

Marcin SZUSTER¹

ROZMYTE STEROWANIE BEHAWIORALNE MOBILNYM ROBOTEM KOŁOWYM W NIEZNANYM ŚRODOWISKU

W artykule zaprezentowano hierarchiczny układ sterowania ruchem mobilnego robota kołowego w nieznanym środowisku ze statycznymi przeszkodami. Układ sterowania składa się z generatora trajektorii realizującego sterowanie behawioralne z zastosowaniem układów z logiką rozmytą oraz neuronowego algorytmu sterowania ruchem nadążnym, w którym zastosowano algorytm aproksymacyjnego programowania dynamicznego. W warstwie planowania trajektorii ruchu zrealizowano sterowanie behawioralne typu „podażaj do celu” oraz „omijaj przeszkody”. W prezentowanym hierarchicznym układzie sterowania sygnały sterowania warstwy planowania trajektorii są generowane przez dwa układy z logiką rozmytą, w których zastosowano model Takagi-Sugeno. W warstwie realizacji ruchu zastosowano algorytm aproksymacyjnego programowania dynamicznego w konfiguracji dualnego heurystycznego programowania dynamicznego, zbudowany z dwóch struktur: aktora i krytyka. Struktury aktora i krytyka zostały zrealizowane z zastosowaniem sztucznych sieci neuronowych. Poprawność zaproponowanych rozwiązań zweryfikowano numerycznie, stosując emulator mobilnego robota kołowego oraz laboratoryjnego toru pomiarowego, zrealizowane w środowisku obliczeniowym Matlab/Simulink.

Słowa kluczowe: aproksymacyjne programowanie dynamiczne, generator trajektorii, sieci neuronowe, sterowanie nadążne, układy z logiką rozmytą

1. Wprowadzenie

Ciągły rozwój przemysłu, poprawa efektywności oraz możliwości sterowania układów wykonawczych robotów, dostępność coraz bardziej złożonych układów sensorycznych, wzrost mocy obliczeniowej mikroprocesorów czy pojemności magazynów energii umożliwiają budowę bardziej zaawansowanych mobilnych robotów kołowych (MRK). Jednocześnie możliwe staje się zastosowanie MRK w zadaniach do tej pory niedostępnych dla tego typu konstrukcji, a znaczny wzrost zainteresowania środowisk naukowych oraz biznesowych potwierdza istotność zagadnień związanych z tematyką projektowania i sterowania

¹ Autor do korespondencji/corresponding author: Marcin Szuster, Politechnika Rzeszowska, al. Powstańców Warszawy 12, 35-959 Rzeszów, tel.: (17) 8651854, e-mail: mszuster@prz.edu.pl

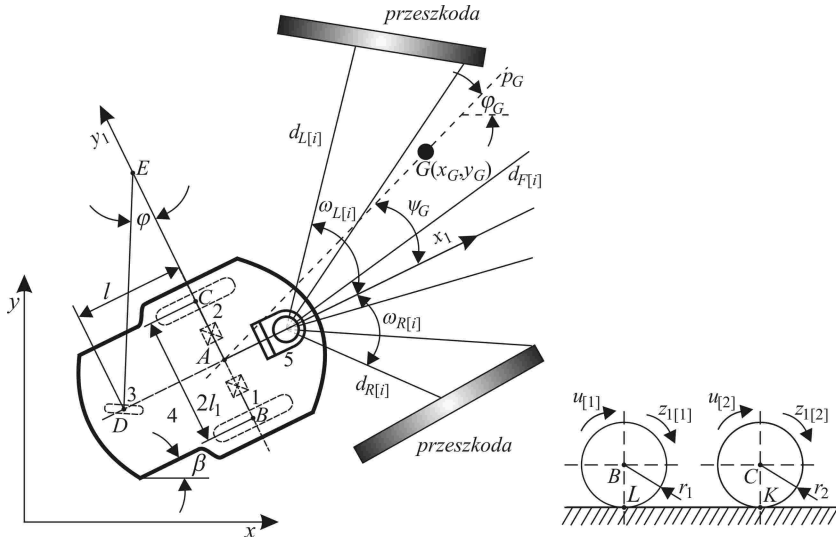
MRK. Autonomiczne MRK mogą być stosowane w wielu zadaniach, takich jak transport, eksploracja nieznanego środowiska, patrolowanie granic czy magazynów. Realizacja tego typu zadań wymaga rozwiązania wielu problemów, z których jednymi z najbardziej istotnych są: generowanie trajektorii ruchu MRK w czasie rzeczywistym na podstawie sygnałów z układu sensorycznego robota, umożliwiające orientację w środowisku ruchu oraz jednoczesna realizacja trajektorii ruchu.

W literaturze można znaleźć wiele metod generowania trajektorii ruchu MRK [1-9], wśród których występuje podział na metody globalne i lokalne. W metodach globalnych proces planowania trajektorii ruchu odbywa się na podstawie znajomości mapy otoczenia robota. W metodach lokalnych przyjęto założenie, że MRK porusza się w nieznanym środowisku, o którym ograniczonej informacji dostarcza układ sensoryczny robota. Przykładem metody lokalnej generowania trajektorii ruchu MRK jest sterowanie behawioralne [3, 5, 6], inspirowane zachowaniami obserwowanymi w świecie zwierząt. Pojęcie to obejmuje grupę metod sterowania umożliwiających realizację elementarnych zachowań, takich jak: „podażaj do celu” (ang. *Goal Seeking* – GS) czy „omijaj przeszkody” (ang. *Obstacle Avoiding* – OA). W algorytmach planowania trajektorii i realizacji ruchu MRK chętnie są stosowane nowoczesne metody sztucznej inteligencji (ang. *Artificial Intelligence* – AI), takie jak układy z logiką rozmytą (ULR) [5, 6], sztuczne sieci neuronowe (SN) [3] czy algorytmy aproksymacyjnego programowania dynamicznego (APD) [10], których idea opiera się na programowaniu dynamicznym Bellmana, a implementacja stała się możliwa przez zastosowanie SN.

W artykule zaprezentowano hierarchiczny układ sterowania ruchem MRK, umożliwiający planowanie trajektorii ruchu w zadaniach typu GS i OA oraz jej jednoczesną realizację. Przedstawione wyniki badań naukowych stanowią kontynuację wcześniejszych prac autora, dotyczących generowania trajektorii ruchu [10,11], czy sterowania ruchem MRK z zastosowaniem metod AI [12,13]. Artykuł składa się z następujących części: po krótkim wprowadzeniu, w części 2. przedstawiono obiekt sterowania. Część 3. zawiera opis hierarchicznego układu sterowania, a część 4. – szczegółowe omówienie warstwy planowania trajektorii ruchu. Warstwy realizacji ruchu przedstawiono w części 5. artykułu. Wyniki przeprowadzonych testów numerycznych procesu generowania i realizacji trajektorii ruchu MRK zamieszczono w części 6. Część 7. podsumowuje artykuł.

