XXIX Seminarium

ZASTOSOWANIE KOMPUTERÓW W NAUCE I TECHNICE 2019

Oddział Gdański PTETiS

doi: 10.32016/1.68.06

MODEL KONSTRUKCYJNY I STEROWANIE ROBOTAMI MOBILNYMI W ŚRODOWISKU Z PRZESZKODAMI

Bartosz MACIĄG, Wojciech MAKOHOŃ, Krzysztof MILEWSKI, Robert PIOTROWSKI

Politechnika Gdańska, Wydział Elektrotechniki i Automatyki,
e-mail: maciagbartosz96@gmail.com; wojtek.makohon@gmail.com; milewskik51@gmail.com; robert.piotrowski@pg.edu.pl

Streszczenie: W dzisiejszych czasach mobilne roboty autonomiczne zyskują coraz większą popularność, zarówno w środowiskach przemysłowych jak i w konsumenckich. Mają one głównie na celu ułatwianie pracy człowieka poprzez przejęcie części jego obowiązków. Z racji różnorodności środowisk, w jakich mogą one pracować, korzystają z algorytmów omijania przeszkód w celu osiągnięcia swobody poruszania się. Artykuł przedstawia model konstrukcyjny oraz wykonanie dwóch autonomicznych robotów mobilnych wraz z planszą, na której były przeprowadzane testy. Następnie po zmontowaniu pojazdów należało je zaprogramować. W tym celu opracowano algorytm omijania przeszkód, oparty na poszukiwaniu bezkolizyjnej ścieżki na podstawie obliczania pochodnej z uśrednionych wartości pomiarów. Dodatkowo, w celu prostej zmiany parametrów działającego algorytmu, opracowano aplikację mobilną na urządzenia z systemem Android. Na koniec przeprowadzono badania testowe oraz dokonano analizy uzyskanych wyników.

Słowa kluczowe: mechatronika, robot mobilny, układ regulacji.

1. WPROWADZENIE

Pojazdy autonomiczne coraz częściej znajdują zastosowanie w różnych dziedzinach życia, poprzez zastąpienie lub wspomaganie człowieka w ciężkich, żmudnych, monotonnych i powtarzalnych pracach. Podstawowe cechy definiujące autonomiczny pojazd to między innymi: niezależność w podejmowaniu decyzji, odbieranie informacji ze środowiska za pomocą różnych urządzeń pomiarowych i przetwarzanie ich oraz oddziaływanie na środowisko. Pojazdy takie wykorzystywane są między innymi w przemyśle [1], a także w zastosowaniach konsumenckich [2].

2. ZAŁOŻENIA PROJEKTOWE

Głównym celem projektu była budowa dwóch autonomicznych pojazdów poruszających się po ograniczonej powierzchni zawierającej statyczne przeszkody o różnym kształcie. Przeszkodami dynamicznymi miały być roboty.

Pojazdy musiały mieć małe rozmiary, charakteryzować się dużą zwrotnością oraz dynamiką. Konieczne było zastosowanie odpowiednich urządzeń pomiarowych pozwalających na zlokalizowanie przeszkód. Wybrana platforma sprzętowa powinna umożliwić implementację prostych, ale efektywnych algorytmów omijania przeszkód dostosowanych do niskobudżetowych konstrukcji.

Zdecydowano się na budowę pojazdów o napędzie różnicowym ze względu na brak konieczności zastosowania mechanizmu różnicowego.

3. MODEL KONSTRUKCYJNY ROBOTÓW MOBILNYCH

Część elektroniczna pojazdu została zaprojektowana za pomocą środowiska EAGLE CAD. Sercem układu był mikrokontroler Atmega 328P-PU wraz z komponentami wymaganymi do zapewnienia stabilnej pracy takimi jak: rezonator kwarcowy o częstotliwości 16 MHz, pasywne układy filtrujące napięcie, czy rezystor podciągający wejście resetujące. Możliwości sprzętowe oferowane przez ten mikrokontroler (32 kB pamięci Flash, 23 linie wejścia/wyjścia, 6 kanałów PWM oraz 6 kanałów 10-bitowego przetwornika A/C) w zupełności wystarczyły do przetwarzania danych i realizacji zadanych algorytmów sterowania [3].

