

# System sterowania grupą robotów ligi Small Size Robot League

Mateusz Kalisch, Wawrzyniec Panfil

Politechnika Śląska, Instytut Podstaw Konstrukcji Maszyn

**Streszczenie:** W artykule przedstawiono oprogramowanie służące do sterowania drużyną robotów piłkarskich klasy Small Size Robot League. Projekt oprogramowania uwzględnia możliwość sterowania drużyną robotów rzeczywistych, których konstrukcja jest zgodna z regulaminem zawodów, a także sterowanie robotami w symulatorze. Podstawą sterowania grupą robotów są strategie bazujące na schemacie Game – Role – Skill. Badania weryfikacyjne oprogramowania i algorytmów przeprowadzono w środowisku symulacyjnym i w najbliższym czasie będą kontynuowane z udziałem rzeczywistych robotów.

**Słowa kluczowe:** roboty mobilne, systemy wielorobotowe, SSL RoboCUP

DOI: 10.14313/PAR\_213/90

Roboty mobilne są coraz powszechniej stosowane w przemyśle i w codziennym życiu człowieka. Mają coraz szersze zastosowanie w wielu dziedzinach, zaczynając od celów rozrywkowych, przez przemysł, aż do zadań zapewnienia bezpieczeństwa oraz działań militarnych. W przemyśle coraz częściej stosowane są wózki transportowe, których konstrukcja bazuje na kołach szwedzkich lub omniskierunkowych. To rozwiązanie pozwala na przemieszczanie się robota w dowolnym kierunku na płaszczyźnie bez potrzeby wykonywania obrotu. Ta cecha (holonomiczność) pomaga podczas poruszania się robota w ciasnej przestrzeni, np. w hali magazynowej. W podobny sposób skonstruowane są roboty piłkarskie należące do klasy *RoboCup Small Size Robot League* [10], gdzie niezależny ruch robota na płaszczyźnie boiska pozwala na dynamiczną zmianę położenia i skuteczniejszą realizację strategii gry. Realizacja ruchu robota wymaga oczywiście opracowania odpowiednich algorytmów sterujących.

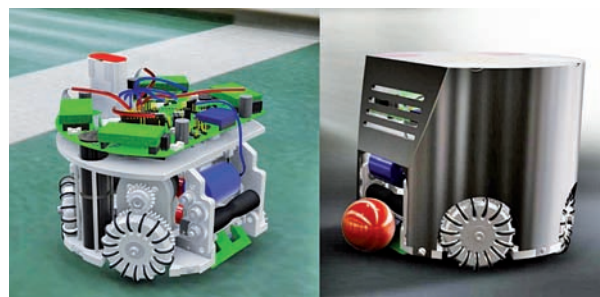
Na skuteczną grę drużyny robotów wpływa nie tylko sposób przemieszczania się robotów na boisku, ale także przyjęta strategia gry i sposób koordynacji, czy kooperacji zadań w systemie wielorobotowym. Zagadnienia te są uwzględniane nie tylko w sterowaniu drużynami robotów piłkarskich [8, 9], ale także podczas zarządzania grupami robotów wykonujących inne zadania [2–4, 6], np. inspekcyjne [5].

Sterowanie ruchem pojedynczego robota, jak również całą drużyną, oprócz algorytmów sterujących wymaga działania odpowiedniej aplikacji. Z racji tego, że żadna z drużyn nie udostępnia takich aplikacji, konieczne było opracowanie własnej. Niniejszy artykuł prezentuje m.in. pierwsze wyniki prac członków Studenckiego Koła Naukowego „Zastosowania Metod Sztucznej Inteligencji AI-METH” Politechniki Śląskiej, zmierzających do zbudowania drużyny robotów mogących w przyszłości wystartować w zawodach *Small Size Robot League* rozgrywek *RoboCup*. W pracy skupiono się na oprogramowaniu sterującym taką drużyną [1].

## 1. Small Size Robot League – SSL

Liga małych robotów piłkarskich pozwala na rozgrywanie meczów robotów o średnicy nie większej niż 180 mm i wysokości 150 mm. Roboty poruszają się zwykle (rys. 1) wykorzystując cztery koła omniskierunkowe (rzadziej trzy), natomiast do przytrzymywania i wybijania piłki służy układ kicker–dribbler. Roboty są sterowane zdalnie za pomocą aplikacji uruchomionej na komputerze należącym do danej drużyny.