2. Mobilny robot kołowy

Obiektem sterowania jest MRK, schematycznie pokazany na rys. 1. w nieznanym otoczeniu. Sterowany obiekt porusza się przez zastosowanie niezależnego napędu kół 1 i 2, koło 3 jest swobodnym kołem podpierającym, 4 to rama, 5 – skaner laserowy przestrzeni, będący częścią układu sensorycznego robota, który umożliwia detekcję przeszkód. Założono ruch MRK w płaszczyźnie xy.



Rys. 1. Schemat mobilnego robota kołowego w nieznanym środowisku

Fig. 1. Scheme of the wheeled mobile robot in the unknown environment

Oznaczenia występujące na rys. 1. są następujące: $u_{[1]}$, $u_{[2]}$ – sygnały sterowania ruchem nadążnym MRK, $z_{1[1]}$, $z_{1[2]}$ – dyskretny kąt obrotu kół napędowych MRK, r_1 , r_2 – promienie kół napędowych, l , l_1 – wymiary wynikające z geometrii ramy, β – kąt obrotu ramy, $A(x_A, y_A)$ – punkt leżący na przecięciu osi kół napędowych i osi ramy MRK, $G(x_G, y_G)$ – cel ruchu MRK o współrzędnych x_G i y_G , p_G – prosta przechodząca przez punkty A i G , ψ_G – kąt zawarty między osią ramy MRK a prostą p_G , ϕ_G – kąt zawarty między osią x a prostą p_G , $d_{L[i]}$, $d_{F[i]}$, $d_{R[i]}$ – pomiary odległości do przeszkód realizowane przez układ sensoryczny MRK, przyporządkowane odpowiednio do grupy pomiarów po lewej stronie ramy, przed MRK i po prawej stronie ramy, $i = 1, 2, 3$, $\omega_{L[i]}$, $\omega_{F[i]}$, $\omega_{R[i]}$ – kąty zawarte między osią ramy MRK a osiami poszczególnych pomiarów odległości.

Odległość do celu G jest wyznaczana jako długość odcinka $|AG|$. W celu wyznaczenia bezkolizyjnej trajektorii ruchu MRK są obliczane minimalne odległości do przeszkód po lewej stronie ramy MRK, $d_{Lm} = \min(d_{L[1]}, d_{L[2]}, d_{L[3]})$, analogicznie przed MRK (d_{Fm}) oraz po prawej stronie ramy (d_{Rm}). Minimalna odległość do przeszkód wokół robota jest wyznaczana na podstawie minimalnych odległości poszczególnych grup pomiarów: $d_O = \min(d_{Lm}, d_{Fm}, d_{Rm})$.

Dynamiczny model ruchu MRK wyprowadzono, stosując formalizm Magiego [14, 15]. Otrzymano równanie w formie:

$$\mathbf{M}\ddot{\mathbf{a}} + \mathbf{C}(\dot{\mathbf{a}})\dot{\mathbf{a}} + \mathbf{F}(\dot{\mathbf{a}}) + \boldsymbol{\tau}_d = \mathbf{u} \quad (1)$$

gdzie: $\dot{\mathbf{a}}$ – wektor prędkości kątowych kół napędowych, $\dot{\mathbf{a}} = [\dot{\alpha}_{[1]}, \dot{\alpha}_{[2]}]$, \mathbf{M} – stała macierz bezwładności, $\mathbf{C}(\dot{\mathbf{a}})\dot{\mathbf{a}}$ – wektor momentów pochodzących od sił odśrodkowych i Coriolisa, $\mathbf{F}(\dot{\mathbf{a}})$ – wektor oporów ruchu, $\boldsymbol{\tau}_d$ – wektor ograniczonych zakłóceń. Ciągły model dynamiki MRK zdyskretyzowano, stosując metodę Eulera, ponieważ w algorytmie sterowania zastosowano dyskretne algorytmy APD. Przyjęto wektor stanu $\mathbf{z}_{\{k\}} = [\mathbf{z}_{1\{k\}}, \mathbf{z}_{2\{k\}}]^T$, gdzie $\mathbf{z}_{2\{k\}}$ odpowiada ciągłemu wektorowi $\dot{\mathbf{a}}$. Otrzymano dyskretny opis dynamiki MRK w postaci:

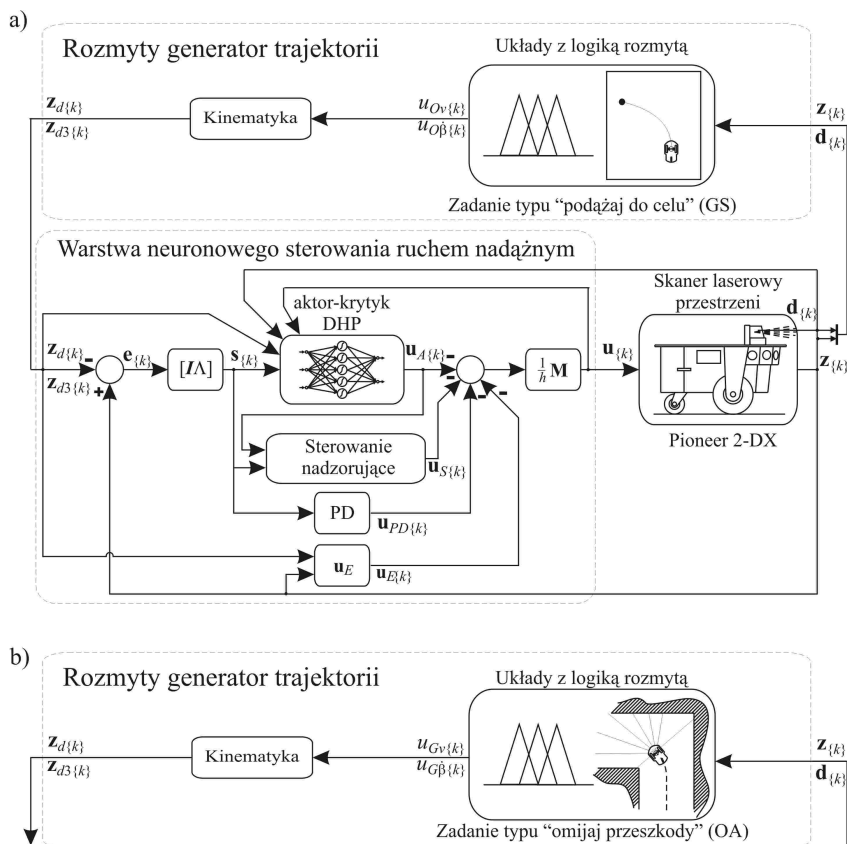
$$\left. \begin{aligned} \mathbf{z}_{1\{k+1\}} &= \mathbf{z}_{1\{k\}} + h\mathbf{z}_{2\{k\}} \\ \mathbf{z}_{2\{k+1\}} &= \mathbf{z}_{2\{k\}} - h\mathbf{M}^{-1} \left[\mathbf{C}(\mathbf{z}_{2\{k\}})\mathbf{z}_{2\{k\}} + \mathbf{F}(\mathbf{z}_{2\{k\}}) + \boldsymbol{\tau}_{d\{k\}} - \mathbf{u}_{\{k\}} \right] \end{aligned} \right\} \quad (2)$$

gdzie: k – indeks kroków iteracji, h – parametr dyskretyzacji czasu.