Kolejną istotną częścią projektu było zasilanie. Zdecydowano się zastosować akumulatory litowo-polimerowe o pojemności 520 mAh, napięciu znamionowym 7,4 V oraz wydajności prądowej 25 C. Akumulatory te charakteryzują się dużą wydajnością przy stosunkowo małej masie i rozmiarach.

W pojazdach wykorzystano silniki firmy Pololu z serii *micro low power*. Cechują się one małymi rozmiarami, niedużym poborem prądu oraz stosunkowo dużą mocą. Zastosowane silniki posiadały przekładnię mechaniczną 30:1, dzięki czemu charakteryzują się one większym momentem obrotowym, tym samym zapewniając dużą dynamikę pojazdu. Silniki zostały wyposażone w enkodery służące do przetwarzania ilości obrotów wału silnika na impulsy elektryczne odczytywane przez mikrokontroler. Dzięki temu zyskano możliwość sterowania prędkością obu silników (a tym samym prędkością i kierunkiem ruchu całego pojazdu) w pętli zamkniętej. Silniki były sterowane za pomocą sterownika silników opartego na układzie podwójnego mostka H.

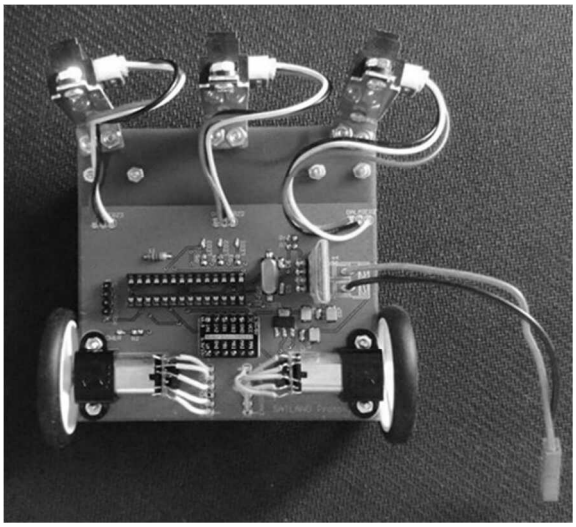
W celu zapewnienia możliwości odbierania informacji o otaczającym środowisku, a tym samym lokalizacji przeszkód, zastosowano analogowe czujniki optyczne firmy Sharp. Urządzenia te są często wykorzystywane przez konstruktorów robotów mobilnych o różnych przeznaczeniach. Zastosowane czujniki pozwalają na wykrywanie obiektów w zakresie 4 – 30 cm.

Ostatnim istotnym elementem projektu jest zapewnienie możliwości komunikacji. W tym celu

zastosowano moduł Bluetooth HC-06, który pozwala na połączenie z urządzeniem mobilnym. W rozdziale 5 opisano proces tworzenia aplikacji mobilnej, która pełniła rolę sterowania nadzorczego.

Oprogramowanie robotów oparto na platformie Arduino. Jest to ustandaryzowana platforma programistyczna przeznaczona dla mikrokontrolerów, zawierająca szereg funkcjonalności i gotowych bibliotek ułatwiających ich programowanie [4].

Kolejnym etapem projektu był montaż robotów. Wybrane części należało umieścić na zaprojektowanej płytce PCB, a następnie sprawdzić działanie robotów w poszukiwaniu ewentualnych błędów konstrukcji związanych z odpowiednim zamontowaniem lub zaprojektowaniem poszczególnych części. Efekt końcowy procesu montażu przedstawiający widok robota z góry został umieszczony na rys. 1.



