

Gra planszowa z robotem o trzech stopniach swobody

Wiktor Bolek
Marcin Goszczyński

Opisano zastosowanie robota o trzech stopniach swobody, sterowanego za pomocą sterownika FPC 101, do gry z człowiekiem w „Kółko i krzyżyk” na planszy 3x3. Algorytm rozgrywki jest oparty na wiedzy eksperta, która składa się z kilku reguł. Algorytm zaimplementowano w języku STL. Zaprogramowany w ten sposób robot przeprowadza bezbłędna rozgrywkę.

Artykuł jest poświęcony opisowi zastosowania robota o trzech stopniach swobody, sterowanego przez PLC, do przeprowadzenia gry planszowej, jaką jest gra w „Kółko i krzyżyk” na planszy 3 x 3. Przeciwnikiem robota w tej grze jest człowiek. Przeciwnicy na przemian ustawiają swoje pionki na planszy. Wygrywa ten, który pierwszy ustawi swoje trzy pionki w jednym rzędzie.

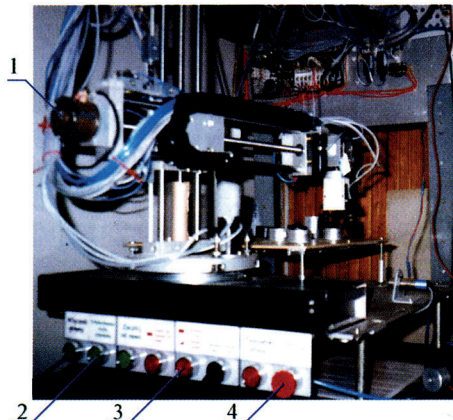
Od robota wymaga się precyzyjnego przenoszenia elementów stanowiących pionki w grze, wykrycia posunięcia wykonanego przez człowieka i wykorzystania każdego taktycznego błędu w celu odniesienia zwycięstwa, ponieważ prawidłowa rozgrywka kończy się remisem. Algorytm rozgrywki robota nie może dopuszczać do sytuacji zwycięstwa człowieka, co będzie potwierdzeniem poprawności algorytmu sterowania, który stanowi zapis sztucznej inteligencji w sterowniku PLC.

Opis stanowiska

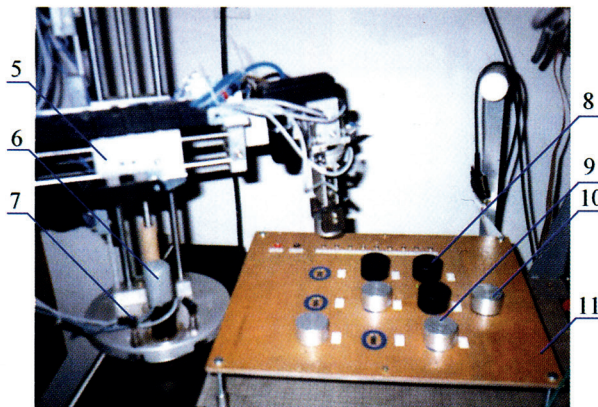
Stanowisko laboratoryjne składa się z robota, sterownika PLC, rozbudowanego o dodatkowy moduł sterowania silnikami krokowymi[1]. Układ kinematyczny robota [5] składa się z jednego przegubu obrotowego napędzanego silnikiem prądu stałego oraz z dwóch przegubów liniowych napędzanych silnikami krokowymi. Chwytnik jest zasilany pneumatycznie, ponadto w skład stanowiska wchodzi: urządzenia transmisyjno-sprzęgające, komputer klasy IBM PC, sprężarka oraz plansza i pionki. Pulpit stanowiska jest wyposażony w lampki i przyciski, których zadaniem jest komunikowanie się z operatorem.

Na rys. 1 i 2 przedstawiono robota w trakcie rozgrywki z człowiekiem. Rozgrywka odbywa się na specjalnie przygotowanej planszy. Na jej powierzchni zaznaczono dziesięć pól, z których dziewięć wyposażono w mechaniczne dwustanowe przełączniki rozłożone na powierzchni kwadratu (rys. 3). Plansza służąca do rozgrywki stanowi zbiór sygnałów wejściowych dla robota. Postawienie pionka na dowolnym polu k1 – k9, jest sygnalizowane „1” logiczną na wejściu PLC i po zarejestrowaniu w odpowiednim kroku programu nastąpi ustawienie odpowiedniej flagi przypisującej to pole człowiekowi lub robotowi.

