



**WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI
POLITECHNIKI RZESZOWSKIEJ**

Kacper Rychel

Robot mobilny z nawigacją autonomiczną oparty na Arduino

PRACA INŻYNIERSKA

Opiekun pracy:
Dr inż. Mariusz Mączka

Rzeszów, 2026

Spis treści

1. Wprowadzenie	5
2. Zakres teoretyczny z dziedziny robotyki	7
2.1. Przegląd stanu wiedzy i techniki w zakresie robotów mobilnych	7
2.1.1. Modelowanie ruchu robota	7
2.1.2. Odtwarzanie danych sensorycznych	8
2.1.3. Technologia mapowania SLAM	8
2.2. Klasyfikacja i architektura robotów mobilnych	9
2.2.1. Klasyfikacja według sposobu lokomocji robota	9
2.2.2. Klasyfikacja według autonomiczności robota	10
2.2.3. Klasyfikacja według architektury rozwiązania	10
2.3. Modelowanie ruchu robotów mobilnych	11
2.4. Układy napędowe i sterowanie silnikami	13
2.4.1. Sterowanie silnikami robota	15
2.5. Percepcja otoczenia	15
2.6. Zaawansowane metody nawigacji	17
3. Część Praktyczna/fizyczna - zmień nazwę	20
4. Część Praktyczna/kodowa - zmień nazwę	21
Literatura	22

1. Wprowadzenie

Roboty mobilne należą do najbardziej dynamicznie rozwijających się obszarów współczesnej robotyki oraz inżynierii mechatronicznej. Ich znaczenie systematycznie rośnie wraz z postępem technologii obliczeniowych, czujnikowych oraz energetycznych, co przekłada się na coraz szerszy zakres zastosowań. Roboty mobilne wykorzystywane są obecnie m.in. w przemyśle, logistyce, eksploracji środowisk niebezpiecznych, a także w systemach autonomicznych. Zgodnie z definicją Międzynarodowej Federacji Robotyki, robot mobilny jest systemem mechatronicznym zdolnym do samodzielnego przemieszczania się w przestrzeni oraz realizacji określonych zadań bez stałego połączenia z infrastrukturą stacjonarną[1].

Rozwój mikrokontrolerów oraz popularyzacja platform open-source istotnie obniżyły próg wejścia w projektowaniu systemów robotycznych. Jak zauważa Monk, dostępność tanich i dobrze udokumentowanych platform, takich jak Arduino, umożliwia szybkie prototypowanie rozwiązań robotycznych również w warunkach edukacyjnych i amatorskich[2]. Dzięki temu możliwe stało się praktyczne łączenie zagadnień z zakresu elektroniki, programowania oraz automatyki w jednym projekcie.

Autonomia robota mobilnego rozumiana jest jako zdolność do samodzielnego podejmowania decyzji na podstawie informacji pozyskiwanych z czujników. System autonomiczny integruje procesy percepji otoczenia, podejmowania decyzji oraz sterowania ruchem. Siegwart, Nourbakhsh i Scaramuzza wskazują, że nawet w przypadku prostych algorytmów reaktywnych niezbędna jest spójna współpraca tych elementów, aby robot mógł funkcjonować w sposób przewidywalny i bezpieczny[3].

Centralnym elementem sterującym robota mobilnego jest mikrokontroler, odpowiedzialny za przetwarzanie danych sensorycznych oraz generowanie sygnałów sterujących elementami wykonawczymi. Platforma Arduino UNO, oparta na mikrokontrolerze ATmega328P, należy do najczęściej wykorzystywanych w projektach edukacyjnych ze względu na bogatą dokumentację techniczną, dostępność bibliotek programistycznych oraz szerokie wsparcie społeczności open-source[4]. Zastosowanie gotowych bibliotek znaczco skraca czas implementacji algorytmów sterowania oraz obsługi czujników.

Istotną rolę w systemach autonomicznych odgrywają czujniki umożliwiające pozyskiwanie informacji o otoczeniu. W robotach niskokosztowych powszechnie wykorzystuje się czujniki ultradźwiękowe, które pozwalają na pomiar odległości na podstawie

czasu przelotu fali akustycznej. Popularnym rozwiązaniem w projektach edukacyjnych jest moduł HC-SR04, który charakteryzuje się niewielkim kosztem, prostą obsługą oraz łatwą integracją z mikrokontrolerami[5]. Dokumentacja producenta wskazuje, że dokładność pomiaru jest wystarczająca dla podstawowych algorytmów nawigacyjnych, choć należy uwzględnić wpływ właściwości powierzchni odbijających oraz warunków środowiskowych[6].

Ruch robota mobilnego realizowany jest najczęściej za pomocą napędu różnicowego, składającego się z dwóch niezależnie sterowanych silników prądu stałego. Taki układ umożliwia realizację manewrów poprzez różnicowanie prędkości obrotowych kół. Jak wskazują Jones, Flynn i Seiger, napęd różnicowy cechuje się niewielką złożonością mechaniczną i sterowniczą, co czyni go jednym z najczęściej stosowanych rozwiązań w robotach mobilnych[7]. Do sterowania silnikami prądu stałego stosuje się układy typu mostek H, takie jak L293D, umożliwiające zmianę kierunku obrotów oraz regulację prędkości z wykorzystaniem sygnału PWM. Tego typu układy stanowią podstawowy element wykonawczy w systemach sterowania napędami niskiej mocy[8].

Praca podzielona jest na trzy rozdziały. W drugim rozdziale przedstawiony jest zakres teoretyczny zagadnień potrzebnych do skompletowania robota. Trzeci rozdział przeznaczony jest skupieniu najważniejszych elementów budowania fizycznej podstawy robota. W czwartym rozdziale opisany został kod programowy robota.

