

Grzegorz FICHT, Robert PIOTROWSKI
 WYDZIAŁ ELEKTROTECHNIKI I AUTOMATYKI, POLITECHNIKA GDAŃSKA
 ul. G. Narutowicza 11/12, 80-233 Gdańsk

Robot typu Micromouse – wykonanie, sterowanie i optymalizacja

Mgr inż. Grzegorz FICHT

Absolwent Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej (kierunek: Automatyka i Robotyka, 2013 r.). Zainteresowania naukowe dotyczą wykorzystania elektroniki mikroprocesorowej w procesach sterowania, robotów humanoidalnych i manipulatorów przemysłowych.



e-mail: ficht.grzegorz@gmail.com

Dr inż. Robert PIOTROWSKI

Absolwent Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej (kierunek: Automatyka i Robotyka) (2001r.). W roku 2005 uzyskał stopień doktora (Automatyka i Robotyka). Obecnie adiunkt w Katedrze Inżynierii Systemów Sterowania. Zainteresowania naukowe dotyczą modelowania i zaawansowanych metod sterowania procesami przemysłowymi.



e-mail: r.piotrowski@eia.pg.gda.pl

Streszczenie

Projektowanie robotów wymaga interdyscyplinarnej wiedzy obejmującej obszary mechaniki, elektroniki, metrologii, automatyki i informatyki. W artykule przedstawiono poszczególne etapy wykonania robota typu Micromouse. Jest to specyficzne urządzenie, którego zadaniem jest zbadać labirynt, dotarcie do jego środka, powrót do początkowej pozycji, a następnie ponowny przejazd przy wykorzystaniu optymalnej trasy. W artykule sformułowano założenia i wykonano projekt techniczny robota. Opracowano algorytm sterowania oraz przedstawiono algorytm rozrostu ziarna, służący do optymalizacji drogi przejścia przez labirynt. Przeprowadzono badania testowe zaprojektowanego robota.

Słowa kluczowe: robot autonomiczny, robotyka, optymalizacja, projektowanie, system sterowania.

Micromouse robot – construction, control and optimization

Abstract

In recent years, it was possible to see a rapid development of robotics. Robots are used in various branches of industries and everyday life. Robot designing requires interdisciplinary knowledge from areas such as mechanics, electronics, metrology, automation and informatics. The paper presents various stages of constructing a Micromouse robot. It is a specific device, whose task is to explore the maze, reach its center, return to the starting position and then repeat the process by using the optimum route. In the paper, the assumptions have been formulated and the technical design of the robot has been accomplished. Next, control algorithm has been proposed and implemented. The algorithm of grain growth, was used to optimize the crossing of the maze. At the end, studies test the robot are presented.

Keywords: autonomous robot, robotics, optimization, designing, control system.

1. Wstęp

Robot Micromouse jest mobilną platformą umożliwiającą testowanie algorytmów sterowania i optymalizacji w czasie rzeczywistym. W ostatnich latach dużą popularnością cieszą się zawody robotów tego typu, w których szuka się drogi wyjścia z labiryntu. Aktualne zasady przewidują labirynt o rozmiarze 16x16 komórek (18x18cm każda), w którym punkt docelowy znajduje się w jego środku. Możliwe jest dojechanie do celu różnymi ścieżkami – droga najkrótsza niekoniecznie jest drogą najszybszą. Zasady wyróżniają dwa rodzaje przejazdów: etap przeszukiwania (ang. search-run) oraz etap rozwiązania labiryntu (ang. speed-run). W pierwszym etapie robot wykonuje mapę labiryntu po czym wraca na jego początek i stara się go pokonać w drugim przejeździe, w jak najkrótszym czasie [1].

Prace dotyczące robotów typu Micromouse trwają od wielu lat, np. [2, 3, 4, 5, 6]. Dotyczą one ich budowy, programowania, sterowania i metod optymalizacji. Tego typu roboty są również bardzo popularne wśród hobbystów i amatorów robotyki.

Zdobyta wiedza może być użyteczna do rozwiązywania złożonych zagadnień, np. bezobsługowych magazynów czy inteligentnych systemów transportowych.

Artykuł jest kontynuacją prac opisanych w [7]. Przedstawiono w nim proces budowy robota typu Micromouse. Opisano elementy mechaniczne i elektroniczne. Następnie zaimplementowano algorytm sterowania oraz przedstawiono algorytm rozrostu ziarna, służący do optymalizacji drogi przejścia przez labirynt. Na koniec przeprowadzono badania testowe i dokonano ich oceny.

2. Wykonanie robota

2.1. Wymagania

Głównym wymaganiem jest budowa robota, który szybko i z dużą powtarzalnością będzie wykonywał manewry w labiryncie. Wymusza to wykorzystanie szybkich napędów. Najlepsze konstrukcje umożliwiają jazdę z prędkością do 3 m/s po prostych. Przy tak dużych prędkościach istotną jest bezwładność całej konstrukcji. Mniejsza bezwładność zapewni lepszą zwrotność robota i pozwoli na pokonywanie zakrętów z większą prędkością. Aby to uzyskać, środek ciężkości konstrukcji musi znajdować się w bliskim otoczeniu osi napędowej robota.

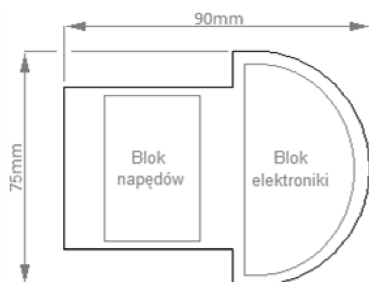
Drugim wymaganiem jest zapewnienie możliwie małej wagi urządzenia. W efekcie zwiększy się stosunek wagi napędu do wagi całej konstrukcji.

