalizacji w labiryncie. Ze względu na brak dostępu do gotowej konstrukcji zdecydowano się zaprojektować i wykonać własnego robota od podstaw.



Rysunek 1.1 Labirynt i jego środek

Rozdział 2

Charakterystyka robota klasy Micromouse

Dokonując przeglądu istniejących robotów klasy Micromouse, spotkać można się z różnorodnością wykorzystywanych konstrukcji. Najczęściej spotykanymi są konstrukcje klasy (2,0) oraz klasy (4,0) [2].

Na rysunkach 2.2 oraz 2.1 przedstawiono przykładowe roboty klasy (2,0) oraz (4,0).

Konstrukcje klasy (2,0) charakteryzują się możliwością wykonywania obrotu w miejscu przy braku występowania poślizgów kół. Cecha ta jest niezwykle przydatna przy sterowaniu robotem, gdyż zapobiega ona błędom w szacowaniu pozycji robota generowanym przez przesuwanie się robota podczas obrotu. Co więcej robot o takiej kinematyce pozwala na poruszanie się w labiryncie przy wykorzystaniu dwóch elementarnych ruchów:

- obrotu robota o zadany kąt,
- ruchu po linii prostej na zadaną odległość.

Zaletą robotów klasy (4,0) jest mniejsza wrażliwość na zakłócenia wynikające z różnic w budowie silników. Wykorzystywane silniki zawsze różnią się między sobą, co utrudnia sterowanie robotem. W wyniku tego, nawet gdy sygnały sterujące silnikami są identyczne, to robot może nie poruszać się idealnie prosto. Konstrukcja czteroślowa skuteczniej niż konstrukcja klasy (2,0) wymusza na robocie lepszą kompensację tego zjawiska.

Sterowanie robotem klasy Micromouse wymaga wiedzy o aktualnym położeniu i orientacji robota. Wiedza ta jest niezbędna do nawigacji w labiryncie. Pozwala ona na odpowiednie poruszanie się. Szczególnie istotna jest właściwa kontrola orientacji robota, aby ją umożliwić konieczne jest wyposażenie robota w odpowiednio szybkie, dokładne oraz odporne na zakłócenia czujniki. W celu określenia swojej pozycji w labiryncie możliwe jest zastosowanie dwóch strategii:

- lokalnej – kiedy to aktualna pozycja określana jest względem pozycji początkowej/ poprzedniej. Strategia ta wymaga wyposażenia robota w czujniki umożliwiające pomiar przemieszczenia i zmiany orientacji względem pozycji początkowej.
- globalnej – kiedy to wykonywany jest pomiar, który w sposób bezpośredni umożliwia określenie pozycji robota i środka labiryntu. Przykładem takiego sensora może być kamera umieszczona odpowiednio wysoko nad robotem, która swoim zasięgiem obejmie labirynt i robot.

Jak już wspomniano roboty klasy Micromouse mają za zadanie dwukrotnie dojechać do środka labiryntu w jak najkrótszym czasie. Robot za pomocą sensorów może mapować i zapamiętywać fragmenty labiryntu, które odwiedził. Wykorzystanie tych informacji może być wykorzystane w kolejnym przejeździe do wyznaczenia ścieżki prowadzącej bezpośrednio do środka labiryntu, zatem zadanie dojechania do środka labiryntu zostaje zamienione w zadanie podążania po zadanej ścieżce. Stąd w zależności od przejazdu można wyróżnić dwa zadania, jakim musi sprostać konstrukcja klasy Micromouse:

- lokalizację robota, akwizycję danych o labiryncie oraz wyznaczanie ścieżki do środka labiryntu,
- podążanie po wyznaczonej ścieżce.

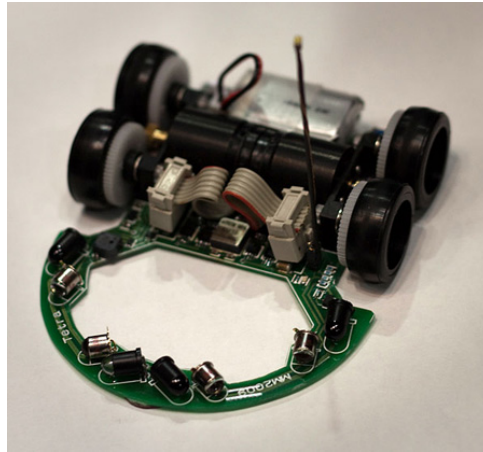
2.1 Charakterystyka robota wykorzystanego do badań

Po przeanalizowaniu dostępnych rozwiązań zdecydowano się na budowę robota klasy (2,0) przystosowanego do realizacji dwóch elementarnych akcji: obrotu o zadany kąt oraz poruszania się po linii prostej.

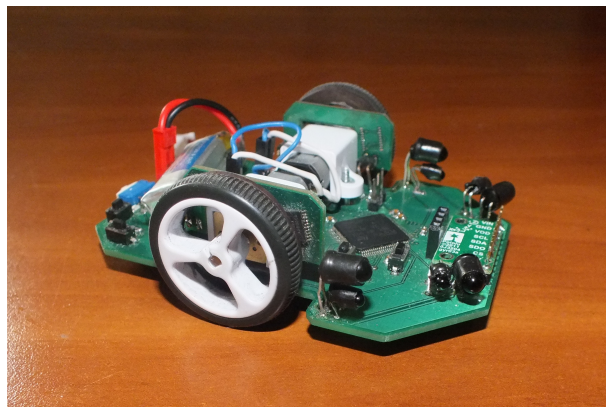
Na rysunku 2.3 przedstawiono funkcjonalność stworzonego robota.

2.2 Wymagania stawiane robotowi

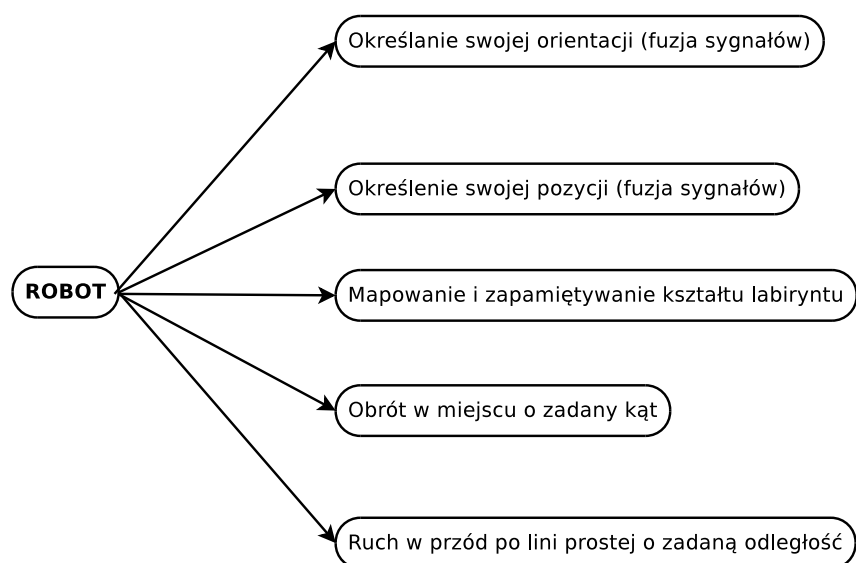
Robot musi być w stanie określić swoją pozycję w labiryncie, planować ścieżkę do jego środka oraz podążać po niej odpowiednio filtrując odczyty z sensorów. Aby robot miał szansę zadziałać musi sprostać zadaniu nawigacji w labiryncie.



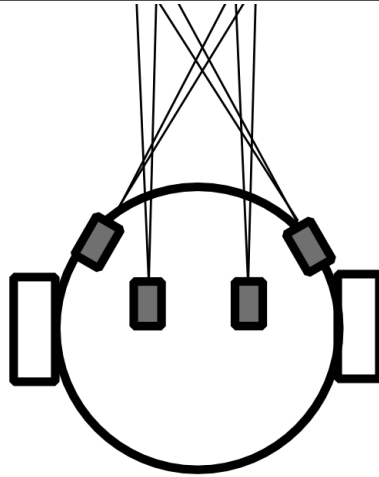
Rysunek 2.1 Micromouse klasy (4,0) Tetra [3]



Rysunek 2.2 Micromouse klasy (2,0) Vampire 2 [4]



Rysunek 2.3 Funkcjonalność stworzonego robota



Rysunek 2.4 Schemat rozłożenia czujników w robocie

2.2.1 Określanie orientacji robota

W celu umożliwienia kontroli orientacji robota konieczne jest wyposażenie go w odpowiednio szybkie i odporne na zakłócenia czujniki. Realizacja zadania określania orientacji robota może być zrealizowana za pomocą następujących czujników:

- enkoderów,
- żyroskopu,
- czujników odległości,

Poniżej pokrótce scharakteryzowano każdy z wymienione czujników.

Żyroskop

Żyroskop jest czujnikiem służącym do pomiaru prędkości kątowej. Poprzez całkowanie tej prędkości możliwe jest wyznaczenie zmiany orientacji robota. Jest to czujnik bardzo dokładny, jednakże wykorzystując go, należy zdawać sobie sprawę z błędów, jakimi obciążone są pomiary, a wynikających w głównej mierze ze zjawiska zwanego dryfem żyroskopu.

Enkodery

Alternatywą dla żyroskopu jest wykorzystanie enkoderów. Poprzez znajomość rozstawu kół robota oraz ich rozmiaru kół możliwe jest obliczenie zmiany orientacji robota na podstawie dróg przejechanych przez obydwa koła. Rozwiązanie takie nie jest jednak dokładne, gdyż nie uwzględnia zakłóceń wynikających np. z poślizgów kół.

Czujniki odległości

Do zadawania i utrzymania orientacji można wykorzystać również czujniki odległości. W szczególnym przypadku, gdy po bokach robota znajdują się ściany labiryntu możliwa jest korekcja orientacji robota poprzez dokonanie pomiaru jego odległości od tychże ścian i umiejscowienie robota w równej odległości od nich. Ustawienie takie gwarantuje, że robot będzie obrócony o 90° względem ściany labiryntu. Przykład rozmieszczenia czujników dla którego opisana zależność występuje przedstawiono na rysunku 2.4.

Pierwsze dwa z omówionych czujników charakteryzują się dużą dokładnością, jednakże realizują one pomiary zadanych wartości w sposób pośredni, w którym wraz z czasem działania robota błąd pomiarowy może narastać. Ostatni z omawianych czujników jest obciążony największym błędem i nie nadaje się do stałej kontroli orientacji robota poza jednym szczególnym przypadkiem. Tym niemniej w przeciwieństwie do błędów żyroskopu i enkoderów. Jest to błąd stały, nie narastający z upływem czasu.

2.2.2 Określanie przejechanej drogi

Aby robot mógł prawidłowo estymować swoje położenie w labiryncie występuje konieczność wyposażenia go w odpowiednie czujniki. Dokonując przeglądu istniejących rozwiązań najczęściej można spotkać się z zastosowaniem następujących czujników:

- enkoderów,
- czujnik optyczny.

Zastosowanie enkoderów pozwala na osiągnięcie dużej dokładności. Znając promień kół robota możliwe jest wyznaczenie jego przemieszczenia. Metoda ta ma jedną podstawową wadę, gdyż pomiar jest obciążony błędami wynikającymi z możliwości poślizgu kół.

W celu usunięcia błędów wynikających z braku możliwości detekcji poślizgów kół robota możliwe jest wykorzystanie czujników odległości. W szczególnym przypadku, gdy przed robotem znajduje się ściana labiryntu robot bazując na odczytach z nich pochodzących może wyregulować swoją pozycję w taki sposób, aby znalazł się on na środku pola labiryntu przed, ścianą, tym samym niwelując narastający błąd pomiaru przemieszczenia.

Alternatywą może być zastosowanie czujnika przemieszczenia np. z myszy optycznej. Pomimo że pomiar taki jest dokładny oraz jest pomiarem bezpośrednim to rozwiązanie takie ma bardzo istotną wadę. Czujnik taki musi znajdować się w stałej odległości od podłoża, co jest trudne do zagwarantowania.

2.2.3 Pomiar odległości oraz wykrywanie ścian labiryntu

Umiejętność oceny przez robota jak daleko znajduje się od ściany labiryntu jest niezbędna do unikania kolizji. Ze względu na to, że robot korzysta z enkoderów, jego zderzenie ze ścianą oraz wynikające z tego ewentualne poślizgi kół mogą uniemożliwić właściwą nawigację w labiryncie. Stąd w celu wykrywania obecności ścian labiryntu oraz pomiaru odległości od nich, należało wyposażyć robota w odpowiednie czujniki.

2.3 Sterowanie robotem klasy Micromouse

Celem robotów startujących w konkurencji Micromouse jest dojechanie do środka labiryntu. W celu nawigacji w labiryncie zdecydowano się na wykorzystanie dwóch akcji:

- obrotu o zadany kąt,
- ruchu po prostej na zadaną odległość.