Celem niniejszej pracy jest zaprojektowanie i wykonanie autonomicznego robota mobilnego opartego na platformie Arduino UNO, zdolnego do samodzielnego poruszania się oraz unikania przeszkód na podstawie pomiarów odległości. Zakres pracy obejmuje analizę teoretyczną zagadnień robotyki mobilnej, projekt i realizację układu sprzętowego, implementację algorytmu sterowania oraz przeprowadzenie testów funkcjonalnych opracowanego rozwiązania.

2. Zakres teoretyczny z dziedziny robotyki

Coś tam

2.1. Przegląd stanu wiedzy i techniki w zakresie robotów mobilnych

Postęp technologiczny w obszarze robotyki mobilnej, obserwowany w ostatnich dekadach, doprowadził do wykształcenia się wielu zróżnicowanych podejść konstrukcyjnych oraz algorytmicznych. Dobór konkretnego rozwiązania zależy przede wszystkim od przeznaczenia robota, charakterystyki środowiska pracy oraz dostępnych zasobów sprzętowych i obliczeniowych. Jak podkreśla się w literaturze przedmiotu, pomimo dynamicznego rozwoju systemów autonomicznych i algorytmów sztucznej inteligencji, w dalszym ciągu istotną rolę odgrywają rozwiązania proste i deterministyczne, szczególnie w systemach o charakterze edukacyjnym oraz prototypowym [9].

2.1.1. Modelowanie ruchu robota

Jednym z podstawowych zagadnień w robotyce mobilnej jest modelowanie ruchu robota, stanowiące fundament projektowania algorytmów sterowania oraz nawigacji. W przypadku najczęściej spotykanego napędu różnicowego ruch robota opisywany jest za pomocą nieliniowego modelu kinematycznego, w którym prędkość liniowa oraz prędkość kątowa platformy są bezpośrednio zależne od prędkości obrotowych kół napędowych. Campion, Bastin oraz D'Andrea-Novel wskazują, że model kinematyczny robota z napędem różnicowym cechuje się prostotą implementacji, jednak wprowadza istotne ograniczenie w postaci braku możliwości realizacji ruchu bocznego[10]. Ograniczenie to wymusza stosowanie odpowiednich strategii planowania trajektorii oraz manewrowania, zwłaszcza w środowiskach o ograniczonej przestrzeni roboczej.

W praktycznych realizacjach robotów mobilnych, szczególnie w systemach dysponujących niewielką mocą obliczeniową, sterowanie ruchem realizowane jest zazwyczaj w oparciu o regulatory dyskretne, bez jawnego uwzględniania pełnego modelu dynamicznego. Regulacja prędkości silników prądu stałego odbywa się najczęściej przy użyciu modulacji szerokości impulsu PWM, natomiast zmiana kierunku obrotów realizowana jest poprzez odpowiednie sterowanie mostkiem H. Jak zauważają Horowitz i Hill, takie podejście, mimo swojej prostoty, jest w pełni wystarczające w systemach, w

których nie jest wymagana wysoka precyzaja pozycjonowania ani zaawansowana regulacja momentu obrotowego[8].

2.1.2. Odtwarzanie danych sensorycznych

Istotnym elementem wpływającym na skuteczność autonomii robota mobilnego jest sposób pozyskiwania oraz przetwarzania danych sensorycznych. W systemach niskokosztowych dominują rozwiązania wykorzystujące pojedyncze lub kilka czujników odległości, których wskazania są bezpośrednio wykorzystywane do podejmowania decyzji ruchowych. Takie podejście określone jest w literaturze mianem nawigacji reaktywnej, w której robot nie buduje globalnej reprezentacji środowiska, lecz reaguje wyłącznie na aktualnie wykryte przeszkody[11]. Metody te, choć nieoptymalne z punktu widzenia długości trajektorii czy efektywności ruchu, charakteryzują się dużą odpornością na zmiany otoczenia oraz niewielkimi wymaganiami obliczeniowymi.

W algorytmach nawigacji reaktywnej powszechnie stosowane są czujniki ultradźwiękowe, cenione za prostą zasadę działania oraz łatwość integracji z popularnymi mikrokontrolerami. Badania porównawcze czujników odległości wskazują, że dokładność pomiarów ultradźwiękowych jest wystarczająca do wykrywania przeszkód o rozmiarach porównywalnych z wymiarami robota, jednak ulega pogorszeniu w przypadku powierzchni pochłaniających fale akustyczne lub ustawionych pod znacznym kątem względem osi pomiaru[12]. Z tego względu zaleca się stosowanie odpowiednich marginesów bezpieczeństwa w algorytmach decyzyjnych oraz filtrację wyników pomiarów w celu zwiększenia odporności systemu na błędne odczyty[13].

2.1.3. Technologia mapowania SLAM

W bardziej zaawansowanych systemach robotów mobilnych stosowane są techniki lokalizacji i mapowania jednocięsnego (SLAM), wykorzystujące dane z czujników LiDARowych¹, kamer wizyjnych oraz jednostek inercyjnych. Metody te umożliwiają jednoczesną budowę mapy otoczenia oraz precyzyjne określanie położenia robota, jednak wiążą się z wysokimi wymaganiami obliczeniowymi oraz znaczną złożonością implementacyjną[14]. Jak podkreśla Thrun, implementacja algorytmów SLAM w systemach edukacyjnych jest często nieuzasadniona, gdyż poziom skomplikowania może przesłaniać podstawowe zagadnienia inżynierskie, takie jak sterowanie ruchem czy integracja czujników[15].

