

Jerzy TCHÓRZEWSKI\*  
Łukasz WOŁYNKA\*

## MOŻLIWOŚCI INFORMATYKI KWANTOWEJ DO POPRAWY DOKŁADNOŚCI MODELOWANIA. CZĘŚĆ 2 – KAE NA PRZYKŁADZIE RUCHU ROBOTA PR-02

Artykuł zawiera wybrane wyniki badań dotyczące próby opracowania kwantowego algorytmu ewolucyjnego i jego implementacji w j. Matlab do poprawy parametrów modelu neuralnego ruchu ramienia robota PR-02. Populację początkową zbudowano na bazie macierzy wag sztucznej sieci neuronowej. Wylosowane wartości poszczególnych chromosomów populacji początkowej zostały przekształcone na wartości binarne, a te z kolei na wartości kwantowe przy wykorzystaniu opracowanej w tym celu funkcji *quantization()*. Wartość kwantowa genu została określona na podstawie silniejszego stanu czystego reprezentowanego przez podchromosomy, do czego została wykorzystana funkcja *dequantization()*. Selekcję osobników przeprowadzono na bazie modelu neuralnego ruchu robota PR-02 zaimplementowanego w j. Matlab jako funkcja *calculationsNeuralNetworks()*.

SŁOWA KLUCZOWE: sztuczne sieci neuronowe, algorytmy ewolucyjne, środowisko MATLABA i Simulinka, robot PR-02, modelowanie systemów, informatyka kwantowa

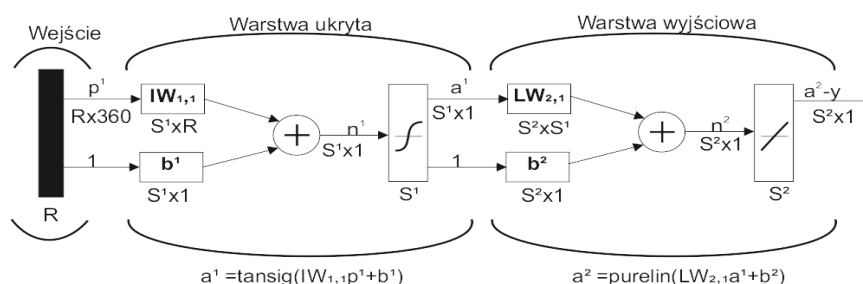
### 1. NEURONALNY MODEL RUCHU RAMIENIA ROBOTA PR-02

Sztuczną Sieć Neuronową (SSN) zaprojektowano jako sieć perceptronową, to jest wielowarstwową i jednokierunkową z czterema wielkościami wejściowymi ( $u_1, u_2, u_3, u_4$  będącymi wektorami ze znormalizowanymi parametrami przegubów robota PR-02), jedną warstwą ukrytą oraz z trzema wielkościami wyjściowymi. Plik uczący oraz plik sterujący zbudowano na bazie wartości czterech stanów przegubów robota PR-02 tworzące macierz wejścia [1, 7–8, 10] oraz współrzędnych przestrzennych położenia końca ramienia robota.

W ten sposób macierz składa się z 360 par uczących i testujących dotyczących wartości czterech wielkości: dwóch obrotów oraz dwóch przesunięć (rys. 1). Do uczenia SSN modelu ruchu ramienia robota PR-02 wykorzystano metodę uczenia Levenberg-Marquardta z regularyzacją Bayesowską [1, 5, 9]. Macierze wartości wielkości wejściowych  $U$  oraz wielkości wyjściowych  $Y$

\* Uniwersytet Przyrodniczo-Humanistyczny, Siedlce.

została znormalizowana przy wykorzystaniu algorytmu MapMinMax wbudowanego w środowisko MATLABA i Simulinka (normalizacja wartości wielkości wejściowych dla każdego stanu przegubu z przedziału  $[-1 \div 1]$ ), przy czym wyznaczone parametry reprezentowane były przez blok *mapminmax* w Simulinku.