Jak wcześniej wspomniano, badany robot należy do klasy (2,0) i posiada dwa współosiowo umieszczone koła. Z tego faktu wynika, że wymienione wyżej akcje są wystarczające do tego, aby nawigować w labiryncie. W celu optymalizacji czasu przejazdu labiryntu, możliwe jest wykorzystanie dodatkowych akcji takich jak jazda po łuku, która pozwala na płynniejsze pokonywanie zakrętów i nie wiąże się z koniecznością zatrzymywania robota podczas pokonywania zakrętów.

2.3.1 Obrót o zadany kąt

W celu pomiaru orientacji oraz wymuszeniu na robocie obrotu o zadany kąt możliwe jest wykorzystanie kilku różnych czujników. Poniżej opisano sposoby pomiaru orientacji robota w sposób pośredni za pomocą enkoderów oraz żyroskopu.

Enkodery

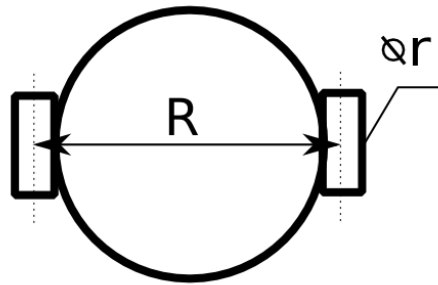
Aby robot mógł obracać się w miejscu bazując na informacjach o położeniu kątowym kół pochodzącej z enkoderów konieczne jest wymuszenie spełnienia dwóch warunków. Pierwszym z nich jest zapewnienie, aby w czasie trwania ruchu koła miały taką samą prędkość, a drugim zadbanie o to, żeby koła kręciły się w przeciwnych kierunkach. Aby zmienić orientację robota klasy (2,0) o zadany kąt można wykorzystać odczyty z enkoderów. W tym celu można skorzystać z zależności określającej jaką drogę muszą pokonać koła robota aby obrócić się on o zadany kąt, gdzie:

$$f(x) = \frac{k \cdot x \cdot 2\pi R}{360 \cdot 2\pi r},$$

- $f(x)$ – liczba impulsów enkodera o jaką mają obrócić się koła (w przeciwnych kierunkach),
- x – zadany kąt [°],
- k – ilość impulsów enkodera na jeden obrót koła,
- R – rozstaw kół robota,
- r – promień koła.

Na rysunku 2.5 przedstawiono schemat robot.

Sterowanie kątem obrotu na bazie sygnałów z enkoderów nie zawsze jest dokładne. Metoda ta nie uwzględnia uginania się kół oraz możliwych poślizgów. O ile zazwyczaj ugięcie kół możemy pominąć, to problem poślizgu może wprowadzać duże błędy w pomiarze orientacji robota.



Rysunek 2.5 Schemat robota

Żyroskop

Alternatywą do sterowania kątem obrotu z wykorzystaniem enkoderów może być sterowanie bazujące na pomiarach pochodzących z żyroskopu. Wówczas w celu określania aktualnej pozycji korzystamy ze wzoru,

$$x = \int_{t_0}^{t_1} \omega(t) dt,$$

gdzie:

- x – aktualne położenie kątowe,
- $\omega(t)$ – odczytana z żyroskopu prędkość kątowa.

Pomiary wykonywane przy pomocy żyroskopu są bardzo dokładne i pozwalają na pomiar nawet bardzo małych zmian orientacji rzędu 0.01 stopnia. Jednakże ze względu na naturę działania żyroskopu oraz fakt, że jest czujnikiem inercyjnym, pomiar wykonywany przy jego użyciu jest obciążony błędem wynikającym z tzw. dryfu żyroskopu. Ze względu na to żyroskop podczas pracy powinien być kalibrowany tak, aby zminimalizować oddziaływanie dryfu na pomiary. Do tego celu można wykorzystać na przykład czujniki odległości.

2.3.2 Ruch po prostej o zadanej odległość

Jest to podstawowa akcja robota, gdyż pozwala na przemieszczanie się robota pomiędzy polami labiryntu. W celu realizacji ruchu wykorzystano enkodery, które pozwoliły na pomiar drogi przejechanej przez każde z kół. Do wyliczenia drogi, jaką pokonało każde koło wykorzystano wzór,

$$f(x) = \frac{2\pi r \cdot x}{k},$$

gdzie:

- k – rozdzielczość enkodera (ilość impulsów na obrót),
- r – promień koła,
- x – liczba impulsów odczytywanych z enkodera podczas przejazdu.

Problemem może okazać się kontrola orientacji robota podczas jazdy do przodu. Poprzez sterowanie wypełnieniem sygnału przekazywanego na silniki nie można zapomnieć o tym, że wykorzystywane silniki mogą się między sobą różnić np. tarcie statycznym przekładni. Skutkuje to tym, że po zadaniu takiego samego wypełniania sygnału sterującego prędkością obrotową silników, koła mogą przebyć różne drogi.

Co więcej masa robota przekłada się na jego inercję. Wraz ze zwiększaniem się masy robota, zwiększa się minimalny moment obrotowy silników, przy którym robot jest w stanie ruszyć się z miejsca. W tym przypadku również przy zadaniu takiego samego sygnału sterującego silnikami mogą one przebyć różne drogi co będzie spowodowane generowaniem różnych momentów obrotowych przez silniki dla takiego samego sygnału sterującego. Przytoczone problemy można wyeliminować poprzez kontrolę drogi przejechanej przez obydwa koła i odpowiednią korekcję sygnałów sterujących pracą silników.

Rozdział 3

Narzędzia

3.1 Czujniki

W robocie wykorzystano następujące czujniki, które pozwoliły na nawigację robota w labiryncie:

3.1.1 Żyroskop L3GD20

Żyroskop L3GD20 jest żyroskopem trzyosiowym [5], jednakże w realizowanej aplikacji pomiary dokonywane są tylko w jednej osi, ze względu na fakt, że robot nie ma możliwości poruszania się wokół dwóch pozostałych. Żyroskop można skonfigurować tak, aby wybrać zakres pomiarów dopasowany do potrzeb użytkownika. Po licznych testach zdecydowano się na zakres 2000 dps ze względu na znaczne prędkości kątowe, jakie osiągał robot. L3GD20 charakteryzuje się małym dryfem, co pozwala robotowi na dłuższą niż w przypadku gorszych jakościowo żyroskopów pracę bez konieczności kompensacji, jednak tak jak w każdym żyroskopie jego wpływ jest w końcu odczuwalny. Przy wyborze żyroskopu kierowano się tym, aby pomiar był realizowany poprzez sygnał cyfrowy, co minimalizuje występowanie zakłóceń związanych z pomiarem sygnału analogowego i zakłócaniu go poprzez inne elementy znajdujące się na płytce robota. Wiązało się to również z brakiem konieczności posiadania stabilnego napięcia referencyjnego na płytce. Dodatkową zaletą był wykorzystywany do komunikacji żyroskopem interfejs SPI jest bardzo szybki i praktycznie nie wprowadza on opóźnień do pętli sterowań. Na rysunku 3.1 przedstawiono omawiany układ scalony.

3.1.2 Enkoder magnetyczny AS5040

Enkoder AS5040 wykorzystano do pomiaru drogi przejechanej przez robota oraz do kontroli trajektorii po jakiej porusza się robot. Wspomniany enkoder dysponuje trzema różnymi zakresami pomiarowymi [6]. W przypadku robota klasy Micromouse wystarcza najmniejszy z możliwych zakresów tj. 512 impulsów na obrót co pozwala na dokonywanie wystarczająco dokładny pomiarów do sterowania robotem. AS5040 jest enkoderem magnetycznym i do jego poprawnego działania wymagane jest umiejscowienie magnesu na kole robota lub wale silnika. Wybrany żyroskop jest niewielki i odporny na zakłócenia. Świetnie radzi sobie nawet, gdy magnes nie jest zamontowany idealnie prostopadle do niego. Do komunikacji z enkoderem spośród trzech dostępnych opcji interfejsu SPI, I2C oraz wyjścia kwadraturowego wybrano ostatnią opcję, a to ze względu na sprzętową obsługę tego typu sygnału przez zastosowany w robocie mikrokontroler. Omawiany enkoder przedstawiono na rysunku 3.2.

3.1.3 Czujnik optyczny na bazie fotodiody i fototranzystora

Do poprawnej nawigacji w labiryncie idealne byłyby czujniki odległości o zakresie pomiarowym 0-288cm wynikającym z rozmiaru labiryntu. Wystarczającą alternatywą byłyby czujniki o zakresie 0-36cm, które pozwoliłyby na wykrywanie ścian labiryntu znajdujących się w odległości 3 pól od miejsca, w którym aktualnie znajduje się robot. Wykrycie ściany labiryntu z takiego dystansu pozwala na rozpoczęcie odpowiednio wcześniej redukcji prędkości robota w celu wyhamowania przed ścianą. Po dokonaniu przeglądu dostępnych na rynku czujników zdecydowano się na zaprojektowanie własnego czujnika obciowego. W tym celu wykorzystano diodę podczerwieni i fototranzystor. Zasada działania czujnika opiera się na wykonaniu pomiaru ilości światła wyemitowanego przez diodę IR, które po odbiciu od ścian labiryntu oświetla fototranzystor. Napięcie, które pojawia się na wyjściu fototranzystora jest odwrotnie proporcjonalne do odległości czujnika od ściany. Dioda IR jest załączana okresowo, co umożliwia pobudzenie jej większym prądem niż w przypadku pracy ciągłej. Większy prąd skutkuje tym, że dioda świeci mocniej, co pozwala na wykonywanie pomiaru z większej odległości. Do konstrukcji czujnika wybrano diodę IR TSAL6100 oraz fototranzystor TEFT4300 ze względu na przedział długości fal, jakie generuje dioda, idealnie wpasowujących się charakterystykę fototranzystora [7, 8]. Na rysunkach 3.3 oraz 3.4 przedstawiono charakterystyki tych elementów:

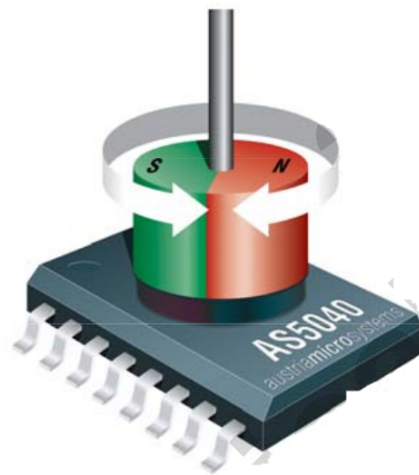
- diody IR TSAL6100,
- fototranzystora TEFT4300.

Wybrany zestaw pozwala na pomiar odległości robota od ścian w zakresie 2-35 cm. Jest to wynik wystarczający, który pozwala robotowi na odpowiednio szybko wykrycie obecności ściany labiryntu i umożliwić mu wyhamowanie przed nią. Czujnik posiada martwą strefę, co oznacza, że dla odległości czujnika od ściany labiryntu mniejszej niż wspomniane 2 cm nie ma możliwości określenia, jak daleko w rzeczywistości od ściany labiryntu znajduje się robot. Defekt ten można kompensować poprzez zapamiętywanie historii kilkunastu ostatnich pomiarów i rozstrzygnięcie na ich podstawie czy generowane na wyjściu czujnika maksymalne napięcie jest spowodowane odległością robota od ściany mniejszą niż 2cm (ściana labiryntu w martwej strefie), czy też odległością większą niż 35cm (ściana labiryntu poza zakresem pomiarowym). Dla czujników mierzących odległość robota od ściany na wprost niego martwa strefa czujnika nie jest problemem. Robot docelowo ma

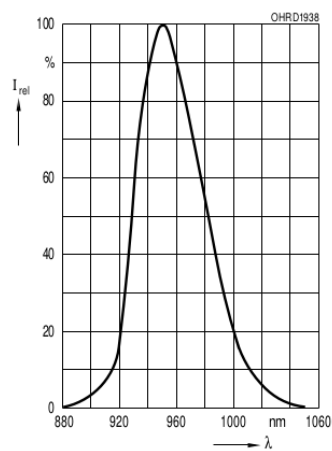


LGA-16 (4x4x1 mm)

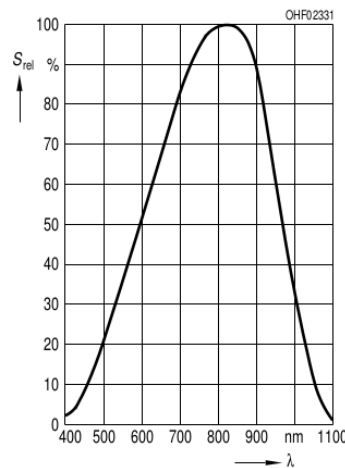
Rysunek 3.1 Żyroskop L3GD20



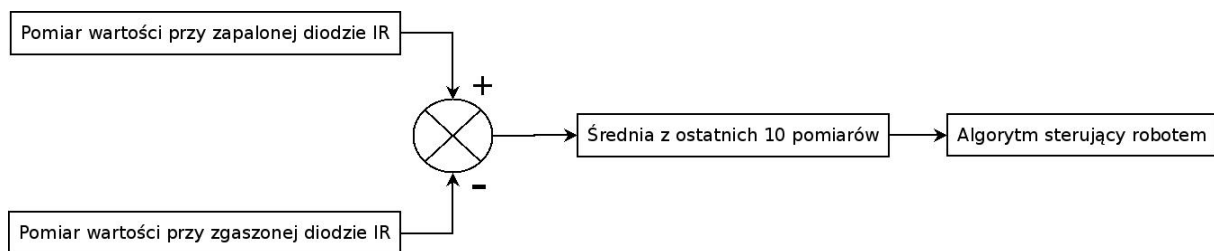
Rysunek 3.2 Enkoder AS5040



Rysunek 3.3 Charakterystyka diody IR TSAL6100



Rysunek 3.4 Charakterystyka fototranzystora TEFT4300



Rysunek 3.5 Filtracja pomiarów pochodzących z czujników odległości

się poruszać między śródkami pól labiryntu, więc sytuacja, gdy robot znajduje się w odległości mniejszej od 2 cm od ściany nie powinna mieć miejsca. Jednakże problemem mogą być czujniki służące do wykrywania odległości robota od ścian znajdujących się po jego bokach. Dlatego zdecydowano się na rozmieszczenie czujników jak na rysunku 2.4 co całkowicie wyeliminowało wpływ martwej strefy na pomiary.