¹LiDAR (Light Detection and Ranging) - metoda pomiarowa używana do określania precyzyjnego dystansu obiektu.

2.2. Klasyfikacja i architektura robotów mobilnych

Roboty mobilne mogą być klasyfikowane według wielu różnych kryteriów, co wynika z dużej różnorodności ich konstrukcji, przeznaczenia oraz środowisk pracy. W literaturze przedmiotu podkreśla się, że brak jednej uniwersalnej klasyfikacji robotów mobilnych jest konsekwencją interdyscyplinarnego charakteru tej dziedziny, łączącej zagadnienia z zakresu mechaniki, elektroniki, informatyki oraz teorii sterowania[16]. Najczęściej spotykane podejścia klasyfikacyjne obejmują podział ze względu na mechanizm lokomocji, stopień autonomii, środowisko eksploatacji oraz zakres realizowanych zadań.

2.2.1. Klasyfikacja według sposobu lokomocji robota

Jednym z podstawowych i najbardziej intuicyjnych kryteriów klasyfikacji robotów mobilnych jest sposób ich poruszania się. W tym ujęciu wyróżnia się roboty kołowe, gąsienicowe, kroczące oraz konstrukcje hybrydowe. Roboty kołowe, wyposażone w jedno lub więcej kół napędowych, charakteryzują się wysoką sprawnością energetyczną, niewielkimi stratami energii wynikającymi z oporów toczenia oraz prostotą konstrukcji mechanicznej. Jak wskazuje Siegwart, łatwość sterowania oraz stosunkowo niski koszt wykonania sprawiają, że roboty kołowe dominują w zastosowaniach edukacyjnych, przemysłowych oraz badawczych[3]. Ich istotnym ograniczeniem jest jednak ograniczona zdolność poruszania się po nierównym lub nieuporządkowanym terenie, co zwiększa zakres potencjalnych zastosowań.

Alternatywą dla konstrukcji kołowych stanowią roboty gąsienicowe, które dzięki większej powierzchni styku z podłożem cechują się lepszą przyczepnością oraz zdolnością pokonywania przeszkód terenowych. Z tego względu znajdują one zastosowanie w robotach eksploracyjnych, ratowniczych oraz wojskowych, gdzie wymagana jest wysoka stabilność ruchu w trudnych warunkach środowiskowych. Jak zauważa Borenstein, zwiększone opory ruchu oraz większa złożoność mechaniczna układów gąsienicowych prowadzą do wyższego zużycia energii oraz utrudniają precyzyjne sterowanie ruchem w porównaniu z robotami kołowymi[17].

Najbardziej złożoną grupę robotów mobilnych pod względem konstrukcyjnym i algorytmicznym stanowią roboty kroczące. Ich lokomocja opiera się na sekwencyjnym przemieszczaniu kończyn, co umożliwia poruszanie się w terenie niedostępnym dla robotów kołowych i gąsienicowych, takim jak schody, gruzowiska czy obszary o znacznych nierównościach. Projektowanie robotów kroczących wymaga zaawansowa-

nego modelowania dynamiki, precyzyjnego sterowania ruchem oraz znacznych zasobów obliczeniowych[18].

2.2.2. Klasyfikacja według autonomiczności robota

Istotnym kryterium klasyfikacji robotów mobilnych, niezależnym od mechanizmu lokomocji, jest stopień autonomii systemu. W tym ujęciu wyróżnia się roboty zdalnie sterowane, półautonomiczne oraz autonomiczne. Roboty zdalnie sterowane wymagają stałej ingerencji operatora i nie podejmują samodzielnych decyzji, roboty autonomiczne są zdolne do samodzielnego analizy danych sensorycznych, planowania działań oraz realizacji zadań bez bezpośredniego nadzoru człowieka, natomiast roboty półautonomiczne są hybrydowym rozwiązaniem łączącym cechy maszyn autonomicznych oraz zdalnie sterowanych. Jak wskazuje Arkin, poziom autonomii robota jest ściśle powiązany z zastosowanymi algorytmami decyzyjnymi oraz architekturą systemu sterowania[11].

2.2.3. Klasyfikacja według architektury rozwiązania

Z punktu widzenia architektury systemowej robot mobilny stanowi złożony system mechatroniczny, który można opisać jako zbiór współpracujących ze sobą warstw funkcjonalnych. Wyróżnia się warstwę sensoryczną, odpowiedzialną za pozyskiwanie informacji o stanie robota i jego otoczeniu, warstwę decyzyjną, w której realizowane są algorytmy sterowania i nawigacji, oraz warstwę wykonawczą, obejmującą układy napędowe i elementy wykonawcze. Taki podział funkcjonalny jest powszechnie stosowany w literaturze i umożliwia modularne oraz skalowalne projektowanie systemów robotycznych[7].

Warstwa sensoryczna obejmuje zestaw czujników, takich jak czujniki odległości, enkodery, czujniki inercyjne czy kamery wizyjne, których zadaniem jest dostarczanie informacji o położeniu robota, jego ruchu oraz otoczeniu. Rodzaj i jakość danych sensorycznych w istotny sposób wpływają na możliwości percepcyjne systemu oraz skuteczność podejmowanych decyzji.