W dalszej części artykułu pozostałe zmienne również będą przedstawiane w postaci dyskretnej z indeksem k .

3. Hierarchiczny układ sterowania

Hierarchiczny układ sterowania ruchem MRK składa się z dwóch warstw: warstwy generowania trajektorii ruchu oraz warstwy sterowania ruchem nadążnym. Warstwę nadrzędną hierarchicznego układu sterowania stanowi generator trajektorii, którego zadaniem jest generowanie w czasie ruchu robota zadanej trajektorii ruchu, realizowanej przez niższą warstwę układu sterowania. Algorytm planowania trajektorii ruchu zrealizowano z zastosowaniem dwóch ULR. Zadaniem warstwy realizacji ruchu jest wytworzenie sygnałów sterowania modułami napędowymi robota, zapewniających realizację wygenerowanej trajektorii ruchu. W układzie sterowania nadążnego zastosowano algorytmy APD w konfiguracji dualnego heurystycznego programowania dynamicznego (DHP). Schemat hierarchicznego układu sterowania ruchem MRK, zastosowanego do realizacji zadania typu GS zamieszczono na rys. 2a. Rysunek 2b przedstawia schemat warstwy generowania trajektorii ruchu w zadaniu typu OA. W hierarchicznym układzie sterowania ruchem MRK zastosowano metody AI, takie jak ULR w warstwie generowania trajektorii czy algorytmy APD w konfiguracji DHP, zrealizowane z zastosowaniem SN w warstwie realizacji ruchu. Poszczególne warstwy układu sterowania zostały szczegółowo opisane w dalszej części artykułu.



Rys. 2. a) Schemat hierarchicznego układu sterowania ruchem mobilnego robota kołowego w zadaniu typu „podążaj do celu”, b) schemat warstwy generowania trajektorii ruchu w zadaniu typu „omijaj przeszkody”

Fig. 2. a) Scheme of the hierarchical control system of the wheeled mobile robot movement in the „goal seeking” task, b) scheme of the layer of the trajectory generating in the „obstacle avoiding” task

4. Rozmyty generator trajektorii

W prezentowanym artykule w warstwie generowania trajektorii ruchu zastosowano koncepcję sterowania behawioralnego. Koncepcja ta opiera się na obserwacji świata zwierząt i naśladowaniu ich prostych zachowań w celu realizacji określonych zadań przez MRK. Idea sterowania behawioralnego obejmuje zachowania elementarne typu GS czy OA, w których zadaniem algorytmu sterowania jest wygenerowanie sygnałów sterowania przeliczanych na podstawie równań kinematyki obiektu sterowania na zadane parametry ruchu. W przypadku MRK w warstwie planowania trajektorii ruchu są wyznaczone wartości dwóch sygnałów sterowania. Pierwszy z nich steruje zadaną prędkością ruchu

MRK (prędkością pkt A ramy, $v_{A\{k\}}$), natomiast drugi steruje zadaną prędkością kątową obrotu ramy. Realizacja zadania typu GS polega na generowaniu sygnałów sterowania warstwy planowania trajektorii ruchu minimalizujących odległość do celu $d_{G\{k\}}$ oraz kąt $\psi_{G\{k\}}$. Znajomość informacji z układu sensorycznego MRK nie jest w tym przypadku istotna. Jeżeli w otoczeniu robota wystąpiłaby przeszkoda, to realizacja zadania typu GS mogłaby doprowadzić do kolizji. W zadaniu typu OA jest generowana bezkolizyjna trajektoria ruchu MRK, przez ruch ramy robota środkiem wolnej przestrzeni, gdzie sygnał sterowania prędkością kątową obrotu ramy robota jest wyznaczany na podstawie minimalnych odległości do przeszkód po prawej i lewej stronie robota, odpowiednio $d_{Lm\{k\}}$ i $d_{Rm\{k\}}$. Wartość sygnału sterowania prędkością ruchu ramy robota zależy od odległości do przeszkód wokół robota $d_{O\{k\}}$. Położenie końcowe ramy MRK w zadaniu typu OA nie jest zdefiniowane, natomiast do prawidłowej realizacji tego zadania konieczne jest pozyskiwanie aktualnej informacji o stanie środowiska z układu sensorycznego robota.

Sterowanie behawioralne typu „podążaj do celu”

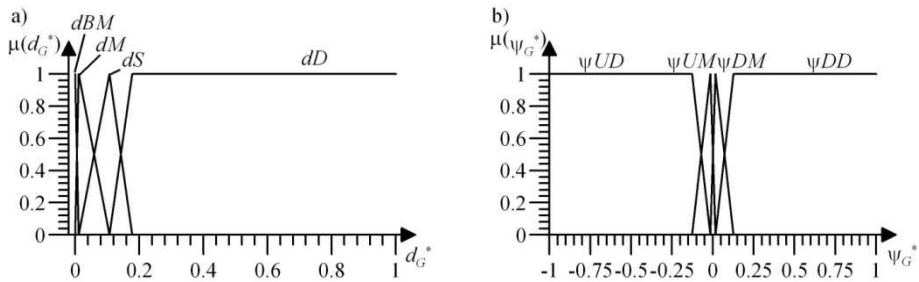
Warstwa generowania trajektorii ruchu w zadaniu typu GS została zrealizowana z zastosowaniem dwóch ULR, w których użyto modelu Takagi-Sugeno, z trójkątnymi bądź trapezowymi funkcjami przynależności do zbiorów rozmytych przesłanek. W konkluzjach zastosowano funkcje przynależności typu singleton. Pierwszy z ULR generuje sygnał sterowania zadaną prędkością pkt A ramy $u_{Gv\{k\}}$ na podstawie odległości $d_{G\{k\}}$. Baza reguł została dobrana w taki sposób, aby generowany sygnał sterowania miał maksymalną wartość, gdy odległość do celu jest duża. Trajektoria ruchu MRK jest wtedy generowana z maksymalną zadaną prędkością v_A^* . Zmniejszanie odległości do celu powoduje nieliniowe zmniejszanie wartości sygnału sterowania $u_{Gv\{k\}}$ do zera, gdy odległość do celu $d_{G\{k\}}$ jest bardzo mała. Drugi ULR generuje sygnał $u_{G\beta\{k\}}$, sterujący zadaną prędkością kątową ramy MRK $\dot{\beta}$. Sygnał ten jest generowany w taki sposób, aby realizacja wyznaczonej trajektorii powodowała minimalizację kąta $\psi_{G\{k\}}$. Przy małej wartości kąta $\psi_{G\{k\}}$ generowana będzie zadana trajektoria ruchu MRK w kierunku celu. Przykładowo, sygnał sterowania $u_{G\beta\{k\}}$ jest generowany na podstawie bazy $m = 4$ reguły typu:

$$R_{G\beta}^m : \text{JEŻELI } \psi_{G\{k\}}^* \text{ JEST } \psi UD \text{ TO } u_{G\beta\{k\}} \text{ JEST } u UD \quad (3)$$

gdzie: $\psi_{G\{k\}}^*$ – znormalizowany kąt zawarty między osią ramy MRK a prostą p_G , ψUD – etykieta lingwistyczna funkcji przynależności do zbioru rozmytego przesłanek, $u UD$ – etykieta lingwistyczna funkcji przynależności konkluzji typu singleton (o wartości $u_{BM} = 0$, $u_M = 0,5$, $u_S = 0,9$, $u_D = 1$ dla pierwszego

