

ŚRODOWISKO SZYBKIEGO PROTOTYPOWANIA ALGORYTMÓW STEROWANIA MOBILNYMI ROBOTAMI KOŁOWYMI

ANDRZEJ BURGHARDT, DARIUSZ SZYBICKI, TOMASZ BURATOWSKI

*Katedra Mechaniki Stosowanej i Robotyki, Politechnika Rzeszowska, Katedra Robotyki i Mechatroniki,
Akademia Górniczo-Hutnicza
e-mail: andrzejb@prz.edu.pl, dszybicki@prz.edu.pl, tburatow@agh.edu.pl*

Streszczenie. Środowiska szybkiego prototypowania pozwalają na testowanie opracowanych rozwiązań programowych oraz sprzętowych na wielu etapach prowadzonych prac badawczych. Poniżej zaprezentowano autorskie oprogramowanie pozwalające na sterowanie i monitorowanie pracy grupy do 256 mobilnych robotów firmy MobileRobots. Wykonane oprogramowanie łączy zalety pakietu Matlab/Simulink oraz oprogramowania Aria. Ponadto w pracy zaprezentowano, wykonany przez autorów, moduł pomiarowy gazów szkodliwych oraz skaner laserowy współpracujący z robotami AmigoBot.

1. WSTĘP

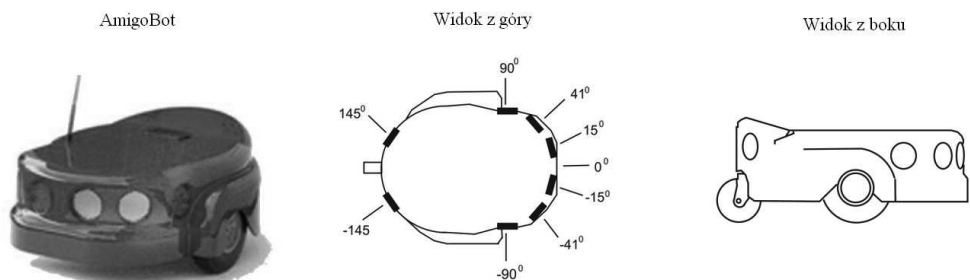
Rozwój robotów mobilnych, wykonujących złożone zadania, jest jednym z motorów napędowych współczesnej robotyki. Pomysłowość konstruktorów oraz wynikająca z niej mnogość aplikacji połączona z różnorodnością algorytmów sterowania owocuje szerokim spektrum zastosowań i aktualnością badań realizowanych z wykorzystaniem środowisk szybkiego prototypowania. W przypadku robotów mobilnych prototypowanie najczęściej dotyczy modelowania zarówno napędów jak i całych konstrukcji, testowania algorytmów sterowania jednego lub grupy robotów, modelowania układów sensorycznych itp. [1,3]. Zaproponowane rozwiązanie zostało wykonane jako połączenie wybranych fragmentów oprogramowania producenta Aria z mex-funkcją Matlaba/Simulinka. Do komunikacji wykorzystano sieć bezprzewodową. Prezentowane środowisko spełnia założenia systemu czasu rzeczywistego, rejestruje dla jednego lub grupy robotów takie parametry jak: prędkości i przemieszczenia kątowe kół napędzających, napięcie baterii, prądy silników napędzających koła, wskazania czujników odległości. Ponadto zapewnia obsługę wejść i wyjść cyfrowych oraz wejść analogowych, w które wyposażone są roboty. Umożliwia programową obsługę urządzeń peryferyjnych robotów mobilnych, wykorzystujących do komunikacji standard RS232 oraz I²C. Dla potrzeb badań roboty AmigoBot wyposażono w zbudowany przez autorów elektroniczny moduł pomiarowy stężenia gazów szkodliwych oraz skaner laserowy.