Warstwa decyzyjna stanowi centralny element architektury robota mobilnego. W prostych konstrukcjach realizowana jest bezpośrednio na mikrokontrolerze, który przetwarza dane sensoryczne i generuje sygnały sterujące w czasie rzeczywistym. W bardziej zaawansowanych systemach warstwa ta może być rozproszona pomiędzy kilka jednostek obliczeniowych, takich jak komputery jednopłytkowe lub procesory dedykowane, co umożliwia implementację złożonych algorytmów planowania trajektorii, lo-

kalizacji czy uczenia maszynowego[19]. Rozwiązań te zwiększą jednak złożoność systemu oraz zapotrzebowanie na energię.

Warstwa wykonawcza obejmuje elementy odpowiedzialne za fizyczną realizację ruchu robota, w tym silniki, przekładnie mechaniczne oraz układy sterujące. Jej projekt musi uwzględniać wymagania mechaniczne i elektryczne, a także ograniczenia wynikające z zastosowanego źródła zasilania. Jak podkreślają Jones i Flynn, prawidłowa integracja warstwy wykonawczej z pozostałymi warstwami systemu jest kluczowa dla zapewnienia stabilnego, przewidywalnego i bezpiecznego zachowania robota mobilnego[7].

2.3. Modelowanie ruchu robotów mobilnych

Modelowanie ruchu robota mobilnego stanowi jeden z kluczowych etapów projektowania systemów sterowania oraz nawigacji, gdyż umożliwia formalny, matematyczny opis zależności pomiędzy sygnałami sterującymi a rzeczywistym ruchem platformy. W robotyce mobilnej wyróżnia się dwa podstawowe podejścia do opisu ruchu: modelowanie kinematyczne oraz dynamiczne. Model kinematyczny koncentruje się na zależnościach geometrycznych i prędkościowych, pomijając wpływ sił oraz momentów działających na robota, natomiast model dynamiczny uwzględnia masę, bezwładność oraz oddziaływanie z otoczeniem[20]. W przypadku większości robotów edukacyjnych i prototypowych stosuje się modele kinematyczne, które zapewniają wystarczającą dokładność przy znacznie mniejszej złożoności obliczeniowej.

Najczęściej spotykanym rozwiązaniem konstrukcyjnym w robotach mobilnych jest napęd różnicowy, składający się z dwóch niezależnie sterowanych kół napędowych umieszczonych po przeciwnych stronach platformy oraz jednego lub więcej elementów podparcia biernego. Ruch robota wyposażonego w taki napęd może być opisany za pomocą zestawu równań kinematycznych, w których prędkość liniowa oraz prędkość kątowa platformy są bezpośrednią funkcją prędkości obrotowych lewego i prawego koła. Model ten cechuje się prostotą formalną oraz intuicyjnością, co czyni go szczególnie przydatnym w projektach dydaktycznych oraz na wczesnych etapach prototypowania[10].

Formalny opis kinematyki robota z napędem różnicowym prowadzi do nielinowego układu równań, w którym pozycja robota opisywana jest zazwyczaj za pomocą współrzędnych płaskich oraz kąta orientacji względem przyjętego układu odniesienia.

Nieliniowość modelu wynika z faktu, że orientacja robota wpływa bezpośrednio na kierunek jego ruchu postępowego. W praktyce oznacza to konieczność ciągłego uwzględniania aktualnego położenia i orientacji robota podczas sterowania ruchem, nawet w przypadku realizacji prostych manewrów, takich jak jazda po łuku lub zmiana kierunku jazdy[21].

Istotnym ograniczeniem wynikającym z kinematyki napędu różnicowego jest brak możliwości realizacji ruchu bocznego. Robot nie jest w stanie przemieszczać się w kierunku prostopadłym do osi swojej orientacji bez wcześniejszego wykonania obrotu. W literaturze ograniczenia tego typu określone są mianem ograniczeń nieholonomicznych i mają one istotny wpływ na projektowanie algorytmów sterowania oraz nawigacji[22]. Ograniczenia nieholonomiczne powodują, że trajektorie ruchu robota muszą spełniać określone warunki ciągłości i gładkości, co komplikuje planowanie ruchu w środowiskach o ograniczonej przestrzeni.

Konsekwencją występowania ograniczeń nieholonomicznych jest konieczność stosowania odpowiednich strategii manewrowania, szczególnie podczas omijania przeszkód. W prostych systemach autonomicznych często wykorzystuje się sekwencje ruchów elementarnych, takich jak jazda do przodu, obrót w miejscu oraz cofanie, które w połączeniu umożliwiają realizację bardziej złożonych manewrów. Jak podkreśla Thrun, takie podejście jest wystarczające w środowiskach o niewielkiej złożoności i stanowi rozsądny kompromis pomiędzy prostotą implementacji a funkcjonalnością systemu[15].

W praktycznych realizacjach robotów mobilnych model kinematyczny wykorzystywany jest głównie pośrednio, jako podstawa do projektowania algorytmów sterowania niskiego poziomu. Zamiast ciągłego rozwiązywania równań ruchu stosuje się sterowanie dyskretne, w którym decyzje podejmowane są w kolejnych krokach czasowych na podstawie aktualnych danych sensorycznych. Takie podejście jest szczególnie korzystne w systemach opartych na mikrokontrolerach o ograniczonej mocy obliczeniowej, gdzie kluczowe znaczenie ma krótki czas reakcji oraz stabilność działania[23].

W literaturze podkreśla się również, że w robotach edukacyjnych dokładność modelu kinematycznego ma często znaczenie drugorzędne w porównaniu z jego czytelnością i łatwością interpretacji. Błędy wynikające z posłizgu kół, nierówności podłoża czy niedokładności wykonania mechanicznego są zazwyczaj akceptowalne, o ile algorytm sterowania wykazuje odpowiednią odporność na zakłócenia[24]. Z tego względu w wielu konstrukcjach rezygnuje się z enkoderów oraz sprzężenia zwrotnego pozycji

na rzecz prostych algorytmów otwartej pętli, opartych na czasie trwania ruchu oraz zadanej prędkości.