Rys. 1. Widok robota z góry

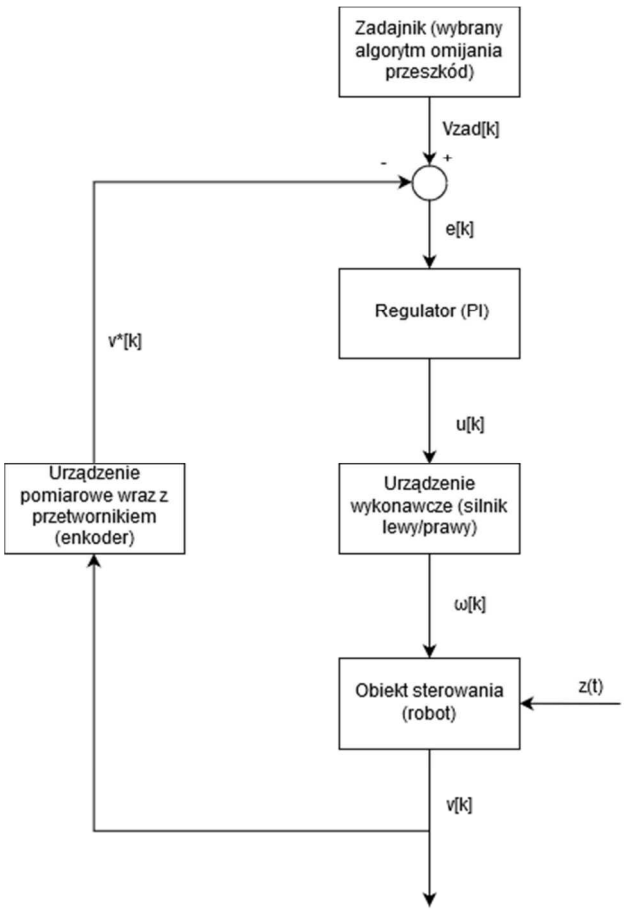
Zdecydowano, aby środowisko, w którym będą poruszać się roboty było ograniczone. Taka decyzja pozwala na lepszą kontrolę pracy urządzeń oraz eliminację części zakłóceń zewnętrznych, takich jak zmienność faktury powierzchni, jej równość, gładkość czy zabrudzenia. W tym celu zbudowano planszę stanowiącą środowisko z przeszkodami do badania działania algorytmu. Ma ona pole powierzchni równe 1,5625 m^2, co zapewnia wystarczająco dużą przestrzeń nieutrudniającą badania zachowania robotów. Przeszkody statyczne stanowiły elementy przestrzenne, różnych kształtów i wielkości, a przeszkodami dynamicznymi były poruszające się roboty.

4. ALGORYTMY STEROWANIA

Strukturę sterowania podzielono na dwie części. Pierwszy, niskopoziomowy moduł odpowiedzialny był za realizację wartości zadanej prędkości oraz kierunku dla obu silników. Zastosowano klasyczny regulator PI o nastawach dobranych w sposób eksperymentalny. Regulator ten zapewniał dużą dynamikę prędkości silnika i eliminację uchybu w stanie ustalonym [5].

Drugi, wysokopoziomowy moduł odpowiedzialny był za realizację algorytmów omijania przeszkód na podstawie odczytów z dalmierzy i wypracowanie wartości zadanej prędkości dla silników realizowanych przez warstwę niskopoziomową.

Strukturę sterowania zastosowaną w obu pojazdach pokazano na rys. 2.



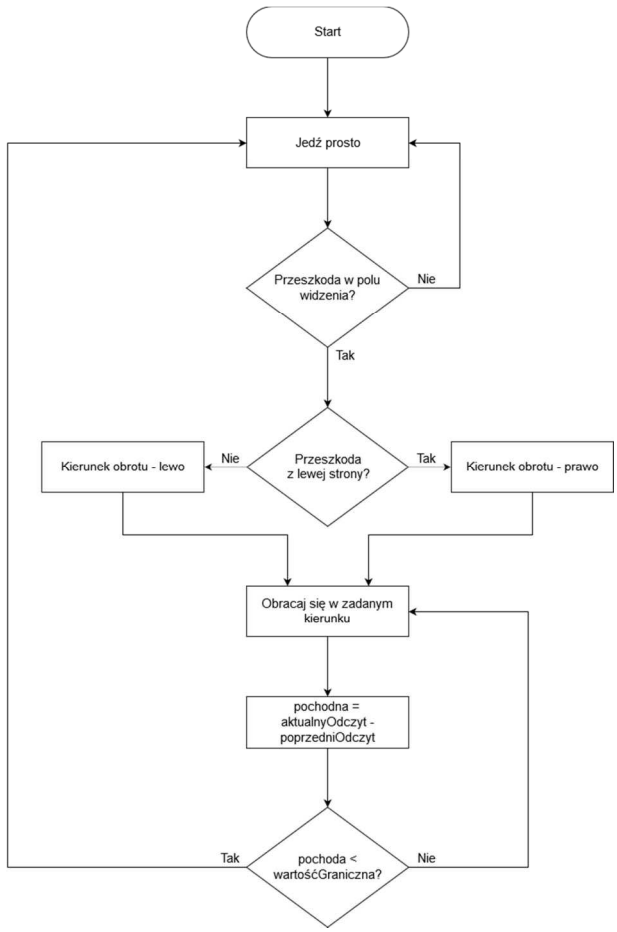
Rys. 2. Schemat struktury sterowania pojazdem