ULR oraz $uUD = -1$, $uUM = -0,5$, $uDM = 0,5$, $uDD = 1$ dla drugiego ULR), UD – ujemny duży, UM – ujemny mały, DM – dodatni mały, DD – dodatni duży, BM – bardzo mały, M – mały, S – średni, D – duży.

Funkcje przynależności do zbiorów rozmytych ULR warstwy generowania trajektorii ruchu w zadaniu typu GS przedstawiono na rys. 3., gdzie $d_{G\{k\}}^*$ – znormalizowana odległość do celu G , $\mu(\cdot)$ – stopień spełnienia funkcji przynależności do zbioru rozmytego.



Rys. 3. Funkcje przynależności do zbiorów rozmytych przesłanek pierwszego układu z logiką rozmytą (a), funkcje przynależności do zbiorów rozmytych przesłanek drugiego układu z logiką rozmytą (b)

Fig. 3. Membership functions of premises of the first fuzzy logic system (a), membership functions of premises of the second fuzzy logic system (b)

Sygnały sterowania warstwy generowania trajektorii ruchu w zadaniu typu GS są przeliczane na wartości zadanych prędkości kątowych kół MRK zgodnie z równaniem kinematyki:

$$\begin{bmatrix} z_{d2[1]\{k\}} \\ z_{d2[2]\{k\}} \end{bmatrix} = \frac{1}{r} \begin{bmatrix} v_A^* & l_1 \dot{\beta}^* \\ v_A^* & -l_1 \dot{\beta}^* \end{bmatrix} \begin{bmatrix} u_{Gv\{k\}} \\ u_{G\dot{\beta}\{k\}} \end{bmatrix} \quad (4)$$

gdzie: $r = r_1 = r_2$, $\dot{\beta}^*$ – maksymalna zadana wartość prędkości kątowej obrotu ramy MRK.

Na podstawie wartości zadanych prędkości kątowych obrotu kół 1 i 2 są obliczane kąty obrotu kół oraz przyspieszenia kątowe. Wygenerowana trajektoria jest realizowana przez układ sterowania ruchem nadążnym.

Sterowanie behawioralne typu „omijaj przeszkodę”

Warstwa generowania bezkolizyjnej trajektorii ruchu w zadaniu typu OA również została zrealizowana z zastosowaniem dwóch ULR. Pierwszy z nich generuje sygnał sterowania zadaną prędkością pkt A ramy MRK: $u_{Ov\{k\}}$, na podstawie znormalizowanej minimalnej odległości do przeszkód wokół robota

$d_{O\{k\}}^*$, uzyskanej z układu sensorycznego. Baza reguł ULR została dobrana w taki sposób, aby sygnał sterowania miał maksymalną wartość, gdy odległość do przeszkód jest duża, co odpowiada generowaniu trajektorii ruchu z maksymalną zadaną prędkością. W miarę zbliżania się do przeszkody jest generowany sygnał sterowania o coraz mniejszej wartości, co odpowiada generowaniu trajektorii ruchu o mniejszej prędkości. Takie zachowanie umożliwia zrealizowanie manewru ominięcia przeszkody. Drugi ULR generuje sygnał sterowania zadaną prędkością kątową obrotu ramy $u_{O\dot{\beta}\{k\}}$ na podstawie błędu:

$$e_{O\{k\}} = d_{R\{k\}}^* - d_{L\{k\}}^* \quad (5)$$

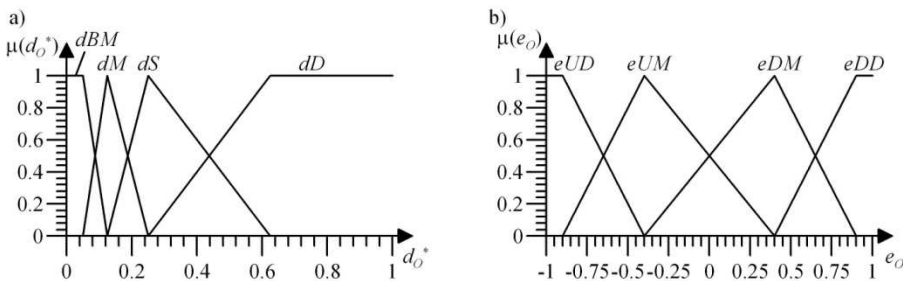
gdzie: $d_{R\{k\}}^*$, $d_{L\{k\}}^*$ – znormalizowane minimalne odległości do przeszkód po prawej i lewej stronie ramy MRK, odpowiednio $d_{Rm\{k\}}$ i $d_{Lm\{k\}}$.

Sygnał sterowania behawioralnego $u_{O\dot{\beta}\{k\}}$ jest generowany na podstawie bazy $m = 4$ reguły typu:

$$R_{O\dot{\beta}}^m: \text{JEŻELI } e_{O\{k\}} \text{ JEST } eDM \text{ TO } u_{O\dot{\beta}\{k\}} \text{ JEST } uDM \quad (6)$$

gdzie: eDM – etykieta lingwistyczna funkcji przynależności do zbioru rozmytego przesłanek, uDM – etykieta lingwistyczna funkcji przynależności konkluzji typu singleton (o wartości $uBM = 0$, $uM = 0,5$, $uS = 0,9$, $uD = 1$ dla pierwszego ULR oraz $uUD = -1$, $uUM = -0,5$, $uDM = 0,5$, $uDD = 1$ dla drugiego ULR).

Funkcje przynależności do zbiorów rozmytych przesłanek ULR warstwy generowania trajektorii ruchu w zadaniu typu OA przedstawiono na rys. 4. Sy-



Rys. 4. Funkcje przynależności do zbiorów rozmytych przesłanek pierwszego układu z logiką rozmytą (a), funkcje przynależności do zbiorów rozmytych przesłanek drugiego układu z logiką rozmytą (b)

Fig. 4. Membership functions of premises of the first fuzzy logic system (a), membership functions of premises of the second fuzzy logic system (b)

gnały sterowania behawioralnego $u_{Ov\{k\}}$ i $u_{O\beta\{k\}}$ w zadaniu typu OA są przeliczane na wartości zadanych prędkości kątowych kół MRK zgodnie z zależnością (4). Wygenerowana trajektoria ruchu jest realizowana przez warstwę sterowania ruchem nadążnym.