Ograniczenia kinematyczne robotów kołowych mają również bezpośredni wpływ na sposób planowania trajektorii. W systemach o wyższym stopniu zaawansowania stosuje się algorytmy planowania ruchu uwzględniające nieholonomiczność, takie jak planowanie w przestrzeni konfiguracyjnej czy metody oparte na krzywych Dubinsa. Jak zauważa LaValle, metody te wymagają precyzyjnego modelu robota oraz znacznych zasobów obliczeniowych, co w praktyce ogranicza ich zastosowanie w prostych platformach mobilnych[25].

Z tego względu w systemach o charakterze edukacyjnym i prototypowym często rezygnuje się z planowania globalnego na rzecz lokalnych algorytmów decyzyjnych, bazujących na bieżących pomiarach sensorycznych. Podejście to, mimo że nie gwarantuje optymalności trajektorii, umożliwia skuteczne poruszanie się robota w nieznanym środowisku i jest zgodne z ograniczeniami sprzętowymi oraz obliczeniowymi prostych platform mobilnych.

2.4. Układy napędowe i sterowanie silnikami

Układ napędowy robota mobilnego stanowi podstawowy element warstwy wykonawczej systemu, bezpośrednio odpowiedzialny za realizację poleceń generowanych przez warstwę decyzyjną. Jego zadaniem jest przekształcenie sygnałów sterujących w rzeczywisty ruch platformy przy zachowaniu możliwie wysokiej sprawności energetycznej, stabilności działania oraz przewidywalnego zachowania robota. W praktycznych realizacjach robotów mobilnych, najczęściej stosowane są silniki prądu stałego z magnesami trwałymi, które łączą prostotę sterowania z korzystnymi parametrami mechanicznymi i niewielkimi wymaganiami sprzętowymi.

Silniki prądu stałego charakteryzują się w przybliżeniu liniową zależnością momentu obrotowego od prądu twornika oraz zależnością prędkości obrotowej od napięcia zasilania. Właściwości te umożliwiają stosunkowo łatwą regulację prędkości obrotowej bez konieczności stosowania złożonych algorytmów regulacji. W robotach mobilnych silniki te zazwyczaj współpracują z przekładniami redukcyjnymi, które zwiększą dostępny moment obrotowy kosztem prędkości.

Regulacja prędkości obrotowej silników prądu stałego realizowana jest najczęściej przy wykorzystaniu modulacji szerokości impulsu PWM (Pulse Width Modu-

lation). Metoda ta polega na okresowym załączaniu i wyłączaniu napięcia zasilającego silnika z ustaloną częstotliwością oraz zmiennym współczynnikiem wypełnienia. Ze względu na indukcyjność uzwojeń silnika oraz jego bezwładność mechaniczną sygnał PWM postrzegany jest jako napięcie o wartości średniej proporcjonalnej do wypełnienia impulsu. Jak podkreślają Horowitz i Hill, modulacja PWM umożliwia efektywną regulację prędkości przy minimalnych stratach mocy, co czyni ją standardowym rozwiązaniem w systemach sterowania napędami niskonapięciowymi[8].

Istotnym zagadnieniem w sterowaniu napędem robota mobilnego jest możliwość zmiany kierunku obrotów silnika. Wymaga to zastosowania układu umożliwiającego odwrócenie polaryzacji napięcia zasilającego, co najczęściej realizowane jest za pomocą mostka H. Układ ten składa się z zestawu elementów przełączających, takich jak tranzystory bipolarne lub MOSFET², które pozwalają na sterowanie kierunkiem przepływu prądu przez silnik. W rozwiązańach edukacyjnych powszechnie stosowane są scalone układy mostków H, integrujące elementy mocy oraz podstawowe zabezpieczenia w jednej obudowie.

Jednym z najczęściej wykorzystywanych układów tego typu jest L293D, który umożliwia niezależne sterowanie czterema silnikami prądu stałego. Układ ten pozwala na realizację zmiany kierunku obrotów poprzez odpowiednie stany logiczne na wejściach sterujących, natomiast regulacja prędkości realizowana jest przez doprowadzenie sygnału PWM do wejścia aktywującego. Takie rozwiązanie jest szczególnie korzystne, gdyż umożliwia prostą i czytelną implementację sterowania ruchem bez konieczności projektowania własnych układów mocy.

W robotach mobilnych o niewielkich rozmiarach istotne znaczenie ma również sposób zasilania układu napędowego. Silniki prądu stałego charakteryzują się stosunkowo dużymi prądami rozruchowymi, które mogą prowadzić do chwilowych spadków napięcia oraz zakłóceń pracy układów logicznych. Z tego względu w literaturze zaleca się separację zasilania części wykonawczej i logicznej lub stosowanie odpowiednich filtrów oraz kondensatorów buforujących[26]. W prostych konstrukcjach edukacyjnych zagadnienie to bywa często pomijane, jednak jego uwzględnienie znacząco poprawia stabilność i niezawodność całego systemu.

²MOSFET - podstawowa technologia produkcji większości półprzewodników z izolowaną bramką stosowanych w komputerach.