2. ŚRODOWISKO PROTOTYPOWANIA ROBOTÓW MOBILNYCH

Prezentowane środowisko szybkiego prototypowania pozwala na sterowanie robotami obsługującymi Arię, czyli obiektową bibliotekę klas (SDK) języka „C++” udostępnianą przez firmę MobileRobots. Napisane oprogramowanie umożliwia sterowanie takimi robotami jak: AmigoBot, Pioneer 3-DX, PowerBot, Seekur, PeopleBot, PatrolBot.

2.1. Obiekt sterowania oraz dostępne oprogramowanie

W omawianym rozwiązaniu obiektem sterowania są roboty AmigoBot (rys.1) napędzane dwoma silnikami prądu stałego z przekładniami zębatymi.



Rys.1. AmigoBot

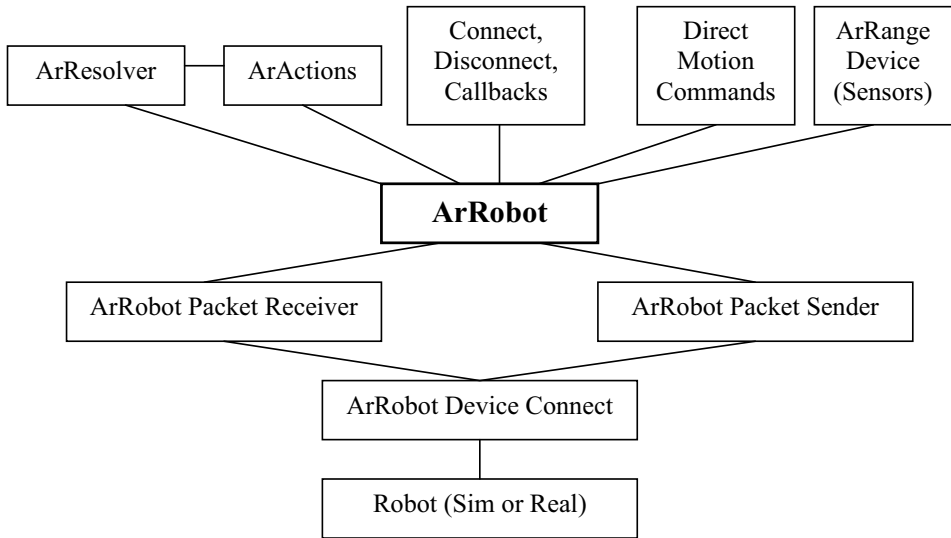
Mobilny robot AmigoBot wyposażony jest w przetworniki obrotowo-impulsowe, ultradźwiękowe czujniki odległości, bezprzewodowy moduł komunikacyjny, wewnętrzny regulator PID. Opcjonalne wyposażenie to: kompas z czujnikami położenia, kamera kolorowa oraz nocna, dalmierz laserowy oraz moduł GPS.

Pracą mobilnego robota zarządza oprogramowanie AmigoSH, które zaimplementowano na sterowniku Hitachi H8/2357, stanowiącym część robota. Za pomocą sieci bezprzewodowej oprogramowanie AmigoSH może wymieniać informację z nadrzędnym komputerem zawierającym oprogramowanie Aria. Oprogramowanie zaimplementowane na robocie pozwala na definiowanie:

- maksymalnych prędkości i przyspieszeń liniowych oraz kątowych ramy robota,
- zakresu pracy czujników,
- nastawów regulatora PID.

Ponadto zarządza portami komunikacji zewnętrznej robota (8 bitów wejść/wyjść port RS232, port I²C).

Oprogramowanie Aria pozwala na kontrolę funkcjonowania robotów mobilnych oraz podłączonych do nich urządzeń. Klasy składowe Arii zawierają funkcje wysyłania do i odbierania od robota prostych instrukcji typu: napięcie baterii, globalna współrzędna „X” (podstawowa klasa ArRobot, rys.2). Dostępne klasy umożliwiają też sterowanie wysokiego poziomu infrastrukturą Actions jak: zadawanie określonego toru jazdy (ArActionGoto), czy obsługa skanera laserowego (ArSick).



