

Jerzy TCHÓRZEWSKI*, Przemysław DOMAŃSKI*

KWANTOWA SZTUCZNA SIEĆ NEURONOWA. CZEŚĆ 2. MODEL RUCHU RAMIENIA ROBOTA PR-02

Artykuł jest kontynuacją pracy pod tym samym tytułem głównym i podtytułem: Część 1. Obliczenia kwantowe. W niniejszej pracy zamieszczono wybrane elementy metody i algorytmu dotyczącego inspirowania modelu neuralnego ruchu robota PR-02 za pomocą rozwiązań informatyki kwantowej. Zwrócono uwagę na występujące problemy w zakresie przeprowadzania obliczeń kwantowych na komputerach klasycznych. Pokazano, że można przeprowadzać kwantyzację, obliczenia kwantowe i dekwantyzację na komputerze klasycznym, co wiąże się z wydłużeniem obliczeń neuronalnych. Dla założonej liczby 1 000 quasi równoległych obliczeń nie uzyskano poprawy przebiegu trajektorii, ale uzyskano bardziej precyzyjny punkt startu i punkt docelowy ruchu ramienia robota PR-02. Badania są kontynuowane dla 10 000 quasi równoległych obliczeń, co na klasycznych komputerach wymaga znacznie dłuższego czasu obliczeń.

SŁOWA KLUCZOWE: dekwantyzacja, kwantyzacja, obliczenia kwantowe, robot PR-02, sztuczna sieć neuronowa.

1. MODEL ANALITYCZNY RUCHU ROBOTA PR-02

1.1. Wprowadzenie

W związku z poszukiwaniem coraz bardziej dokładnych metod modelowania systemów w ostatnich latach wzrosło ponownie zainteresowanie metodami sztucznej inteligencji z punktu widzenia ich podatności na inspirowanie metodami informatyki kwantowej, a zwłaszcza metod ewolucyjnych i neuronalnych [15, 23]. Dotyczy to też m.in. metod modelowania analitycznego oraz neuralnego tego typu systemów jak ruch końca ramienia robota PR-02. Modele te są stosunkowo dokładne, niemniej podjęto próbę zwiększenia dokładności ruchu końca ramienia robota w zakresie zarówno zwiększenia dokładności punktu startu i końca położenia jak też przebiegu trajektorii. Metodę poprawiania ruchu ramienia robota zamieszczono w części 1 pracy pod tym samym tytułem głównym i podtytułem „Obliczenia kwantowe” [20], w tym metodę kwantyzacji, dekwantyzacji oraz metodę prowadzenia obliczeń kwantowych na kompute-

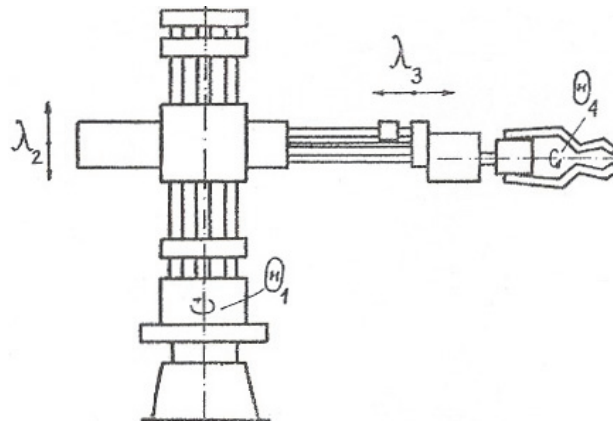
* Uniwersytet Przyrodniczo-Humanistyczny w Siedlcach

rach klasycznych. W tej części pracy odniesiono ww. metodę obliczeń kwantowych do poprawy modelu neuralnego ruchu końca ramienia robota PR-O2.

1.2 Robot PR-02

W badaniach eksperymentalnych został wykorzystany robot PR-02 o czterech stopniach swobody, który został zaprojektowany w Przemysłowym Instytucie Automatyki i Pomiarów w Warszawie w latach 1976-80 w celu prowadzenia przy jego wykorzystaniu badań komparatystycznych i symulacyjnych nowych metod i modeli systemów.

Roboty tego typu charakteryzują się nieskomplikowanym programowaniem trajektorii oraz prostym układem zasilania (rys. 1). Głównymi cechami robota PR-02 są m.in.: prosty napęd pneumatyczny, modułowa budowa łatwa w programowaniu, możliwość zarządzania sygnałami sterującymi znajdującymi się w standardach przemysłowych (0-24V), dyskretna przestrzeń robocza, itp. [2-3, 8-9, 13, 17].