2.4.1. Sterowanie silnikami robota

Sterowanie silnikami w robotach mobilnych może być realizowane zarówno w pętli otwartej, jak i zamkniętej. W systemach edukacyjnych i prototypowych dominującym rozwiązaniem jest sterowanie w pętli otwartej, w którym prędkość oraz czas trwania ruchu ustalane są na podstawie zadanych wartości sygnału PWM. Brak sprzężenia zwrotnego z enkoderów oznacza, że system nie kompensuje odchyłek wynikających z poślizgu kół, nierówności podłoża czy różnic parametrów silników. Jak zauważają Horowitz i Hill, w wielu zastosowaniach tego typu uproszczenie jest w pełni akceptowalne i nie wpływa istotnie na funkcjonalność systemu [8].

W bardziej zaawansowanych konstrukcjach stosuje się enkodery obrotowe, które umożliwiają pomiar prędkości oraz kąta obrotu kół. Pozwala to na implementację regulatorów prędkości, najczęściej typu PID, oraz na dokładniejsze sterowanie ruchem robota. Należy jednak podkreślić, że zastosowanie sprzężenia zwrotnego wiąże się ze wzrostem złożoności zarówno sprzętowej, jak i programowej, co może być niepożądane w projektach o charakterze dydaktycznym.

Z punktu widzenia architektury systemowej układ napędowy stanowi interfejs pomiędzy warstwą decyzyjną a fizycznym ruchem robota. Prostota jego realizacji ma istotne znaczenie dla przejrzystości projektu oraz możliwości dalszej rozbudowy systemu. W konstrukcjach opartych na mikrokontrolerach, takich jak Arduino, wykorzystanie standardowych bibliotek do obsługi PWM oraz gotowych sterowników silników pozwala skoncentrować się na zagadnieniach algorytmicznych, zamiast na niskopoziomowych detaliach sprzętowych.

2.5. Percepcja otoczenia

Percepcja otoczenia stanowi jeden z fundamentalnych elementów autonomii robota mobilnego, gdyż to na jej podstawie system podejmuje decyzje dotyczące ruchu oraz reakcji na zmieniające się warunki środowiskowe. W przeciwieństwie do systemów sterowanych zdalnie, robot autonomiczny musi samodzielnie interpretować sygnały pochodzące z czujników i przekształcać je w działania zgodne z przyjętą strategią sterowania. W literaturze podkreśla się, że jakość procesu percepcji ma bezpośredni wpływ na bezpieczeństwo oraz skuteczność poruszania się robota, niezależnie od stopnia zaawansowania zastosowanych algorytmów decyzyjnych[7].

Wśród czujników wykorzystywanych w systemach reaktywnych szczególnie szerokie zastosowanie znajdują czujniki ultradźwiękowe. Ich zasada działania polega na emisji krótkiego impulsu fali akustycznej o wysokiej częstotliwości oraz pomiarze czasu, po którym sygnał powraca do odbiornika po odbiciu od przeszkody. Znając prędkość rozchodzenia się dźwięku w powietrzu, możliwe jest wyznaczenie odległości do obiektu.

Badania porównawcze czujników odległości wskazują, że czujniki ultradźwiękowe zapewniają wystarczającą dokładność pomiaru dla podstawowych zadań unikania przeszkód, szczególnie w zakresie odległości od kilkunastu do kilkudziesięciu centymetrów[12]. Skuteczność tych czujników zależy jednak w dużym stopniu od właściwości powierzchni odbijającej falę akustyczną. Powierzchnie miękkie, porowate lub ustawione pod znacznym kątem względem osi pomiaru mogą powodować osłabienie lub rozproszenie sygnału odbitego, prowadząc do błędnych odczytów. Na dokładność pomiarów wpływają również warunki środowiskowe, takie jak temperatura powietrza czy obecność zakłóceń akustycznych.

Z tego względu istotnym elementem przetwarzania danych sensorycznych jest odpowiednia filtracja wyników pomiarów. W prostych systemach stosuje się najczęściej uśrednianie kilku kolejnych odczytów lub odrzucanie wartości skrajnych, które mogą być skutkiem chwilowych zakłóceń. Zaleca się także wprowadzanie progów decyzyjnych, zwiększających odporność algorytmu sterowania na pojedyncze błędne pomiary[13]. Przykładowo, decyzja o zmianie kierunku ruchu może być podejmowana dopiero po kilkukrotnym potwierdzeniu obecności przeszkody w określonym zakresie odległości.

W bardziej rozbudowanych rozwiązaniach możliwe jest łączenie danych pochodzących z kilku czujników odległości, rozmieszczonych w różnych kierunkach wokół robota. Takie podejście pozwala na uzyskanie bardziej kompletnej informacji o otoczeniu bez konieczności stosowania zaawansowanych czujników. Integracja wielu źródeł danych sensorycznych, określana mianem fuzji sensorycznej, może istotnie zwiększyć niezawodność procesu percepcji, jednak wiąże się ze wzrostem złożoności algorytmów decyzyjnych oraz wymaga starannego doboru parametrów czasowych.

W kontekście mikrokontrolerów, takich jak Arduino UNO, przetwarzanie danych sensorycznych musi być realizowane w sposób efektywny czasowo. Pomiar odległości z wykorzystaniem czujnika ultradźwiękowego wymaga precyzyjnego odmierzania czasu trwania impulsu powrotnego, co realizowane jest z wykorzystaniem liczników sprzętowych lub mechanizmów przerwań. Odpowiednia organizacja programu, obejmująca

cykliczny odczyt czujników oraz sekwencyjne podejmowanie decyzji, ma kolosalne znaczenie dla zapewnienia reakcji robota w czasie zbliżonym do rzeczywistego.