Rys. 1. Architektura SSN wykorzystanej do nauczania modelu neuralnego ruchu końca ramienia robota PR-02. Oznaczenia w tekście, [8, 10]

Ważniejsze oznaczenia do rys. 1:

R – wymiar macierzy zawierającej wartości wielkości wejściowych  $u_i$ ,

Rx360 – wymiar macierzy zawierającej wartości wielkości wejściowych  $u_i$  (w środowisku MATLAB plik uczący *JointsInputStates*),

$p^1$  – liczba wielkości wejściowych  $u_i$

$IW_{1,1}$ ,  $LW_{2,1}$  – macierze wag odpowiednio warstwy pierwszej (ukrytej) o wymiarze  $S^1 \times R$  oraz drugiej (wyjściowej) o wymiarze  $S^2 \times S^1$ ,

$b^1$ ,  $b^2$  – wektory biasów odpowiednio warstwy pierwszej o wymiarze  $S^1 \times 1$  oraz warstwy drugiej o wymiarze  $S^2 \times 1$ ,

$a^1 = \text{tansig}(IW_{1,1}p^1 + b^1)$  – funkcja tangensoidalna (tangens hiperboliczny)<sup>1</sup>,

$a^2 = \text{purelin}(LW_{2,1}a^1 + b^2)$  – funkcja liniowa,

n (net) – wartość sygnału z pierwszej lub z drugiej warstwy,

$y_j$  – j-ta wielkości wyjściowa.

## 2. KAE DO POPRAWY PARAMETRÓW MODELU

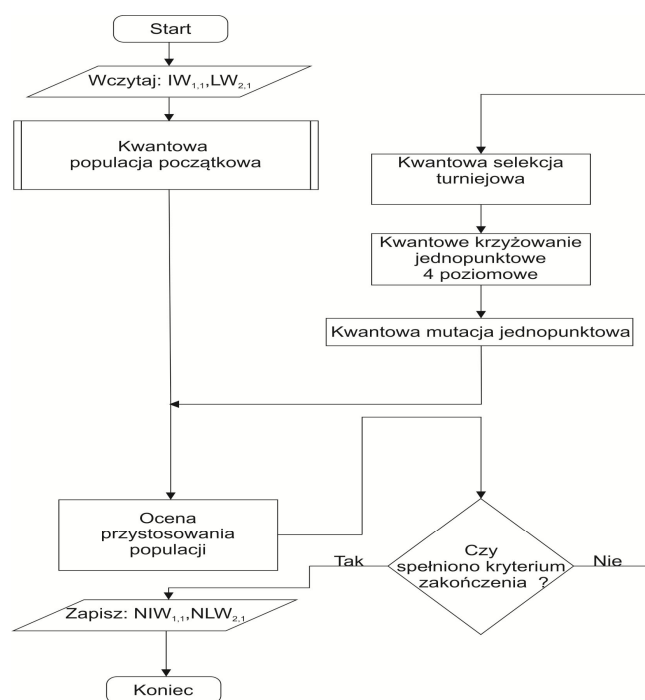
W wyniku studiów i badań w zakresie możliwości informatyki kwantowej do poprawy dokładności modelowania [2–3, 8] opracowano konkretny kwantowy algorytm ewolucyjny (KAE), który zweryfikowano praktycznie na przykładzie modelu neuralnego ruchu ramienia robota PR-02 (KAE) zaimplementowanego w j. Matlab [8, 10] (rys. 2).

<sup>1</sup>

$$f(\text{net}^l) = a^l(\text{net}^l) = \frac{2}{1 + e^{-2n}} - 1$$

Populację początkową (PP) KAE zbudowano na podstawie wartości elementów obu macierzy wag w ustalonych przedziałach wartości obu warstw SSN ( $IW_{1,1}$  i  $LW_{2,1}$ ) wykorzystując operację kwantyzacji zdefiniowaną w pracy [8]. Proces wyznaczania PP został szczegółowo opisany w pracy [10].

W bloku pn. Ocena przystosowania (rys. 2) znajduje się algorytm dokonujący dekwantyzacji, to jest zamiany uzyskanych stanów kwantowych na wartości binarne, a te na wartości dziesiętne, a następnie wyznaczania dla każdego chromosomu reprezentującego obie macierzy wag odległości euklidesowej od rzeczywistych wartości reprezentujących konkretne położenie końca ramienia robota PR-02 podczas ruchu po trajektorii w celu ustalenia polepszenia efektywności i skuteczności KAE.