3.2 Filtracja i fuzja pomiarów

Robot dysponuje różnego rodzaju czujnikami. Każdy pomiar obarczony jest zakłóceniami wynikającymi z dalekiego od ideału działania przyrządów pomiarowych, dlatego też wymagane jest przygotowanie odpowiednich algorytmów pozwalających na filtrację takich pomiarów, dodatkowo pojawia się konieczność przeprowadzenia fuzji pomiarów, ponieważ sensory pozwalają na pomiar jednego parametru robota na kilka różnych sposobów. W tym celu rozważono możliwe rozwiązania i zdecydowano się na wybór optymalnych.

3.2.1 Filtracja pomiarów pochodzących z czujnika odległości

Ze względu na, to że pomiary pochodzące z czujników odległości są mocno zakłócone zdecydowano się na ich filtrację. Jest ona realizowana na następujące sposoby:

- pomiar różnicowy – w celu określenia odległości przeszkody od robota wykonywany jest pomiar napięcia na wyjściu fototranzystora przy zapalanej i zgaszonej diodzie. Na wyjściu algorytmu wystawiana jest różnica tych wartości – taka filtracja pozwala na zmniejszenie wpływu oświetlenia labiryntu na pomiary,
- uśrednianie pomiarów – dane otrzymane z pomiaru różnicowego są uśredniane w celu eliminacji zakłóceń losowych.

Na rysunku 3.5 przedstawiono schemat algorytmu realizujący pomiar różnicowy.

3.2.2 Filtracja pomiarów pochodzących z żyroskopu

W celu filtracji danych z żyroskopu zdecydowano się na pomijanie kilku najmniej znaczących bitów wartości mierzonej. Zdecydowano się na pominięcie 3 z 12 bitów, co pozwoliło na 100% eliminację zakłóceń generowanych przez dryf żyroskopu, gdy robot znajdował się w stanie spoczynku.

Zastosowana metoda filtracji nie przyczyniła się do całkowitego usunięcia dryfu żyroskopu, a jedynie zmniejszyła jego wpływ na pomiary kosztem dokładności. Rozwiązanie to pozwoliło na zmniejszenie częstotliwości dokonywania korekcji pomiarów orientacji robota.

3.2.3 Fuzja sygnałów przy pomiarze orientacji robota

Do pomiaru orientacji możliwe jest wykorzystanie zarówno enkoderów jak i żyroskopu, w które wyposażony jest robot.

Żyroskop jest inercyjnym sensorem, który idealnie nadaje się do pomiaru orientacji robota, poprzez całkowanie prędkości kątowej wokół osi Z. Jednakże jednocześnie całkowany jest błąd tzw. dryf. Skutkuje to tym, że po pewnym czasie orientacja mierzona może znacznie różnić się od rzeczywistej orientacji robota.

Alternatywą dla wykorzystania żyroskopu są wspomniane enkodery. Do sterowania orientacją robota można wykorzystać zależność opisaną w sekcji 2.3.1. Jednakże jak wcześniej wspomniano pomiary bazujące na odczytach z enkoderów są wrażliwe na zjawisko poślizgów kół robot i podobnie jak w przypadku żyroskopu błąd z tego wynikający narasta z czasem.

Obydwa z rozważanych sposobów pomiaru orientacji robota niosą ze sobą problem kompensacji trudnych do odfiltrowania zakłóceń, których wpływ na pomiar orientacji narasta wraz z upływem czasu. W celu kompensacji powstałych błędów dokonano przeglądu możliwych rozwiązań i wybrano trzy najbardziej popularne metody filtracji:

- wykorzystując filtr Madgwicka,
- wykorzystując filtr Kalmana,
- wykorzystując odczyty z czujników odległości.

Filtr Madgwicka

Filtr Madgwicka jest stosunkowo nowym filtrem, który przy wykorzystaniu żyroskopu, akcelerometru i magnetometru pozwala na określenie orientacji robota w przestrzeni [9]. Bazuje on głównie na żyroskopie, a dodatkowe czujniki wykorzystywane są do kompensacji dryfu żyroskopu. Ze względu na to, że filtr ten dokonuje fuzji pomiarów z 3 czujników idealnie nadawałby się do wykorzystania w przygotowanej konstrukcji, jednakże nie została ona wyposażona w magnetometr co dyskwalifikuje możliwość wykorzystania tego filtru.

Filtr Kalmana

Można spotkać się z opiniami, że do kompensacji opisanych zakłóceń nadaje się wykorzystanie filtru Kalmana, co nie do końca jest prawdą. Filtr ten może być wykorzystywany do wygładzania charakterystyk pochodzących z pomiarów [10,11] lecz sam w sobie nie potrafi poradzić sobie np. z dryfem żyroskopu. Filtr Kalmana można wykorzystać dla układu pomiarowego, który już taką fuzję realizuje. Poskutkowałoby to uzyskaniem stabilnych pomiarów np. w stanie spoczynku gdy robot się nie porusza przy dobrze nastrojonym filtrze odczyty również byłyby stałe. Poprzez filtrowanie pomiarów np. z samego żyroskopu uzyskano by jedynie bardziej gładką charakterystykę pomiarów z ograniczoną ilością szumów pomiarowych.

Kompensacja dryfu za pomocą czujników odległości

Aby skompensować zakłócenia spowodowane dryfem żyroskopu i poślizgami kół w przypadku enkoderów po rozważeniu innych opcji zdecydowano się na wykorzystanie do tego celu czujników odległości.

Czujniki te w sposób bezpośredni pozwalają na dokonywanie pomiarów, pomimo że są one obciążone dużym błędem jest to błąd stały, który nie zwiększa się wraz z upływem czasu. Ta własność pozwala na dokonywanie okresowej korekcji odczytów pochodzących z innych czujników, obciążonych błędem, który rośnie w czasie. Ze względu na to że podczas startu robot umieszczony jest w polu z którego istnieje tylko jedna droga wyjazdu możliwe jest wykonanie pomiaru wartości zwracanych przez czujniki odległości od ścianek znajdujących się po bokach robota. Wykonywana jest seria pomiarów próbnych z czujników, a następnie odrzucone zostają skrajne pomiary z próby – z pozostałych zostaje wyznaczona wartość średnia. Pomiary uzyskane w ten sposób są pomiarami referencyjnymi. Ze względu na to że rozmiary pola labiryntu są takie same, robot może starać się kalibrować swoją orientację przy każdej okazji, gdy po jego bokach znajdują się ściany labiryntu.

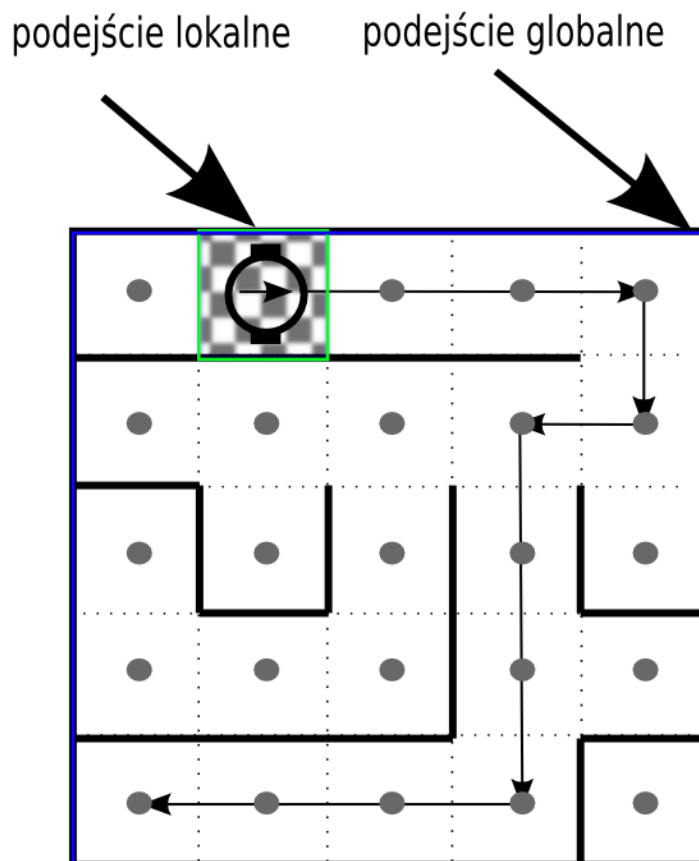
Zdecydowano się na implementację i porównanie jakości pomiarów dla fuzji pomiarów pochodzących z:

- enkoderów i czujników odległości,
- żyroskopu i czujników odległości.

Enkodery i żyroskop wykorzystane będą do określania orientacji robota. zaś informacje pochodzące z czujników odległości wykorzystane zostaną do okresowej kompensacji błędów. Błędy te mogą być spowodowane przez dryf w przypadku zastosowania żyroskopu lub poślizgi kół, gdy stosowane będą enkodery.

3.2.4 Fuzja sygnałów podczas pomiarze przemieszczenia

Do pomiaru przemieszczenia wykorzystać można enkodery. W tym przypadku również pojawia się problem kompensacji błędów wynikających z poślizgów kół robota. Kompensację zdecydowano się przeprowadzić poprzez wykorzystanie fuzji sygnałów pochodzących z enkoderów i czujników odległości. W trakcie przemieszczania się robota pozycja mierzona będzie na podstawie informacji pochodzących z enkoderów, czujniki odległości zostaną natomiast wykorzystane do kompensacji błędów wynikających z poślizgu kół robota. Kompensacja taka może być przeprowadzana każdorazowo, gdy przed robotem w zasięgu czujników odległości znajdują się ściany labiryntu. Robot będzie porównywał szacowaną na podstawie znajomości swojej pozycji i rozmiaru pól labiryntu odległość do ściany labiryntu, a następnie porównywał ją z odległością mierzona przy pomocy czujników optycznych.



Rysunek 3.6 Metody lokalizacji robota

3.3 Metoda lokalizacji

3.4 Dwuwarstwowy interfejs algorytmu

Aby móc poprawnie poruszać się w labiryncie robot musi posiadać możliwie dokładne informacje o swoim położeniu. Zadanie lokalizacji można zrealizować w dwóch etapach:

- określenie pola labiryntu, w którym znajduje się robot,
- wyznaczanie położenia robota względem pola labiryntu, w którym aktualnie się znajduje.

Na rysunku 3.6 zilustrowano te dwie metody.

3.4.1 Lokalizacja pola labiryntu, w którym znajduje się robot dokładnością do pola labiryntu

Lokalizacja pola, w którym znajduje się aktualnie robot polega na traktowaniu robota jako punktu, który porusza bezpośrednio pomiędzy środkami sąsiednich pól labiryntu. Robot wykorzystując informacje o kształcie labiryntu uzyskane za pomocą sensorów wyznacza kolejne pola labiryntu, które mają zostać odwiedzone. Ten etap lokalizacji jest wykorzystywany przy planowaniu ścieżki przechodzącej przez pola których, robot jeszcze nie odwiedził.