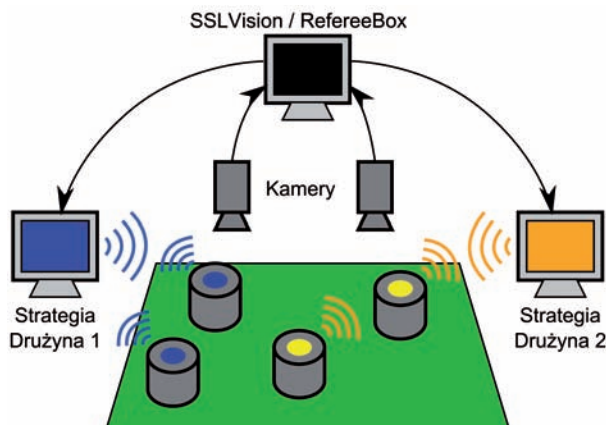
Ogólny schemat komunikacji elementów systemu SSL w trakcie rozgrywanego meczu przedstawiono na rys. 2. Pozycja robotów obliczana jest za pomocą systemu wizyj-



Rys. 1. Model CAD robota ligi SSL

Fig. 1. CAD model of the SSL robot

nego składającego się z dwóch kamer umieszczonych nad boiskiem i serwera, na którym jest uruchomiona aplikacja *SSLVision* [7] zajmująca się akwizycją i rozpoznawaniem obrazu. Informacje z systemu wizyjnego wysyłane są



**Rys. 2.** Schemat komunikacji w czasie meczu SSL  
**Fig. 2.** Scheme of communication during the SSL match

za pomocą protokołu UDP do komputerów obu drużyn, gdzie na ich podstawie podejmowane są decyzje o taktyce, a następnie w ostatnim kroku wysyłane są rozkazy sterujące do robotów. W skład danych wchodzi informacje o pozycji robotów oraz piłki, geometria boiska i dodatkowe dane dotyczące systemu wizyjnego. Algorytm sterujący drużyną robotów działa w pełni autonomicznie i w trakcie meczu nie jest możliwy wpływ na jego działanie. W trakcie meczu sędziowie obsługują aplikację sędziowską o nazwie RefereeBox. Aplikacja ta umożliwia zatrzymanie gry lub wymuszenie na każdej z drużyn określonego zachowania za pomocą dodatkowych rozkazów. Każda z drużyn musi uwzględniać te rozkazy w swoich algorytmach sterowania, aby roboty prawidłowo reagowały w przypadku wystąpienia jednej z sytuacji wyjątkowych, jak np. faul lub gol.

W rozgrywce bierze udział do sześciu robotów w każdej drużynie, jednak istnieje możliwość rozegrania meczu mniejszą liczbą robotów, jeżeli obie drużyny wyrażą taką wolę. Czas trwania jest ustalany przez zespoły przed rozpoczęciem meczu, najczęściej są to dwie połowy po 10 minut. W przypadku zakończenia drugiej połowy wynikiem remisowym, przewidziana jest w regulaminie dogrywka składająca się również z dwóch części, o odpowiednio krótszym czasie trwania. Podobnie jak w przypadku prawdziwego meczu, jeżeli dogrywka również zakończy się remisem, można rozegrać serię rzutów karnych w celu wyłonienia zwycięzcy.

## 2. Oprogramowanie sterujące drużyną robotów

Przed przystąpieniem do opracowania projektu oprogramowania, a następnie jego implementacji przyjęto szereg założeń, z których wiele wynika z regulaminu SSL oraz udostępnionych modułów oprogramowania, z którymi opracowywane oprogramowanie powinno być kompatybilne. Do najważniejszych założeń należy zaliczyć:

- język programowania C++,
- system operacyjny Linux,
- komunikacja z symulatorem *grSim* [11],
- kompatybilność z aplikacją sędziowską *RefereeBox* [12] oraz systemem wizyjnym *SSL-Vision* [7],

- opracowanie protokołu umożliwiającego komunikację z rzeczywistym(i) robotem(ami),
- różne sposoby sterowania ruchem robota (aplikacja użytkownika, klawiatura, kontroler gier),
- wizualizacja sytuacji na boisku w oknie aplikacji operatora,
- sterowanie z wykorzystaniem współrzędnych punktu docelowego.

