

METODY SZYBKIEGO PROTOTYPOWANIA ZE SPRZEŻENIEM ZWROTNYM ROBOTÓW MOBILNYCH Z WYKORZYSTANIEM HARDWARE IN THE LOOP

Głównym celem artykułu jest przedstawienie najbardziej rozbudowanych zdaniem autora programów komputerowych służących do symulacji zachowania robotów mobilnych. Skupiono się na trzech wybranych symulatorach, w których możliwa jest implementacja algorytmu dla każdego poruszającego się obiektu. Oprogramowanie zostało przedstawione pod kątem funkcjonalności oraz możliwości zastosowania w projekcie badawczym. Na koniec zaprezentowano koncepcję robota mobilnego z systemem wizyjnym, do budowy którego wykorzystano program Webots.

METHODS OF RAPID PROTOTYPING FOR MOBILE ROBOTS WITH FEEDBACK USING METHODOLOGY HARDWARE IN THE LOOP

The main goal of this article is to present the most extensive programs to simulate behavior of mobile robots. Author focused on three selected simulators, in which it is possible to implement the algorithm for each robot model. The software has been presented in terms of functionality and the ability to use in a research project. At the end presents a concept of mobile robot with vision system, which was used to build by the Webots program.

*Słowa kluczowe: robot mobilny, symulator, wirtualne środowisko
Keywords: mobile robot, simulator, virtual environment*

1. WPROWADZENIE

Budowa robotów mobilnych jest kompleksowym zadaniem. Jedno urządzenie integruje elektronikę, mechanikę oraz oprogramowanie. Utrudniona dostępność oraz cena wyspecjalizowanych podzespołów zwiększa koszty projektu oraz ogranicza jego częstą walidację. Z pomocą przychodzą aplikacje do realistycznego odwzorowania wyglądu oraz pracy robotów. Symulatory układów mechatronicznych znajdują zastosowanie przy tworzeniu modelu mechanicznego, układu sensorycznego oraz oprogramowania dla dowolnego robota. Przy tworzeniu projektu nie musimy się martwić o dostępność podzespołów czy miejsca na testowanie budowanego urządzenia. Budowany pojazd można złożyć z przygotowanych wcześniej modeli 3D, a układy sensoryczne – pobrać z bogatej bazy danych czujników. Symulacje służą do testowania programu dla robota, ale bez konieczności narażania na uszkodzenia rzeczywistego sprzętu podczas błędnego funkcjonowania programu. Artykuł w pierwszej części prezentuje wybrane programy symulujące odruchy i interakcje robotów w wirtualnym oraz rzeczywistym środowisku. Oprogramowanie zostało przedstawione pod kątem funkcjonalności oraz możliwości zastosowania w projekcie badawczym. Analizowano

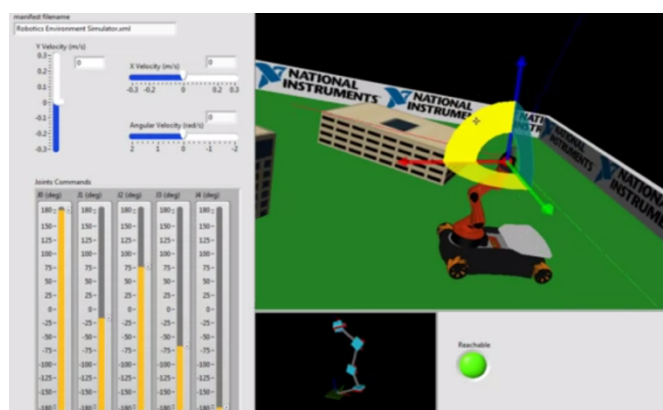
głównie budowę systemu, funkcjonalność techniczną oraz obszary zastosowań. W następnej części skupiono się na oprogramowaniu Webots, które posłuży do wykonania kolejnej wersji robota mobilnego z systemem wizyjnym.

2. PRZEGLĄD OTWARTEGO OPROGRAMOWANIA

Symulatory robotów mobilnych pozwalają głównie na: szybkie prototypowanie robota z użyciem modeli CAD, wykorzystanie silnika fizycznego w celu uzyskania realistycznych ruchów, nakładanie tekstur i renderowanie scen środowiska z robotem oraz implementację kodu źródłowego – zwykle takiego samego jak dla rzeczywistego pojazdu. Zbiór symulatorów dzieli się na programy darmowe i komercyjne. Do darmowych aplikacji zaliczają się: *Player Project*, *Simbad*, *UchiSim*, *SimRobot*. Wśród komercyjnych programów tego typu rozwijane są między innymi: *Webots*, *Microsoft Robotics Developer Studio*, *Virtual Robot Experimentation Platform*, *Cogmation robotSim*, *LabVIEW Robotics Simulator* czy *WorkCellSimulator*.