Ostatnim wymaganiem jest duża sztywność i wytrzymałość podstawy oraz głównej ramy konstrukcji. Pozwoli to zabezpieczyć robota przed uszkodzeniem w wyniku ewentualnych zderzeń wewnątrz labiryntu. Po analizie podjęto decyzję o zastosowaniu laminatu, który pełni również rolę płyty głównej sterownika i części elektronicznych.

2.2. Elementy mechaniczne

W pierwszej kolejności określono wymiary robota ograniczone wielkością labiryntu, w których się on porusza. W celu skrócenia czasu przejazdu, założono jazdę robota po przekątnych komórek labiryntu. Dzięki temu urządzenie wykonuje mniej zakrętów oraz zmniejsza się stosunek przebytej drogi o $\sqrt{2}$ (skrócenie jej o ponad 30%). Wymaga to jednak budowy wąskiej konstrukcji. Szerokość trasy podczas jazdy po przekątnych wynosi 110,309 mm. Poza tym należy pozostawić margines bezpieczeństwa (10÷20) mm z lewej i prawej strony robota. Mniejsza szerokość to również mniejszy promień skrętu i lepsza dynamika ruchu. Długość urządzenia wpływa głównie na działanie czujników odległości. Montowane na przodzie robota, posiadają tzw. martwą strefę. Mniejsza długość pozwoli więc uniknąć pojawienia się czujników w tej strefie. Wstępne wymiary robota pokazano na rysunku 1.

Chcąc zapewnić dobrą przyczepność robota do podłoża, wybrano koła z oponami o niskiej twardości (rzędu 10). Średnica koła z oponą wyniosła 22 mm.



Rys. 1. Wstępne wymiary robota
Fig. 1. Preliminary dimensions of the robot

Niewielkie rozmiary robota wymuszają zastosowanie małego napędu z dużym momentem. Po analizie zdecydowano się wykorzystać szczotkowe silniki prądu stałego z pustym rdzeniem. Kolejnym zagadnieniem jest konfiguracja napędów. Minimalizacja wymiarów oraz wagi robota nie pozwala na użycie dużej liczby napędów. Zdecydowano się na konfigurację z dwoma silnikami. Aby uzyskać prędkość 3 m/s konieczne było zastosowanie przekładni. Jej doboru dokonano uwzględniając możliwą średnicę zębatego. Wybrano przełożenie 1:4, a średnica zębatego wyniosła 21 mm. Dodatkowo, wykorzystanie przełożenia będącego potęgą liczby 2 ułatwiło w późniejszym etapie wyznaczenie prędkości w mikrokontrolerze.

Ostatecznie, wykonano model silnika i zębatego w środowisku Autodesk Inventor [8] oraz sporządzono projekt obejmy. Pełni ona dwie funkcje – mocowania silnika oraz jego chłodzenia. Obejme wykonano z aluminium, które odznacza się dobrymi właściwościami rozpraszania ciepła, małą wagą i prostotą w obróbce. Wykonanie elementów zlecono profesjonalnej firmie. Do osi koła przyklejono magnes, co pozwoliło uzyskać odczyt o prędkości obrotowej z enkoderów.

2.3. Elementy elektroniczne

Z punktu widzenia możliwości implementacji algorytmów sterowania najbardziej istotnym komponentem jest jednostka centralna. Wybrano 32-bitowy mikrokontroler ARM, który cechuje się dużą ilością pamięci flash (128 kB), wysoką częstotliwością taktowania (do 72 MHz) i dużą liczbą linii wejście/wyjście (51). Poza tym zaletą układu są rozbudowane układy czasowe pozwalające na sprzętową obsługę enkoderów z interfejsem kwadraturowym.

Do sterowania napędami potrzebny jest pośredniczący układ mocy (tzw. driver). Zdecydowano się na użycie dwóch mostków typu H. Pojedynczy driver dostarcza ciągły prąd o wartości do 2,8 A. Jednocześnie wymiary układu są bardzo małe.

Następnie dokonano doboru trzech urządzeń pomiarowych. Do efektywnej jazdy po labiryncie konieczne jest śledzenie układu ścian. Sama informacja, czy ściana w danej komórce labiryntu się znajduje (lub nie), może okazać się niewystarczająca. Wiedza o odległości robota od ścian może okazać się bardzo pomocna, np. w unikaniu kolizji ze ścianami oraz korekcy położenia robota. Jako czujniki odległości wybrano pary fototranzystor – dioda pracująca w paśmie podczerwieni. Na nadajnik wybrano diody o wąskiej ($\pm 3^\circ$) i bardzo skupionej wiązce światła. Jako odbiornik użyto fototranzystor, odznaczający się liniowością w dużym zakresie. Przyjęto, że wystarczą cztery pary tych czujników – dwa mierzące odległość z przodu robota oraz dwa mierzące odległość robota od bocznych ścian. Zastosowanie dwóch par pozwoliło obliczać różnicę między odległościami, co przydało się do korekcy błędów podczas jazdy w labiryncie.

Do orientacji w labiryncie potrzebna jest także informacja o przebytej drodze. W projekcie przewidziano pomiar prędkości

i pozycji robota. W tym celu wykorzystano inkrementalne enkodery magnetyczne, które odznaczają się dużą rozdzielczością sięgającą 12-bitów, prostą implementacją i małą wagą. Pomiar prędkości polega na zliczeniu liczby impulsów w odstępie czasu równym 1 ms. Pomiar pozycji następuje pośrednio, poprzez sumowanie prędkości. Do poprawnego działania tego typu enkoderów konieczne jest zamocowanie odpowiedniego magnesu na elemencie (koło), którego prędkość obrotową chcemy mierzyć.