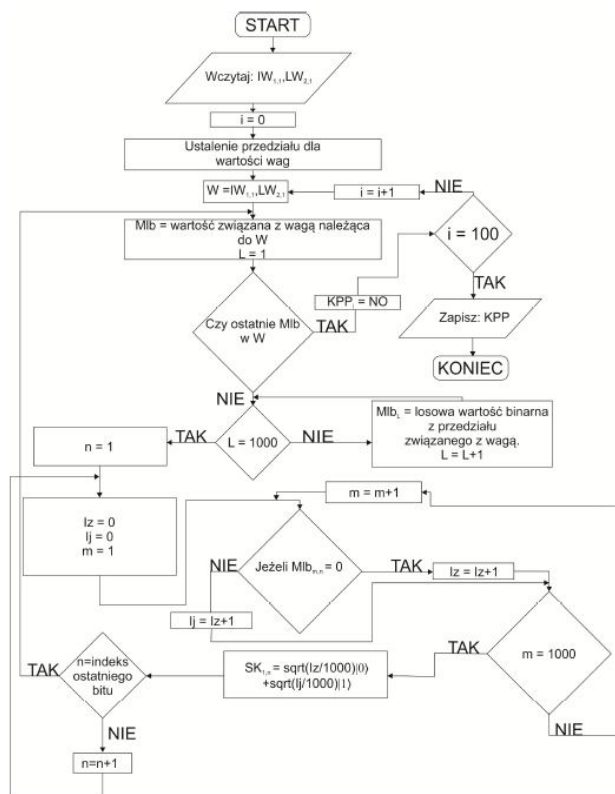


Rys. 2. Kwantowy Algorytm Ewolucyjny do poprawy parametrów modelu neuralnego ruchu końca ramienia robota PR-02. Oznaczenia w tekście, [8, 10]

Następnie na podstawie odległości danego chromosomu od średniej wartości wszystkich chromosomów w danym pokoleniu wyznaczane jest przystosowanie chromosomów do dalszego uczestnictwa w KAE. Blok decyzyjny dotyczący kryterium zakończenia algorytmu KAE został zbudowany na bazie ustalonej dokładności najlepszego chromosomu od średniej odległości wszystkich chromosomów z danej epoki. W KAE wykorzystano selekcję turniejową z powtór-

rzeniami, krzyżowanie jednopunktowe oraz mutację dotyczącą zmiany wartości genu na wartość przeciwną (np. ze stanu  $|0\rangle$  na  $|1\rangle$ , ze stanu  $l_{k0}=0,6|0\rangle+0,8|1\rangle$  na stan  $l_{k1}=0,8|0\rangle+0,6|1\rangle$ ).

Wyżej wymienione bloki KAE wykonują działania na stanach kwantowych. W wyniku działania KAE otrzymano bardziej dokładne wartości wag obu warstw, to jest macierzy:  $IW_{1,1}$  i  $LW_{2,1}$ . Operator kwantyzacji stwarza przy tym możliwość przejścia z systemu dziesiętnego na system dwójkowy, a z systemu dwójkowego na system kwantowy. Następnie dla każdego chromosomu wylosowano nieskończoną liczbę jego stanów (np. wyrażona przez liczbę 1 000), co pozwalało wyznaczyć liczbę wystąpień stanów czystych  $|0\rangle$  oraz liczbę wystąpień  $|1\rangle$ , które wykorzystywano do wygenerowania stanów mieszanych. Było to podyktowane koniecznością uzyskania na komputerze klasycznym zjawiska wystąpienia nieskończonej liczby stanów dla każdego bitu jednocześnie. Wyżej wymieniony algorytm został zamieszczony w postaci schematu blokowego na rys. 3 (blok dotyczący generowania PP).



Rys. 3. Schemat blokowy generowania PP w KAE wykorzystanym do poprawy parametrów modelu neuralnego ruchu końca ramienia robota PR-02. Oznaczenia w tekście, [3, 8, 10]

W losowaniu wykorzystano przedziały z dokładnością większą o jeden rząd wartości. Na tak określonych przedziałach wartości wag wylosowano 1 000 wartości poszczególnych wag, które konwertowano do postaci binarnej (na rys. 3 – macierz  $Mlb$ , gdzie wartości  $L$  i  $m$  stanowią indeksy wiersza omawianej macierzy, a wartości  $n$  indeksy kolumn). Wartości wag (każda waga w liczbie 1 000) zostały wygenerowane z ww. przedziałów oraz przekonwertowane na postać binarną za pomocą m-pliku skryptowego opracowanego w j. Matlab pn. `RandQuantumPopulationGPU` [10].

Wszystkie wartości binarne zostały następnie zakodowane zgodnie ze standardem IEEE 754 dla danej liczby zmiennoprzecinkowej 64 bitowej (podwójnej precyzji)<sup>2</sup> oznaczonych jak w tabeli 1.

Odpowiadające sobie wartości wag pochodzą ze zbioru wartości znajdujących się w danym przedziale wyznaczonym dla konkretnej wartości wagi. Dla każdego takiego zbioru bitów związanych z wartością wagi wyznaczone zostało prawdopodobieństwo występowania danego stanu binarnego. Operacja kwantyzacji związana jest z symulacyjnym wyznaczeniem stanu kwantowego, gdzie prawdopodobieństwo wystąpienia jednego ze stanów binarnych odpowiada amplitudzie stanu bazowego tworzącego jednokubitowy stan kwantowy, co prowadzi do uzyskania stanów mieszanych [8].