2.1. Labview robotics simulator

Oprogramowanie zaproponowane przez LabVIEW pozwala na efektywną walidację projektu zarówno pod względem oprogramowania dla przygotowanego modelu, jak i rozmieszczenia czujników. Zawiera biblioteki umożliwiające obsługę popularnych układów sensorycznych, elementów wykonawczych czy protokołów komunikacyjnych.



Rys. 1. Projekt środowiska dla robota KUKA wraz z graficznym interfejsem
Źródło: World Wide Web: topicnow.info

Pakiet zawiera narzędzia do analizy obrazu, implementacji regulatorów PID oraz projektowania interfejsu użytkownika do sterowania robotem. Symulator działa w oparciu o otwartoźródłowy silnik fizyki *Open Dynamics Engine* (<http://www.ode.org>) pozwalający na odwzorowanie dynamicznych ruchów robota. Program dzieli się na trzy sekcje:

- *CAD Model Importer* – służący do konwertowania modeli typu CAD,
- *Robot Simulation Model Builder* – pozwala na dodanie czujników oraz aktuatorów,
- *Robotics Environment Simulator Wizard* – umożliwia symulację w wybranym otoczeniu.

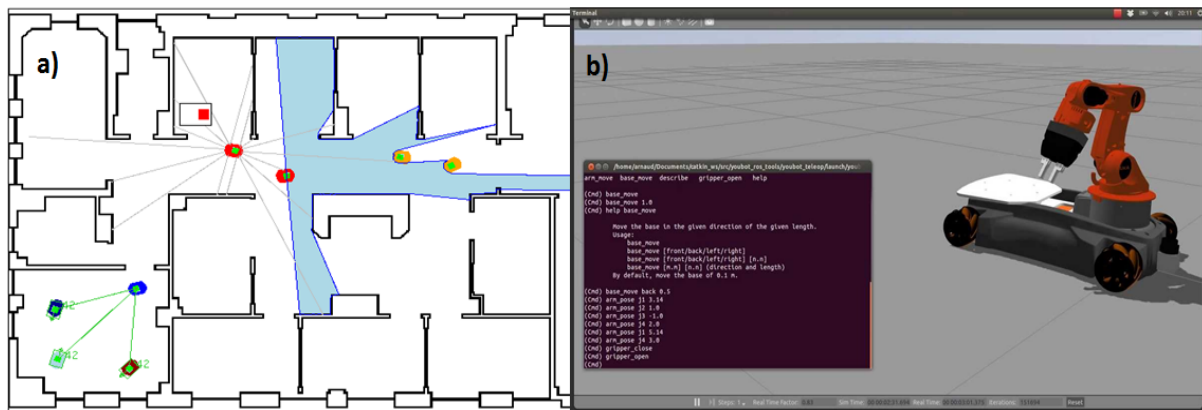
Ostatecznie *LabVIEW Robotics Simulator* pozwala na przeniesienie algorytmu z maszyny wirtualnej do rzeczywistego robota mobilnego z wbudowanym procesorem typu ARM, RT OS czy FPGA. Oszczędza to dużo czasu, podczas gdy podobnego algorytmu używamy do programowania różnych typów układów scalonych.

2.2. Project Player/Stage/Gazebo

Player Project jest jednym z najbardziej popularnych środowisk open source związanych z symulacją robotów mobilnych. Projekt powstał w 2000 roku na Uniwersytecie Południowej Kalifornii. Dzięki otwartej licencji wzbudził zainteresowanie wielu inżynierów, oferujących swój wkład w rozwój oprogramowania. Cały pakiet składa się z trzech współpracujących ze sobą części:

- Player – pełniący rolę serwera TCP/IP,
- 2D Stage – symulator dwuwymiarowy,
- 3D Gazebo – symulator trójwymiarowy.