Rys.2. Architektura klasy ArRobot

Ponadto Aria zawiera bibliotekę ArNetworking pozwalającą na budowanie systemu komunikacji robotów opartej na protokole TCP/IP. Dzięki temu możliwa jest, bezprzewodowa kontrola robotów i podłączonych do nich urządzeń. Biblioteka ta jest napisana w języku „C++”, ma dostęp do API (ang. Application Programming Interface, czyli interfejs programowania aplikacji) dla języków Java i Pyton. Biblioteka pozwala na korzystanie w systemie GNU/Linux z kompilatora GCC (wersja 3.4 albo wyższa, prekompilowana w GCC 3.4). W systemie Windows Arię można kompilować w MS Visual C++ NET 2003 (7.1) i Visual C++ 2008 (9.0) [2].

Omawiane oprogramowanie pomimo niewątpliwych zalet ma liczne wady, nie zapewnia bowiem sterowania formacją robotów, nie pozwala na zapis danych pomiarowych, nie pozwala na wygodne testowanie opracowanych algorytmów i rozwiązań, a interfejs użytkownika jest nieprzyjazny i wymaga znajomości szeregu komend (rys.3).

```

C:\WINDOWS\system32\cmd.exe - demo -rh 192.168.1.13
You are in 'position' mode currently.
Mode is one of two values:
  both: heading and move can happen simultaneously
  either: only heading or move is active (move holds heading)

up arrow: forward 1.0 meter(s)
down arrow: backward 1.0 meter(s)
page up: increase distance by 1/2 meter
page down: decrease distance by 1/2 meter
left arrow: turn left 90 degrees
right arrow: turn right 90 degrees
space bar: stop
'r' or 'R': reset position to (0, 0, 0)
'x' or 'X': switch heading/velocity mode

Position mode shows the position stats on a robot.
  x      y      th      comp      volts      mps     mode
-335   -930   -121.73    0      12.40     191   both
  
```

Rys.3. Wygląd interfejsu użytkownika oprogramowania Aria

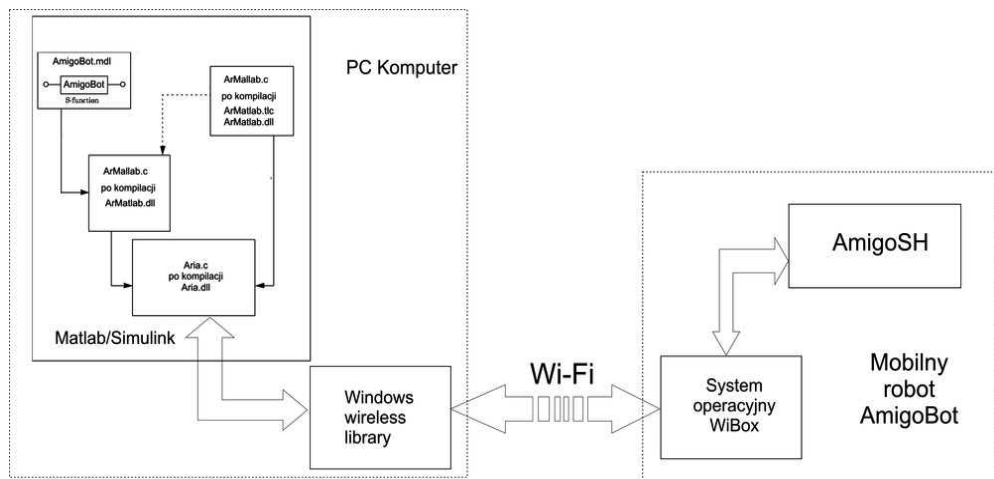
W celu wyeliminowania tych wad konieczne było opracowanie własnego rozwiązania.