### 2.1. Udostępnione oprogramowanie

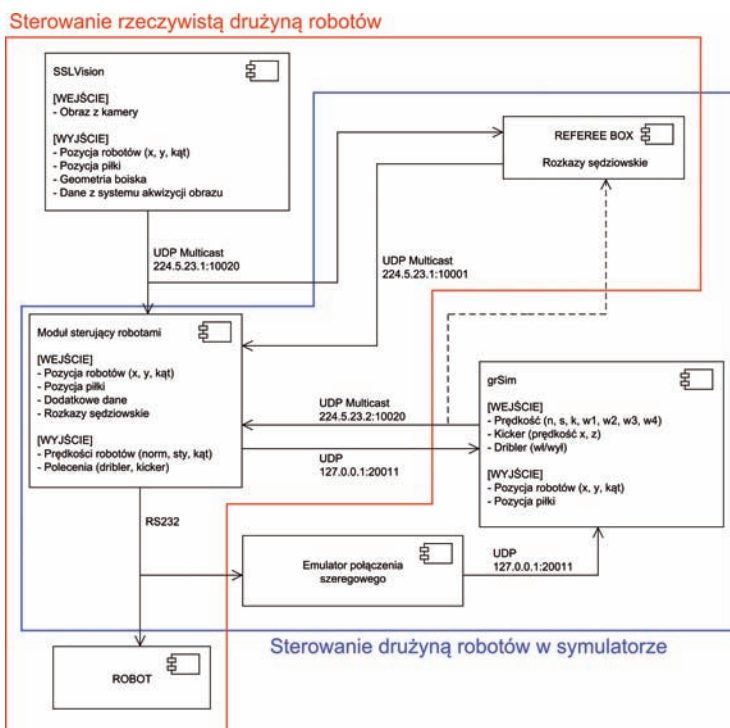
Ze względu na brak istniejącego rzeczywistego robota prace wykonywane w trakcie opracowywania oprogramowania były weryfikowane za pomocą symulatora.

Podczas realizacji prac skorzystano z następującego oprogramowania:

- Symulator *grSim* [11] – oprogramowanie, za pomocą którego zostały przeprowadzone badania weryfikacyjne algorytmów sterowania grupą i pojedynczymi robotami.
- Aplikacja sędziowska *RefereeBox* [12] – moduł oprogramowania, który jest obsługiwany przez sędziów i umożliwia za pomocą prostych rozkazów decydować o zatrzymaniu lub wznowieniu rozgrywki. Aplikacja ta została opracowana przez osoby zaangażowane w rozwój ligi z myślą o pełnej automatyzacji meczów i ma na celu zminimalizowanie udziału człowieka w procesie rozgrywki (ręczne usuwanie, rozstawianie robotów na boisku).
- System wizyjny *SSL-Vision* [7] – system wizyjny zamontowany nad boiskiem, którego zadaniem jest przesyłanie informacji o położeniu wszystkich robotów i piłki do komputerów drużyn. W niniejszej pracy przedstawiono wyniki badań weryfikacyjnych wykonanych za pomocą symulatora. Należy zaznaczyć, że opracowane oprogramowanie współdziała już z systemem *SSL-Vision*, dzięki czemu możliwe jest dalsze rozwijanie systemu i oprogramowania dla rzeczywistej grupy robotów.

### 2.2. Projekt systemu sterowania

Standardowa architektura systemu (rys. 3), podczas trwania zawodów ligi małych robotów piłkarskich, wymaga połączenia aplikacji sterującej z systemem wizyjnym w celu zebrania informacji o rozmieszczeniu robotów i piłki, połączenia z aplikacją sędziowską oraz połączenia z drużyną robotów. W celu umożliwienia weryfikacji algorytmu sterującego – bez konieczności budowania całej drużyny robotów, pełnowymiarowego boiska i kalibracji systemu wizyjnego – postanowiono dodać kolejny moduł pełniący funkcję symulatora sceny rozgrywek (boisko, piłka, roboty, bramki) i systemu wizyjnego, przez co komunikacja między aplikacją sterującą a symulatorem jest dwustronna. Docelowo algorytm ma współpracować z rzeczywistymi robotami za pomocą protokołu połączenia szeregowego, dlatego dodano kolejny element pełniący funkcję emulatora pozwalającego na testowanie połączenia szeregowego i porównania poprawności jego działania z protokołem wysyłającym rozkazy do środowiska symulacyjnego. Oficjalna aplikacja sędziowska musi być obsługiwana w pełnym zakresie przez sędziów, nie ma więc zaimplementowanych żadnych mechanizmów pracujących autonomicznie. Dlatego też, na potrzeby projektu, zbudowano nowy *RefereeBox*, będący rozbudowaną wersją