3.4.2 Lokalizacja robota względem środka danej komórki

Lokalizacja robota względem środka komórki, w której aktualnie znajduje się robot skupia się na sposobie poruszania się robota w labiryncie. Celem tego etapu nawigacji jest lokalizacja robota wewnątrz komórki, w której aktualnie się znajduje. Zadanie to jest realizowane poprzez kontrolę położenia robota jego otoczenia. Robot zbiera i analizuje informacje o swojej orientacji, położeniu oraz odległości od ścian labiryntu. Informacje te są niezbędne do kontrolowania położenia i orientacji robota przy przemieszczaniu się robota pomiędzy sąsiednimi polami labiryntu.

Rozdział 4

Konstrukcja robota

Aby robot bez problemu mógł poruszać się w labiryncie jego wymiary muszą być mniejsze od wymiarów pola labiryntu. Stąd zdecydowano się na budowę robota w kształcie koła o średnicy 8cm. Taki rozmiar gwarantuje swobodne nawigowanie pomiędzy ścianami labiryntu.

Zdecydowano że konstruowany robot powinien charakteryzować się:

- możliwością detekcji ścian labiryntu,
- możliwością pomiaru przemieszczenia,
- możliwością pomiaru zmian orientacji,
- silnikami pozwalającymi na osiągnięcie przyspieszenia robota rzędu $2\frac{m}{s^2}$,
- jak najmniejszą wagą, która wpływa na zmniejszenie się siły odśrodkowej przy pokonywaniu zakrętów,
- odpowiednio lekkim akumulatorem o pojemności > 300 mAh,
- dwukanałowym mostkiem H z wydajnością prądową $> 0.5A$, w celu zapewnienia odpowiedniego momentu obrotowego silników,
- nisko położonym środkiem ciężkości,
- mikrokontroler z jak największą liczbą układów peryferyjnych, dużą ilością pamięci (> 150 kB RAM) oraz obsługą kwarcu o częstotliwości co najmniej 50 MHz,
- układem zasilania dostarczającym odpowiednie poziomy napięcie dla wszystkich podzespołów.

4.1 Elementy składowe robota

Do pomiaru przemieszczenia, zmiany orientacji oraz pomiaru odległości robota od ścian labiryntu wybrano czujniki opisane w sekcji narzędzia. Czujniki odległości rozmieszczono jak na rysunku 2.4. Enkodery zostały umieszczone na zewnętrznej części kół robota tak aby wartością mierzoną była pozycja kątowna koła. Ze względu na charakter siły Coriolisa [12], na bazie której wyznaczana jest prędkość kątowna wokół wybranej osi robota, umieszczenie żyroskopu na płytce z elektroniką może być dowolne.

Aby umożliwić poruszanie się robota w labiryncie konieczne jest wyposażenie go w silniki o odpowiednio dużym momencie i prędkości obrotowej. Zdecydowano się na silniki Pololu HP z przekładnią 1:30. Wybór ten był motywowany wcześniejszymi doświadczeniami w stosowaniu ich w budowanych robotach mobilnych.

Przy konstrukcji robota klasy Micromouse ważnym aspektem jest waga robota. Roboty cięższe charakteryzują się większą inercją przy rozpędzaniu i hamowaniu, dlatego zdecydowano się na maksymalne ograniczenie wagi konstrukcji. Z tego względu robot nie posiada żadnej obudowy, a elementem nośnym całej konstrukcji jest płytka z elektroniką. Ponadto w celu zapewnienia dobrej przyczepności robota zdecydowano się na jak najniższe położenie środka ciężkości robota.

Przy wyborze mikrokontrolera kierowano się dostępnymi w nim układami peryferyjnymi oraz mocą obliczeniową. Założono, że główna pętla sterowania będzie działała z częstotliwością 1kHz. Z tego względu wymagane było zastosowanie mikrokontrolera taktowanego z częstotliwością co najmniej kilkukrotnie wyższą tak, aby był on w stanie obsłużyć wszystkie układy peryferyjne i być w stanie kontrolować pracę elementów wykonawczych robota. Po analizie wymienionych we wstępie do tego rozdziału potrzeb można stwierdzić, że niezbędnym jest aby robot posiadał:

- 3 timery z możliwością modulacji sygnału PWM oraz sprzętową obsługą sygnału kwadraturowego pochodzącego z enkoderów,
- interfejs SPI oraz I2C do komunikacji z żyroskopem,
- 2 interfejsy USART do komunikacji z użytkownikiem oraz programowania mikrokontrolera (poprzez bootloader),
- 5 wejść analogowych – 4 do pomiaru napięcia pochodzącego z czujników odległości oraz 1 do pomiaru napięcia akumulatora,

- interfejs pozwalający na programowanie i debugowanie mikrokontrolera z poziomu systemu operacyjnego Linux,
- obsługę rezonatora kwarcowego pozwalającego na taktowanie procesora z częstotliwością $> 50\text{MHz}$.

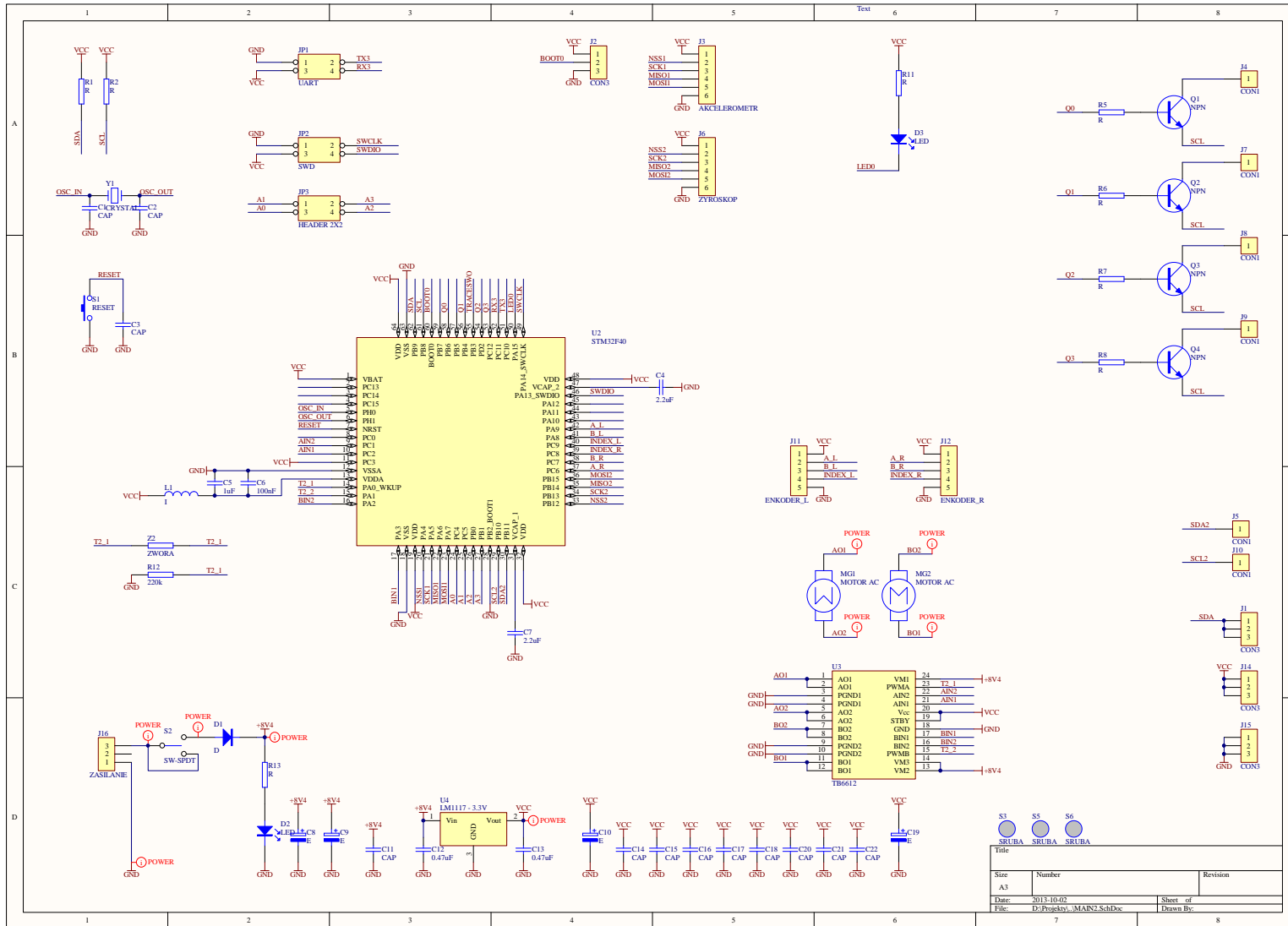
Tego typu funkcjonalność dostarcza układ STM32F4 więc wybrano go na jednostkę obliczeniową robota.

Do sterowania silnikami zdecydowano się wykorzystać mostek H TB6612. Mostek ten charakteryzuje się następującymi parametrami:

- napięciem zasilania 2.7 - 5.5 V,
- ciągłym prądem wyjściowym 0.6 A,
- sygnałem sterującym PWM,
- układem zabezpieczającym mostek przed przegrzaniem.

Sekcja zasilania robota składa się z akumulatora o napięciu 7.4 V oraz stabilizatora napięcia 3.3V. Wszystkie wykorzystywane układy mogą być zasilane napięciem 3.3V z tego względu zdecydowano się na wybór stabilizatora który, zapewnił odpowiednie napięcie zasilania i charakteryzował się wystarczającą wydajnością prądową.

Schemat ideowy płytki z elektroniką przedstawiono na rysunku 4.1



Rysunek 4.1 Schemat ideowy elektroniki

4.2 Oprogramowanie

Nadrzędną funkcją algorytmu jest znalezienie środka labiryntu, a następnie dotarcie do niego w jak najkrótszym czasie. W celu realizacji tego zadania należy wykonać szereg akcji pomocniczych, które zilustrowano na rysunku 4.2 i opisano poniżej.

4.2.1 Aktualizacja pomiarów

Robot co 1us dokonuje odczytu z czujników i uaktualnia informacje o swojej pozycji i orientacji. Algorytm komunikuje się z żyroskopem poprzez interfejs SPI i odczytuje wartość aktualnej prędkości kątowej. Na podstawie jej wartości i znajomości częstotliwości wykonywania się pętli algorytmu możliwe jest uaktualnienie orientacji robota.

Do gromadzenia informacji o kształcie labiryntu i odległości jego ścian od robota wykorzystano czujniki odległości. Odległość od ścian labiryntu jest proporcjonalna do napięcia, które pojawia się na wyjściu zaprojektowanego czujnika odległości. Do jego pomiaru wykorzystano przetwornik analogowo cyfrowy, który jest wbudowany w mikrokontroler.

Uaktualnianie pozycji robota jest realizowane poprzez wykorzystanie sygnału kwadraturowego pochodzącego z enkoderów. Do odczytu aktualnego położenia kątowego kół robota wykorzystywany jest jeden z timerów mikroprocesora, który potrafi dekodować sygnał kwadraturowy, w wyniku czego wartość licznika dla danego enkodera odpowiada jego pozycji kątowej.

4.2.2 Eliminacja poślizgów - profiler prędkości

W robocie klasy Micromouse zarówno przemieszczenie jak i zmiana orientacji jest osiągnięta w wyniku odpowiedniego sterowania pracą silników. Zbyt duża zmiana sygnału sterującego może skutkować ślizganiem się kół robota.

Przy sterowaniu robotami kołowymi korzystając z enkoderów istotnym problemem jest występowanie poślizgów. Generują one błędy trudne do odfiltrowania i zmniejszają dokładność i niezawodność pomiaru. W celu minimalizacji ilości poślizgów można wykorzystać tzw. profil prędkości [13]. Najbardziej popularnym profilem prędkości jest profil trapezoidalny. Profil taki ogranicza wartość przyspieszenia robota oraz jego maksymalną prędkość.

Celem stosowania profilu prędkości jest ograniczenie maksymalnego przyspieszenia przy starcie oraz hamowaniu robota oraz wybór maksymalnej dozwolonej dla robota prędkości. W celu minimalizacji prawdopodobieństwa wystąpienia poślizgu przy hamowaniu należy wybrać maksymalną prędkość, z jaką może poruszać się robot oraz odpowiednie nachylenie krzywej hamowania, która pozwoli zatrzymać się w zadanym miejscu bez wystąpienia poślizgu kół.

Przykład trapezoidalnych profili prędkości przedstawiono na rysunku 4.3.

4.2.3 Kolejka poleceń

W robocie zastosowano podział algorytmu sterowania na dwie warstwy:

- warstwa realizacji celu,
- warstwa obsługi sprzętowej.