gdzie:

$V_{zad}[k]$ – zadana prędkość silników, wypracowana przez nadrzędny algorytm omijania przeszkód,
e[k] – sygnał uchybu,
u[k] – sygnał sterujący silnikami (wypełnienie sygnału PWM),
ω[k] – prędkość obrotowa silników,
z(t) – sygnały zakłócające,
v[k] – prędkość liniowa całego robota,
v*[k] – prędkość liniowa robota po przetworzeniu przez enkoder.

Jednym z dwóch zaimplementowanych algorytmów był algorytm nazwany Derivative Search. Zasada jego działania polegała na poszukiwaniu kierunku jazdy poprzez obliczanie pochodnej ze średniej arytmetycznej pomiarów odległości. Robot porusza się prosto, dopóki w jego polu nie znajdzie się przeszkoda. W momencie jej wykrycia zaczyna obracać się w miejscu i skanować otoczenie, obliczając przy tym różnicę pomiędzy aktualną, a poprzednią uśrednioną wartością odczytu z dalmierzy. Skanowanie zostaje zakończone w momencie uzyskania odpowiednio małej różnicy (pochodnej), co jest równoznaczne z odnalezieniem najdłuższej bezkolizyjnej ścieżki. Warto zwrócić uwagę, że algorytm w przedstawionej postaci kończy swoje działanie w momencie odnalezienia najdłuższej bezkolizyjnej ścieżki, ale z matematycznego punktu widzenia, jest ona ekstremum lokalnym. Odnaleziona ścieżka niekoniecznie jest najdłuższą ścieżką w sensie globalnym Podeście takie sprawia, że

algorytm działa szybciej, a ruch robota jest płynny. Schemat blokowy algorytmu pokazano na rys. 3.



Rys. 3. Schemat blokowy algorytmu Derivative Search

Wymienione algorytmy zostały zaimplementowane w języku C++. Dzięki temu wykorzystano zalety programowania obiektowego do stworzenia czytelnego kodu, łatwego w zarządzaniu i rozbudowie, a także rozdzieleniu odpowiedzialności na pojedyncze moduły oprogramowania [6].

W celu implementacji algorytmów omijania przeszkód stworzono klasę abstrakcyjną o nazwie ObstacleAvoidanceAlgorithm, której kod przedstawiono na listingu 1.

Listing 1. Kod klasy ObstacleAvoidanceAlgorithm

```
{
protected:
bool obstacleDetected;
float hysteresis;
float boundary;
float sensorReadings[3];
bool checkForObstacles();virtual void
setLEDs(Result &r) = 0;
public:
ObstacleAvoidanceAlgorithm();
void GetDistances
       (float left, float middle, float right);
void SetBoundary(float newBoundary);
void SetHysteresis(float newHysteresis);
float GetBoundary();
float GetHysteresis();
virtual Result Run() = 0;
};
```

Klasa ta zawiera deklaracje pól oraz metod wykorzystywanych przez wszystkie algorytmy omijania przeszkód. Są to między innymi: metoda wykonująca jedną iterację algorytmu, metoda przetwarzająca informacje otrzymane z dalmierzy, metoda ustawiająca odpowiedni stan diod LED służących do sygnalizacji, a także pola określające zasięg widzenia robota oraz szerokość pętli histerezy przy przechodzeniu pomiędzy stanami braku przeszkody i wykrycia przeszkody. Dzięki zastosowaniu histerezy wyeliminowano wpływ szumów pomiarowych z dalmierzy, który powodował oscylacje pomiędzy stanami w przypadku, gdy przeszkoda znajdowała się na granicy pola widzenia. Metody służące do zmian tych parametrów zostały stworzone z myślą o użyciu ich z poziomu aplikacji SCADA (ang. *Supervisory Control And Data Acquisition*).