Dr inż. Wiktor Bolek jest pracownikiem Zakładu Automatykacji Procesów Przemysłowych w Instytucie Techniki Ciepłej i Mechaniki Płynów Politechniki Wrocławskiej. Mgr inż. Marcin Goszczyński jest tegorocznym absolwentem Wydziału Mechaniczno-Energetycznego Politechniki Wrocławskiej.



Rys. 1. Stanowisko robota (widok z przodu): 1 – silnik krokowy, 2 – przyciski, 3 – lampki sygnalizacyjne, 4 – wyłącznik



Rys. 2. Stanowisko robota (widok z boku) 5 – wózek suportowy, 6 – silnik prądu stałego, 7 – czujnik położenia stołu obrotowego, 8 – pion człowieka, 9 – pion komputera, 10 – magazyn pionów komputera, 11 – plansza

K1	K2	K3
K4	K5	K6
K7	K8	K9

Rys.3. Sposób numeracji pól planszy

Strategia rozgrywki

Wiedza eksperta sprowadza się do kilku prostych reguł [3]. W trakcie gry rozróżnia się sześć różnych sytuacji, które uporządkowano według hierarchii ważności.

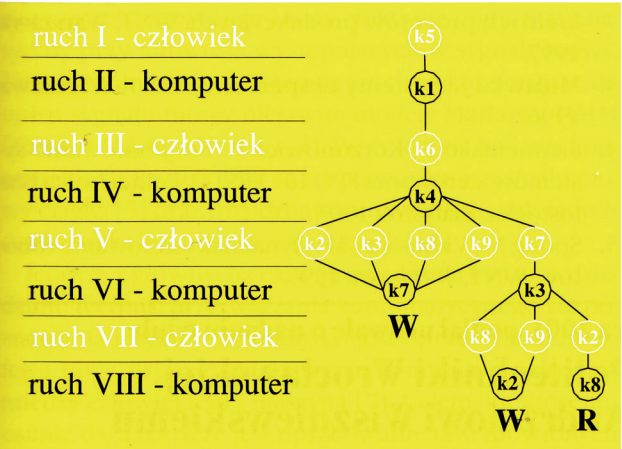
1. wygrana własna,

- 2. wygrana przeciwnika,
- 3. wolny środek,
- 4. sytuacje szczególne A,
- 5. wolny lewy górny róg,
- 6. sytuacje szczególne B.

Poprawność przedstawionych reguł zweryfikowano analizując wszystkie możliwe rozgrywki. Różne przebiegi rozgrywki można przedstawić graficznie za pomocą tzw. drzewa gry. W węzłach podaje się pola, na które został wykonany ruch. Kolorem czerwonym zaznaczono ruch wykonany przez człowieka, a czarnym – ruch wybrany przez algorytm. Ruchy robota są wybierane w oparciu o wiedzę eksperta. Przyjmuje się natomiast, że człowiek może wykonywać ruch na dowolne pole. Litery R i W na końcu drzewa oznaczają odpowiednio remis lub wygraną robota. Na rys. 4 przedstawiono fragment jednego z drzew po rozpoczęciu przez człowieka i postawieniu przez niego pionków w kolejno na pola k5 i k6. Liczbę i możliwości zakończenia wszystkich gier przedstawia tabl. 1. Inne sytuacje nigdy nie wystąpią w przypadku wykorzystania przedstawionego algorytmu.

Tablica 1. Liczba gier wynikająca z przyjętej strategii

	Rozpoczęcie przez		Dwa przypadki łącznie
	komputer	człowieka	
Wygrana komputera — W	102	393	495
Remis — R	6	128	134
Wygrana człowieka	0	0	0
Suma możliwych gier	108	521	629



Rys.4. Fragment przykładowego drzewa gry rozpoczynanej przez człowieka

Prawidłowości dobranej strategii dowodzi liczba przypadków dająca zwycięstwo człowiekowi, która wynosi 0. Dodatkowo, gdyby przyjąć prawdopodobieństwo każdej sytuacji za jednakowe, to komputer wygrywa w 78,7 %, a remisuje tylko w 21,3 %. Potwierdza to wykorzystanie powstałych w czasie gry błędów człowieka. Należy pamiętać jednak, że stworzenie strategii dającej zwycięstwo z dobrze grającym człowiekiem jest niemożliwe, ponieważ bezbłędna rozgrzywka zawsze kończy się remisem, co z kolei udowodniono przez niemożliwość pokonania komputera.