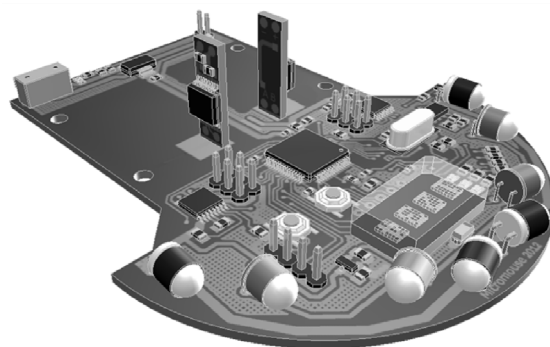
Ostatnim z urządzeń pomiarowych jest żyroskop. Ponieważ napęd różnicowy narażony jest na poślizgi i gubienie impulsów z enkoderów, szczególnie przy skrętach, żyroskop spełnia istotną rolę przy korekcy tych błędów. Zwiększa to tym samym precyzję i niezawodność nawigacji robota. Wybrano żyroskop mierzący prędkość obrotową robota w osi z, z rozdzielczością 2000 $^\circ/s$.

Ze względu na dużą ilość danych przetwarzanych podczas przejazdu, istotny jest dobry interfejs komunikacji z robotem. W tym celu wykorzystano 2 przyciski membranowe, 3 diody LED, 4-znakowy wyświetlacz ledowy oraz sygnalizator dźwiękowy. Różne tonacje dźwięków, zależne od częstotliwości generowanego sygnału, są znacznie bardziej zauważalne niż sygnał świetlny. Ponadto wyprowadzono połączenia pozwalające na komunikację przez interfejs USART (ang. Universal Synchronous and Asynchronous Receiver and Transmitter) i usuwanie błędów przy użyciu interfejsu SWD (ang. Serial Wire Debugging).

Micromouse jest robotem mobilnym więc musi posiadać własne zasilanie. Biorąc pod uwagę niewielkie rozmiary i wagę konstrukcji oraz dostatecznie dużą pojemność akumulatora, po analizie wybrano akumulator litowo-polimerowy. Użycie dwóch ogniw pozwoliło na uzyskanie bardzo dużej wydajności prądowej przy niewielkich wymiarach i wadze. Napięcie znamionowe akumulatorów wynosi 7,4 V, a w stanie pełnego naładowania: 8,4 V. Przy ich użytkowaniu nie można ich zbyt rozładować (poniżej 3,2 V na ogniwo), gdyż grozi to ich spalaniem lub wybuchem. Konieczne było zatem wykorzystanie układu nadzorującego stan naładowania baterii, aby w razie potrzeby zminimalizować zużycie prądu.

Wszystkie komponenty dobrano z myślą o niskim poborze energii na poziomie napięć równym 3,3 V. Napięcie znamionowe silników pobierane jest bezpośrednio z baterii przez mostek H. Odseparowano zasilanie urządzeń pomiarowych (części analogowej) od pozostałych elementów elektronicznych (części logicznej) w celu zmniejszenia zakłóceń, mając na uwadze maksymalne pobory prądu przez wszystkie urządzenia.

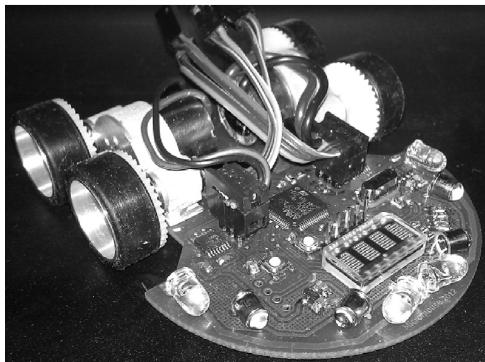
Łączenie komponentów, na podstawie schematu, rozpoczęto od narysowania założonego obrysu płytki. Następnie rozmieszczono wszystkie komponenty i połączenie rozpoczęto od linii zasilania. Linie wysokoprądowe prowadzono grubymi ścieżkami z dużymi odległościami od pozostałych ścieżek. Następnie połączono kolejne komponenty i rozplanowano położenie urządzeń pomiarowych. Projekt płytki drukowanej robota wykonano w środowisku CAD-SOFT Eagle [9], a jej wizualizację w programie Autodesk Inventor [8] (rysunek 2).



Rys. 2. Wizualizacja płytki drukowanej
Fig. 2. Visualization of the circuit board

2.4. Końcowy wynik prac

Na rysunku 3 pokazano końcowy rezultat prac nad konstrukcją robota Micromouse, którego funkcjonalność jest zgodna z przyjętymi wymaganiami.



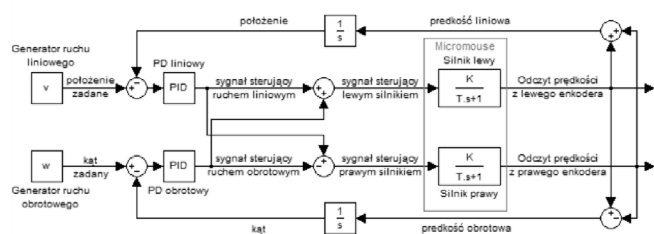
Rys. 3. Widok robota Micromouse
Fig. 3. A view of the Micromouse robot

3. Sterowanie robotem

3.1. Układ regulacji

Układ regulacji zaprojektowano z myślą o specyfice napędu różnicowego. Pojazdy wykorzystujące taką konfigurację są sterowane kierunkiem jazdy poprzez podawanie różnych prędkości na oba koła, tym samym nie wymagają one osobnych układów kierowniczych. Cały zakres ruchów robota można zatem opisać kombinacją ruchów: liniowego oraz obrotowego. Pierwszy z nich jest uśrednioną sumą prędkości z obu silników, podczas gdy ruch obrotowy jest ich uśrednioną różnicą.