5. Układ sterowania ruchem nadążnym

W warstwie sterowania ruchem nadążnym na podstawie znajomości zadanych ($\mathbf{z}_{d\{k\}}$) i zrealizowanych parametrów ruchu ($\mathbf{z}_{\{k\}}$) wyznacza się błędy nadążania:

$$\left. \begin{aligned} \mathbf{e}_{1\{k\}} &= \mathbf{z}_{1\{k\}} - \mathbf{z}_{d1\{k\}} \\ \mathbf{e}_{2\{k\}} &= \mathbf{z}_{2\{k\}} - \mathbf{z}_{d2\{k\}} \end{aligned} \right\} \quad (7)$$

a na ich podstawie uogólnione błędy nadążania są minimalizowane przez algorytm realizacji ruchu:

$$\mathbf{s}_{\{k\}} = \mathbf{e}_{2\{k\}} + \mathbf{\Lambda} \mathbf{e}_{1\{k\}} \quad (8)$$

gdzie $\mathbf{\Lambda}$ – diagonalna, dodatnio określona, stała macierz projektowa.

W warstwie realizacji ruchu zastosowano algorytm DHP, należący do rodziny algorytmów APD [16-18], nazywanych również algorytmami neuronowego programowania dynamicznego (NPD) lub algorytmami typu aktor-krytyk. Ich geneza wywodzi się z chęci implementacji programowania dynamicznego Bellmana w aplikacjach optymalnego sterowania obiektami dynamicznymi online. Metoda programowania dynamicznego Bellmana umożliwia wyznaczenie optymalnego prawa sterowania i optymalnej trajektorii stanu obiektu dynamicznego od ostatniego kroku procesu do kroku pierwszego. Odbywa się to przez wybór trajektorii stanu i sterowania o najmniejszej wartości funkcji celu, nazywanej również funkcją wartości, spośród wszystkich fizycznie osiągalnych trajektorii ruchu sterowanego obiektu. Wadą tej metody, oprócz złożoności obliczeniowej, jest konieczność wyznaczania optymalnego prawa sterowania od ostatniego kroku procesu do kroku pierwszego. Wady tej są pozbawione algorytmy APD, w których funkcja wartości jest aproksymowana przez SN krytyka, co umożliwia generowanie suboptymalnego prawa sterowania online przez SN aktora. Rodzina algorytmów APD składa się z sześciu struktur, z których podstawowe to heurystyczne programowanie dynamiczne (HDP), DHP i globalne dualne heurystyczne programowanie dynamiczne (GDHP). W algorytmie HDP aktor generuje suboptymalne prawo sterowania, natomiast krytyk aproksymuje funkcję wartości. W algorytmie DHP funkcja aktora jest taka sama, krytyk aproksymuje pochodną funkcji wartości względem stanu sterowanego obiektu. Algorytm GDHP ma taką samą strukturę jak HDP, jednak metoda adaptacji wag SN

krytyka jest bardziej złożona, gdyż jest połączeniem metod adaptacji wag krytyka algorytmów HDP i DHP. Pozostałe trzy algorytmy to wersje zależne od sterowania (ang. *Action Dependant* – AD) podstawowych algorytmów, w których sygnał sterowania widziany jako wyjście z SN aktora jest jednocześnie wejściem do SN krytyka.

W prezentowanym układzie sterowania zastosowano algorytm typu DHP, gdzie strukturę zarówno aktora, jak i krytyka zrealizowano w formie SN liniowych ze względu na wagi warstwy wyjściowej *Random Vector Functional Link* (RVFL), o sigmoidalnych, bipolarnych funkcjach aktywacji neuronów i stałych, dobieranych w sposób losowy w procesie inicjalizacji sieci wartościach wag warstwy wejściowej. Zastosowano SN o 8 neuronach w warstwie ukrytej każdej z sieci. Dobór liczby neuronów warstwy ukrytej SN zależy od wielu czynników. W pracy heurystycznie dobrano liczbę neuronów na podstawie wcześniejszych badań autora, biorąc pod uwagę rozmiar SN oraz zakładaną jakość sterowania ruchem nadeżnym.

Warstwa sterowania ruchem nadeżnym składa się z algorytmu APD w konfiguracji DHP, który generuje sygnał sterowania $\mathbf{u}_{A\{k\}}$, regulatora PD generującego sygnał sterowania $\mathbf{u}_{PD\{k\}}$, członu generującego sygnał sterowania nadzorującego $\mathbf{u}_{S\{k\}}$ oraz dodatkowego członu o strukturze wynikającej z dyskretyzacji modelu MRK i sygnale sterowania $\mathbf{u}_{E\{k\}}$. Struktura sygnału sterowania nadzorującego wynika z syntezy algorytmu sterowania nadeżnego z zastosowaniem teorii stabilności Lyapunova. Całkowity sygnał sterowania ruchem nadeżnym MRK wyrażono wzorem:

$$\mathbf{u}_{\{k\}} = -\frac{1}{h} \mathbf{M} [\mathbf{u}_{A\{k\}} + \mathbf{u}_{PD\{k\}} + \mathbf{u}_{S\{k\}} + \mathbf{u}_{E\{k\}}] \quad (9)$$

przy czym:

$$\left. \begin{aligned} \mathbf{u}_{PD\{k\}} &= \mathbf{K}_D \mathbf{s}_{\{k\}} \\ \mathbf{u}_{S\{k\}} &= \mathbf{I}_S \mathbf{u}_{S\{k\}}^* \\ \mathbf{u}_{E\{k\}} &= [\Lambda \mathbf{e}_{2\{k\}} - \mathbf{z}_{d3\{k\}}] h \end{aligned} \right\} \quad (10)$$

gdzie: \mathbf{K}_D – dodatnio określona, diagonalna macierz współczynników wzmocnień regulatora PD, $\mathbf{u}_{S\{k\}}^*$ – wektor sygnałów sterowania nadzorującego, \mathbf{I}_S – macierz diagonalna o współczynnikach $I_{S[j,j]} = 1$, gdy $|s_{[j]\{k\}}| \geq \rho_{[j]}$ lub $I_{S[j,j]} = 0$ w przeciwnym przypadku, $j = 1, 2$, $\rho_{[j]}$ – stała wartość projektowa.

Algorytm sterowania ruchem nadeżnym MRK opisano szczegółowo w pracach [12, 13].