Z punktu widzenia tematu pracy, pierwsza z nich zajmuje się badaniem aktualnej pozycji oraz orientacji robota. Na podstawie dotychczas przejechanej i zapamiętanej trasy wyznacza ona kolejne pola labiryntu, które ma odwiedzić robot. Po wyznaczeniu pól, które ma odwiedzić robot, algorytm czeka na informację zwrotną na to, że ruch się zakończył i wykonywana jest jego kolejna iteracja. Algorytm jest zupełnie odseparowany od warstwy sprzętowej robota. Nie wykorzystuje on bezpośrednio żadnego z sensorów, a jedynie operuje na przetworzonych wcześniej informacjach – o tym, które pola już odwiedził i o możliwościach przemieszczenia się z nich do pól sąsiednich. Do komunikacji z drugą warstwą wykorzystywany jest specjalny interfejs nazwany kolejką poleceń.

Do kolejki dodawane są dwa rodzaje poleceń, które są rozumiane przez warstwę sprzętową:

- ruch do przodu na zadaną odległość,
- obrót o zadany kąt.

Druga z warstw dba o wykonywanie zadawanych jej poleceń. Kolejno wykonuje polecenia znajdujące się w kolejce. Po wykonaniu danej instrukcji zgłasza to i sięga po kolejną. Taka separacja zadań przyczyniła się do zwiększenia czytelności kodu i uniezależniła rozwój tych dwóch warstw algorytmów od siebie.

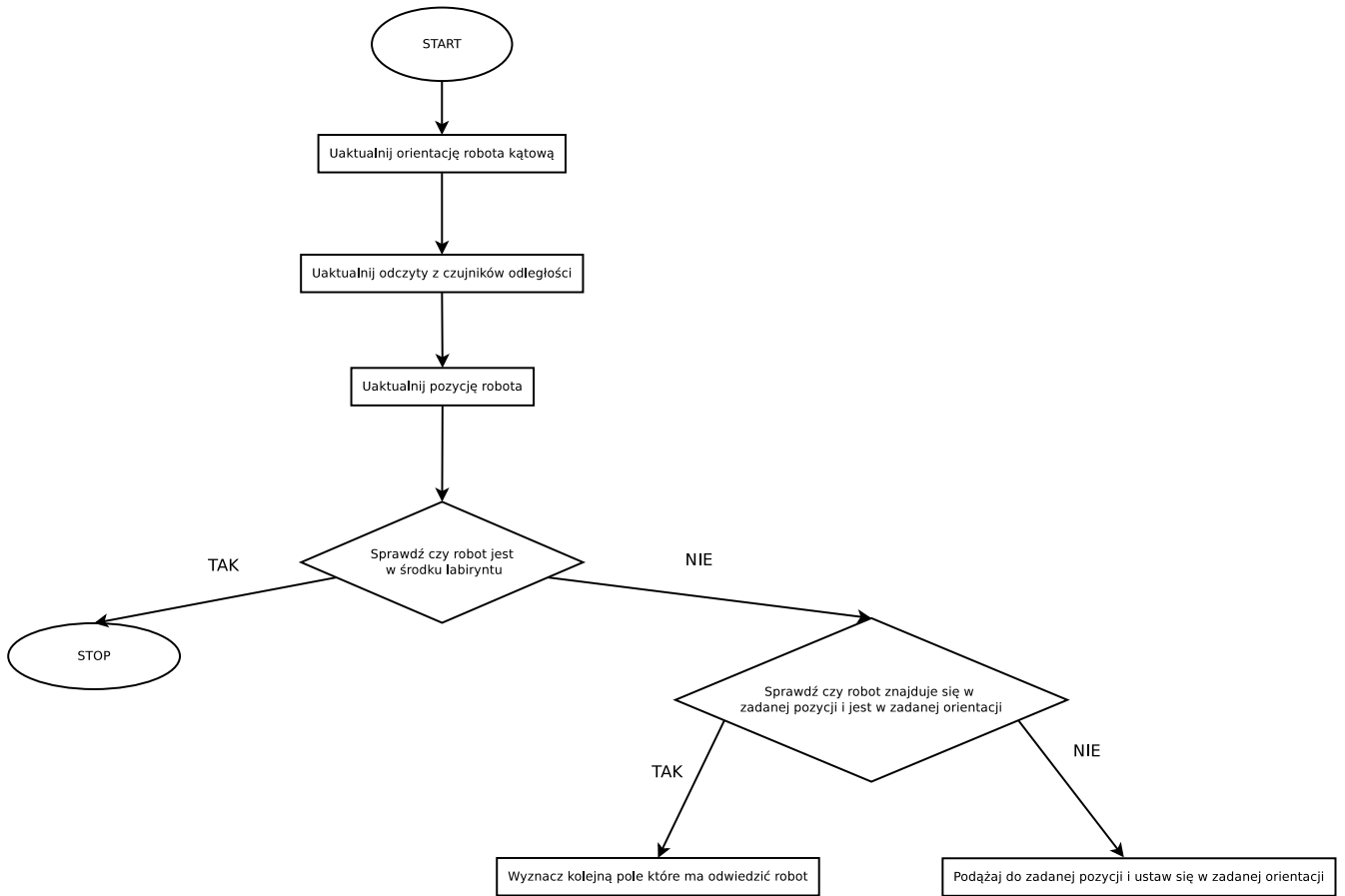
4.2.4 Algorytm sterowania przemieszczeniem i orientacją robota

Przy sterowaniu przemieszczeniem i orientacją robota wykorzystano regulator PID (proporcjonalno-całkująco-różniczkujący). Regulator ten pozwala na uzyskanie zerowego uchybu regulacji i jest stosunkowo prosty w implementacji [14]. Największym problemem przy wykorzystaniu tego typu regulatorów jest dobór wzmocnień członów: różniczkującego, całkującego i proporcjonalnego. Do nastrojenia regulatorów wykorzystano program Matlab. Poniżej przedstawiono schematy algorytmów sterowania przemieszczeniem i orientacją robota.

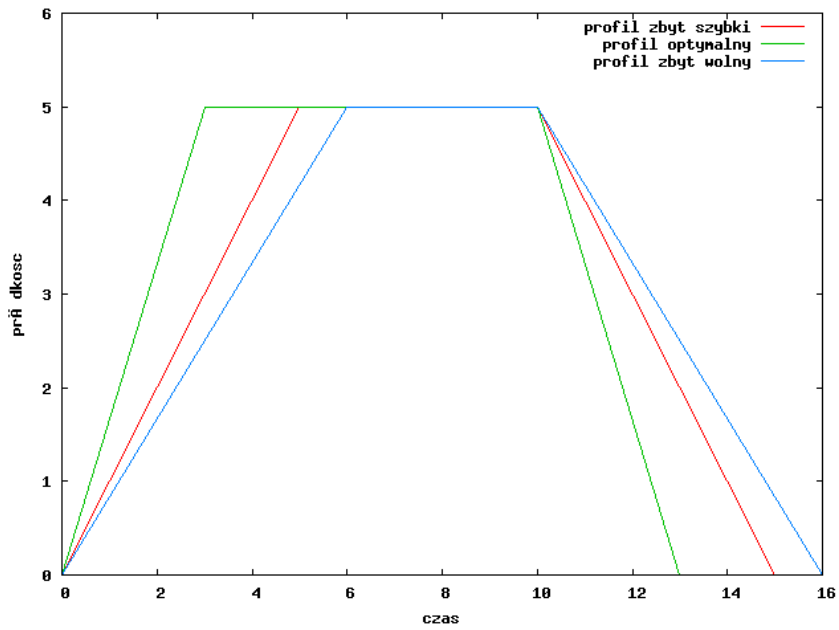
Sterowanie orientacją robota

Jak wcześniej wspomniano w zadaniu sterowania orientacją robota można wykorzystać enkodery lub żyroskop. Przy wykorzystaniu żyroskopu w celu regulacji kąta obrotu robota, na wejście regulatora PID doprowadzana jest różnica między zadana i aktualną orientacją robota. Na podstawie wielkości uchybu (różnicy między wartością zadaną i aktualną) obliczane są nowe sygnały sterujące pracą silników. Schemat algorytmu regulacji orientacji robota przedstawiono na rysunku 4.4.

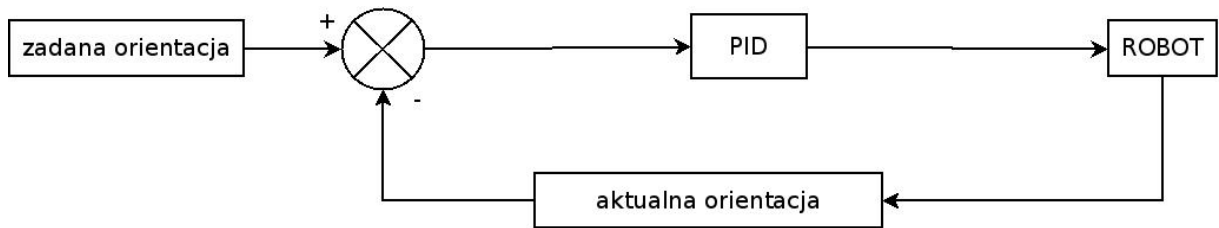
Nieco bardziej rozbudowany jest algorytm sterowania orientacją wykorzystujący odczyty z enkoderów. W celu regulacji porównuje on aktualną pozycję kół robota z pozycją zadaną. W związku z tym, że robot ma obracać się w miejscu, koła w czasie trwania ruchu powinny poruszać się z dokładnie taką samą prędkością lecz w przeciwnym kierunku, w wyniku czego droga przebyta przez koła powinna w każdej chwili być taka sama. W celu zapewnienia tego warunku zastosowano dodatkowy regulator PID, który wzmacnia sygnał sterujący silnika, który przebył krótszą drogę. Schemat takiego regulatora przedstawiono na rysunku 4.5