W labiryncie istotne jest położenie robota oraz jego orientacja, które w procesie regulacji są wartościami regulowanymi. Wartości zadane przekazywane są z generatorów ruchów w postaci naliczeń enkoderów odpowiadających odcinkom jakie ma przejechać robot. Sprzężenie zwrotne realizowane jest za pomocą enkoderów. Całością sterują dwa regulatory PD: pierwszy odpowiedzialny za ruch liniowy, a drugi za obrotowy (rysunek 4). Blok sterowania pojedynczym napędem pokazano na rysunku 5.



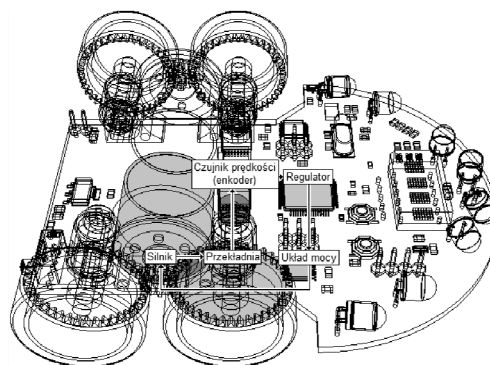
Rys. 4. Schemat układu regulacji
Fig. 4. Scheme of the control system

Podanie stałego zasilania na wyprowadzenia silnika wymusi obroty silnika o proporcjonalnej do zasilania prędkości, z drobnym opóźnieniem związanym z jego bezwładnością. Takie zachowanie obiektu można opisać za pomocą transmitancji pierwszego rzędu, w której parametrami jest wzmocnienie sygnału wejściowego K oraz opóźnienie T . Zgodnie z zasadą superpozycji, suma (ruch liniowy) oraz różnica (ruch obrotowy) obiektów o charakterze liniowym, jest również obiektem liniowym. Parametry pojedynczego silnika nie są istotne z punktu widzenia przyjętego układu regulacji, ze względu na znacznie bardziej znaczącą bezwładność robota. Ponieważ celem regulacji jest położenie robota, jego prędkość obrotowa i liniowa są całkowane. Zatem zarówno

ruch liniowy jak i obrotowy zamodelowano za pomocą transmitancji II-rzędu, postaci:

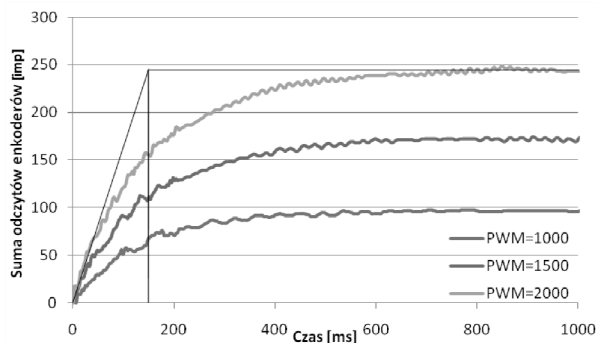
$$G = \frac{K}{s(Ts+1)} \tag{1}$$

gdzie: K – wzmocnienie, T – stała czasowa.

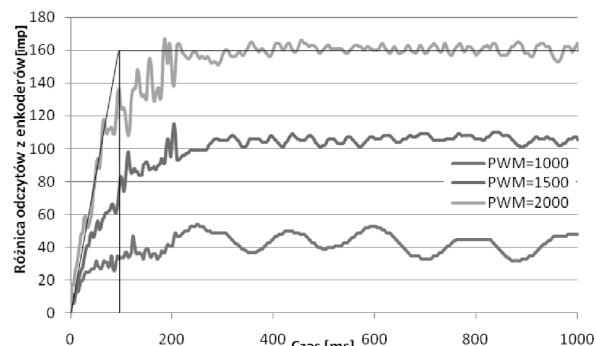


Rys. 5. Układ sterowania pojedynczym napędem
Fig. 5. Scheme of the individual drive control

Identyfikację ruchu robota przeprowadzono podając na oba silniki taki sam, 12-bitowy sygnał sterujący PWM (ang. Pulse Width Modulation) oraz zadając jednakowy (w przypadku ruchu liniowego) lub przeciwny (dla ruchu obrotowego) kierunek obrotów. Wynik identyfikacji dla trzech różnych wypełnień sygnałów przedstawiono na rysunkach 6 i 7.



Rys. 6. Identyfikacja ruchu liniowego
Fig. 6. Linear motion parameter identification



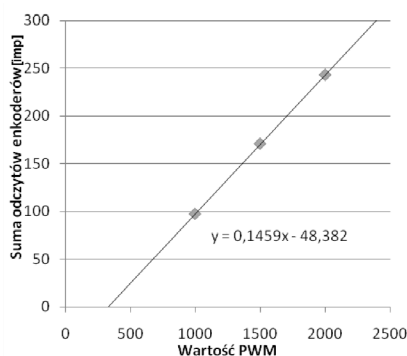
Rys. 7. Identyfikacja ruchu obrotowego
Fig. 7. Rotational motion parameter identification

Należy odnotować, że kształt każdej z krzywych jest taki sam. Jest to lekko zasumowana forma funkcji eksponentialnej opisanej zależnością:

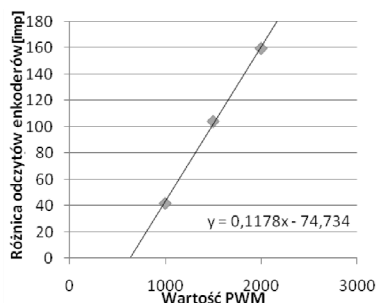
$$y(t) = K \left(1 - e^{-\frac{t}{T}}\right) * u(t) \tag{2}$$

W ruchu obrotowym widać niewielkie odchylenia przy niższych prędkościach. Zanikają one wraz ze wzrostem prędkości, gdyż robot obraca się szybciej. Amplituda oscylacji maleje, a ich częstotliwość rośnie. Powodem powstawania tych odchylenia mogą być błędy w mechanice robota. Po dokładniejszej analizie można zauważyć, że prawy zestaw kół robota swoim ruchem zamiast kreślić prostą linię, zatacza wąską cyfrę „8”. Innym powodem może być nierównomierne rozłożenie wagi, przez co środek ciężkości nie leży na osi symetrii robota.

Stałą czasową T można określić prowadząc styczną do narastającego sygnału. Jest ona równa wartości, w której poprowadzona styczna przecina się z linią wyznaczającą wartość odpowiedzi w stanie ustalonym (patrz rysunek 6). Dla uzyskanych danych pomiarowych wyniosła ona odpowiednio: 153 ms (ruch liniowy) i 81 ms (ruch obrotowy). Wzmocnienie statyczne K wyznacza się obliczając stosunek wartości wyjścia obiektu, do jego sygnału sterującego. Wykonano pomiary dla trzech różnych sygnałów sterujących. Następnie wartości odpowiedzi w stanie ustalonym wykreślono w funkcji wypełnienia (rysunki 8-9).



Rys. 8. Określenie wzmocnień transmitancji ruchu liniowego
Fig. 8. Determining the static gain value of linear motion



Rys. 9. Określenie wzmocnień transmitancji ruchu obrotowego
Fig. 9. Determining the static gain value of rotational motion

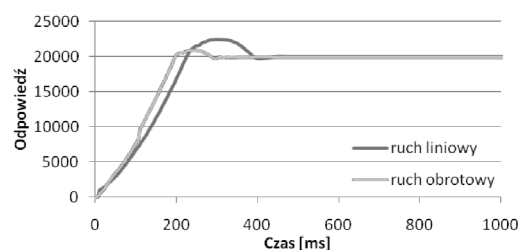
Jak widać trend obu funkcji jest liniowy i przechodzi przez sam środek wszystkich jej punktów, co pozwala jednoznacznie stwierdzić, iż obiekt jest liniowy. Ze względu na bezwładność robota, potrzebna jest pewna wartość wypełnienia sygnału sterującego silnikami, aby robot w ogóle ruszył z miejsca. Widać to również na rysunkach 8-9 (funkcja nie przechodzi przez punkt (0,0)). Punkt przecięcia jest związany z minimalną wartością sygnału PWM, od którego robot powinien się poruszać. Wzmocnienie obiektu można odczytać z rysunków 8-9 – jest to wartość nachylenia funkcji. Ostatecznie, ruch liniowy (3) i obrotowy (4) robota opisano za pomocą transmitancji postaci:

$$G = \frac{0,1459}{s(0,153s+1)} \quad (3)$$

$$G = \frac{0,1178}{s(0,081s+1)} \quad (4)$$

Wykorzystanie regulatorów PD wynika z możliwości skorzystania z pełnej informacji pomiarowej dostępnej z enkoderów: położenia i prędkości. Użycie regulatora z członem całkującym jest zbędne ze względu na zerowy uchyb w stanie ustalonym. Sterowanie odbywa się z częstotliwością 1 kHz, a nastawy regulatorów dobrano w wyniku badań testowych i wynoszą one odpowiednio: $K_p = 6,863$, $K_d = 0,3125$ (regulator ruchu liniowego), $K_p = 9,128$, $K_d = 0,53125$ (regulator ruchu obrotowego).

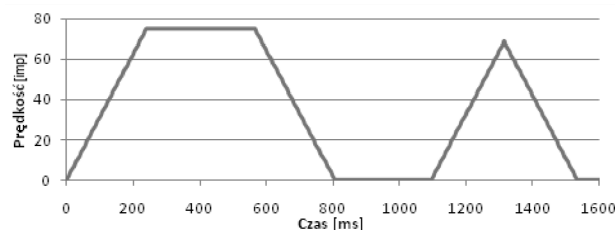
Wyniki działania układów regulacji na wymuszenia o wartości 20000 przedstawia rysunek 10. Wymuszenie podano zamiennie na jeden z regulatorów, rolą drugiego regulatora jest stabilizacja wokół punktu zerowego.



Rys. 10. Odpowiedź regulowanego obiektu na wymuszenie równe 20000
Fig. 10. Response of the controlled robot to a set point of 20000

Wyniki sterowania są dobre mimo występowania przeregulowań (maksymalnie 12,38% dla ruchu liniowego i 4,985% dla ruchu obrotowego). Nie da się ich do końca wyeliminować, ze względu na bezwładność robota. Niemożliwe jest, aby rozpedzony robot natychmiast wyhamował.

Oprócz kontroli pozycji robota konieczna jest jeszcze kontrola jego prędkości. Odpowiednio zmieniając pozycję zadaną robotu, można kontrolować prędkość jego jazdy. Zamiast przesunąć punkt zadany o jeden większy odcinek, jest on dzielony na mniejsze, odpowiadające pożądanej prędkości w danej chwili. Wartość zadana regulatora aktualizowana jest w każdej iteracji pętli sterującej. Długość tych odcinków w czasie wykreśla pożądany profil ruchu. Przykładowy ruch wygenerowany przez taki generator (tzw. profiler) obrazuje rysunek 11.