Rys. 1. Robot PR-02 o czterech stopniach swobody, dwóch obrotach (θ_1 i θ_4) oraz dwóch przesunięciach (przez λ_2 ; λ_3). Oznaczenia: θ_1 – parametr wpływający na obrót platformy, λ_2 – parametr wpływający na aktualną wysokość zawieszenia ramienia, λ_3 – parametr wpływający na aktualną długość wysunięcia ramienia, θ_4 – parametr wpływający na obrót chwytaka [17]

Przyjęty do eksperymentów robot przemysłowy PR-02 składa się z dwóch zespołów: manipulacyjnego i sterowania. Wyposażony został w cztery przeguby, po dwa zapewniające ruch liniowy oraz ruch obrotowy [2-3, 7-9, 13, 17]. Równania proste kinematyki ruchu chwytaka robota PR-02 przedstawiane jest za pomocą grafu przekształceń za pomocą macierzy X postaci:

$$\mathbf{X} = \begin{bmatrix} C_1 S_4 & C_1 C_4 & -S_1 & l_2 C_1 - S_1 \lambda_3 \\ S_1 S_4 & S_1 C_4 & C_1 & l_2 C_1 + S_1 \lambda_3 \\ C_4 & -S_4 & 0 & l_3 + \lambda_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

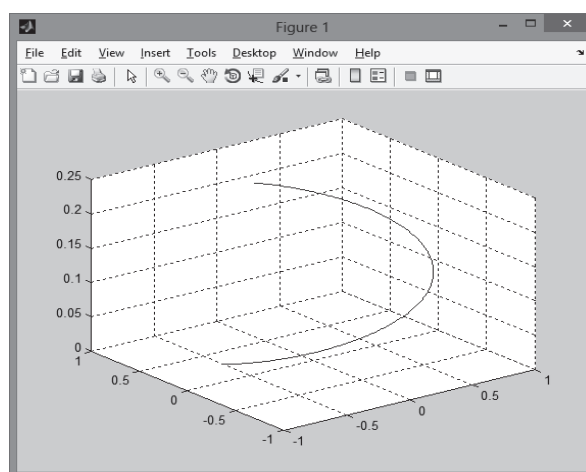
przy czym parametrami C_i oraz S_i oznaczono wartości cosinusów i sinusów odpowiednich kątów na przegubie, które szczegółowo opisano w tabeli 1 [2-3, 7-9, 13, 17].

Tabela 1. Wartości parametrów tzw. Hartenberga-Denevita (H-D) wykorzystywanych do opisu manipulatora PR-02 [17].

Nr	α_i [°]	l_i [m]	λ_i [m]	θ_i [°]
1	0	0	0	-300 ÷ 0
2	-90	-0,11	0 ÷ 0,200	0
3	0	0	0,367 ÷ 0,676	-90
4	0	0	0	0 ÷ 360

1.3. Trajektoria ruchu końca ramienia robota PR-02

Zmiana parametrów H-D steruje w bezpośredni sposób ruchem końca ramienia robota PR-02 (położeniem punktu oznaczającego koniec ramienia robota), w wyniku czego koniec ramienia robota kreśli w przestrzeni kartezjańskiej XYZ trajektorię pokazaną na rys. 2 [5, 17].

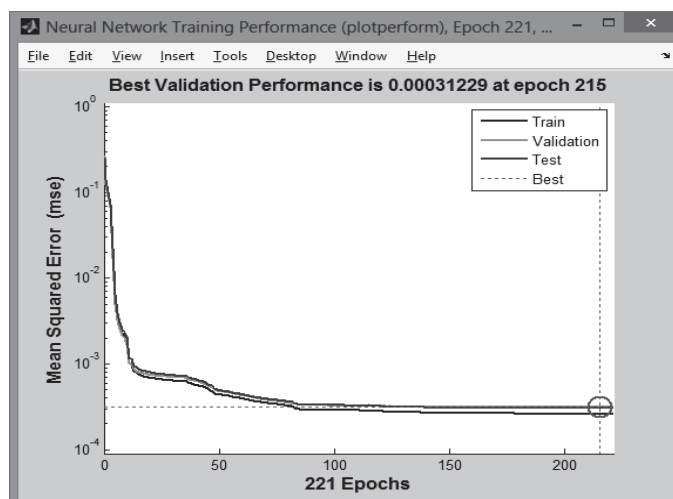


Rys. 2. Przebieg trajektorii ruchu robota PR-02. Oznaczenia: X,Y,Z – układ kartezjański [m], przy czym oś pionowa jest osią Z [5]