**Rys. 3.** Projekt systemu sterowania drużyną robotów SSL

**Fig. 3.** Project of SSL control system

oficjalnego programu z dodatkowymi funkcjami pozwalającymi na autonomiczną pracę bez ingerencji człowieka.

Elementem docelowo tworzonym i rozwijanym przez drużyny konstruktorów jest *Moduł sterowania robotami* (rys. 4). W jego skład wchodzi następujące elementy:

- Tracker – predykcja danych z systemu wizyjnego,
- Moduł budowania strategii,
- Moduł wyznaczania ścieżki ruchu robota,
- Moduł wizualizacji.

Dane z systemu wizyjnego są odbierane bezpośrednio przez moduł *Tracker*, który przygotowuje dane wykorzystywane następnie w module strategii. Do jego zadań należy między innymi predykcja pozycji piłki i robotów w przypadku odebrania niekompletnych danych z systemu wizyjnego, niwelowanie opóźnień powstałych w wyniku akwizycji obrazu i transmisji informacji o pozycji robotów, piłki oraz geometrii boiska, a także usuwanie zakłóceń i szumu wynikających z procesu akwizycji obrazu. *Moduł budowania strategii* odbiera dane z *Trackera* oraz z aplikacji sędziowskiej, na ich podstawie decyduje o przydziale zadań poszczególnym robotom na boisku. Jeżeli wśród zadań pojawia się rozkaz przemieszczenia robota do innego punktu na boisku, to współrzędne tego punktu są wysyłane do *Algorytmu wyznaczającego ścieżkę ruchu robota*, który zwraca zestaw punktów należących do ścieżki, którą powinien podążać robot, aby ominąć wszystkie przeszkody. W końcowej fazie algorytm uwzględnia punkty w ścieżce i wysyła bezpośrednio do robotów rozkazy mówiące o ich prędkości w kierunku normalnym i stycznym oraz kątową prędkość obrotu robota. Moduł wizualizacji nie jest bezpośrednio połączony z żadnym z modułów, korzysta ze zmiennych dostępnych w środowisku aplikacji sterującej i na tej

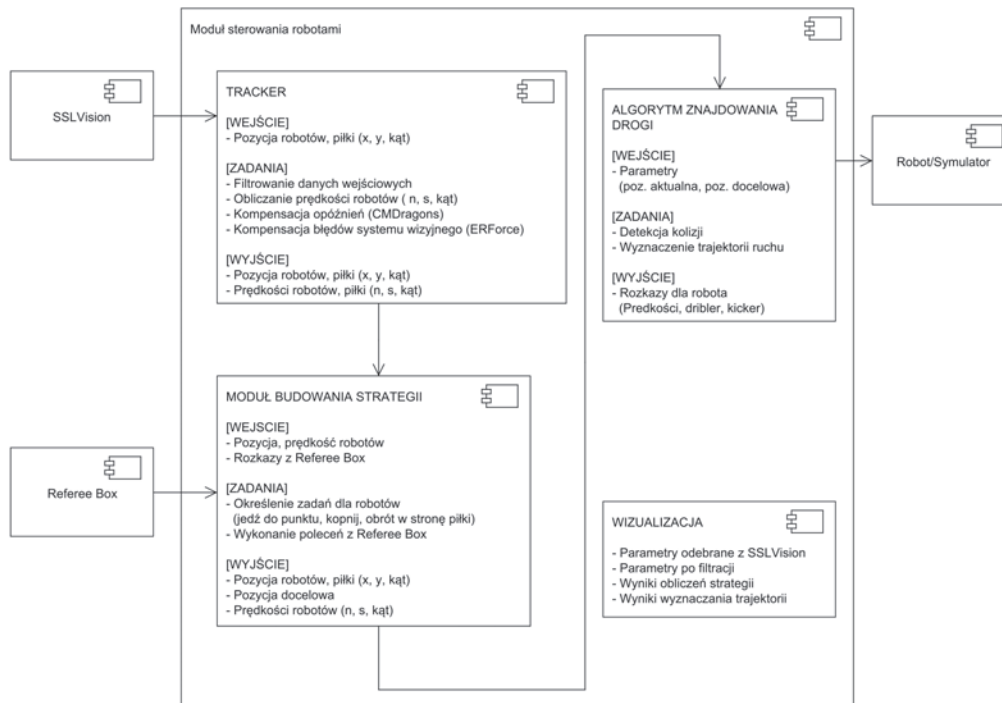
podstawie wyświetla informację na ekranie w formie graficznej lub tekstowej.