Rys. 11. Przykładowy profil prędkości robota
Fig. 11. Example of the robots speed profile

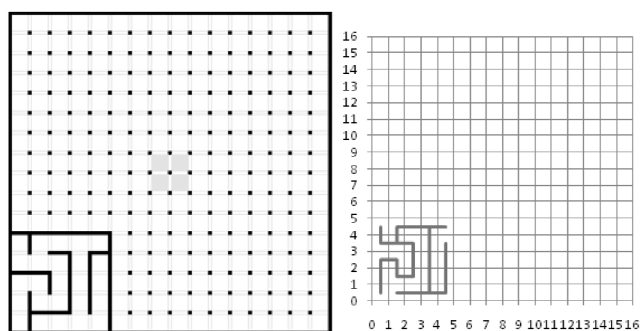
W czasie przyspieszania, prędkość rośnie liniowo. Po osiągnięciu prędkości zadanej, robot poruszać się będzie z tą prędkością, aż do momentu, gdy nie zbliży się do końca drogi zadanej, kiedy to zacznie hamować. Po wyznaczeniu prędkości z jaką ma poruszać się robot, należy zaktualizować zadane położenie i przekazać je do regulatora. Przedwczesne lub spóźnione hamowanie spowoduje, że przebyta droga będzie niezgodna z zamierzoną. Problem przeregulowań, który widać na rysunku 10, nie występuje przy profilowaniu prędkości.

3.2. Pomiary

Odczytanie wartości zwracanych przez urządzenia pomiarowe polega na pomiarze napięcia z przetwornika analogowo-cyfrowego znajdującego się w mikrokontrolerze. Do poprawnej pracy fototranzystorów potrzeba jedynie źródła światła (diod podczerwonych), które odbijane od ścian labiryntu będzie informować robota o pozostałej do nich odległości. Przy wykonywaniu

pomiaru żyroskopem, największą uwagę należy poświęcić jego kalibracji. Po uruchomieniu robota napięcie na baterii jest stabilne, gdyż nie pracują urządzenia pobierające dużą ilość prądu, np. silniki czy diody nadawcze. Następuje wtedy kalibracja odczytów z żyroskopu, ponieważ wartość napięcia na wyjściu regulatora analogowego nie podlega wahaniom – punkt odniesienia dla przetwornika analogowo-cyfrowego jest jednakowy. Pomiaru są wykonywane wielokrotnie, a następnie uśredniane.

Mapowanie labiryntu polega na sprawdzeniu, czy odczyt z urządzenia pomiarowego wskazuje na odległość mniejszą od progowej, przy której przyjęto, że powinna zostać wskazana obecność ściany. W przypadku przednich ścian, sprawdzane są odczyty z dwóch czujników. Aby dodać ścianę, oba przednie czujniki muszą wskazywać, z pewnym marginesem błędów, podobny odczyt. W tym samym momencie odnotowane jest, że robot zwiedził komórkę i poznał układ jej ścian. Następnie uzupełniane są szczegółowe informacje o komórkach sąsiednich. Labirynt zapisany jest w formacie tablicy 256-elementowej. W każdej komórce należy zaznaczyć obecność maksymalnie 4 ścian. Używając zmiennych 8-bitowych, gdzie jeden bit reprezentuje obecność lub brak ściany, pozostałe cztery bity można wykorzystać do innych zastosowań. Przykładem może być sygnalizacja, czy robot był w danej komórce lub nie i czy powinien do danej komórki wjeżdżać. Przykładowy zapisany labirynt przedstawiono na rysunku 12.



Rys. 12. Przykładowy labirynt oraz przejazd
Fig. 12. Example of a maze and taken route

Wszystkie pomiary z przejazdów zapisywane są na karcie microSD. Wykorzystując taką rejestrację danych możliwy jest zapis danych z przejazdów robota, do ich dalszej analizy, udoskonalania konstrukcji i oprogramowania. Przykładowe wykresy mogą przedstawiać, np. zależność napięcia na baterii od sygnałów sterujących silnikami lub zwiedzone przez robota komórki w trakcie jazdy. Ponadto, odpowiednie sformatowanie danych pomiarowych pozwala na wizualizację układu ścian labiryntu.

4. Optymalizacja czasu przejścia labiryntu

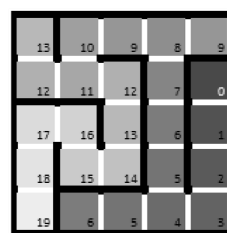
4.1. Algorytm rozrostu ziarna

Celem robota Micromouse jest jak najszybsze przejechanie labiryntu. Należy zatem obliczyć najkrótszą ścieżkę. Zadanie to wymaga zastosowania algorytmów optymalizacji, a jednym z nich jest algorytm rozrostu ziarna. Używa się go np. w programach graficznych do wypełniania zamkniętych obszarów bitmapy kolorem [10]. Odpowiednio go modyfikując możliwe jest, aby wskazywał odległości pomiędzy dwiema komórkami, poprzez uwzględnienie ścian labiryntu.