2. MODEL NEURONALNY RUCHU ROBOTA PR-02

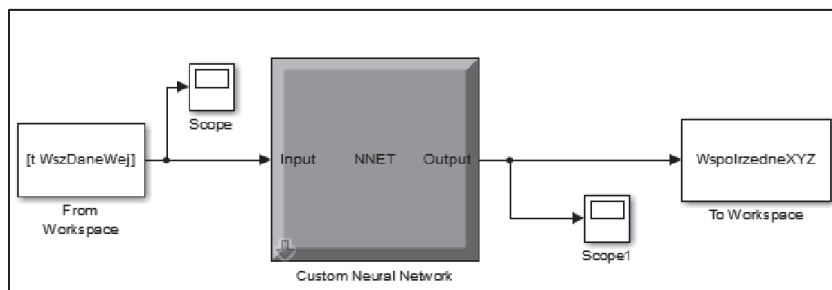
W celu uzyskania modelu neuralnego ruchu końca ramienia robota PR-02 zaprojektowano perceptronową sztuczną sieć neuronową (SSN) szczegółowo opisaną m.in. w pracach [12, 19, 27] z jedną warstwą ukrytą zawierającą cztery neurony, przyjmując na wejściu (warstwa wejściowa neuronów) cztery neurony adekwatnie do wielkości parametrów H-D oraz trzy neurony na wyjściu SSN odpowiadające przestrzennym współrzędnym położenia punktu końca ramienia robota (XYZ). Do uczenia SSN wykorzystano wartości parametrów zamieszczone w tabeli 1 przyjmując liczbę epok 1 164.

W wyniku uczenia SSN uzyskano błąd MSE rzędu $10^{-3} - 10^{-4}$ (rys. 3) już po ok. 80 epokach (train), przy czym uczenie trwało przez 215 epok, kiedy SSN osiągnęła najlepszy wynik uczenia [5, 16, 20-22, 26].



Rys. 3. Wykres krzywych uczenia, testowania i walidacji. Oznaczenia: Epochs-epoki, Train - przebieg krzywej uczenia (tzw. błąd MSE), Validation - przebieg krzywej walidacji, Test- przebieg krzywej testowania, Best- przebieg optymalny, MSE (ang. Mean Squared Error) - błąd średniokwadratowy

W wyniku uczenia przeprowadzonego z wykorzystaniem GUI środowiska Neural Network Toolbox-a uzyskano model neuronalny w postaci sztucznej sieci neuronowej, który dla potrzeb badań symulacyjnych wygenerowano za pomocą funkcji gensim() – rys. 4.



Rys. 4. Model sztucznej sieci neuronowej do badań symulacyjnych i komparatystycznych. Oznaczenia: From Workspace – blok do wprowadzania wartości wielkości wejściowych do SSN, To Workspace – blok do wyprowadzania wyników do przestrzeni roboczej (ang. Workspace) [5]

3. MODEL KWANTOWY RUCHU ROBOTA PR-02

3.1. Kwantowa Sztuczna Sieć Neuronowa

W literaturze przedmiotu istnieje już bogaty dorobek w zakresie inspirowania metod sztucznej inteligencji rozwiązaniami informatyki kwantowej, co zostało opisane m.in. w pracach [1, 4, 6-7, 11, 14]. W niniejszych badaniach wykorzystano teorię starowania i systemów, zwłaszcza w zakresie precyzowania stanów kwantowych [10, 24-25] uzupełniona rozwiązaniami informatyki kwantowej [1, 4, 6-7]. W tym duchu w celu zbudowania kwantowej sztucznej sieci neuronowej (KSN) przyjęto model neuronalny ruchu robota PR-02, w którym wprowadzono kwantyzację, dekwantyzację oraz obliczenia kwantowe na kwantowych liczbach mieszanych opisanych szczegółowo w pracy [20].

Wiązało się to m.in. z zamianą liczb rzeczywistych na liczby kwantowe (proces kwantyzacji), przeprowadzeniem obliczeń na kwantowych liczbach mieszanych oraz zamianą uzyskanych wyników obliczeń w postaci kwantowych liczb mieszanych na liczby rzeczywiste (proces dekwantyzacji).

Kwantowy model neuronalny został szczegółowo opisany w pracy [16, 20-22, 26] i z racji przyjętej funkcji aktywacji typu tansig() był następujący:

$$y(net_i^k(t)) = \frac{1 - e^{-2 \cdot net_i^k(t)}}{1 + e^{-2 \cdot net_i^k(t)}}, \quad (2)$$

gdzie: net_i^k – sumator kwantowy i-tego neuronu w k-tej warstwie wag neuronów wyznaczony jako suma ważonych kwantowych wartości sygnałów wejściowych doprowadzonych do k-tej warstwy neuronów wyrażony następująco [5, 16, 20-22, 26]:

$$net_i^k(t) = \sum_{i,lm,k} \begin{bmatrix} net_{i,lim}^k & net_{i,lm}^k \\ net_{i,lm}^k & net_{i,lm}^k \end{bmatrix}, \quad (3)$$

przy czym: $net_{i,lm}^k$ - element sumatora netki o indeksie lm jako kwantowy ważony sygnał wejściowy do k -tej warstwy neuronów sztucznej sieci neuronowej o naturze stanu czystego wynikającego z dwóch stanów mieszanych (kwantowej liczby mieszanej sygnału wejściowego oraz kwantowej liczby mieszanej wagi).