Budowa programu

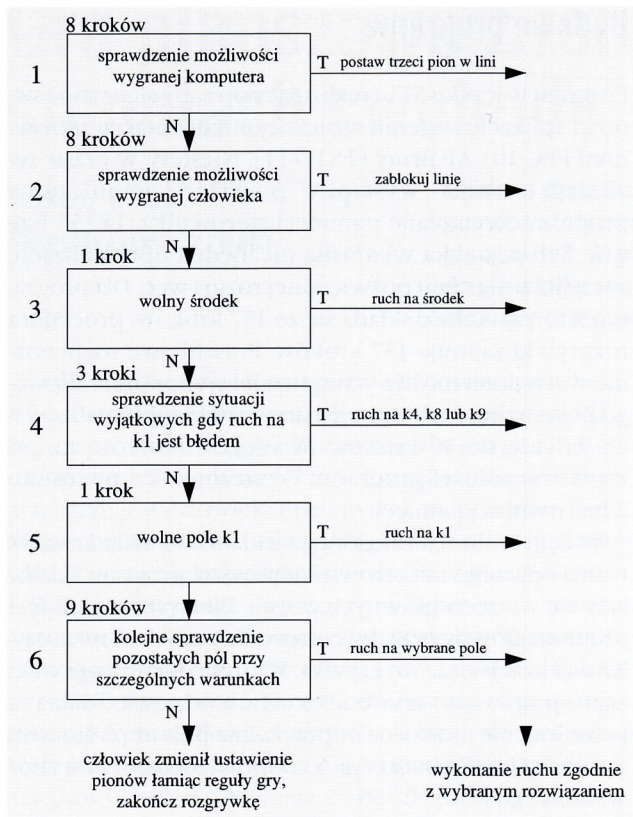
Program w języku STL (realizujący opisany algorytm) stanowi aplikację systemu sztucznej inteligencji na sterowniku FPC 101 AF firmy FESTO [1]. Niestety, w czasie realizacji zadania wystąpiły poważne ograniczenia programu wywołane pamięcią sterownika, 12255 bajtów. Sytuacja taka wymusiła niezbędną optymalizację w części programu poświęconej rozgrywce. Dla programu, którego całość składa się ze 157 kroków, procedura rozgrywki zajmuje 137 kroków. Prawidłowe rozpoznanie wszystkich możliwych sytuacji i wybór prawidłowego pola ruchu, czyli strategia gry, została zoptymalizowana jedynie do 30 kroków. W efekcie końcowym, po zapisaniu całości programu do sterownika, pozostało 2 bajty wolnej pamięci.

W celu realizacji założonego zadania dla układu sterowania opartego na sterowniku powstał program składający się z trzech głównych części. Pierwszą z nich jest program główny oraz dwie procedury użytkownika, czyli inicjalizacja oraz rozgrzywka. W procedurze rozgrywki zaprogramowano sekwencje ruchów odpowiedzialną za przeniesienie pionka na odpowiednie pole oraz algorytm sztucznej inteligencji (rys. 5) odpowiedzialny za wybór właściwego pola.

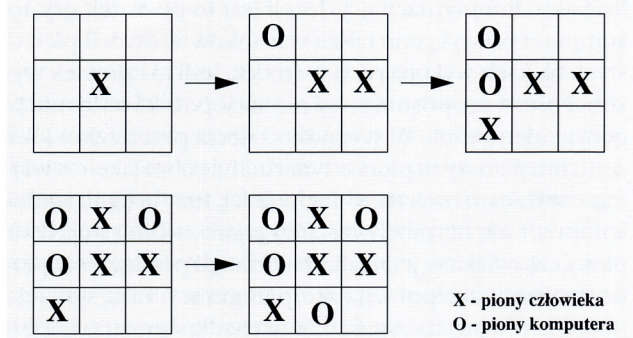
Robot po pobraniu pionka z magazynu wykona ruch na jedno z 9 pól planszy. Pole to zostanie wybrane po analizie aktualnej sytuacji gry. Jeżeli jest to początek gry, to komputer po wykryciu fałszu warunków w dwóch pierwszych blokach wykona ruch w środek. Jeśli to człowiek wykonał pierwsze posunięcie i nie zajął pola k5 to sytuacja będzie identyczna. W wypadku zajęcia przez człowieka centrum planszy w pierwszym ruchu, robot jako rozwiązanie wybierze ruch na wolne pole k1. Przy drugim ruchu komputer nie ma możliwości wygrania, a jeżeli stoją dwa pionki człowieka w jednej linii to nastąpi postawienie pionka komputera zapobiegające przegranej. Taka sytuacja jest zilustrowana na rys. 4 i 6. W wypadku wystąpienia fałszu zapisanego warunku zostaną zanalizowane w algorytmie sytuacje wyjątkowe zapobiegające popełnieniu błędu przez postawienie pionka na pole k1.