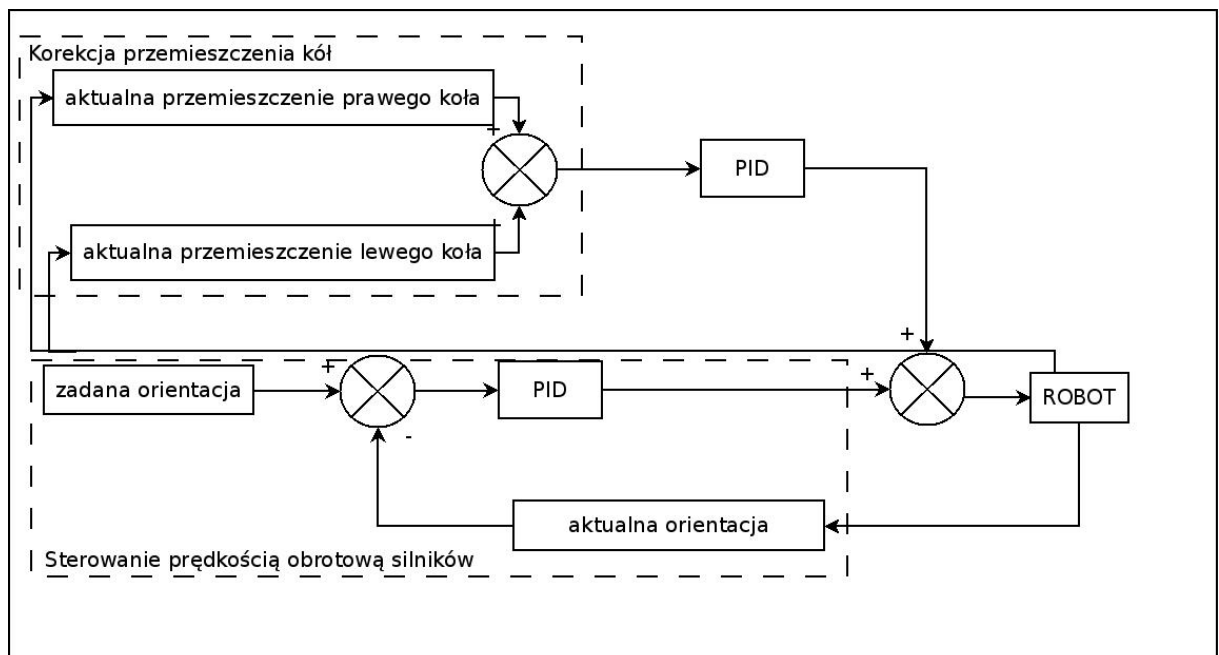
Rysunek 4.2 Algorytm robota



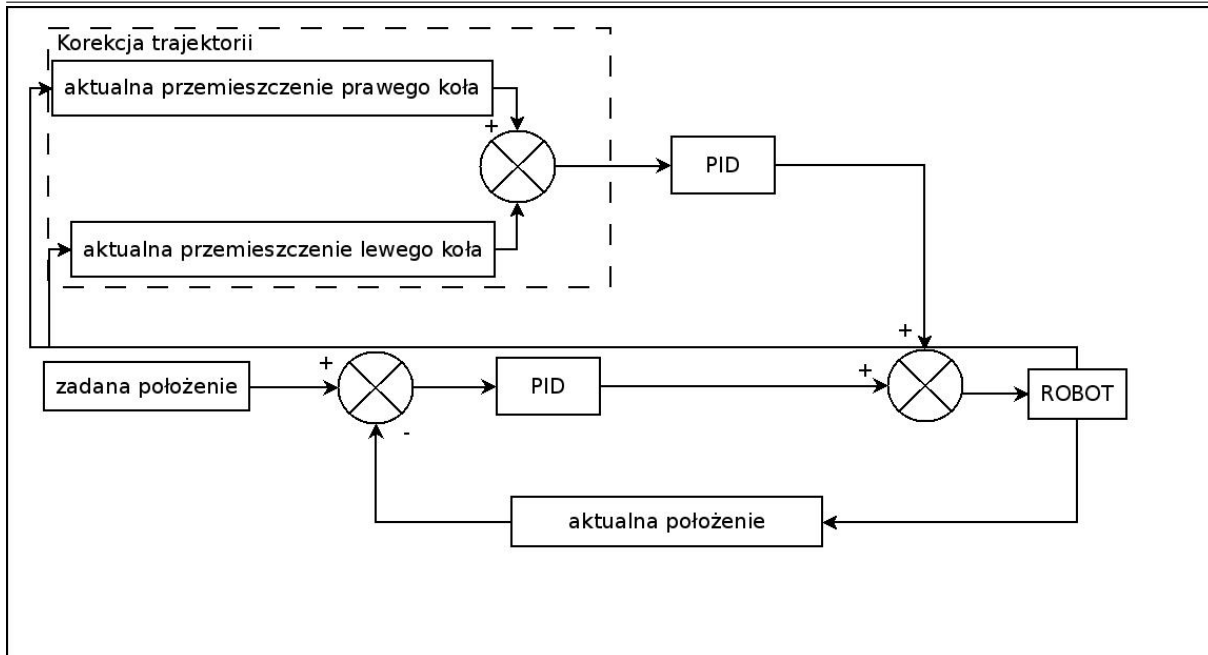
Rysunek 4.3 Przykładowy profil prędkości



Rysunek 4.4 Schemat algorytmu sterowania orientacją robota przy wykorzystaniu żyroskopu



Rysunek 4.5 Schemat algorytmu sterowania orientacją robota przy wykorzystaniu żyroskopu i korekcji pochodzącej z enkoderów



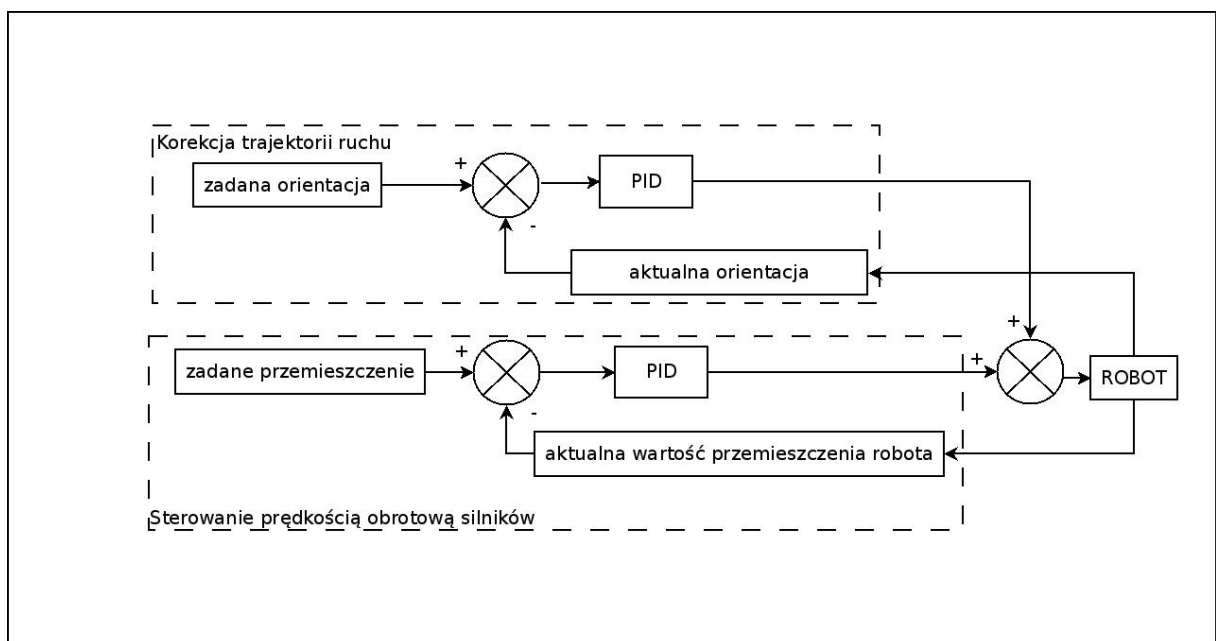
Rysunek 4.6 Schemat algorytmu sterowania orientacją przy pomocy enkoderów

Sterowanie przemieszczeniem robota po linii prostej

Do sterowania przemieszczeniem robota wykorzystywane są odczyty z enkoderów. Aktualne przemieszczenie jest porównywane z zadaniem i uchyb regulacji jest doprowadzony na wejście sterownika PID. W celu kontroli i ewentualnej korekcji orientacji robota podczas jazdy po linii prostej można wykorzystać enkodery lub żyroskop.

Aby robot poruszał się po linii prostej koniecznym jest aby podczas przemieszczania się robota w kolejnych odstępach czasu koła pokonywały takie same odległości. Korekcja bazująca na odczytach z enkoderów polega na porównywaniu drogi przejechanej przez koła robota i dbanie by różnica tychże dróg była równa 0. Jest to warunek konieczny do poprawnego sterowania przemieszczeniem robota z wykorzystaniem enkoderów. Schemat regulacji przemieszczenia robota przy wykorzystaniu korekcji pochodzącej z odczytów z enkoderów przedstawiono na rysunku 4.6.

Alternatywą jest wykorzystanie korekcji bazującej na pomiarach pochodzących z żyroskopu. W celu jazdy po linii prostej badana jest orientacja robota podczas rozpoczęcia ruchu, a następnie przez cały czas trwania ruchu algorytm dba, by była ona równa orientacji początkowej. Schemat tak działającego regulatora przedstawiono na rysunku 4.7.



Rysunek 4.7 Schemat algorytmu sterowania przemieszczeniem robota

Rozdział 5

Badania

Sterowanie orientacją robota może być realizowane za pomocą żyroskopu oraz enkoderów co opisano w sekcji 2.3.1. W celu pomiaru jakości sterowania przygotowano stanowisko pomiarowe składające się z:

- robota,
- kolorowych znaczników przyklejanych do robota służących do detekcji robota,
- umieszczonej nad stanowiskiem kamery,
- programu pozwalającego na odczyt parametrów robota za pomocą informacji z kamery.

5.1 Sposób dokonywania pomiarów przemieszczenia i orientacji robota

W celu pomiaru przemieszczenia robota przymocowano do niego kolorowe znaczniki, a do ich detekcji wykorzystano autorski program napisany przy wykorzystaniu biblioteki OpenCV. Zaimplementowany algorytm odnajduje na obrazie znacznik i zwraca współrzędne odpowiadające jego środkowi. Współrzędne te są interpretowane jako położenie robota. Na rysunku 5.1 przedstawiono obraz przed i po przetworzeniu przez omawiany algorytm. Na potrzeby pomiarów orientacji, na powierzchni robota umieszczono wspomniane znaczniki. Algorytm na podstawie ich wzajemnego położenia zwraca aktualną orientację robota. Przykład działania detekcji orientacji robota przedstawiono na rysunku 5.2

5.2 Porównanie jakości sterowania orientacją robota przy wykonywaniu obrotu w miejscu

Aby porównać jakość kontroli orientacji robota wymuszono na nim kilkukrotne obrót o zadany kąt w lewo oraz w prawo tak aby po zakończeniu sekwencji ruchów jego orientacja była taka sama jak przed jej rozpoczęciem. Następnie porównano orientację początkową z końcową robota, a różnicę między nimi przyjęto jako błąd pomiaru orientacji robota. Po ustabilizowaniu się orientacji robota wymuszono na nim korekcję orientacji bazującą na pomiarze odległości robota od ścian labiryntu.

Na rysunkach 5.3 oraz 5.4 zaprezentowano zmianę orientacji robota w czasie dla sterowania opartego na odczytach z żyroskopu oraz enkoderów.

Po zakończeniu pomiarów odczytano, że końcowy błąd określania orientacji robota wyniósł odpowiednio:

- $\approx 1^\circ$ dla sterowania opartego na odczytach z żyroskopu,
- $\approx 4.1^\circ$ dla sterowania opartego na odczytach z enkoderów.

Pomiar bazujący na odczytach z żyroskopu okazał się być dokładniejszy od pomiaru opartego na odczytach z enkoderów. Nie jest to wynik zaskakujący ze względu na to, że żyroskop charakteryzuje się większą dokładnością pomiarową.

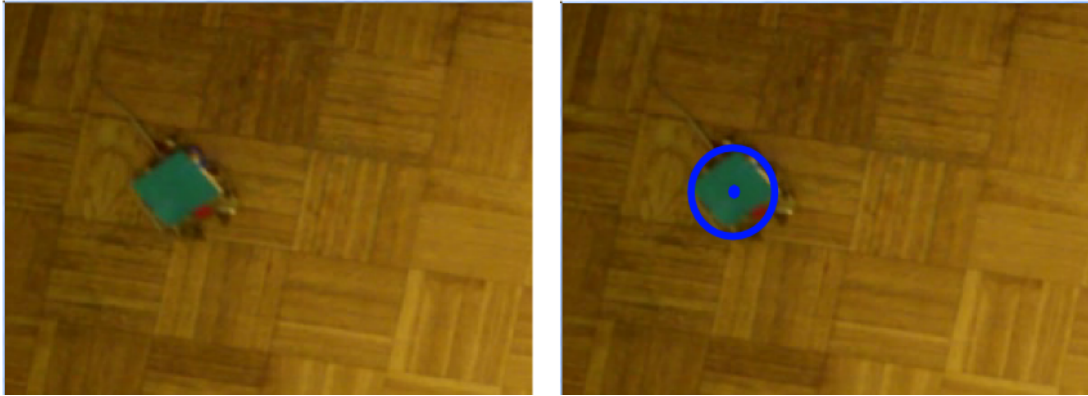
Na rysunkach 5.5 oraz 5.6 przedstawiono działanie korekcji orientacji bazującej na pomiarach z czujników odległości.

Korekcja taka pozwoliła w obydwu przypadkach na zmniejszenie wartości błędu orientacji robota. Ewentualna różnica pomiędzy osiągniętymi orientacjami dla sterowań wykorzystujących pomiary z różnych czujników może wynikać z niedoskonałości aparatury pomiarowej oraz niesprzyjających warunków pomiarowych np. nierównomiernego oświetlenia pokoju, w którym dokonywano pomiarów.

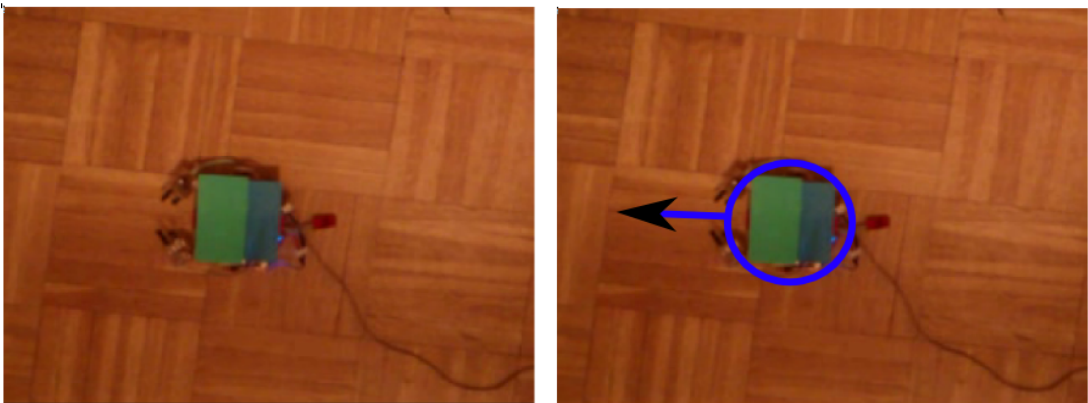
W celu pomiaru jakości algorytmu sterującego przemieszczaniem robota po linii prostej również wykorzystano system wizyjny. Na podstawie położenia robota w kilku pierwszych sekundach ruchu wyznaczono prostą po której poruszał się robot. Następnie porównywano odległość robota od tej prostej (w pikselach), aż do końca ruchu. Badanie to pozwoliło na porównanie algorytmów sterowania bazujących na odczytach z enkoderów i żyroskopu. Na rysunku 5.7 przedstawiono uzyskane pomiary.