2.2. Budowa środowiska szybkiego prototypowania

Dzięki użyciu dostępnych w Arii obiektów, funkcji oraz rozwiązań programistycznych możliwe było napisanie autorskiego oprogramowania łączącego zalety pakietu Matlab/Simulink z zaawansowanym, bezprzewodowym sterowaniem jednym lub grupą robotów i funkcjami kontrolno-pomiarowymi. Prezentowane rozwiązanie pozwala na monitorowanie pracy grupy mobilnych robotów, rejestrację wartości parametrów ruchu, wskazań z czujników odległości oraz sterowanie napędami. Ponadto jest ono elastyczne, pozwala na pracę z różną liczbą robotów, zmianę ich ustawień konfiguracyjnych oraz spełnia założenia systemu czasu rzeczywistego. Prezentowane środowisko kontrolno-pomiarowe:

- zapewnia sterowanie autonomiczne,
- realizuje procesy kontrolno-pomiarowe dla jednego lub kilku robotów,
- umożliwia pomiar i zapis następujących wielkości:
 - prędkości kątowych kół napędzających,
 - przemieszczeń kątowych kół napędzających,
 - wskazań czujników odległości,
 - pomiar prądu silników napędzających koła,
 - pomiar napięcia baterii,
- zostało wykonane jako oprogramowanie w postaci biblioteki pakietu Simulink.

Środowisko szybkiego prototypowania zarządzające pracą grupy n-mobilnych robotów zostało wykonane jako połączenie wybranych fragmentów oprogramowania producenta Aria z S-funkcją Simulinka, wykorzystując do komunikacji bezprzewodową sieć komputerową. W programie Simulink napisano bibliotekę zawierającą bloczek AmigoBot (rys.5), którego użycie w modelu umożliwia połączenie budowanych przez użytkowników algorytmów sterowania z obiektem rzeczywistym. Ogólny schemat realizacji środowiska kontrolno-pomiarowego przedstawiono na rys.4.



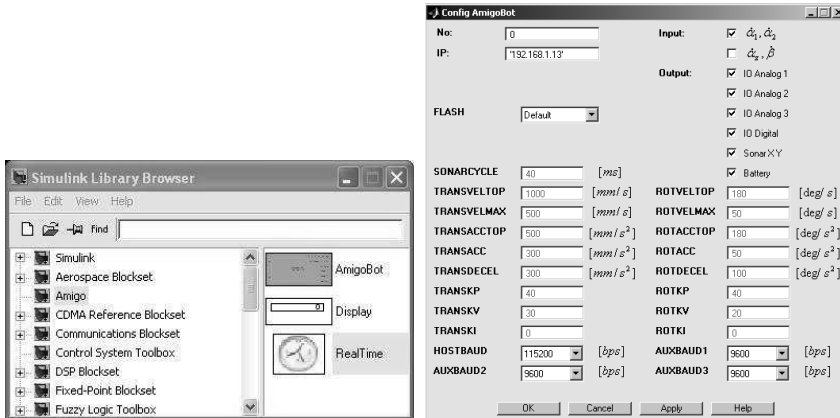
Rys.4. Schemat środowiska szybkiego prototypowania

Do zmiany wartości parametrów robota AmigoBot w podsystemie zostało stworzone specjalne okno konfiguracyjne, którego kod źródłowy znajduje się w m-pliku funkcyjnym (rys.5). Pozwala ono na zmianę parametrów robota takich jak:

- numer robota,

- adres IP robota,
- wartości zdefiniowanych w Amigo-SH, nastawy PID, prędkości maksymalne, itd.
- szybkości transmisji dla wskazanego portu szeregowego.

Wygląd okna wyboru autorskiej biblioteki AmigoBot oraz konfiguracyjnego podsystemu wykonanego w programie Matlab/Simulink przedstawia rys.5.