### 2.3. Strategia sterowania drużyną robotów

Po przeanalizowaniu informacji udostępnionych przez zespoły startujące w zawodach postanowiono zrealizować koncepcję przyjętą przez większość drużyn – kaskadową metodę budowania strategii (rys. 5) opartą na trzech krokach:

- Strategia (Gra),
- Role,
- Zadania (Umiejętności).

Na podstawie ogólnej sytuacji na boisku, w module strategii podejmowana jest decyzja o taktyce gry całej drużyny, czyli określane jest, czy drużyna przyjmuje postawę defensywną, neutralną czy ofensywną. W tym miejscu analizowane są również rozkazy płynące z aplikacji sędziowskiej, ponieważ tego typu polecenia wynikają z regulaminu gry i mają nadrzędny priorytet. Role przypisywane robotom definiują jego zachowanie w danej sytuacji na boisku. Wśród ról można wymienić „Napastnik”, która oznacza, że ten konkretny robot ma piłkę i jego priorytetowym zadaniem jest zdobycie punktu. Inną rolą może być „Obróńca”, którego zadaniem jest przeszkadzanie w zdobyciu punktu przez przeciwną drużynę. To, jaka rola zostanie przypisana do zawodnika, zależy od zastosowanej metody przydziału. Drużyny należące do aktualnej czołówki zawodów stosują różne koncepcje. Wśród niżej plasujących się w rankingu zespołów można zaobserwować naśladowanie rozwiązań stosowanych w zespołach odnoszących już sukcesy. Zadania stanowią najniższy poziom algorytmu sterowania, a ich celem jest obliczenie wartości podstawowych parametrów



**Rys. 4.** Schemat modułu algorytmu sterującego drużyną robotów  
**Fig. 4.** Scheme of SSL team control algorithm

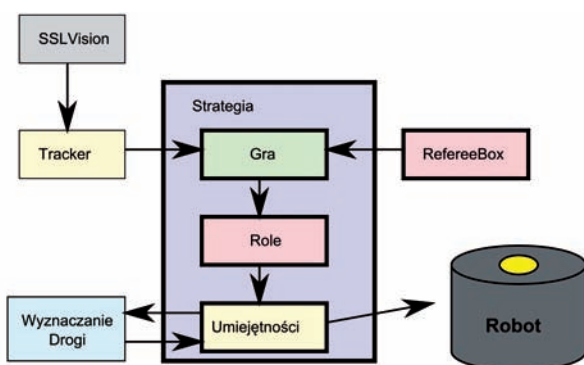
sterujących robotami, jak np. prędkość ruchu robota. Docelowo moduł ma określić miejsce, w którym ma się znaleźć robot oraz czas, po jakim to nastąpi, co przekłada się na jego prędkość. Znając jego rolę moduł musi przygotować pozostałe podzespoły zawodnika do jej wykonania, np. jeżeli rolą robota jest oczekiwanie na podanie i strzał na bramkę, to powinien przygotować odpowiedni ładunek energii w kondensatorach i ustawić się pod odpowiednim kątem w stosunku do podającego i bramki. Jest to system z koordynatorem, dlatego wszystkie elementy strategii (gra, role, umiejętności) są realizowane na komputerze klasy PC, na którym uruchomiona jest odpowiednia aplikacja sterująca. Umiejętności znajdujące się na najniższym poziomie strategii, odpowiadają za wysyłanie określonego zbioru param-

trów do każdego z robotów sterowanej drużyny. Wśród tych parametrów znajdują się informacje o prędkości liniowej oraz kątowej robota, rozkaz dotyczący stanu driblera (włączony lub wyłączony), informacja o sile i rodzaju kopnięcia.