3.2. Liczby kwantowe

W celu przeprowadzenia obliczeń kwantowych należało przeprowadzić proces kwantyzacji liczb rzeczywistych na kwantowe liczby mieszane, co zostało przeprowadzone na 1 000 obliczeniach quasi równoległych. Szczegółowy opis przebiegu algorytmu kwantyzacji zawierają m.in. prace [16, 20-22, 26]. Przyjęto, że liczba rzeczywista zamieniana jest najpierw na liczbę binarną składającą się z 17 bitów, przy czym pierwszy bit odpowiada za znak liczby (0 lub 1), następnie przyjęto 9 bitów na reprezentację części całkowitej liczby, po czym jeden znak za występowanie przecinka (jedenasty), a w końcu sześć następnym za część ułamkową liczby [5].

Zamiana liczby rzeczywistej na liczbę kwantową wymaga zatem w pierwszej kolejności zamiany jej na liczbę binarną, której poszczególne bity są stanami czystymi w zapisie liczb kwantowych. Następnie zaś należy zamienić ją ze stanów czystych liczby kwantowej na stany mieszanych liczb kwantowych [5, 16, 20-22, 26], przy czym:

$$|k_m\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle \quad (4)$$

Z zasady superpozycji wyznaczenie prawdopodobieństwa wystąpienia ket 0 lub ket 1 dla $\alpha = \beta$ wynika, że:

$$2 \cdot \alpha^2 = 1 \quad (5)$$

a więc:

$$\alpha = \frac{\sqrt{2}}{2} \quad (6)$$

czyli:

$$\alpha = \beta = 0,71 \quad (7)$$

Losowanie wartości z przewagą stanu czystego ket 0 odbywa się zatem z przedziału:

$$0,71 \leq \alpha \leq 1 \quad (8)$$

W ten sposób ustalono średnią wartość wylosowanych wartości α dla 1 000 przypadków (losowanie quasi równoległe). A jako uzupełnienie stan β wyznaczany jest z zasady superpozycji stanów:

$$\alpha^2 + \beta^2 = 1 \quad (9)$$

Podobna sytuacja wystąpi, gdy stanem czystym jest ket 1. Przykład kwantowych liczb mieszanych zamieszczono w tabeli 2.

3.3. Obliczenia kwantowe

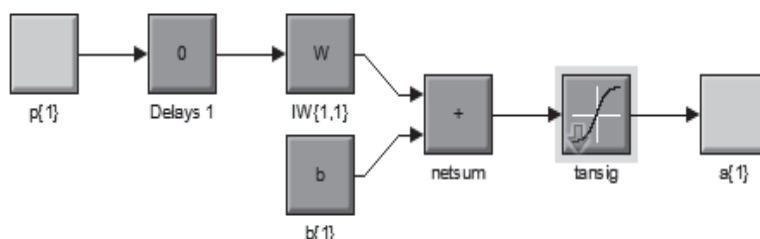
W celu przeprowadzenia obliczeń kwantowych wykorzystano algorytm opisany w pracach [5,20] co wymagało:

- zamiany wartości wielkości wejściowych i wag na kwantowe liczby mieszane,
- przeprowadzenia obliczeń kwantowych dla obu warstw kwantowej sztucznej sieci neuronowej, a więc wyznaczyć sumator na każdym neuronie warstwy ukrytej oraz warstwy wyjściowej oraz wyznaczyć wartości funkcji aktywacji, której argumentem był sumator.

Jako funkcję aktywacji przyjęto funkcję tansig wyrażoną wzorem (2) - rys. 5.

Tabela 2. Przykład kwantowych liczb mieszanych dla 6 qubitów w zapisie kolumnowym (po dwie kolejne kolumny dotyczą wartości α związanego z ketem 0 kolejnego qubitów i wartości β związanego z ketem 1).