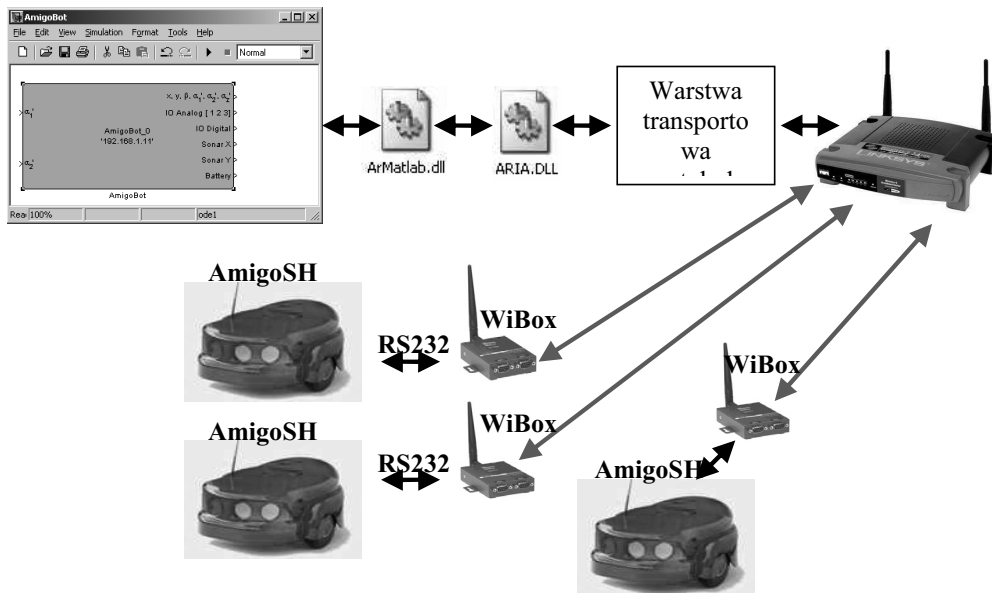


Rys.5. Okno konfiguracyjne środowiska prototypowania

Okno konfiguracyjne umożliwia zdefiniowanie wektora parametrów, jakie będą odczytywane z mobilnego robota.

2.3. Komunikacja środowiska szybkiego prototypowania

Do sterowania grupą autonomicznych robotów mobilnych niezbędna jest komunikacja pomiędzy komputerem nadrzędnym a robotami. Dla zapewnienia poprawnej komunikacji niezbędna jest poprawna konfiguracja urządzeń sieciowych: karty bezprzewodowej lub Access Point-a oraz modułu bezprzewodowego zainstalowanego w robocie. Komunikacja z robotami zostaje nawiązana w chwili uruchomienia symulacji, a zakończona podczas zakończenia symulacji. Zasadę działania środowiska prototypowania oraz drogę przepływu informacji przedstawia rys.6.



Rys.6. Schemat komunikacji pomiędzy środowiskiem Matlab/Simulink a robotami

Po uruchomieniu symulacji następuje nawiązanie połączenia z robotem (robotami), następnie wektor wejściowy podsystemu AmigoBot zostaje przesłany poprzez następujące moduły programowe i urządzenia sieciowe:

- za pomocą odpowiednio zaprojektowanego podsystemu AmigoBot, który jest maską bloku S-function, wektor wejściowy podsystemu AmigoBot oraz ustawienia zapisane poprzez interfejs zarządzający zostają przesłane do mex-funkcji ArMatlab [4],
- następnie wektor wejściowy oraz ustawienia parametrów operacyjnych robota, poprzez bibliotekę Aria, zostają przesłane za pomocą protokołu TCP/IP do Access Point-a,
- z Access Point-a do modułu bezprzewodowego zainstalowanego w robocie dane są przesyłane drogą radiową, z użyciem protokołu transmisji bezprzewodowej 802.11 b,
- moduł bezprzewodowy w robocie przesyła dane poprzez wbudowany port szeregowy z szybkością 115200 bps do kontrolera AmigoBot, gdzie następuje interpretacja otrzymanych danych oraz na ich podstawie sterowanie robotem.