Ten trójstopniowy algorytm jest stosowany przez wiele zespołów, a na pewno przez drużyny odnoszące największe sukcesy w zawodach, dlatego postanowiono zbudować koncepcję systemu sterującego bazując na tym schemacie. Wśród wszystkich opisanych metod doboru ról do zawodnika, zdecydowano skupić się w pierwszej kolejności na koncepcji stosowanej przez zespół *Odens*, opierającej się na tabelarycznym zapisie reguł. Ta metoda została bardzo dobrze opisana, dodatkowo jest stosunkowo prosta w stosowaniu oraz umożliwia modyfikację strategii w bardzo szerokim zakresie.

Opracowując algorytm sterowania autorzy skorzystali jedynie ze sposobu zapisywania strategii zespołu *Odens* [9], natomiast role oraz definiujące je umiejętności zostały opracowane samodzielnie. Poniżej zestawiono zaproponowane role, a w nawiasach podano umiejętności charakterystyczne dla tych ról:

- Główny napastnik (Odbierz podanie, Jedź do piłki, Kopnij, Kopnij górą, Podaj do kopnięcia),
- Napastnik/Pomocnik (Wspomagaj napastnika, Odbierz podanie),
- Ofensywny pomocnik (Jedź do piłki, Podaj górą, Podaj dołem, Blokuj podanie, Podaj do kopnięcia, Wspomagaj Napastnika),
- Defensywny pomocnik (Jedź do piłki, Blokuj podanie, Blokuj robota 1, Blokuj robota 2),
- obrońca (Odbierz podanie, Broń bramki, Blokuj robota 1, Blokuj robota 2).



**Rys. 5.** Ogólny schemat komunikacji modułu budowania strategii z pozostałymi elementami systemu SSL

**Fig. 5.** General scheme of communication between strategy building module and the rest of SSL system

## 2.4. Aplikacja użytkownika

Elementem scalającym system sterowania drużyną robotów piłkarskich jest aplikacja użytkownika (wraz z interfejsem). Aplikacja pełni kilka funkcji, z których najważniejsze to:

- Konfiguracja systemu – możliwe jest konfigurowanie wymiarów boiska, parametrów systemu komunikacji, profili prędkości robotów, parametrów modułu testowania umiejętności, konfiguracji strategii (w tym ról i umiejętności),
- Wizualizacja aktualnej sytuacji na boisku – obserwowanie położenia i ruchu robotów oraz piłki, na bieżąco rysowane są ścieżki ruchu robotów, wektory prędkości robotów, wolne strefy między robotem posiadającym piłkę a bramką przeciwnika,
- Sterowanie ręczne wybranym robotem,
- Uruchamianie trybu autonomicznego sterowania drużyną robotów.

## 3. Badania weryfikacyjne

W pierwszej kolejności przeprowadzono badania weryfikacyjne działania aplikacji. Sprawdzano, czy prawidłowo działają wszystkie moduły oprogramowania, system komunikacji, czy między modułami przesyłane są właściwe dane. Następnie dokonano weryfikacji działania umiejętności. Polegało to na wyznaczeniu wartości miar skuteczności działania umiejętności dla różnych wartości parametrów definiujących

**Tab. 1.** Wyniki rozgrywek dla różnych konfiguracji strategii: a) neutralna–neutralna, b) ofensywna–neutralna, c) defensywna–neutralna, d) defensywna–ofensywna

**Tab. 1.** Matches results of different strategy types: a) neutral–neutral, b) offensive–neutral, c) defensive–neutral, d) defensive–offensive

	a) 1 połowa		2 połowa			b) 1 połowa		2 połowa	
	N	N	N	N		O	N	O	N
M1	6	6	9	9	2	4	5	7	
M2	5	5	9	9	2	5	4	8	
M3	3	4	7	9	3	6	7	11	
M4	5	3	10	7	5	7	10	11	

	c) D		N			d) D		O	
	D	N	D	N		D	O	D	O
M1	5	2	12	5	2	4	9	6	
M2	4	6	10	9	6	2	12	7	
M3	1	3	6	6	4	4	10	8	
M4	5	2	11	7	4	1	11	4	