W celu wybrania lepszego algorytmu sterowania orientacją robota posłużono się kryterium całki kwadratów odchyień. Otrzymane wartości przedstawiono w tabeli 5.1.

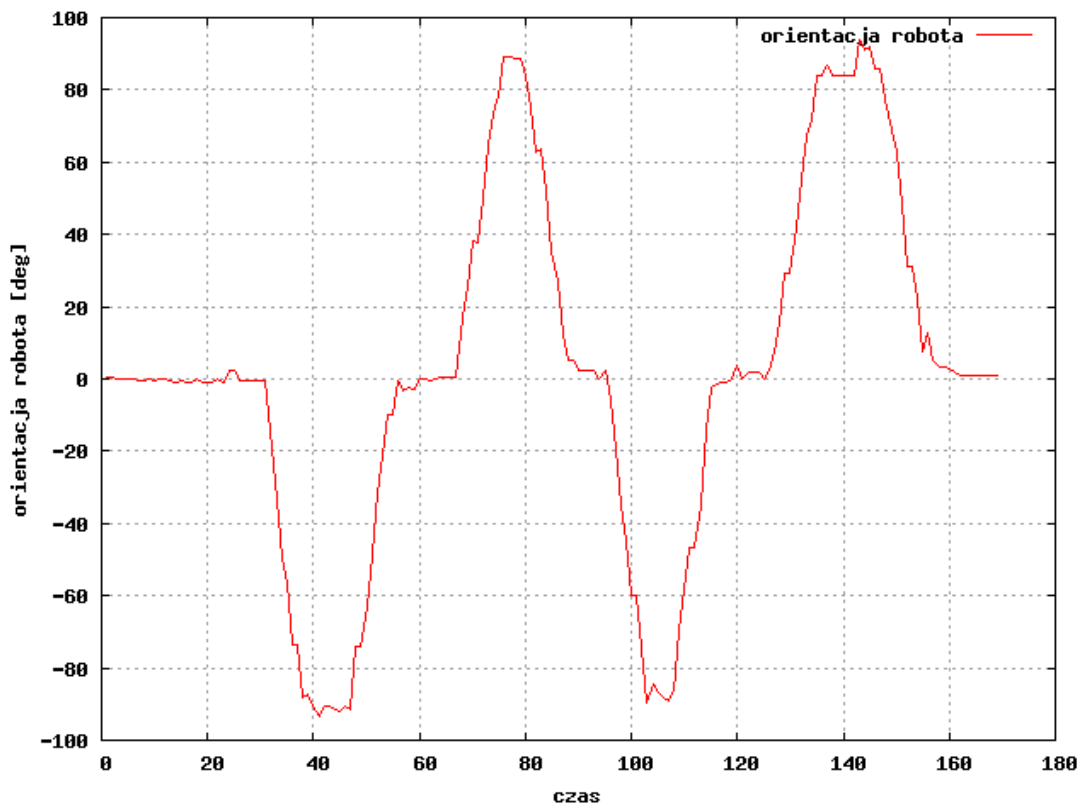
W przypadku sterowania położeniem robota podobnie jak w przypadku sterowania orientacją robota mniejsze błędy udało się uzyskać przy sterowaniu bazującym na pomiarach z żyroskopu.



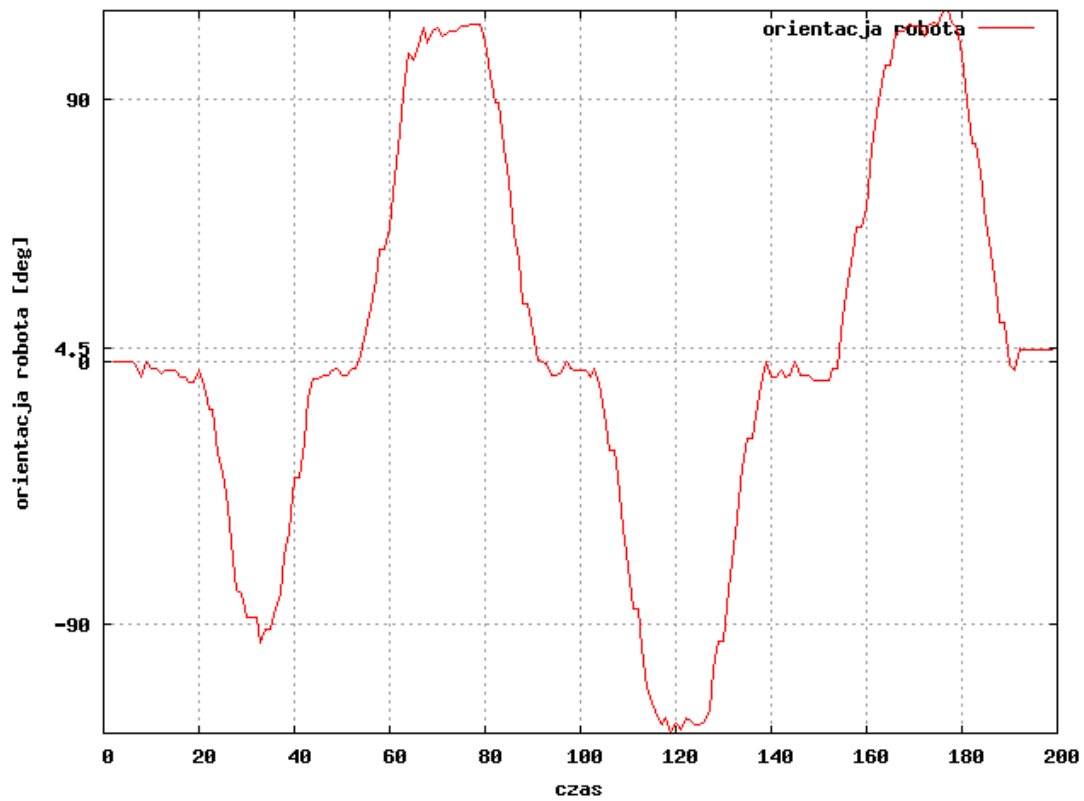
Rysunek 5.1 Detekcja położenia robota przy pomocy biblioteki OpenCV



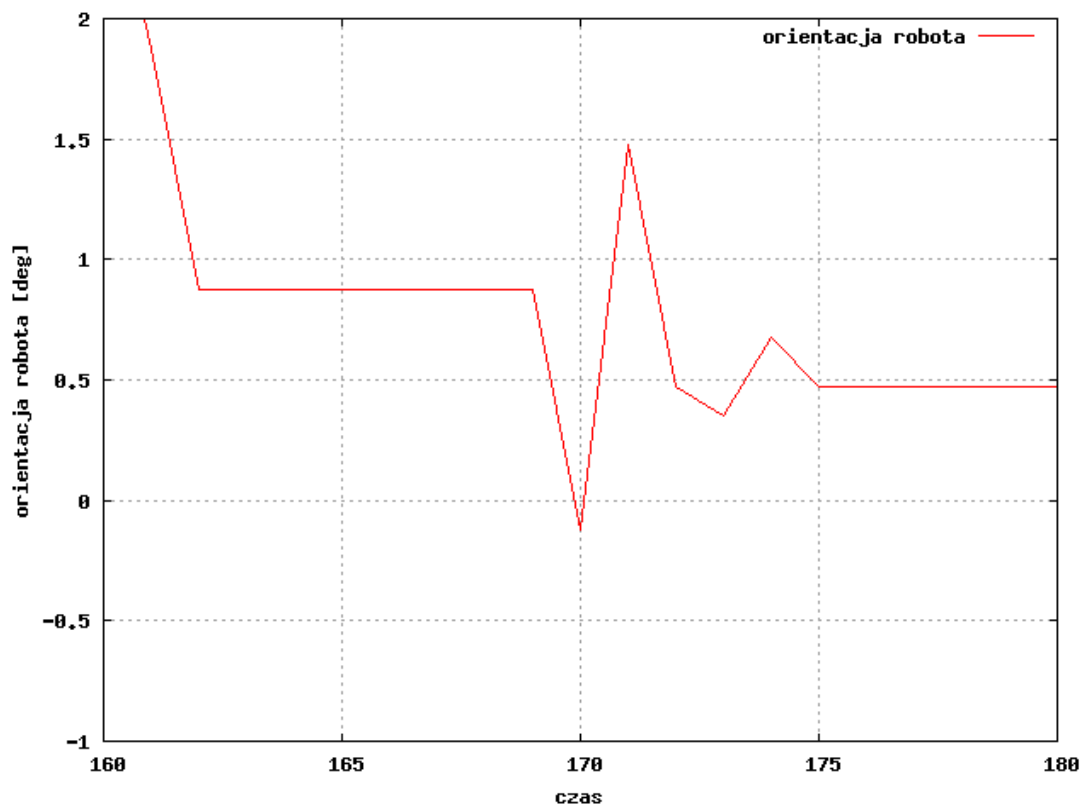
Rysunek 5.2 Detekcja orientacji robota przy pomocy biblioteki OpenCV



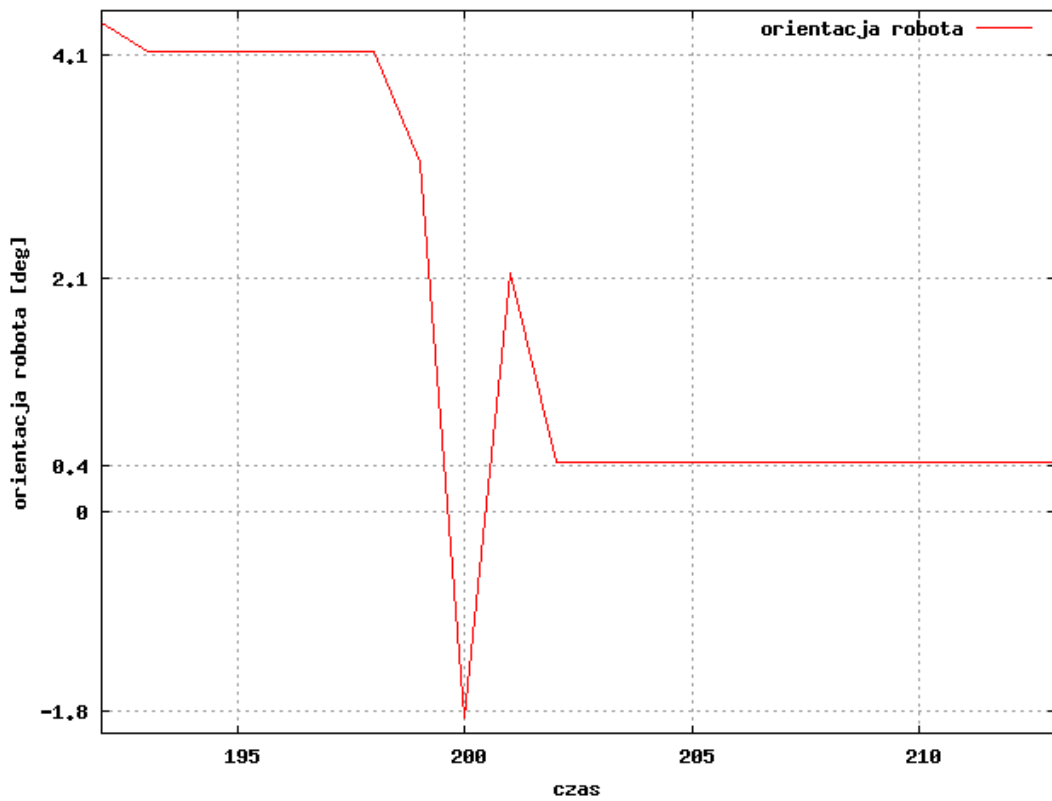
Rysunek 5.3 Zmiana orientacji robota w czasie przy sterowaniu opartym na odczytach z żyroskopu