Opisane w szóstym bloku kombinacje odpowiadają za postawienie pionka, którego celem jest zaatakowanie. Atak oznacza sytuację, gdy w linii jednego pionka komputera są dwa wolne pola, a dzięki sekwencyjnej analizie warunków wcześniej odrzucone zostały możliwości bezpośredniego zwycięstwa i ataku człowieka. Nie spełnienie żadnego z wyżej zapisanych warunków oznacza modyfikację ustawienia pionów na planszy w czasie gry przez człowieka i zakończenie działania procedury rozgrywki.

Jako kryterium optymalizacji strategii wybrano ilość bajtów potrzebnych do jej zapisania, a w zastosowanej regułowej reprezentacji wiedzy wybrano konteksty. Jako przykład przedstawiono trzy sytuacje, gdy robot może wygrać wykonując ruch na k7. Sprawdzenie iloczynów flag pól k3 i k5 lub k1 i k4, lub k8 i k9 w jednym kroku odpowiadającym za ruch na wspólne wygrywające pole jest obrazem przedstawionych uproszczeń.



Rys. 5. Algorytm strategii rozgrywki



Rys. 6. Przykład bezbłędnej rozgrywki zakończonej remisem po ruchu robota

Podsumowanie

W pracy przedstawiono zastosowanie sztucznej inteligencji w sterowniku FPC 101 do sterowania robota grającego z człowiekiem w „Kółko i krzyżyk”.

Program sterujący robota rozpoznaje wszystkie sytuacje powstające podczas rozgrywki i wykonuje ruch. Każdorazowa odpowiedź układu nie różni się niczym dla zewnętrznego obserwatora od tej, jaką udzieliłby dobrze grający człowiek, który nigdy nie przegrywa. Sytuacja taka dowodzi zapisania sztucznej inteligencji w postaci programu sterownika, również na podstawie rzeczywiście przeprowadzonych rozgrywek.

Ograniczenia są spowodowane zbyt małą pamięcią sterownika dla przyjętych założeń. Efektem tego jest brak wykrywania sytuacji awaryjnych. Człowiek podejmujący grę i mający przeciwnika w postaci robota musi przestrzegać reguł gry, np.: usunięcie z planszy pionu komputera w czasie rozgrywki nie spowoduje jej przerwania, a aktywna flaga przypisana do tego pola wciąż będzie miała wartość „1” logicznej. Prostem i skutecznym rozwiązaniem pozwalającym na udoskonalenie stanowiska może być zastosowanie drugiego sterownika i powierzenia mu jedynie funkcji kontroli. Zastosowanie najnowszego sterownika PLC pozwoliłoby również całkowicie wyeliminować to ograniczenie.

Bibliografia:

1. Eberhardt M., Schule R.: Positioner Remote Module – Handbuch. FESTO Didactic, 1990.
2. Mikulczyński T., Samsonowicz Z.: Automatyzacja dyskretnych procesów produkcyjnych. WNT, Warszawa 1997.
3. Mulawka J.: Systemy ekspertowe. WNT, Warszawa 1996.
4. Siemieniako F., Korzunowicz P.: Programowanie systemów sterowania FPC 101 FESTO. Politechnika Białostocka, Białystok 1993.
5. Spong M., Vidyasagar M.: Dynamika i sterowanie robotów. WNT, Warszawa 1997.

Gra planszowa z robotem o trzech stopniach swobody,
Marcin Goszczyński, Wiktor Bolek — s. 26

Przedstawiono zastosowanie robota o trzech stopniach swobody, sterowanego sterownikiem FPC 101, do gry z człowiekiem w grę „Kółko i krzyżyk” na planszy 3 x 3. Algorytm rozgrywki oparto na bazie wiedzy eksperta składającej się z kilku reguł. Algorytm zaimplementowano w języku STL. Zaprogramowany robot przeprowadza bezbłędną rozgrywkę.

Panel game against the robot with three degrees of freedom,
Marcin Goszczyński, Wiktor Bolek — p. 26

The robot with three degrees of freedom, controlled by the FPC 101 controller, plays the tictactics game with the 3 x 3 game board against the human. The robot's algorithm for plying game bases on the expert knowledge, described in few rules. The algorithm has been implemented in the STL language. The robot controlled by the mentioned algorithm makes perfect moves.