**Tab. 2.** Ogólne wyniki dla testowanych strategii

**Tab. 2.** General results of each strategy

Strategia	Wygrane	Remis	Przegrane
Defensywna	7	1	0
Neutralna	4	1	3
Ofensywna	0	0	8

cych te umiejętności. Przykładowo, dla umiejętności *Kopnij* modyfikowano takie parametry jak margines kątowy (umożliwiający wykonanie kopnięcia) oraz prędkość obrotu robota wokół piłki, natomiast miarą skuteczności – procent celnych strzałów na bramkę.

Główne badania weryfikacyjne przeprowadzono dla oprogramowania, w szczególności algorytmów sterujących zachowaniem drużyny. Badanie to polegało na tym, że w pierwszej kolejności zdefiniowano trzy strategie gry drużyny (defensywna, neutralna i ofensywna), a następnie przeprowadzono testy w symulatorze dla różnych konfiguracji tych strategii (tab. 1). Strategie zostały skonfigurowane zgodnie z przyjętym schematem zespołu Odens [9]. Rozróżnienie strategii wynika z różnych ról przypisanych robotom dla podobnych sytuacji na boisku, a także modyfikowaniu wag umiejętności definiujących poszczególne role. Definiowanie strategii następowało na drodze doświadczalnej na podstawie obserwacji zachowań użytkownika. Badania prowadzono dla rozgrywek składających się z dwóch 5-minutowych połów. Wymiary boiska przyjęto zgodnie z regulaminem SSL. W tabeli 1 zestawiono wyniki rozegranych meczów. Dla każdej z kombinacji dwóch strategii rozegrano cztery mecze oznaczone w tabeli skrótami M1, M2, M3 i M4. W samej tabeli zawarto wynik meczu w postaci zdobytych przez każdą z drużyn punktów.

Na podstawie rozegranych meczów opracowano ranking strategii (tab. 2), w którym zebrano liczby wygranych, zremisowanych i przegranych meczów przez drużyny robotów. Analiza wyników badań weryfikacyjnych strategii pokazuje, że najskuteczniejszym typem strategii, spośród wszystkich testowanych, jest strategia defensywna. Na osiem rozegranych meczów, siedem z nich (gdzie stosowano strategię defensywną) zostało wygranych, a jeden zakończył się remisem.

Kolejną strategią, charakteryzującą się dużą liczbą wygranych meczów, jest strategia neutralna. Spośród ośmiu zrealizowanych meczów, cztery zostały wygrane, jeden zremisowany a trzy przegrane. Drużyna sterowana strategią neutralną zyskała znaczną przewagę nad drużyną ofensywną. Drużyna stosująca strategię ofensywną nie mogła wygrać żadnego meczu, przez co ta strategia została oceniona najgorzej, ponieważ wszystkie rozegrane mecze zostały przegrane.

Analizując wszystkie wyniki, można zauważyć, że jednym z najistotniejszych elementów strategii jest obrona. Najskuteczniejszą okazała się drużyna dysponująca największą liczbą robotów defensywnych pomagających bramkarzowi w obronie bramki.

## 4. Podsumowanie

W ramach badań powstały zarówno aplikacja jak i algorytm pozwalające na sterowanie drużyną robotów piłkarskich klasy Small Size Robot League. Opracowane moduły oprogramowania zostały przygotowane w sposób umożliwiający sterowanie drużyną robotów, zarówno w środowisku symulacyjnym jak i w warunkach rzeczywistych. Jako podstawę opracowania strategii zachowań drużyny zastosowano schemat *Game – Role – Skill*. Przeprowadzone w środowisku symulacyjnym badania weryfikacyjne wykazały, że najkorzystniejsza jest strategia defensywna, trochę mniej neutralna, a najmniej ofensywna. Zdaniem autorów na podstawie obserwacji zachowania drużyny robotów w trakcie gry można wywnioskować, że przewaga strategii defensywnej wynika z tego, że roboty większość czasu – w stosunku do pozostałych strategii – przebywają w pobliżu pola karnego swojej drużyny. Istotnym elementem dalszej pracy jest przetestowanie algorytmów sterujących na grupie robotów w warunkach rzeczywistych. Dotychczasowe badania weryfikacyjne zostały przeprowadzone w warunkach wirtualnych, dlatego ważne jest, aby porównać zachowania drużyn robotów rzeczywistych z wirtualnymi i uwzględnić różnice działania między nimi. Inną ścieżką rozwoju oprogramowania będzie opracowanie modułu, który w sposób adaptacyjny do zachowania drużyny przeciwnej mógłby modyfikować parametry algorytmów sterujących tak, aby możliwa była zmiana strategii w trakcie gry.