6. Wyniki testów numerycznych

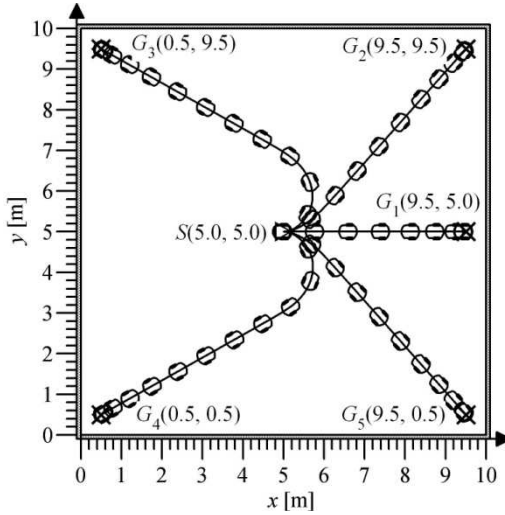
Testy numeryczne hierarchicznego układu sterowania ruchem MRK przeprowadzono w środowisku obliczeniowym Matlab/Simulink. Zaprogramowano model dynamiki MRK, zasymulowano działanie układu sensorycznego robota zrealizowanego z zastosowaniem skanera laserowego przestrzeni oraz zamodelowano laboratoryjne środowisko pomiarowe. Przygotowane autorskie algorytmy symulacji umożliwiły modelowanie ruchu MRK oraz symulację interakcji ze środowiskiem z zastosowaniem modelu układu sensorycznego robota. W ramach testów numerycznych przeprowadzono symulację działania hierarchicznego układu sterowania podczas realizacji zadań typu GS i OA. W celu uproszczenia zapisu zmiennych, w niniejszym rozdziale zrezygnowano z umieszczania indeksu k w oznaczeniach zmiennych. Przebiegi wartości wszystkich zmiennych przedstawiono w ciągłej dziedzinie czasu.

Test numeryczny układu sterowania w zadaniu typu „podążaj do celu”

Testy numeryczne hierarchicznego układu sterowania ruchem MRK w zadaniu typu GS przeprowadzono w symulowanym torze pomiarowym, którego mapę pokazano na rys. 5. Symbolem trójkąta oznaczono położenie początkowe pkt A MRK, $S(5,0; 5,0)$, przy czym orientacja ramy robota była taka, że jej oś symetrii była równoległa do osi x układu współrzędnych, a przód robota skierowany w stronę dodatniej gałęzi osi x . Zadane położenia końcowe przyjęto w punktach $G_1(9,5; 5,0)$, $G_2(9,5; 9,5)$, $G_3(0,5; 9,5)$, $G_4(0,5; 0,5)$, $G_5(9,5; 0,5)$ i oznaczono znakiem X na mapie otoczenia. Tor ruchu pkt A ramy zaznaczono linią ciągłą. Podczas testu numerycznego krok dyskretyzacji czasu $h = 0,01$ s.

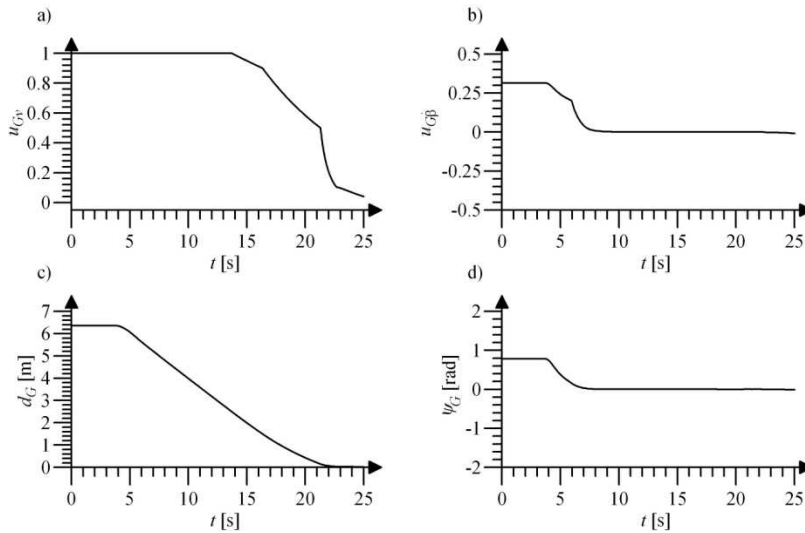
Zadanie uznawano za zrealizowane poprawnie, jeżeli po zakończeniu ruchu MRK odległość do celu $d_G \leq 0,03$ m. Wszystkie przeprowadzone testy numeryczne potwierdziły poprawność procesu generowania trajektorii ruchu do zadanego celu. Dalej przedstawiono szczegółowo przebiegi wartości poszczególnych sygnałów otrzymanych w czasie symulacji ruchu MRK do celu $G_2(9,5; 9,5)$.

Sygnały sterowania warstwy generowania trajektorii ruchu w zadaniu typu GS, odpowiednio u_{Gv} i $u_{G\beta}$, przedstawiono na rys. 6a i b. Na rysunku 6c zobrażowano przebieg wartości odległości do celu d_G , natomiast na rys. 6d – przebieg wartości kąta ψ_G . Sygnał sterowania zadaną prędkością pkt A ramy przyjmuje wartość 1, gdy odległość do celu d_G jest duża. Gdy odległość do celu maleje, zmniejsza się również wartość sygnału sterowania, powodując zatrzymanie MRK w położeniu zapewniającym poprawną realizację zadania. Wartość sygnału sterowania $u_{G\beta}$ zależy od wartości kąta ψ_G . Gdy wartość kąta jest „duża dodatnia”, generowana jest odpowiednia wartość sygnału sterowania. Gdy wartość



Rys. 5. Mapa symulowanego środowiska pomiarowego w zadaniu typu „podążaj do celu”

Fig. 5. Map of the simulated measuring environment in the „goal seeking” task

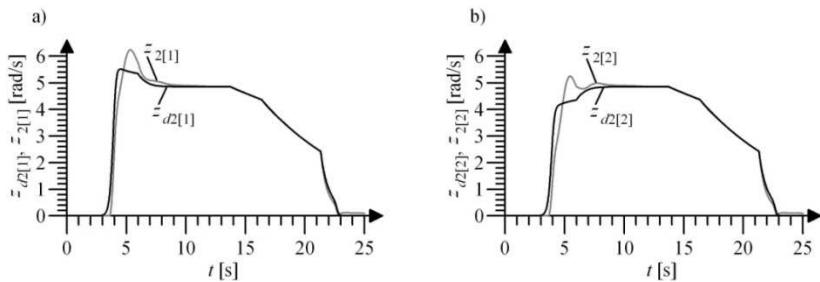


Rys. 6. Sygnał sterowania warstwy generowania trajektorii u_{Gv} (a), sygnał sterowania warstwy generowania trajektorii $u_{G\beta}$ (b), odległość do celu d_G (c), wartość kąta ψ_G (d)

Fig. 6. The control signal u_{Gv} of the trajectory generator layer (a), the control signal $u_{G\beta}$ of the trajectory generator layer (b), distance to the goal d_G (c), value of the ψ_G angle (d)

kąta ψ_G maleje do wartości równej zero, oznacza to, że MRK porusza się w kierunku celu, a oś symetrii ramy robota pokrywa się z prostą p_G .