Nie jest istotne, czy cały labirynt został już zwiedzony – algorytm rozwiązuje labirynt w oparciu o informacje, które są dostępne w danej chwili. Dzięki temu możliwe jest wygenerowanie ruchu dla robota i umieszczenie go w kolejce, aż do następnej iteracji algorytmu, która wykonywana jest w każdym ruchu robota. Algorytm cyklicznie skanuje labirynt, uzupełniając trasę o nowe wartości. W najgorszym wypadku, podczas jednej iteracji, algorytm może zeskanować labirynt 255 razy. W takim przypadku najprawdopo-

dobnie istnieje błąd w labiryntcie uniemożliwiający dojechanie do celu. Labirynt jest skanowany dopóki algorytm wykazuje, że mapa trasy ulega zmianie między kolejnymi skanami.

Algorytm rozrostu ziarna wyznacza odległości w komórkach od celu postawionego przed robotem na początku zadania. Podczas przejazdu (podczas wyznaczania kolejnego ruchu robota) należy kierować się zarówno odległością do celu, jak i obecnością ścian w labiryntcie. Przykład działania algorytmu przedstawiono na rysunku 13. W poszczególnych komórkach zapisana jest odległość od komórki docelowej oraz jej graficzna reprezentacja w postaci gradientu. Robot startując z lewego dolnego rogu musi dojechać do komórki o funkcji celu (odległości) równej 0. W przypadku nieuwzględnienia obecności ścian, robot wystartowałby skracając w prawo i uderzając w ścianę.



Rys. 13. Wizualizacja algorytmu rozrostu ziarna
Fig. 13. Flood-fill algorithm visualization

4.2. Korekcja pozycji w labiryntcie

Korekcja pozycji robota jest kolejnym istotnym zagadnieniem. Drobne szumy pomiarowe z enkoderów mogą powodować uchyb pomiędzy położeniem zadaniem a rzeczywistym. Próby wykorzystania filtrów pomiarowych niestety nie usprawniły pracy robota, a nawet ją pogorszyły. Dodanie filtrów wprowadzało dodatkową dynamikę do układu, powodując oscylacje robota. Implementacja zaawansowanych metod filtracji, np. filtr Kalmana okazała się niemożliwa do realizacji z powodu niewystarczającej dokładności mikrokontrolera i zbyt dużego obciążenia obliczeniowego procesora.

Korekcję pozycji robota rozwiązano poprzez wykorzystanie analogowych odczytów z czujników odległości. Zależnie od konkretnego przypadku, celem jest wygenerowanie ruchu, gdy wystąpi uchyb. W przypadku korekcji położenia względem bocznych ścian oraz kąta obrotu, generowany jest ruch obrotowy. Jeżeli konieczna jest poprawa pozycji względem przedniej ściany komórki to robot musi wykonać korygujący ruch liniowy.

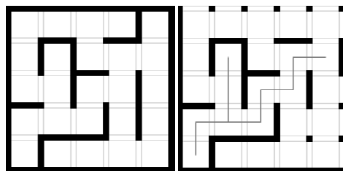
5. Badania testowe

Zwieńczeniem prac konstrukcyjno-programistycznych jest ocena osiągnięć robota. W tym celu przygotowano dwa różne układy labiryntów do rozwiązania. Elementami decydującymi o ocenie robota będzie fakt wykonania poprawnej mapy labiryntu, dojechania do celu i wyznaczenia optymalnej ścieżki. Za wskaźnik jakości posłuży średnia procentowa wartość uchybu, oraz czas w jakim robot przejechał labirynt. Podczas każdego przejazdu robot będzie startował z lewego dolnego rogu labiryntu, zgodnie z zasadami zawodów Micromouse. Ponieważ pełnowymiarowy labirynt wykorzystywany na zawodach ma wymiary 2,88x2,88m, przy testach posługiwano się mniejszym labiryntem, o wymiarach 5x5 komórek i rozmiarze 0,9x0,9m. Jako komórkę docelową wybrano komórkę znajdującą się bezpośrednio poniżej komórki w prawym górnym rogu. Pierwszy labirynt testowy i wyniki przejazdu pokazano na rysunkach 14-16.

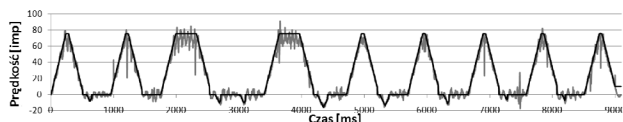
Przejazd trwał 9,1 s a wskaźnik jakości wyniósł 2,73%. Można zauważyć błędy w śledzeniu zadanej trajektorii, które wynikają z niedociągnięć w fizycznej realizacji napędów. Drobne niedokładności przy ząbieniu się przekładni powodują szarpanie robota wokół punktu zadanego (rysunki 15-16). Poza tym robot z dużym powodzeniem jeździ zgodnie z wyznaczoną trajektorią –

średni uchyb jest mniejszy niż 3%. Wszelkie błędy korygowane są przy każdym możliwym napotkaniu ściany (z dowolnej strony robota) – widoczne są na przebiegach jako drobne ruchy generowane między większymi.

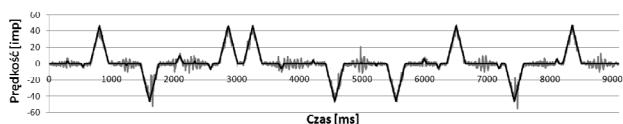
Na rysunkach 17-19 przedstawiono drugi labirynt testowy i wyniki przejazdu.