## Bibliografia

1. Kalisch M., *Algorytmy autonomicznego sterowania grupą robotów ligi Small Size Robot League*, Praca Dyplomowa Magisterska, Politechnika Śląska, Instytut Podstaw Konstrukcji Maszyn, Gliwice 2013.
2. Skrzypczyński P., *Multi-agent software architecture for autonomous robots: a practical approach*, "Management and Production Engineering Review", Vol. 1, No. 4, 55–66, 2010.
3. Dahl T.S., Mataric M., Sukhatme G.S., *Multi-robot task allocation through vacancy chain scheduling*, "Robotics and Autonomous Systems", Vol. 57, Issues 6–7, 674–687, 2009.
4. Zhang Y., Parker L.E., *Multi-Robot Task Scheduling*, Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2013.
5. Panfil W., *System sterowania grupą inspekcyjnych robotów mobilnych bazujący na zachowaniach*, Politechnika Śląska, Katedra Podstaw Konstrukcji Maszyn, Zeszyt 145, Gliwice 2012.
6. Turek W., Cetnarowicz K., Zaborowski W., *Software Agent Systems for Improving Performance of Multi-Robot Groups*, "Fundamenta Informaticae", Vol. 112, 103–117, IOS Press, Amsterdam 2011.
7. Zickler S., Laue T., Birbach O., Wongphati M., Veloso M., *SSL-Vision: The Shared Vision System for the RoboCup Small Size League*, Springer 2009, 425–436.
8. Zickler S., Biswas J., Luo K., Veloso M., *CMDragons 2010 Team Description*, <http://robocupssl.cpe.ku.ac.th/tdp/2010/cmdragons.pdf>.
9. Fukumoto Y., Nakajima M., Masutani Y., *ODENS 2011 Team Description*, <http://robocupssl.cpe.ku.ac.th/tdp/2011/Odens TDP 2011.pdf>.
10. RoboCup Small Size League: SSL Web Site., <http://small-size.informatik.uni-bremen.de>, 2013.
11. Parsian: RoboCup Small Size Robot Soccer Simulator. <https://github.com/mani-monaj/grSim>.
12. Referee Box Users Manual. <http://robocupssl.cpe.ku.ac.th/referee:manual>. ■

## Control System of the Small Size Robot League Robots Group

**Abstract:** The paper presents the software intended for controlling the robots team of the Small Size Robot League. The software project takes into account the possibility of control of the robots team both in the 3D simulator and in the real conditions. The *Game – Role – Skill* schema serves as the foundation of the team's control algorithms. Verification tests of the software and the control algorithms have been carried-out in the simulator and in the near future they will take place in the real conditions.

**Keywords:** mobile robots, multi-robot systems, SSL RoboCUP

Artykuł recenzowany, nadesłany 16.05.2014 r., przyjęty do druku 17.10.2014 r.

---

### mgr inż. Mateusz Kalisch

Jest asystentem naukowym w Instytucie Podstaw Konstrukcji Maszyn Politechniki Śląskiej. Pracę magisterską obronił na Wydziale Mechanicznym Technologicznym Politechniki Śląskiej w 2013 r.  
*e-mail: mateusz.kalisch@polsl.pl*



### dr inż. Wawrzyniec Panfil

Jest adiunktem w Instytucie Podstaw Konstrukcji Maszyn Politechniki Śląskiej. W 2011 r. obronił rozprawę doktorską dotyczącą sterowania grupą inspekcyjnych robotów mobilnych. Jego zainteresowania obejmują zagadnienia robotyki mobilnej (w szczególności systemów wielorobotowych), a także zastosowań poszerzonej rzeczywistości w diagnostyce i eksploatacji maszyn.  
*e-mail: wawrzyniec.panfil@polsl.pl*