Na podstawie sygnałów sterowania warstwy planowania trajektorii ruchu u_{Gv} i $u_{G\beta}$ były wyznaczane zadane wartości prędkości kątowych kół 1 i 2: $z_{d2[1]}$ i $z_{d2[2]}$. Jednocześnie trajektoria zadana była przekazywana do warstwy sterowania nadążnego, która generowała sygnały sterowania powodujące realizację zadanych parametrów ruchu. Przebiegi wartości zadanych i zrealizowanych $z_{d2[1]}$ i $z_{d2[2]}$ prędkości kątowych kół przedstawiono na rys. 7a i b.

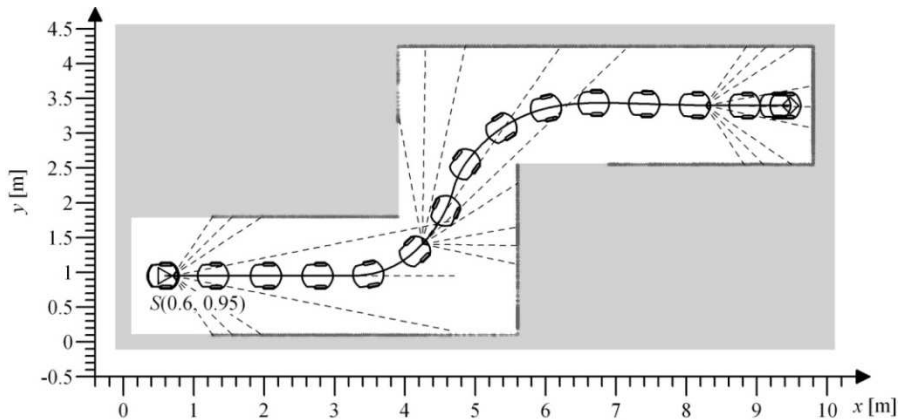


Rys. 7. Wartość zadanej $z_{d2[1]}$ i zrealizowanej $z_{2[1]}$ prędkości kątowej koła 1 (a), wartość zadanej $z_{d2[2]}$ i zrealizowanej $z_{2[2]}$ prędkości kątowej koła 2 (b)

Fig. 7. Value of the required $z_{d2[1]}$ and obtained $z_{2[1]}$ angular velocities of the wheel 1 (a), value of the required $z_{d2[2]}$ and obtained $z_{2[2]}$ angular velocities of the wheel 2 (b)

Test numeryczny układu sterowania w zadaniu typu „omijaj przeszkody”

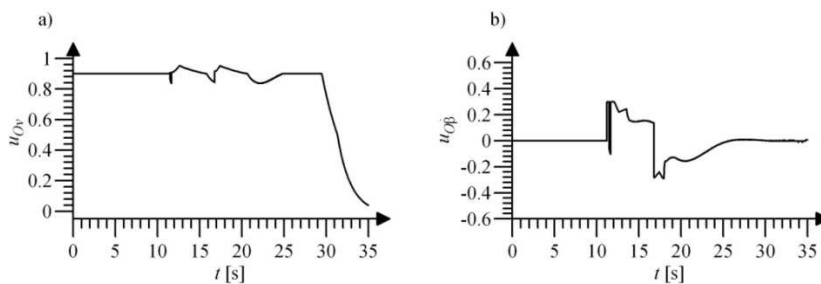
Testy numeryczne hierarchicznego układu sterowania ruchem MRK w zadaniu typu OA przeprowadzono w symulowanym torze pomiarowym, którego mapę przedstawiono na rys. 8. Symbolem trójkąta oznaczono położenie początkowe pkt A MRK, $S(0,6; 0,95)$, przy czym oś symetrii ramy robota była równoległa do osi x układu współrzędnych. Zadane położenia końcowego nie zdefiniowano. Tor ruchu pkt A ramy MRK zaznaczono linią ciągłą, liniami przerywanymi natomiast – osie wybranych pomiarów zrealizowanych przez układ sensoryczny MRK w chwilach $t = 0$ s, $t = 14$ s i $t = 28$ s. Szarymi kropkami zaznaczono położenia przeszkód wykrytych przez układ sensoryczny robota. Podczas testu numerycznego krok dyskretyzacji czasu $h = 0,01$ s.



Rys. 8. Mapa symulowanego środowiska pomiarowego w zadaniu typu „omijaj przeszkodę”

Fig. 8. Map of the simulated measuring environment in the „obstacle avoiding” task

Sygnaly sterowania warstwy generowania bezkolizyjnej trajektorii ruchu w zadaniu typu OA, odpowiednio u_{Ov} i $u_{O\beta}$, przedstawiono na rys. 9a i b. W końcowej fazie ruchu robot znajduje się w pobliżu przeszkód, wartość sygnału sterowania u_{Ov} zmniejsza się w celu umożliwienia realizacji manewru ominięcia przeszkody. Wartość sygnału sterowania $u_{O\beta}$ zmienia się w sposób powodujący generowanie trajektorii zapewniającej ruch MRK środkiem wolnej przestrzeni.

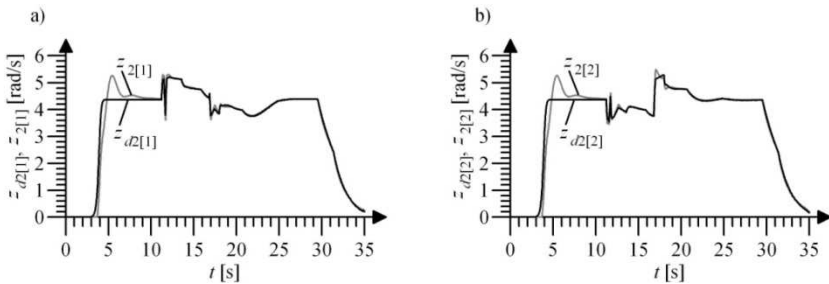


Rys. 9. Sygnał sterowania warstwy generowania trajektorii u_{Ov} (a), sygnał sterowania warstwy generowania trajektorii $u_{O\beta}$ (b)

Fig. 9. The control signal u_{Ov} of the trajectory generator layer (a), the control signal $u_{O\beta}$ of the trajectory generator layer (b)

Sygnaly sterowania warstwy generowania trajektorii ruchu, zgodnie z zależnościami (4), są przeliczane na zadane wartości prędkości kątowych kół MRK $\dot{z}_{d2[1]}$ i $\dot{z}_{d2[2]}$, i realizowane przez warstwę sterowania ruchem nadążnym. Przebie-

gi zadanych i zrealizowanych prędkości kątowych kół MRK przedstawiono na rys. 10a i b. W początkowym okresie ruchu można zauważyć rozbieżności między zadanymi i zrealizowanymi wartościami prędkości kątowych obrotu kół, co wynika z zastosowanej metody sterowania ruchem nadążnym i przyjęcia zerowych wartości początkowych wag warstwy wyjściowej SN struktury DHP.