Należy również zwrócić uwagę na kompromis pomiędzy częstotliwością odczytu czujników a stabilnością działania algorytmu sterowania. Zbyt częste pomiary mogą prowadzić do nadmiernej liczby decyzji ruchowych i niestabilności toru jazdy, natomiast zbyt rzadkie odczyty zwiększały ryzyko kolizji z przeszkodami. W literaturze podkreśla się konieczność doboru parametrów czasowych w sposób dostosowany do prędkości robota oraz dynamiki środowiska pracy.

2.6. Zaawansowane metody nawigacji

Postęp w zakresie mocy obliczeniowej systemów wbudowanych oraz rosnąca dostępność zaawansowanych czujników pomiarowych przyczyniły się do dynamicznego rozwoju metod lokalizacji i nawigacji robotów mobilnych. W odróżnieniu od podejść reaktywnych, które bazują wyłącznie na bieżących pomiarach sensorycznych, metody zaawansowane zakładają istnienie wewnętrznej reprezentacji środowiska oraz ciągłą estymację położenia robota w czasie. Centralnym zagadnieniem w tym obszarze jest problem jednoczesnej lokalizacji i mapowania, znany w literaturze jako SLAM (Simultaneous Localization and Mapping).

Algorytmy SLAM umożliwiają stopniowe tworzenie mapy nieznanego środowiska przy jednoczesnym określaniu pozycji robota względem tej mapy. Złożoność tego problemu wynika z faktu, że zarówno dane sensoryczne, jak i modele ruchu obarczone są niepewnością, której kumulacja prowadzi do narastania błędów lokalizacji. Jak podkreśla Thrun, SLAM stanowi jedno z fundamentalnych wyzwań współczesnej robotyki mobilnej, łącząc elementy probabilistyczne, estymacji stanu oraz sztucznej inteligencji [15].

W praktycznych implementacjach algorytmy SLAM wykorzystują dane pochodzące z wielu źródeł sensorycznych. Najczęściej stosowane są czujniki LiDARowe, dostarczające precyzyjnych informacji o geometrii otoczenia, kamery wizyjne umożliwiające analizę cech wizualnych środowiska oraz jednostki inercyjne, pozwalające na estymację przyspieszeń i prędkości kątowych robota. Fuzja danych z różnych czujników zwiększa dokładność lokalizacji oraz poprawia odporność systemu na zakłócenia lub chwilową utratę sygnału z jednego z kanałów pomiarowych[3].

W literaturze wyróżnia się kilka klas algorytmów SLAM, różniących się sposobem reprezentacji niepewności oraz złożonością obliczeniową. Do najwcześniej stosowanych należą metody oparte na filtrze Kalmana oraz jego rozszerzeniach, takich jak rozszerzony filtr Kalmana (EKF), które znajdują zastosowanie głównie w systemach o ograniczonej liczbie stanów. Algorytmy cząsteczkowe, w tym FastSLAM, umożliwiają lepsze odwzorowanie nieliniowości oraz wielomodalności rozkładów prawdopodobieństwa, jednak wiążą się ze znacznym wzrostem zapotrzebowania na moc obliczeniową. W nowoczesnych systemach autonomicznych coraz częściej stosuje się metody grafowe, charakteryzujące się wysoką dokładnością estymacji, kosztem złożonej implementacji oraz dużych wymagań sprzętowych.

Poza lokalizacją i mapowaniem istotnym elementem zaawansowanej nawigacji jest planowanie ruchu. W systemach dysponujących mapą środowiska wykorzystuje się algorytmy planowania globalnego, których zadaniem jest wyznaczenie trajektorii pomiędzy pozycją początkową a celem. Do najczęściej stosowanych należą algorytmy grafowe, takie jak Dijkstra czy A*, a także algorytmy losowe, w tym RRT (Rapidly-exploring Random Trees). Planowanie globalne umożliwia efektywne omijanie przeszkód oraz minimalizację długości trajektorii, jednak jego skuteczność jest ściśle uzależniona od jakości mapy oraz dokładności lokalizacji robota[3].

Uzupełnieniem planowania globalnego są algorytmy planowania lokalnego, odpowiedzialne za reagowanie na dynamiczne zmiany w otoczeniu, takie jak pojawienie się przeszkód ruchomych lub nieprzewidziane zakłócenia. W praktycznych systemach autonomicznych często stosuje się architektury hybrydowe, w których plan globalny stanowi punkt odniesienia, natomiast algorytmy lokalne realizują bieżącą korektę ruchu. Takie podejście zwiększa skuteczność działania robota w złożonych środowiskach, lecz jednocześnie prowadzi do znacznego wzrostu złożoności całego systemu sterowania.

Pomimo licznych zalet, implementacja zaawansowanych metod lokalizacji i nawigacji napotyka na istotne ograniczenia. Algorytmy SLAM wymagają nie tylko znacznych zasobów obliczeniowych, lecz także precyzyjnych czujników oraz starannej kalibracji, co może utrudniać analizę działania systemu i przesłaniać podstawowe cele dydaktyczne. Jak zauważa Thrun, nadmierna złożoność rozwiązań może ograniczać ich przydatność w kontekście nauczania podstaw robotyki mobilnej[15].

Z tego względu często świadomie rezygnuje się z implementacji pełnych algorytmów SLAM na rzecz uproszczonych metod nawigacji. Takie podejście umożliwia kon-

centrację na zagadnieniach integracji czujników, projektowania algorytmów sterowania ruchem oraz eksperymentalnej analizie zachowania robota w rzeczywistym środowisku. Jednocześnie znajomość zaawansowanych metod lokalizacji i nawigacji stanowi istotne zaplecze teoretyczne, pozwalające właściwie ocenić możliwości oraz ograniczenia prostych rozwiązań stosowanych w systemach o ograniczonych zasobach.