α_1	β_2	α_2	β_2	α_3	β_3	α_4	β_4	α_5	β_5	α_6	β_6
0.8273	0.4157	0.4123	0.8300	0.4149	0.8278	0.8208	0.4233	0.4151	0.8277	0.8318	0.4101
0.8280	0.4147	0.4129	0.8296	0.4160	0.8269	0.8338	0.4077	0.4076	0.8339	0.8279	0.4148
0.8326	0.4092	0.4151	0.8277	0.4143	0.8284	0.8335	0.4080	0.4111	0.8310	0.8298	0.4123
0.8278	0.4149	0.4159	0.8270	0.4146	0.8281	0.8333	0.4082	0.4089	0.8328	0.8314	0.4108
0.8267	0.4163	0.4164	0.8266	0.4203	0.8233	0.8290	0.4135	0.4092	0.8326	0.8288	0.4138
0.8273	0.4155	0.4156	0.8273	0.4126	0.8298	0.8270	0.4159	0.4106	0.8314	0.4141	0.8285
0.8293	0.4132	0.4142	0.8284	0.4143	0.8283	0.8258	0.4173	0.4115	0.8306	0.4191	0.8243
0.8292	0.4133	0.4123	0.8300	0.4087	0.8330	0.8272	0.4157	0.4169	0.8262	0.4059	0.8352
0.8266	0.4165	0.4112	0.8310	0.4098	0.8321	0.8265	0.4165	0.4125	0.8298	0.4169	0.8262
0.8279	0.4149	0.4077	0.8338	0.4120	0.8302	0.8247	0.4187	0.4124	0.8299	0.4190	0.8243
0.8288	0.4138	0.4097	0.8321	0.4121	0.8302	0.8298	0.4126	0.4106	0.8314	0.4137	0.8285
0.8272	0.4157	0.4127	0.8297	0.4141	0.8285	0.8303	0.4120	0.4156	0.8273	0.4117	0.8305
0.8312	0.4108	0.4166	0.8265	0.4134	0.8291	0.8284	0.4143	0.4140	0.8286	0.4149	0.8275
0.8308	0.4113	0.4097	0.8321	0.4105	0.8315	0.4171	0.8261	0.8328	0.4089	0.8289	0.4137
0.8297	0.4127	0.4186	0.8248	0.4141	0.8285	0.4082	0.8334	0.8320	0.4099	0.8249	0.4184
0.8318	0.4101	0.4100	0.8319	0.4126	0.8297	0.4113	0.8308	0.8263	0.4168	0.8328	0.4088
0.8267	0.4163	0.4121	0.8301	0.4150	0.8278	0.4160	0.8269	0.8326	0.4091	0.8298	0.4124
0.8216	0.4224	0.4137	0.8289	0.4083	0.8333	0.4173	0.8258	0.8317	0.4103	0.8277	0.4150
0.8240	0.4195	0.4164	0.8266	0.4099	0.8320	0.4112	0.8310	0.8288	0.4137	0.8313	0.4108
0.8313	0.4108	0.4118	0.8304	0.4130	0.8294	0.4151	0.8277	0.8282	0.4144	0.8262	0.4168
0.8282	0.4144	0.4120	0.8302	0.4147	0.8280	0.4138	0.8288	0.8276	0.4152	0.8312	0.4108

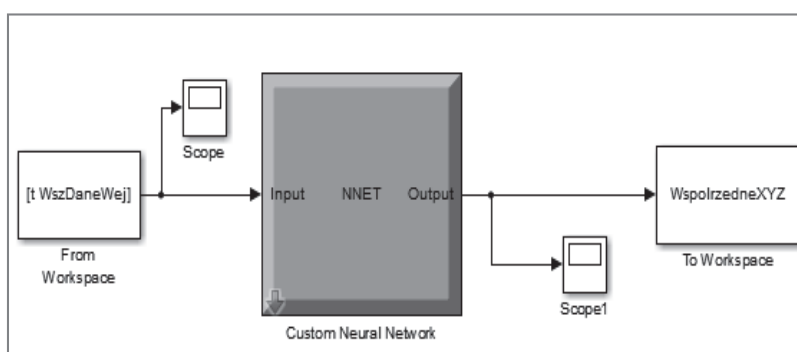


Rys. 5. Model sztucznej sieci neuronowej nauczonej ruchu robota PR-02 dla pierwszej warstwy (ukrytej) SSN. Oznaczenia środowiska MATLAB: $p\{1\}$ – wielkości wejściowe, Delays 1 – opóźnienie na pierwszej warstwie neuronów, $IW\{1,1\}$ – macierz wag na pierwszej warstwie neuronów (na warstwie ukrytej), $b\{1\}$ – wektor biasów na warstwie ukrytej, netsum – sumator ważonych wejść do pierwszej warstwy neuronów, tansig – funkcja aktywacji na pierwszej warstwie neuronów, $a\{1\}$ – sygnał wyjściowy z warstwy ukrytej, który jest jednocześnie sygnałem wejściowym do drugiej warstwy SSN (warstwy wyjściowej)

W wyniku wyznaczania sumatorów na poszczególnych neuronach należało pomnożyć kwantowe liczby mieszane wartości wejściowych przez wartości kwantowe wag i następnie na każdy neuronowe dodać je do siebie, w wyniku czego np. otrzymano ostatecznie np. dla neuronu 1 warstwy ukrytej:

$$y_1^1(\text{net}_1^1) = \frac{1 - e^{\begin{pmatrix} -92,9326 & -43,547 \\ -43,7272 & -27,0664 \end{pmatrix}}}{1 + e^{\begin{pmatrix} -92,9326 & -43,547 \\ -43,7272 & -27,0664 \end{pmatrix}}} \quad (10)$$

Po przeprowadzeniu obliczeń należało ostatecznie dokonać dekwantyzacji, co wykonano za pomocą zbudowanej w tym celu SSN w układzie jak na rys. 6.