Serwer Player działa na wszystkich popularnych systemach operacyjnych (Linux, Mac OS X, Windows) i służy głównie do sterowania robotem. Dzięki obsłudze TCP/IP możemy podłączyć wielu klientów jednocześnie, którymi mogą być zarówno wirtualne, jak i rzeczywiste maszyny. Przeniesienie systemu kontroli robotów pomiędzy wirtualnym a rzeczywistym nie wymaga modyfikacji kodu. Stage natomiast stanowi prosty symulator przestrzeni dwuwymiarowej, który dzięki mniejszym wymaganiom sprzętowym pozwala na symulację zachowania dużej liczby robotów. Zrezygnowano w nim ze szczegółowych modeli na rzecz wydajności aplikacji. Roboty są reprezentowane jako prostokąty, a posiadane przez nie cechy nie uwzględniają tarcia czy bezwładności. Dzięki takiemu uproszczeniu możliwe jest symulowanie kilkudziesięciu robotów na przeciętnym komputerze klasy PC.



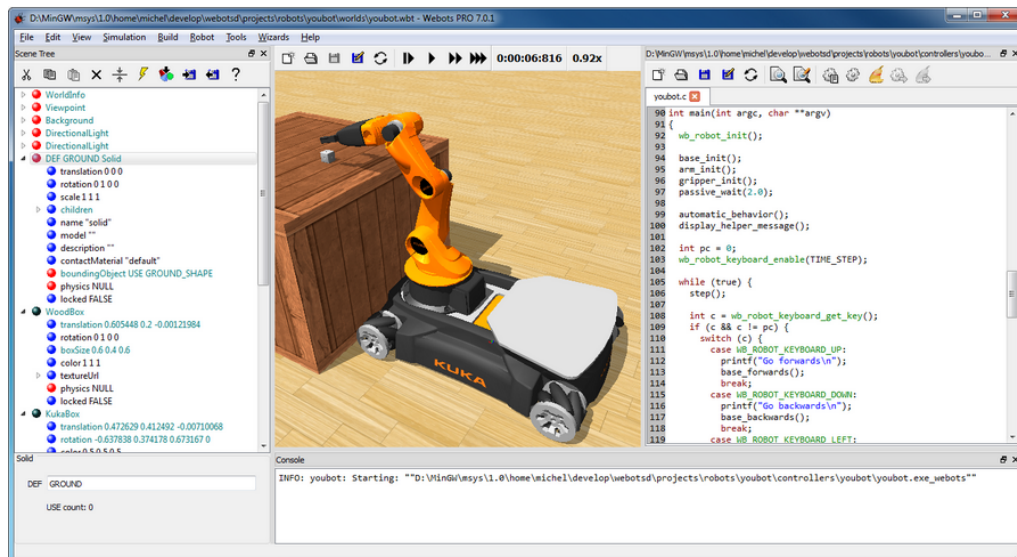
Rys. 2. a) Stage – zaprojektowane środowisko wraz z robotami różnego typu,
 b) Gazebo – model robota wraz z widocznym kodem źródłowym
 Źródło: World Wide Web: playerstage.sourceforge.net

Moduł Gazebo również umożliwia symulację jednego lub wielu robotów mobilnych. Różnica polega na tym, że symulowane środowisko jest trójwymiarowe, uwzględnia cechy fizyczne, dynamikę oraz oddziaływania między modelami. Pozwala na symulację standardowych sensorów takich jak: czujniki odległości, skaner 3D, kamery czy moduł GPS. Program

umożliwia tworzenie własnych modeli oraz importowanie. Ponadto daje możliwość przypisania do każdego modelu kontrolera, który będzie pozwalał na sterowanie elementem.

2.3. Symulator Webots

Środowisko Webots powstało w 1996 roku w Szwajcarii w celach rozwojowych. Obecnie program dostępny jest w wersji edukacyjnej oraz profesjonalnej. Wykorzystuje silnik fizyczny Open Dynamics Engine (ODE) w celu detekcji kolizji oraz symulowania ruchu. Wizualizacja odbywa się natomiast przy pomocy OpenGL. Program posiada rozbudowaną gamę bibliotek i rozszerzeń dla produkowanych obecnie układów sensorycznych takich jak: akcelerometry, kamery, dalmierze, moduł GPS, detektory światła i dotyku, mikrofony czy radiodbiorniki. Niestety nie przewidziano możliwości obsługi własnych czujników, dlatego wszelkie dodatkowe moduły mogą być implementowane jedynie przez producenta. Oprogramowanie dla robotów może być napisane w językach C, C++, Python, Matlab oraz Java, a także innych poprzez protokół TCP/IP. Interfejs programu podzielony jest na cztery okna, z których główne odpowiada za wizualizację fragmentu sceny. Istnieje możliwość wyboru perspektywy obserwacji oraz źródła obrazu z kamery.