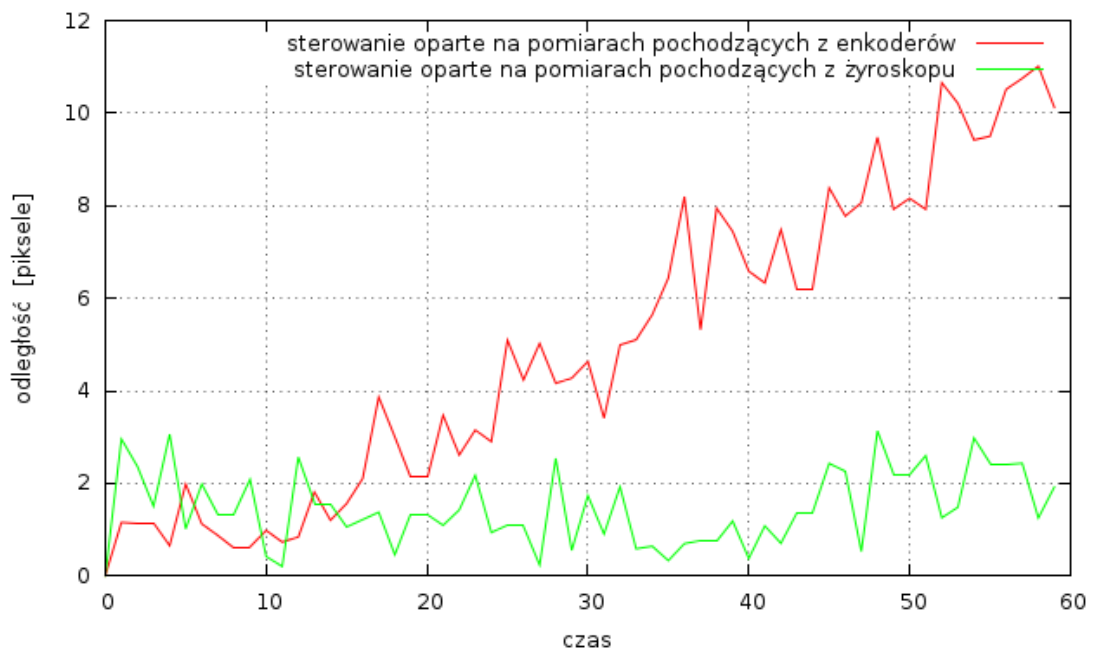
Rysunek 5.4 Zmiana orientacji robota w czasie przy sterowaniu opartym na odczytach z enkoderów



Rysunek 5.5 Korekcja pomiaru pochodzących z żyroskopu za pomocą czujników odległości



Rysunek 5.6 Korekcja pomiaru pochodzących z enkoderów za pomocą czujników odległości



Rysunek 5.7 Błędy przemieszczenia podczas jazdy do przodu, przy wykorzystaniu enkoderów

Tabela. 5.1 Porównanie jakości algorytmów sterowania położeniem robota

Algorytm	Całka kwadratów odchyień
bazujący na pomiarach pochodzących z żyroskopu	165.63
bazujący na pomiarach pochodzących z enkoderów	2111.6

Rozdział 6

Wnioski

Celem projektu było rozwiązanie problemu lokalizacji robota klasy Micromouse w labiryncie oraz zaprojektowanie platformy mobilnej, na której mogłyby zostać przetestowane zaimplementowane algorytmy. Tak przedstawiony cel został zrealizowany, udało się dokonać przeglądu możliwych problemów i ich rozwiązań dla zadania lokalizacji robota klasy Micromouse w labiryncie. Po dokonaniu rozważań teoretycznych dotyczących budowy i wyposażenia omawianej konstrukcji, wykonano prototyp robota klasy Micromouse przedstawiony na rysunku 6.1. Wykorzystanie konstrukcji klasy (2,0) uprościło zadanie sterowania robotem, gdyż pozwoliło wydzielić dwie akcje: ruch po linii prostej oraz obrót w miejscu, dzięki którym robot mógł przemieszczać się w labiryncie. Dobrym pomysłem jest wykorzystanie algorytmu PID do sterowania kołami robota. Duża ilość dostępnych materiałów odnośnie regulatora PID oraz możliwość wykorzystania programu matlab do doboru nastaw regulatora przyczyniła się do implementacji niezawodnego algorytmu sterowania robotem. Stworzonego robota wyposażono w zestaw szeregu czujników:

- enkodery,
- żyroskop,
- czujniki odległości.

W celu kontroli położenia robota w labiryncie wykorzystano enkodery, które ze względu na dużą dokładność świetnie sprawdziły się w tej roli. Jedynym mankamentem stosowania enkoderów jest konieczność kontrolowania przyspieszenia robota w celu minimalizacji poślizgów jego kół. W tym celu wykorzystano trapezoidalny profil prędkości. Następnie dokonano porównania możliwych rozwiązań pozwalających na kontrolę orientacji robota w labiryncie. Zdecydowano się na implementację algorytmu do kontroli orientacji robota bazującego na odczytach z:

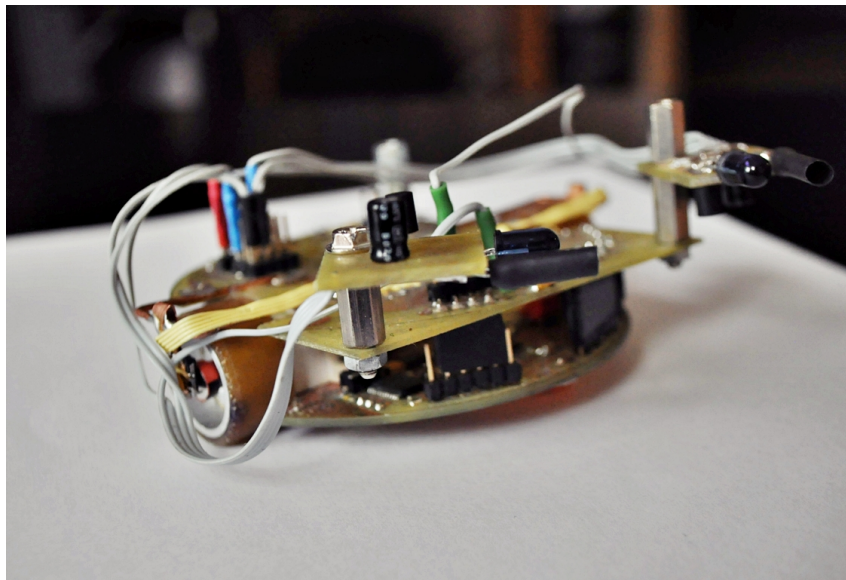
- enkoderów,
- enkoderów z korekcją przy wykorzystaniu czujników odległości,
- żyroskopu,
- żyroskopu z korekcją przy wykorzystaniu czujników odległości,

spośród których najlepszy okazał się być ostatni z wymienionych. Wykorzystanie żyroskopu zapewniło wysoką dokładność pomiaru, a czujniki odległości wykorzystane zostały do odfiltrowania zakłóceń wywołanych przez dryf żyroskopu. Zaimplementowany algorytm pozwolił na określanie z dużą dokładnością pozycji oraz orientacji robota w labiryncie.

W przygotowanym robocie klasy Micromouse wykorzystano: żyroskop L3GD20, enkoder AS5040 oraz czujnik odległości zbudowany z diody IR TSAL6100 i fototranzystora TEFT4300. Wszystkie wymienione czujniki okazały się być odporne na zakłócenia i pozwoliły na dokładny pomiar parametrów robota.

Zastosowany w konstrukcji mikrokontroler również był trafnym wyborem: sprzętowa obsługa zmiennych typu float, duża częstotliwość zegara oraz zrozumiała dokumentacja przyczyniła się do braku problemów przy programowaniu mikrokontrolera i implementacji algorytmów sterowania. Dodatkową zaletą mikrokontrolera są dostępne w internecie biblioteki do obsługi jego urządzeń peryferyjnych. Przy implementacji algorytmów zrezygnowano z wykorzystania ich ze względu na to, że są napisane w sposób nieoptymalny i samo dołączenie ich do projektu zwiększa objętość zajmowanego przez kod miejsca 10 krotnie.

Planowany jest dalszy rozwój projektu, który będzie obejmował implementację algorytmów nie wymuszających na robocie zatrzymywania się przy każdorazowej zmianie orientacji. Rozwiązanie takie powinno znacznie skrócić czas, w jakim robot będzie w stanie dojechać do środka labiryntu. Ponadto dobrym pomysłem wydaje się być zmiana aktualnie wykorzystywanego algorytmu filtracji odczytów z żyroskopu na filtr Kalmana. Co powinno zwiększyć dokładność pomiarów. W celu usprawnienia programowania robota planowane jest dołączenie do robota modułu Bluetooth który pozwoli na bezprzewodowe programowanie robota i podgląd parametrów robota w czasie rzeczywistym za pomocą komputera lub telefonu komórkowego wyposażonego w odbiornik bluetooth.



Rysunek 6.1 Stworzony na potrzeby projektu robot klasy Micromouse

Bibliografia

- [1] Rysunek przedstawiający labirynt. <http://www.forbot.pl/forum/topics20/micromouse-metody-przeszukiwania-labiryntu-vt2246.htm>.
- [2] K. Tchoń i inni. *Manipulatory i Roboty Mobilne: Modelowanie, planowanie ruchu, sterowanie*.
- [3] Rysunek przedstawiający robota Tetra. <http://www.engadget.com/2009/12/12/tetra-the-micromouse-needs-no-cheese-completes-maze-in-new-reco/>.
- [4] Rysunek przedstawiający robota Vampire2. <http://www.forbot.pl/forum/topics7/micromouse-vampire2-vt8531.htm>.
- [5] STMicroelectronics. L3GD20. <http://www.pololu.com/file/0J563/L3GD20.pdf>.
- [6] Austria Microsystems. AS5040. <http://www.altotech.ee/Miscs/dsc/4020v3/pc4020v3.pdf>.
- [7] Vishay. TEFT4300. www.vishay.com/docs/81549/teft4300.pdf.
- [8] Vishay. TSAL6100. www.vishay.com/docs/81009/tsal6100.pdf.
- [9] Sebastian O. H. Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. http://sharenet-wii-motion-trac.googlecode.com/files/An_efficient_orientation_filter_for_inertial_and_inertialmagnetic_sensor_arrays.pdf, 2010.
- [10] Jan Kędziński. *Filtr Kalmana - zastosowania w prostych układach sensorycznych*. http://sharenet-wii-motion-trac.googlecode.com/files/An_efficient_orientation_filter_for_inertial_and_inertialmagnetic_sensor_arrays.pdf, 2007.
- [11] Filtr Kalmana - Teoria i Praktyka. <http://www.forbot.pl/forum/topics20/inne-filtr-kalmana-teoria-i-praktyka-vt7789.htm>, 2012.
- [12] Wikipedia. Efekt Coriolisa. http://pl.wikipedia.org/wiki/Efekt_Coriolisa.
- [13] Bartosz Wawrzacz. Sterowanie robotem mobilnym klasy (2,0). <http://www.forbot.pl/forum/topics20/programowanie/sterowanie/robotem-mobilnym-klasy-20-vt7787.htm>.
- [14] Wikipedia. Regulator PID. [pl/forum/topics7/micromouse-vampire2-vt8531.htm](http://www.forbot.pl/forum/topics7/micromouse-vampire2-vt8531.htm).

Spis rysunków

1.1	Labirynt i jego środek	2
2.1	Micromouse klasy (4,0) Tetra [3]	4
2.2	Micromouse klasy (2,0) Vampire 2 [4]	4
2.3	Funkcjonalność stworzonego robota	4
2.4	Schemat rozłożenia czujników w robocie	5
2.5	Schemat robota	7
3.1	Żyroskop L3GD20	9
3.2	Enkoder AS5040	9
3.3	Charakterystyka diody IR TSAL6100	9
3.4	Charakterystyka fototranzystora TEFT4300	10
3.5	Filtracja pomiarów pochodzących z czujników odległości	10
3.6	Metody lokalizacji robota	12
4.1	Schemat ideowy elektroniki	15
4.2	Algorytm robota	17
4.3	Przykładowy profil prędkości	17
4.4	Schemat algorytmu sterowania orientacją robota przy wykorzystaniu żyroskopu	18
4.5	Schemat algorytmu sterowania orientacją robota przy wykorzystaniu żyroskopu i korekcji pochodzącej z enkoderów	18
4.6	Schemat algorytmu sterowania orientacją przy pomocy enkoderów	19
4.7	Schemat algorytmu sterowania przemieszczeniem robota	20
5.1	Detekcja położenia robota przy pomocy biblioteki OpenCV	22
5.2	Detekcja orientacji robota przy pomocy biblioteki OpenCV	22
5.3	Zmiana orientacji robota w czasie przy sterowaniu opartym na odczytach z żyroskopu	22
5.4	Zmiana orientacji robota w czasie przy sterowaniu opartym na odczytach z enkoderów	23
5.5	Korekcja pomiaru pochodzących z żyroskopu za pomocą czujników odległości	23
5.6	Korekcja pomiaru pochodzących z enkoderów za pomocą czujników odległości	24
5.7	Błędy przemieszczenia podczas jazdy do przodu, przy wykorzystaniu enkoderów	24
6.1	Stworzony na potrzeby projektu robot klasy Micromouse	26

Spis tabel

5.1 Porównanie jakości algorytmów sterowania położeniem robota	24
--	----