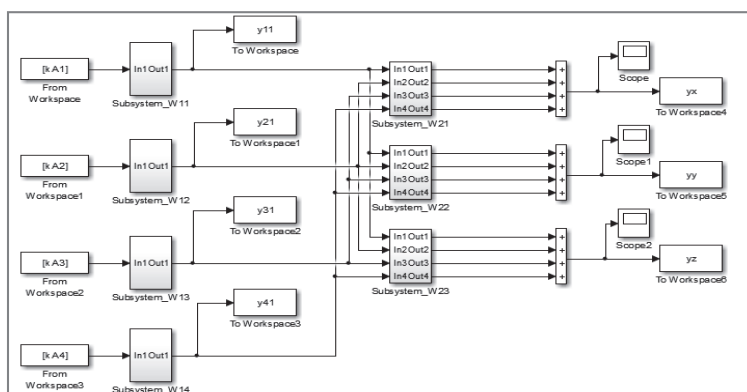


Rys. 6. Model Sztucznej Sieci Neuronowej do dekwantyzacji kwantowych liczb mieszanych na liczby rzeczywiste. Oznaczenia zamieszczono w tabeli 3 [5]

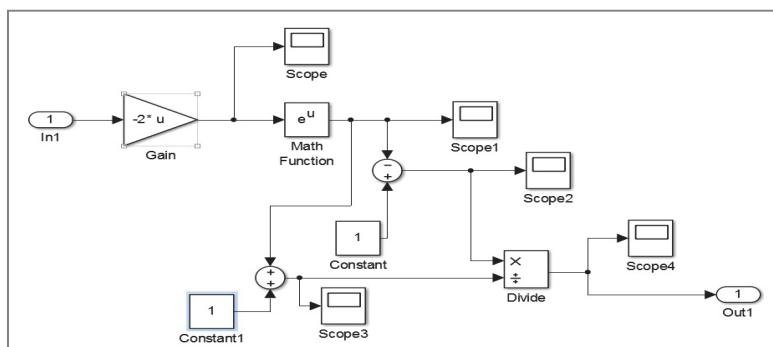
Tabela 3. Oznaczenia do rys. 5. [5].

Lp.	Blok	Objaśnienie
1	From Workspace	Blok wprowadzania danych z przestrzeni roboczej
2	To Workspace	Blok wyprowadzania danych do przestrzeni roboczej
3	Input	wejście
4	Output	wyście
5	t	macierz t sterująca wprowadzaniem wartości wielkości wejściowych
6	WspolrzedneXYZ	macierz wielkości wyjściowych
7	WszDaneWej	macierz wielkości wejściowych
8	Scope1	Wizualizacja wielkości wyjściowych

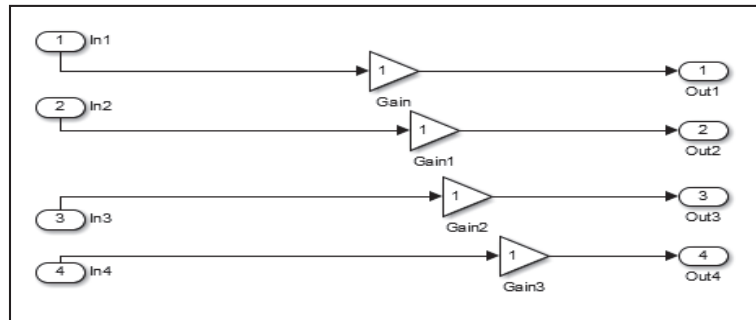
Model kwantowej sztucznej sieci neuronowej zamieszczono na rys. 7, a jego subsystemy na rys. 8 i rys. 9 z objaśnieniami do nich zamieszczonych w tabeli 4.



Rys. 7. Model Kwantowej Sztucznej sieci neuronowej, oznaczenia zawarte w tabeli 4 [5]



Rys. 8. Zawartość: Subsystem_W11, Subsystem_W12, Subsystem_W13, Subsystem_W14 oznaczenia zawarte w (tabeli 4)



Rys. 9. Zawartość: Subsystem_W21, Subsystem_W22, Subsystem_W23
oznaczenia zawarte w (tabela 4)

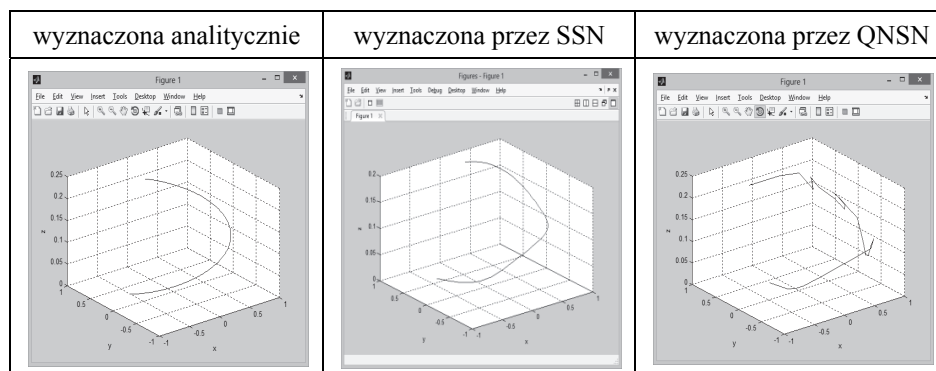
Tabela 4. Oznaczenia do rys. 6 - rys. 8 [5].