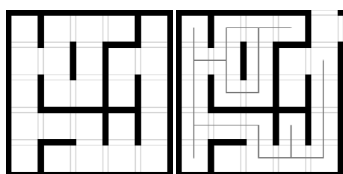
Rys. 14. Testowy labirynt nr 1 (lewa strona), zwiedzony labirynt (prawa strona)
Fig. 14. Test maze no. 1 (left side), explored maze (right side)



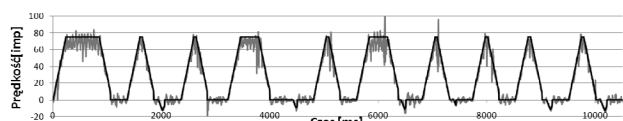
Rys. 15. Ruch liniowy robota wewnątrz pierwszego labiryntu
Fig. 15. Linear motion of the robot inside the first maze



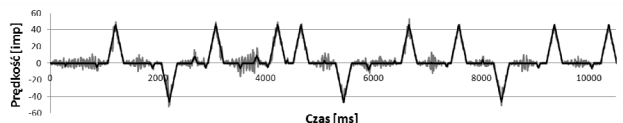
Rys. 16. Ruch obrotowy robota wewnątrz pierwszego labiryntu
Fig. 16. Rotational motion of the robot inside the first maze



Rys. 17. Testowy labirynt nr 2 (lewa strona), zwiedzony labirynt (prawa strona)
Fig. 17. Test maze no. 2 (left side), explored maze (right side)



Rys. 18. Ruch liniowy robota wewnątrz drugiego labiryntu
Fig. 18. Linear motion of the robot inside the second maze



Rys. 19. Ruch obrotowy robota wewnątrz drugiego labiryntu
Fig. 19. Rotational motion of the robot inside the second maze

Przejazd robota został ukończony w 20,2 s. Dłuższy czas przejazdu wynika z działania algorytmu – robot „preferuje” jazdę wprost, by nie tracić czasu na hamowanie przed i rozpędzanie po zakręcie. Z tego powodu, robot pojechał w „ślepy zaułek” szukając alternatywnej drogi do celu, po czym wrócił i sprawdził drugą (właściwą) ścieżkę. Wskaźnik jakości wyniósł 2,76% - wynik podobny do poprzedniego przejazdu.

W zamieszczonych obu przejazdach, zadana prędkość liniowa wynosiła 0,63 m/s. Na obecnym etapie prac jest to prędkość bezpieczna, przy której robot jedzie pewnie, a ewentualne kolizje nie zagrażają jego dalszemu działaniu. Kolejne przejazdy charakteryzowały się podobnymi wartościami wskaźnika jakości (poniżej 3%).

6. Podsumowanie

Prace nad nowoczesnymi formami transportu prowadzone są intensywnie na całym świecie, a badania nad robotami mobilnymi realizowane są w wielu placówkach na świecie.

Micromouse sprawdził się jako platforma do implementacji algorytmów sterowania i optymalizacji. Zbudowano obiekt fizyczny, który zamodelowano, dokonano identyfikacji parametrów i zweryfikowano. Następnie zaimplementowano metodę sterowania jego ruchami (liniowy, obrotowy) z wykorzystaniem układów regulacji PD. Uzyskano zadowalające wyniki sterowania. Na koniec zaimplementowano algorytm optymalizacji do wyszukiwania ścieżki i planowania ruchów robota.

Niewątpliwie robot wymaga dalszych modernizacji. Udoskonalenie obecnej formy napędu pozwoli uzyskać większą stabilność i liniowość ruchów. Zmniejszenie masy robota pozytywnie wpłynie na jego jazdę (mniejsza bezwładność). Zastosowanie mikrokontrolera z jednostką zmiennoprzecinkową pozwoli zwiększyć możliwości robota z punktu widzenia implementacji bardziej złożonych algorytmów filtracji, sterowania i optymalizacji.

7. Literatura

- [1] IEEE Micromouse rules, <http://iee.ucsd.edu/projects/micromouse/rules.php/>, dostęp 20.08.2014.
- [2] Bradbeer R.: The Experience of Using Micromouse and Robot Ping Pong for Teaching Mechatronics. Proc. of the 2nd Asian Conference on Robotics and Its Application, Beijing, China, October 13-15, 1994.
- [3] Wyard-Scott L., Meng Q.-H.M.: A potential maze solving algorithm for a micromouse robot. Proc. of the IEEE Pacific Rimm Conference on Communications, Computers and Signal Processing, Victoria, Canada, May 17-19, 1995.
- [4] Mishra S.: Maze Solving Algorithms for Micro Mouse. Proc. of IEEE International Conference on Signal Image Technology and Internet Based Systems, Bali, Indonesia, November 30-December 3, 2008.
- [5] Jianping C., Wan X., Huo M., Wu J.: An Algorithm of MicroMouse Maze Solving. Proc. of IEEE 10th Int. Conf. on Computer and Information Technology (CIT), Bradford, England, June 29-July 1, 2010.
- [6] Kibler S.G., Hauer A.E., Giessel D.S., Malveaux C.S., Raskovic D.: IEEE Micromouse for Mechatronics Research and Education. Proc. of the IEEE Int. Conf. on Mechatronics, Istanbul, Turkey, April 13-15, 2011.
- [7] Ficht G., Piotrowski R.: Micromouse robot – technical design and construction. Mat. konferencyjne III ogólnopolskiego Forum Innowacji Młodych Badaczy – FIMB 2012, Łódź, Polska, 16-17.11.2012.
- [8] Autodesk Inventor, <http://usa.autodesk.com/autodesk-inventor/> dostęp 20.08.2014.
- [9] CADsoft, <http://www.cadsoftusa.com/>, dostęp 20.08.2014.
- [10] Shaw J.R.: QuickFill: An efficient flood fill algorithm, <http://www.codeproject.com/>, dostęp 20.08.2014.

otrzymano / received: 23.08.2014

przyjęto do druku / accepted: 03.11.2014

artykuł recenzowany / revised paper