Tabela 1. Wartości wag wygenerowane w liczbie 1 000 i przekonwertowane na postać binarną [10]

Indeksy bitów	63	62	...	52	51	50	...	2	1	0
Przykładowe wartości binarne	0	0	...	1	0	1	...	0	0	0
Części wartości binarnej:	Bit znaku	Wykładnik			Mantysa					

Wygenerowano zatem 100 chromosomów (macierzy wag). Na klasycznym komputerze prowadzenie takich równoległych obliczeń jest związane z wysokim kosztem algorytmu (utworzenie kwantowej populacji początkowej trwało 30 min). W celu przyspieszenia obliczeń część kodu została zrównoleglona poprzez wykorzystanie `Parallel Computing Toolboxa` (PCT). Procesorem wykonującym obliczenia był procesor Intel Core i7 3610QM, 2,3 MHz, posiadający 4 rdzenie, na których równolegle uruchomione były oddzielne oraz niezależne procesy obliczeniowe, wykonujące równolegle ww. skrypt<sup>3</sup>. Procesy działały

<sup>2</sup> Zgodnie z oznaczeniami w przyjętym standardzie bit znaku odpowiada czy dana wartość jest dodatnia czy ujemna (0 wartość dodatnia, 1 wartość ujemna), a wykładnik składa się z 11 bitów, przy czym nadmiar wynosi 1023, tzn., że najmniejszą wartością jest -1023 a największą 1024. Z kolei mantysa jest zapisana w kodzie U1, są w niej jedynie bity ułamkowe.

<sup>3</sup> W skrypcie napisanym w j. Matlab zostały użyte pętle `parfor` dla każdego z zadeklarowanych procesów., które powodują, że są one wykonywane na każdym z zadeklarowanych procesów.

niezależnie od głównego procesu obsługiwanego przez kompilatora MATLABA. Operator dekwantyzacji daje z kolei możliwość przejścia z postaci kwantowej na postać binarną, a następnie na postać dziesiętną.

Na początku dla każdego chromosomu reprezentowanego przez jeden kubit wyznaczono stan bazowy, jako stan, który będzie miał większe prawdopodobieństwo wystąpienia. W tym wypadku stosując wzór na amplitudy stanu jednokubitowego można wyznaczyć przedział  $[0,1]$ , a prawdopodobieństwa amplitud stanów bazowych przedstawić np. jako obszary, które zajmują, przy czym, gdy wyznaczony jest jeden stan dotyczący wartości chromosomu, wówczas drugi stan można wyznaczyć z zasady superpozycji stanów [3, 8, 10]. W tym przypadku można dokonać losowego wyboru wartości z przedziału  $\langle 0,1 \rangle$ , a następnie sprawdzić, w którym obszarze (czy w obszarze stanu  $|0\rangle$ , czy w obszarze stanu  $|1\rangle$ ) mieści się dana wartość. Zwiększa to szansę uzyskania stanu, którego wartość prawdopodobieństwa wystąpienia amplitudy stanu bazowego jest większa. Z drugiej strony nie jest wykluczone wylosowanie mniejszej wartości z obszaru tworzonego przez prawdopodobieństwo wystąpienia amplitudy stanu bazowego. W taki sposób może zostać też wyznaczony stan czysty  $|0\rangle$  lub  $|1\rangle$ .

Amplitudy związane ze stanami czystymi są następnie porównywane ze sobą, aby wyodrębnić stan bardziej prawdopodobny jako wartość stan mieszanego genu w chromosomie. W celu uproszczenia opisu stany mieszane oraz czyste związane z danym chromosomem przyjęto jako podchromosomy.

W tym wypadku jest możliwe odwzorowanie czystych stanów na bity, odpowiednio: 0 na  $|0\rangle$  oraz 1 na  $|1\rangle$  tworzących jedną wartość wagi. Następnie dekodowanie 64 bitów będących jednym chromosomem do postaci wartości należących do liczb rzeczywistych przy użyciu standardu IEEE 754. Strukturę chromosomu pokazano na przykładzie pierwszego chromosomu  $ch_1$  i ostatniego chromosomu  $ch_{100}$  zamieszczono w tabeli 2.