Lp.	Blok	Objaśnienie
1	In1-In4	wejścia
2	Out1-Out4	wyjścia
3	A1,A2,A3,A4	macierze wejściowe
4	yx, yy,yz	macierze wyjściowe
5	k	macierz k
6	WspolrzedneXYZ	macierz wyjściowa
7	WszDaneWej	macierz wejściowa
8	Scope,Scope1,Scope2	wyjście danych
9	Subsystem_W11,Subsystem_W12 Subsystem_W13, Subsystem_W14	Schemat realizacji obliczeń w Simulinku pokazano na rys. 8
10	Subsystem_W21, Subsystem_W22 Subsystem_W23	Schemat realizacji obliczeń w Simulinku pokazano na rys. 9
11	Constant	Stała
12	Divide	Blok dzielenia
13	Gain	Blok mnożenia
14	Math Function	Blok funkcji matematycznej e^u

4. BADANIA SYMULACYJNE I KOMPARATYSTYCZNE MODELI

W wyniku porównania funkcjonowania trzech modeli ruchu końca ramienia robota PR-02, a więc modelu analitycznego, modelu neuralnego i modelu kwantowego (neuralnego inspirowanego rozwiązaniami informatyki kwantowej - tabela 5) okazało się, że model neuronalny uzyskany w wyniku nauczania sztucznej sieci neuronowej ruchu końca ramienia robota był bardzo zbliżony do modelu analitycznego, a model kwantowy odtwarzał co prawda tendencje zachowania się końca ramienia robota PR-02, ale wystąpiły w nim nawroty i brak płynności, przy czym sygnał był ciągły.

Tabela 5. Przebiegi trajektorii ruchu końca ramienia robota PR-02 otrzymane w modelach: analitycznym (rys. 2), neuronalnym (rys. 5) oraz kwantowo-neuronalnym (rys. 6-8) [5].



Spowodowane to było małą liczbą obliczeń quasi-równoległych (tylko 1 000) jak na wymagania stawiane w informatyce kwantowej (występowanie stanów mieszanych o nieskończonej liczbie), stąd w dalszych badaniach należałoby zwiększyć liczbę stanów mieszanych zarówno w kwantyzacji jak też w obliczeniach kwantowych przynajmniej o rząd wielkości, czyli np. do 10 tys. kwantowych stanów mieszanych i obliczeń kwantowych. Badania są kontynuowane.

5. WNIOSKI KOŃCOWE I KIERUNKI DALSZYCH BADAŃ

Przeprowadzone badania pokazały, że opracowana metoda kwantyzacji, dekwantyzacji i prowadzenia obliczeń kwantowych może zostać wykorzystana m.in. do inspirowania modelu neuralnego ruchu ramienia robota PR-02 za pomocą rozwiązań informatyki kwantowej. Pokazano, że można przeprowadzać kwantyzację, obliczenia kwantowe i dekwantyzację na komputerze klasycznym, co wiąże się ze znacznym wydłużeniem obliczeń neuronalnych. Dla założonej liczby 1 000 quasi równoległych obliczeń nie uzyskano poprawy przebiegu trajektorii, ale uzyskano bardziej precyzyjny punkt startu i punkt docelowy ruchu ramienia robota PR-02. Kierunek dalszych badań dotyczy przeprowadzenia obliczeń dla 10 000 a nawet 100 000 quasi równoległych obliczeń kwantowych, co na klasycznych komputerach wymaga znacznie dłuższego czasu prowadzenia obliczeń.