Dzięki zastosowanej architekturze oprogramowania implementacja kolejnych, dodatkowych algorytmów jest bardzo prosta, a nowy moduł oprogramowania nie wpływa na istniejący i przetestowany kod. Implementacja nowego algorytmu omijania przeszkód wygląda następująco:

1. Stworzenie nowej klasy dziedziczącej po klasie ObstacleAvoidanceAlgorithm,
2. Przesłonięcie metod wirtualnych,
3. Stworzenie dowolnej ilości pól i metod pomocniczych właściwych dla danego algorytmu.

5. APLIKACJA MOBILNA

Kolejnym etapem, po zaimplementowaniu algorytmów omijania przeszkód w robotach, było stworzenie aplikacji mobilnej, która pozwalałaby na nadrzędne sterowanie poprzez możliwość włączania i wyłączania robota oraz zmianę aktualnie działającego algorytmu omijania przeszkód i jego kluczowych parametrów. Komunikacja pomiędzy telefonem, a robotem odbywa się poprzez standard Bluetooth. Do zaprojektowania graficznego interfejsu użytkownika oraz stworzenia odpowiedniego programu aplikacji wykorzystano środowisko MIT App Inventor. Jest to oprogramowanie stworzone przez studentów oraz pracowników Massachusetts Institute of Technology pozwalające na łatwe i szybkie projektowanie aplikacji mobilnych na urządzenia z systemem Android.

Następnie zintegrowano aplikację i wysyłane przez nią komunikaty z oprogramowaniem robota, co umożliwiło przetwarzanie informacji w celu uzyskania pożądanego efektu. Mając na uwadze potencjalną rozbudowę w przyszłości, zapewniono możliwość obsługi wielu mediów komunikacyjnych przez jeden moduł oprogramowania. W tym celu stworzono klasę o nazwie Communication, korzystającą z mechanizmu wstrzykiwania zależności (ang. *Dependency Injection*). Klasa ta korzysta z abstrakcyjnego wskaźnika typu *Stream* – bazowego typu dla wszystkich rodzajów mediów korzystających z szeregowej transmisji danych. Klasa ta zawiera zestaw metod niezbędnych do dwukierunkowej komunikacji z systemem sterowania nadzorczego. Jej kod przedstawiono na listingu 2.

Listing 2. Kod klasy Communication

```
{
private:
Stream *stream;
public:
String ReadMessage();
bool Available();
void SendMessage(String msg);
void Begin(Stream *str);
};
```

6. BADANIA TESTOWE

Zdecydowano się przetestować działanie algorytmu omijania przeszkód w trzech różnych środowiskach: tylko z przeszkodami statycznymi, tylko z przeszkodami dynamicznymi oraz w środowisku z dwoma rodzajami przeszkód. Każdorazowo próby trwały po dziesięć minut. Sprawdzano również wpływ zmiany wartości parametrów granicy wykrywania przeszkód oraz histerezy.

Podczas badania algorytmu omijania z przeszkodami statycznymi robotowi czasem zdarzało się zderzać z przedmiotami w wyniku bezpośredniego najechania na nie, nie były to jednak otarcia. Zauważalne było chwilowe przyspieszanie po „oderwaniu się" od najechanej przeszkody, co wynika z zastosowanego regulatora prędkości. W rejonach planszy, w których odległości między przeszkodami były mniejsze, pojazd często zatrzymywał się i poruszał zygzakiem. Zmniejszenie wartości granicy powodowało redukcję tych objawów, robot jeździł zdecydowanie bardziej płynnie i rzadziej korygował kierunek ruchu.

Działanie algorytmu w środowisku z przeszkodami dynamicznymi było satysfakcjonujące. Roboty zazwyczaj omijały się, do zderzeń dochodziło sporadycznie. Zmniejszenie wartości granicy wykrywania przeszkód ponownie powodowało bardziej płynne poruszanie się robotów, lecz nie wpływało znacznie na liczbę zderzeń.