2.4. Prototypowanie autorskich urządzeń rozszerzających możliwości robotów

Środowisko prototypowania pozwala na zmianę liczby robotów, bez konieczności ingerencji w oprogramowanie oraz umożliwia wymianę danych, sterowanie i zarządzanie urządzeniami peryferyjnymi w jakie każdy z robotów może być wyposażony.

Dla potrzeb badań roboty AmigoBot wyposażono w zbudowane w Katedrze Mechaniki Stosowanej i Robotyki elektroniczny moduł pomiarowy stężenia gazów szkodliwych (rys.7a) oraz skaner laserowy (rys.7b).



a) b)
Rys.7. Autorskie urządzenia obsługiwane przez środowisko szybkiego prototypowania

Elektroniczny moduł pomiarowy gazów szkodliwych jest to urządzenie, którego celem jest pomiar stężenia gazów niebezpiecznych dla człowieka (tlenek węgla, metanu). Ponadto pozwala na pomiar temperatury otoczenia oraz temperatur silników robota inspekcyjnego.

Zaprojektowany i zbudowany dla robota AmigoBot skaner laserowy 2D pełni funkcję systemu percepcyjnego robota mobilnego w zamkniętych lokalizacjach. Przewaga skanerów laserowych nad innymi sensorami, m.in. ultradźwiękowymi, w jakie wyposażony jest robot AmigoBot, polega na możliwości szybkiego pozyskiwania dokładnej informacji na temat otoczenia robota. Opracowany skaner laserowy 2D jest źródłem informacji w procesie budowania mapy otoczenia w środowisku szybkiego prototypowania.

3. WNIOSKI

Zaprezentowane autorskie środowisko szybkiego prototypowania umożliwia prowadzenie prac badawczych z wykorzystaniem rzeczywistych robotów AmigoBot oraz oprogramowania Matlab/Simulink. Dedykowane jest dla prac dotyczących testowania algorytmów sterowania ruchem zarówno pojedynczego robota jak i formacji. Integracja z pakietem Matlab/Simulink i jego rozbudowanymi narzędziami do modelowania oraz bezprzewodowa komunikacja z robotami są niekwestionowanymi zaletami omawianego oprogramowania. Możliwość odczytu parametrów oraz obsługa zewnętrznych urządzeń podłączonych do robotów umożliwiają testowanie opracowanych rozwiązań sprzętowych i programowych oraz integrację zewnętrznych sensorów w czasie rzeczywistym.

LITERATURA

1. Burghardt A., Hendzel Z.: Sterowanie behawioralne mobilnymi robotami kołowymi. Rzeszów: Ofic. Wyd. Pol. Rzesz., 2007.
2. Giergiel J., Szybicki D.: System Linux w robotyce. Rzeszów: Ofic. Wyd. Pol. Rzesz., 2010.
3. Hendzel Z., Żylski W., Burghardt A.: Autonomiczne mobilne roboty kołowe. Rzeszów: Ofic. Wyd. Pol. Rzesz., 2008.
4. Mrozek B., Mrozek Z., Matlab i Simulink.: poradnik użytkownika. Gliwice: Wyd. Helion, 2004.

**RAPID PROTOTYPING ENVIRONMENT FOR ALGORITHMS
OF MOBILE ROBOTS CONTROL**

Summary. The rapid prototyping environments allow for testing the developed software and hardware solutions at many stages of the research. The current work presents original software allowing controlling a group of up to 256 mobile robots manufactured by MobileRobots. The proposed solution was developed as the combination of chosen elements from software produced by Aria and the mex-function from Simulink. The current work presents the model research work carried out with the use of an original rapid prototyping environment, among other things being the results of working with mobile robot peripherals,

Pracę wykonano w ramach realizacji projektu badawczego nr N N501 068838 finansowanego ze środków na naukę w latach 2010-2012.