LITERATURA

- [1] Byron F. W., Fuller R. W., *Matematyka w fizyce klasycznej i kwantowej*, PWN, Warszawa 1973.
- [2] Dindrof R., *Modernizacja układu sterowania robota przemysłowego PR-02 z napędem pneumatycznym*, *Pomiary Automatyka Robotyka* nr 12, PIAP, Warszawa 2007.
- [3] *Dokumentacja techniczno-ruchowa robota PR-02*. WSK, Kalisz 1979, s.121.
- [4] Giaro K., Kamiński M., *Wprowadzenie do algorytmów kwantowych*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2003.
- [5] Domański P., *Model i implementacja Kwantowej Sztucznej Sieci Neuronowej*
- [6] *na przykładzie ruchu robota PR-02*. Praca inżynierska napisana pod kierunkiem prof. nzw. dr hab. inż. Jerzego Tchórzewskiego w Instytucie Informatyki na Wydziale Nauk Ścisłych UPH w Siedlcach. UPH. Siedlce 2017.
- [7] Heller M., *Mechanika kwantowa dla filozofów*, OBI, Kraków 1996.
- [8] Hirvensalo M., *Algorytmy kwantowe*, WSiP, Warszawa 2004.
- [9] Honczarenko J., *Roboty przemysłowe: Budowa i zastosowanie*. WNT, Warszawa 2004, s. 336.
- [10] *Instrukcja laboratoryjna: Stanowisko dydaktyczne: Manipulator PR-02*, Instytut Cybernetyki Technicznej Politechniki Wrocławskiej, Wrocław 2002.
- [11] Kaczorek T., *Podstawy teorii sterowania*, WNT, Warszawa 2005.
- [12] Kasabov N., *Neuro-, Genetic-, and Quantum Inspired Evolving Intelligent Systems*, *International Symposium on Evolving Fuzzy Systems*, 2006, pp. 63 – 73.
- [13] Osowski S., *Sieci neuronowe do przetwarzania informacji*. OW PW, Warszawa 2013, s. 422.
- [14] Palewski A., Wnuk M., *Stanowisko dydaktyczne manipulator PR-02*, Instytut cybernetyki technicznej Politechniki Wrocławskiej, Raport serii: Sprawozdania nr 1_2002, Wrocław 2002.
- [15] Pittenger A.O., *An Introduction to Quantum Computing Algorithms*, Birkhauser, Boston 2000.
- [16] Pisarek Ł.: *Modernizacja systemu sterowania manipulatora pneumatycznego PR-02*. Praca dyplomowa. Politechnika Świętokrzyska, Kielce 2007.
- [17] Ruciński D., *Neural-evolutionary Modelling of Polish Electricity Power Exchange*, XPlore Digital Library, EPNet, PWr., 2016.
- [18] Szkodny T., *Kinematyka robotów przemysłowych*, wydawnictwo Politechniki Śląskiej, Gliwice 2013.
- [19] Tadeusiewicz R., Szaleniec M., *Leksykon sieci neuronowych*, Wyd. Fundacji „Projekt Nauka”, Wrocław 2015.
- [20] Tchórzewski J., Ruciński D., *Kwantowa sztuczna sieć neuronowa. Obliczenia kwantowe*. Poznan University of Technology. Academic Journals, Poznan 2018.
- [21] Tchórzewski J., Ruciński D., *Quantum inspired evolutionary algorithm to improve the accuracy of a neuronal model of the electric power exchange*, *Proceedings of IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, IEEE Xplore Digital Library, pp. 638-643.

- [22] Tchórzewski J., Ruciński D., Modeling and simulation inspired by quantum methods of the Polish Electricity Stock Exchange, 2017 Progress in Applied Electrical Engineering (PAEE), IEEE XPlore Digital Library, pp. 1-5.
- [23] Tchórzewski J., Marłęga R., Systemowy Algorytm Ewolucyjny do poprawy modelu Towarowej Giełdy Energii Elektrycznej. Część 1. Istota i możliwości algorytmu. Poznan University of Technology Academic Journals Electrical Engineering. Vol. 90/2017, Published by Poznan University of Technology, str. 277-288
- [24] Tchórzewski J., Cybernetyka życia i rozwoju systemów, Monografie nr 22, Wydawnictwo AP, Siedlce 1992.
- [25] Tchórzewski J., Inżynieria rozwoju systemów. Monografie nr 18, Wydawnictwo Uczelniane WSR-P w Siedlcach, Siedlce 1990.
- [26] Tchórzewski J., Wołonka Ł., Możliwości informatyki kwantowej do poprawy dokładności modelowania. Część 2, Ruch robota PR-02. Poznan University of Technology Academic Journals, No 88, Electrical Engineering, 2016.
- [27] Żurada J., Barski M., Jędruch W. Sztuczne sieci neuronowe. Podstawy teorii i zastosowania, Wydawnictwo PWN, Warszawa 1996.

**THEA SYSTEMIC ARTIFICIAL NEURAL NETWORK INSPIRED WITH
QUANTUM TECHNOLOGY. PART 2. THE MODEL OF THE ROBOT
MOVEMENT PR-02**

The article is a continuation of the work under the same main title and subtitle: Part 1. Quantum calculations. This paper presents selected elements of the method and algorithm for inspiring the PR-02 robot's neural motion model with the help of quantum computing solutions. Attention was paid to the problems occurring in the field of quantum computing on classical computers. It has been shown that quantization, quantum computation and dequantization on a classical computer can be performed, which is associated with the prolongation of neuronal calculations. For the assumed number of 1 000 quasi parallel calculations, no improvement in the path trajectory was achieved, but a more precise starting point and target point of the robot arm movement PR-02 were obtained. The research is continued for 10,000 quasi parallel computations, which requires much longer calculation time on classic computers

(Received: 31.01.2018, revised: 18.03.2018)