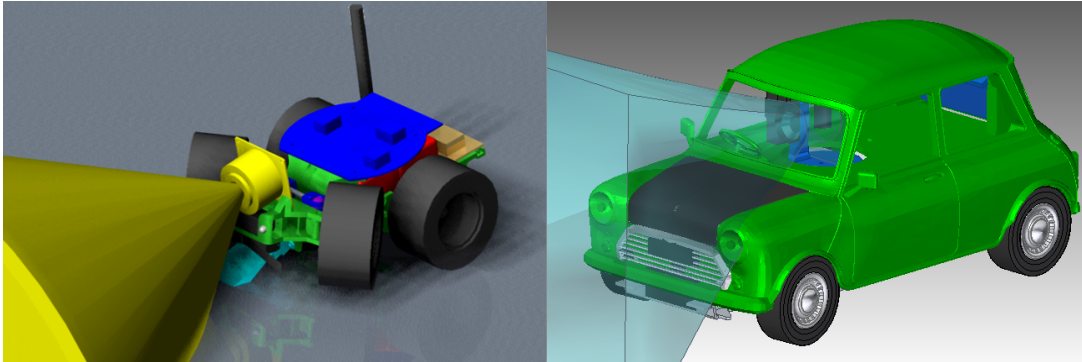


Rys. 3. Zaprojektowane środowisko dla robota mobilnego KUKA w programie Webots
Źródło: World Wide Web: cyberbotics.com

Oprogramowanie robota odpowiadające za zachowanie jest przypisane bezpośrednio do danego modelu. Każdy pojazd znajdujący się w wirtualnym środowisku może mieć przypisany własny indywidualny program. Webots umożliwia wgranie algorytmu bezpośrednio do rzeczywistego robota, a także utworzenie programu wynikowego, który może zostać zainstalowany przy pomocy zewnętrznego programatora. Użytkownik ma możliwość nadania kształtów, kolorów, tekstur, masy czy współczynnika tarcia dla każdego elementu, w celu jak najlepszego odwzorowania rzeczywistego zachowania. Symulator umożliwia renderowanie scen oraz nagrywanie symulacji do pliku wideo.

3. KONCEPCJA ROBOTA

Powyżej przedstawiono trzy wybrane środowiska umożliwiające symulowanie pracy robotów mobilnych. Do budowy kolejnego robota wykorzystano program Webots. Celem dalszej części pracy będzie wykonanie modelu i przeprowadzenie symulacji autonomicznego robota mobilnego w oparciu o uniwersalną i otwartą architekturę ROS (Robot Operating System) oraz symulator Webots. Nowy model będzie projektowany w oparciu o doświadczenia związane z budową poprzednich wersji robotów.



Rys. 4. Modele 3D poprzednich wersji robotów mobilnych z systemem wizyjnym
Źródło: opracowanie własne

Opracowany robot będzie pozwalał na sprzężenie z fizycznymi układami sensorycznymi i wykonawczymi. Projekt będzie integrował część mechaniczną z programową, co umożliwi wykonanie testów funkcjonowania w wielu konfiguracjach i w różnym środowisku.



Rys. 5. Model miejskiego pojazdu autonomicznego z kamerą w programie Webots
Źródło: opracowanie własne

Symulacja będzie wykorzystywać szczegółowo odwzorowane modele dynamiki robota oraz środowiska otoczenia. Tak przygotowana część mechaniczna i środowisko programowe ułatwi wprowadzenie zmian i przeniesienie projektu do rzeczywistego układu. Efektem wymiernym projektu będzie szczegółowy model autonomicznego robota mobilnego wraz z zaimplementowaną częścią logiczną. W celu sprawdzenia funkcjonowania przygotowanych zostanie wiele scenariuszy środowisk, w których robot miałby zastosowanie. Pozwoli to na

dobranie optymalnego układu sensorycznego, wykonawczego oraz kontrolera. Elementy mechaniczne pojazdu zostaną wydrukowane na drukarce 3D. Opis robota oraz wyniki badań użytkowych zostaną zawarte w sprawozdaniu z pracy badawczej.

LITERATURA

- [1] Chen I.Y., Macdonald B.A., Wünsche B.B.: *Mixed reality simulation for mobile robots*, New Zealand, 2009.
- [2] Dokumentacja Player [online – dostęp: 1.3.2015]: www.playerstage.sourceforge.net
- [3] Dokumentacja Webots [online – dostęp: 1.3.2015]: www.cyberbotics.com