3. Część Praktyczna/fizyczna - zmień nazwę

4. Część Praktyczna/kodowa - zmień nazwę

Literatura

- [1] International Federation of Robotics: World Robotics 2023 – Service Robots. IFR, Frankfurt am Main, 2023.
- [2] Monk S.: Programming Arduino: Getting Started with Sketches. McGraw-Hill Education, New York, 2016.
- [3] Siegwart R., Nourbakhsh I. R., Scaramuzza D.: Introduction to Autonomous Mobile Robots. MIT Press, Cambridge, 2011.
- [4] Arduino: Arduino Uno Rev3 – Technical Documentation. Arduino SA, 2022.
- [5] Cytron Technologies: HC-SR04 Ultrasonic Sensor User Manual. Dokumentacja techniczna, 2013.
- [6] Adafruit Industries: Adafruit Motor Shield. Dokumentacja techniczna, 2018.
- [7] Jones J. L., Flynn A. M.: Mobile Robots: Inspiration to Implementation. A K Peters, Natick, 1999.
- [8] Horowitz P., Hill W.: The Art of Electronics. Cambridge University Press, Cambridge, 2015.
- [9] Alimisis D.: Educational Robotics: Open Questions and New Challenges. Themes in Science and Technology Education, Vol. 6, No. 1, 2013, s. 63–71.
- [10] Campion G., Bastin G., D'Andrea-Novel B.: Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. IEEE Transactions on Robotics and Automation, Vol. 12, No. 1, 1996, s. 47–62.
- [11] Arkin R. C.: Behavior-Based Robotics. MIT Press, Cambridge, 1998.
- [12] Kurniawan A., Hadiyoso S.: Performance Comparison of Ultrasonic Sensors for Obstacle Detection. International Journal of Engineering Research, Vol. 7, No. 4, 2018, s. 210–215.
- [13] Welch G., Bishop G.: An Introduction to the Kalman Filter. University of North Carolina at Chapel Hill, 2006.

- [14] Durrant-Whyte H., Bailey T.: Simultaneous Localization and Mapping: Part I. IEEE Robotics & Automation Magazine, Vol. 13, No. 2, 2006, s. 99–110.
- [15] Thrun S., Burgard W., Fox D.: Probabilistic Robotics. MIT Press, Cambridge, 2005.
- [16] Borenstein J., Everett H. R., Feng L.: Where am I? Sensors and Methods for Mobile Robot Positioning. University of Michigan, 1996.
- [17] Borenstein J.: Control and kinematic design of multi-degree-of-freedom mobile robots. IEEE Transactions on Robotics and Automation, Vol. 14, No. 2, 1998, s. 315–322.
- [18] Raibert M.: Legged Robots That Balance. MIT Press, Cambridge, 1986.
- [19] Brooks R. A.: A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation, Vol. 2, No. 1, 1986, s. 14–23.
- [20] Siciliano B., Sciavicco L., Villani L., Oriolo G.: Robotics: Modelling, Planning and Control. Springer, London, 2010.
- [21] Canudas-de-Wit C., Siciliano B., Bastin G.: Theory of Robot Control. Springer, London, 1996.
- [22] Bloch A. M.: Nonholonomic Mechanics and Control. Springer, New York, 2003.
- [23] Kenjo T.: Stepping Motors and Their Microprocessor Controls. Oxford University Press, Oxford, 1994.
- [24] Eguchi A.: Educational Robotics for Promoting 21st Century Skills. Journal of Automation, Mobile Robotics & Intelligent Systems, Vol. 8, No. 1, 2014, s. 5–11.
- [25] LaValle S. M.: Planning Algorithms. Cambridge University Press, Cambridge, 2006.
- [26] Erickson R. W., Maksimović D.: Fundamentals of Power Electronics. Springer, New York, 2020.

STRESZCZENIE PRACY DYPLOMOWEJ INŻYNIERSKIEJ

ROBOT MOBILNY Z NAWIGACJĄ AUTONOMICZNĄ OPARTY NA ARDUINO

Autor: Kacper Rychel, nr albumu: EF-173701

Opiekun: Dr inż. Mariusz Mączka

Słowa kluczowe: Arduino Uno, Robot mobilny, Czujnik HC-SR04, Czujnik Ultradźwiękowy Odległości

Cel pracy: Stworzenie robota mobilnego, który potrafi omijać przeszkody i poruszać się autonomicznie w środowisku. Zakres pracy: zaprojektowanie platformy robota z silnikami i czujnikami ultradźwiękowymi do detekcji przeszkód; zaprogramowanie algorytmu omijania przeszkód i nawigacji w środowisku; możliwość rozbudowy o funkcje śledzenia wyznaczonego celu lub powrotu do bazy. Testy robota w różnych scenariuszach, takich jak omijanie przeszkód czy poruszanie się po określonej trasie.

BSC THESIS ABSTRACT

MOBILE ROBOT WITH AUTONOMOUS NAVIGATION BASED ON ARDUINO

Author: Kacper Rychel, Student's ID: EF-173701

Supervisor: Prof. Mariusz Mączka

Key words: Arduino Uno, Mobile robot, HC-SR04 sensor, Ultrasonic distance sensor

Objective: To create a mobile robot that can avoid obstacles and navigate autonomously within its environment. Scope of work: Designing a robot platform with motors and ultrasonic sensors for obstacle detection; programming an algorithm for obstacle avoidance and navigation within the environment; and the potential for expansion with target tracking or return-to-base functionality. Testing the robot in various scenarios, such as obstacle avoidance and navigating along a defined route.