W środowisku z przeszkodami mieszanymi robotom również zdarzało się uderzać w przeszkody. Czasami było to spowodowane zmianą kierunku ruchu po wykryciu zbliżającej się drugiej konstrukcji. Rzadziej dochodziło do kontaktów między samymi robotami. Tak samo jak w poprzednich przypadkach, zmniejszenie wartości parametrów wpływało na poprawę jakości działania algorytmu. Wyniki badań zostały przedstawione w tabeli 1.

Tabela 1. Wyniki badań dla algorytmu DerivativeSearch

Algorytm	Rodzaj przeszkód	Parametr	
		Pobór energii (różnica napięcia na akumulatorze) przez 10 minut działania [V]	Skuteczność omijania (liczba kontaktów z przeszkodą lub barierą)[-]
DerivativeSearch	Statyczne	0,10	6
	Dynamiczne	0,11	5
	Mieszane	0,11	11

Zaimplementowany algorytm omijania przeszkód spełnił swoje zadanie. Ograniczenia sprzętowe i dokładność urządzeń pomiarowych nie pozwalają na osiągnięcie bezbłędnych wyników podczas badań, aczkolwiek uzyskane działanie jest zadowalające. Częstym powodem kontaktu z przeszkodami są wystające poza obrys pojazdu koła lub nawet przewody łączące czujniki i akumulator, które zahaczając o obiekt sprawiają, że cała konstrukcja obraca się. Kolejnym powodem jest zakres pomiarowy urządzeń pomiarowych. Kiedy robot obracał się w odległości od przeszkody mniejszej niż około 4 cm, nie była ona wykrywana i konstrukcja uderzała w nią.

7. PODSUMOWANIE

Prace związane z zaprojektowaniem i wykonaniem robotów autonomicznych zdolnych do omijania przeszkód zakończyły się sukcesem. Zostały spełnione założenia projektowe.

Proponowaną kontynuacją prac może być próba implementacji kolejnych, bardziej zaawansowanych algorytmów omijania przeszkód, opartych na metodach optymalizacji np. CVM (ang. *Curvature Velocity Method*).

8. BIBLIOGRAFIA

1. Shuai Guo, Quzhuo Diao, Fengfeng Xi: Vision Based Navigation for Omni-directional Mobile Industrial Robot, 2016 IEEE International Symposium on Robotics and Intelligent Sensors, IRIS 2016, 17-20 December 2016, Tokyo, Japan.
2. David Bernstein, Remy Michaud, Brian Silvia,: Consumer Robotics: State of the industry and Public Opinion, Worcester Polytechnic institute, May 5, 2010.
3. Datasheet Atmega https://www.sparkfun.com/datasheets/Components/SMD/ATMega328.pdf.
4. Louis, L. Working principle of Arduino and using it as a tool for study and research. International Journal of Control, Automation, Communication and Systems. 2016, 1(2), 21-29. doi: 10.5121/ijcacs.2016.1203.
5. Tochukwu, U. Effects of PID Controller on a Closed Loop Feedback System. Proceedings of Conference: Ternopil National Technical University. Ukraine, November, 2013. doi:10.13140/2.1.2650.0167.
6. Urdhwareshe, A. Object-Oriented Programming and its Concepts. International Journal of Innovation and Scientific Research. volume 26, no. 1, pages 1–6, August 2016.

CONSTRUCTION MODEL AND CONTROL OF MOBILE ROBOTS IN AN ENVIRONMENT WITH OBSTACLES

Nowadays autonomous robots gain increasingly more popularity in both consumer and industrial applications. Their main goal is to support human's work. Due to high diversity of environments these robots work in, there is a need for utilizing obstacle avoidance algorithms which allow them for safe maneuvering. This paper presents a complete process of designing and assembling two autonomous robots as well as the board used for tests. One algorithm was proposed, which was based on calculating derivative of averaged sensor measurements. Additionally, various software development aspects are covered, including a mobile application for Android devices, which was used as a supervisory control. In the end, a series of performance tests were conducted, along with results analysis.

Keywords: mechatronics, mobile robot, control system.