Rys. 10. Wartość zadanej $z_{d2[1]}$ i zrealizowanej $z_{2[1]}$ prędkości kątowej koła 1 (a), wartość zadanej $z_{d2[2]}$ i zrealizowanej $z_{2[2]}$ prędkości kątowej koła 2 (b)

Fig. 10. Value of the required $z_{d2[1]}$ and obtained $z_{2[1]}$ angular velocities of the wheel 1 (a), value of the required $z_{d2[2]}$ and obtained $z_{2[2]}$ angular velocities of the wheel 2 (b)

7. Podsumowanie

Zaprezentowano hierarchiczny układ sterowania ruchem MRK, składający się z dwóch warstw. Warstwa nadrzędna generuje trajektorię ruchu MRK w zadaniach typu GS i OA. W warstwie tej zastosowano metody AI w postaci ULR. Generowana trajektoria jest jednocześnie realizowana przez drugą z warstw układu sterowania, którą jest układ sterowania nadążnego. W prezentowanym przykładzie zastosowano neuronowy układ sterowania z algorytmem APD w konfiguracji DHP, w którym struktury aktora i krytyka zrealizowano w postaci SN RVFL. Zastosowano SN liniowe ze względu na wagi warstwy wyjściowej, z wagami warstwy wejściowej stałymi, dobieganymi w sposób losowy w procesie inicjalizacji sieci i z sigmoidalnymi, bipolarnymi funkcjami aktywacji neuronów. Przedstawiony algorytm sterowania poprawnie generuje i realizuje trajektorię ruchu do zdefiniowanego celu w zadaniu typu GS. W zadaniu typu OA układ sterowania na podstawie sygnałów z modelowanego układu sensorycznego MRK, zrealizowanego w postaci skanera laserowego przestrzeni, generuje bezkolizyjną trajektorię ruchu. Zastosowanie ULR w algorytmach sterowania behawioralnego warstwy generowania trajektorii ruchu umożliwia planowanie sposobu generowania trajektorii ruchu na podstawie wiedzy eksperta, popartej obserwacją analogicznych zachowań organizmów żywych. Zastosowa-

nie algorytmów APD w konfiguracji DHP umożliwia uzyskanie wymaganej jakości realizacji ruchu nadążnego.

Literatura

- [1] Arkin R.C.: Behavior-based robotics. MIT Press, Cambridge 1998.
- [2] Borenstein J., Koren Y.: Real-time obstacle avoidance for fast mobile robots. IEEE Transactions on Systems, Man Cybernetics, 19 (1989), 1179-1187.
- [3] Burghardt A.: Implementation of Kohonen network in behavioral control of the amigobot wheeled mobile robot. Mechanics Mech. Engi., 12 (2008), 337-347.
- [4] Faihimi F.: Autonomous robots. Modeling, path planning, and control. Springer, New York 2009.
- [5] Hendzel Z.: Fuzzy reactive control of wheeled mobile robot. J. Theor. Appl. Mech., 42 (2004), 503-517.
- [6] Hendzel Z., Burghardt A.: Rozmyte sterowanie odruchowe elementarnymi zachowaniami mobilnego robota. PAK, 11 (2004), 23-25.
- [7] Maaref H., Barret C.: Sensor-based navigation of a mobile robot in an indoor environment. Robotics Autonomous Systems, 38 (2002), 1-18.
- [8] Millan J.: Reinforcement learning of goal-directed obstacle-avoiding reaction strategies in an autonomous mobile robot. Robotics and Autonomous Systems, 15 (1995), 275-299.
- [9] Sarkar S., Shome S.N., Nandy S.: An intelligent algorithm for the path planning of autonomous mobile robot for dynamic environment. Communications in Computer and Information Science, 103 (2010), 202-209.
- [10] Hendzel Z., Szuster M.: Neuronowe programowanie dynamiczne w sterowaniu behawioralnym mobilnym robotem kołowym. Acta Mech. Automat., 5 (2011), 28-36.
- [11] Szuster M., Hendzel Z., Burghardt A.: Fuzzy sensor-based navigation with neural tracking control of the wheeled mobile robot. Lecture Notes in Artificial Intelligence, 8468 (2014), 302-313.
- [12] Hendzel Z., Szuster M.: Discrete model-based adaptive critic designs in wheeled mobile robot control, Lecture Notes Artificial Intelligence, 6114 (2010), 264-271.
- [13] Hendzel Z., Szuster M.: Discrete neural dynamic programming in wheeled mobile robot control. Communications Nonlinear Sci. Numerical Simul., 16 (2011), 2355-2362.
- [14] Giergiel M.J., Hendzel Z., Żylski W.: Modelowanie i sterowanie mobilnych robotów kołowych. PWN, Warszawa 2002.
- [15] Giergiel J., Żylski W.: Description of motion of a mobile robot by Maggie's equations. J. Theor. Appl. Mech., 43 (2005), 511-521.
- [16] Prokhorov D.V., Wunsch D.C.: Adaptive critic designs, IEEE Transactions on Neural Networks, 8, 5 (1997), 997-1007.
- [17] Powell W.B.: Approximate dynamic programming: solving the curses of dimensionality. Wiley, New Jersey 2007.
- [18] Si J., Barto A.G., Wunsch D.C.: Handbook of learning and approximate dynamic programming. Wiley, IEEE Press, 2004.

FUZZY BEHAVIOURAL CONTROL OF THE WHEELED MOBILE ROBOT IN THE UNKNOWN ENVIRONMENT

Summary

In the article the hierarchical control system of the wheeled mobile robot movement in the unknown environment with static obstacles was presented. The control system consists of the trajectory generator that realises the behavioural control using fuzzy logic system, and the neural tracking control system in which approximate dynamic programming algorithm is implemented. In the planning layer of movement trajectory the behavioural control tasks of the „goal-seeking” type and the „obstacle avoiding” type were realised. In the presented hierarchical control system the control signals of the trajectory planning layer were generated using two fuzzy logic systems in which Takagi-Sugeno model was utilized. In the movement control layer the approximate dynamic programming algorithm in the dual heuristic dynamic programming configuration was applied. It consists of two structures: the actor and the critic. Both the actor and the critic structures were realised using artificial neural networks. Performance of the proposed control algorithm was verified numerically using emulator of the wheeled mobile robot and the laboratory measuring track in the Matlab/Simulink computational environment.

Keywords: approximate dynamic programming, trajectory generator, neural networks, tracking control, fuzzy logic systems

DOI: 10.7862/rm.2014.63

Otrzymano/received: 26.06.2014 r.

Zaakceptowano/accepted: 15.11.2014 r.