Tabela 2. Struktura chromosomu z przykładowymi wartościami wag wygenerowanych w liczbie 1 000 (chromosom nr 1 oraz chromosom nr 100) [3, 10]

Indeksy	Numer genu	0	1	...	63	
chromosom: $ch_1$	podchromosom $pch_1^1$	$\alpha 0\rangle$	0.6	0.4	...	0.9949
		$\beta 1\rangle$	0.8	0.9165	...	0.1
		Stan czysty	$ 1\rangle$	$ 1\rangle$	...	$ 1\rangle$
chromosom: $ch_{100}$	podchromosom $pch_{100}^1$	$\alpha 0\rangle$	0.77659	0.3	...	0.9165
		$\beta 1\rangle$	0.63	0.9539	...	0.4
		Czysty stan	$ 1\rangle$	$ 0\rangle$	...	$ 0\rangle$

Chromosom  $ch_1$  składa się z jednego podchromosomu  $pch_1^1$  otrzymanego w procesie kwantyzacji. Każdy z podchromosomów składa się ze stanu mieszanego i uzyskanego w procesie dekwantyzacji stanu czystego. Natomiast chromosom  $ch_{100}$  zbudowany jest z podchromosomu  $pch_{100}^1$ . Podchromosomy chromosomu  $ch_{100}$  mają taką samą strukturę jak w przypadku chromosomu  $ch_1$ .

Po wykonaniu dekwantyzacji dokonano oceny przystosowania 100 chromosomów reprezentujących model neuronalny w postaci dwóch macierzy wag. Do oceny przystosowania wykorzystano funkcję zbudowaną jako wektor wielkości wyjściowych sztucznej sieci neuronowej [6, 8, 10]. Wygenerowane wartości przez model neuronalny w postaci SSN są punktami trajektorii ruchu końca ramienia robota PR-02. Aby można było określić ich przystosowanie, a dokładniej określić błąd względem rzeczywistej trajektorii, należy obliczyć odległość euklidesową każdego punktu otrzymanego z prostego zadania kinematyki dla robota PR-02 [6, 8, 10] w procesie dostosowania prowadzonego przez KAE [3, 6].

Przyjęta metoda selekcji pozwala na większą eksplorację danych, poprzez wybór w  $n$ -turniejach, losowo wybranej  $k$ -elementowej grupy, z której wyłaniany jest najlepiej przystosowany jeden chromosom, który będzie brał udział w krzyżowaniu [1–2, 5, 8–10]. W przypadku przeprowadzonego eksperymentu wyodrębniono 100 turniejów z grupą pięciu chromosomów. Za pomocą metody selekcji otrzymuje się 100 chromosomów do operacji krzyżowania.

Przyjęto operator krzyżowania jednopunktowego, a proces krzyżowania przeprowadzono w postaci krzyżowania z zanurzeniem czteropoziomowym (poziom najwyższy model neuronalny, następnie pomiędzy macierzami wag, jej wierszami oraz na najniższym poziomie pomiędzy elementami macierzy zapisanymi w postaci liczb kwantowych) [3, 8, 10]. Operator mutacji przyjęto także jako operator jednopunktowy [5, 8–10] polegający na zamianie jednego stanu czystego na drugi względnie na zamianie stanu mieszanego na stan mieszany z nim sprzężony.

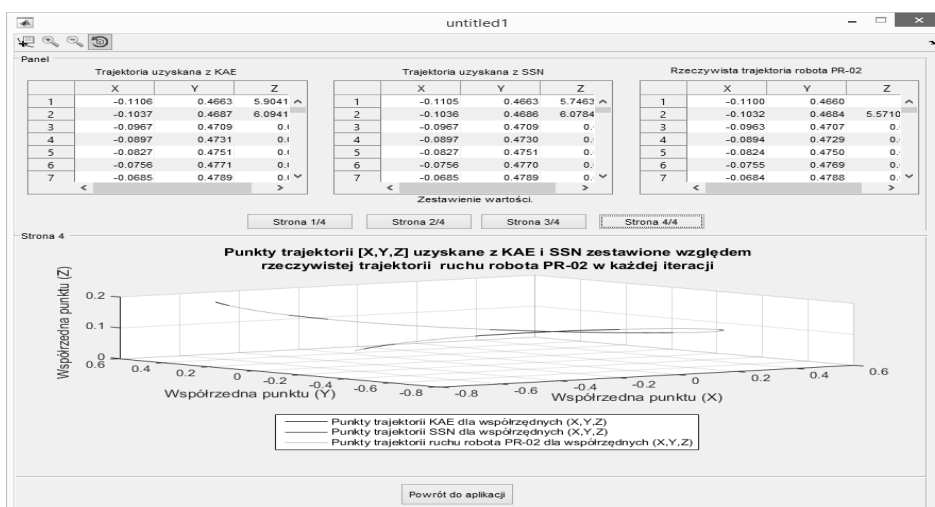
### 3. BADANIE WRAŻLIWOŚCI KAE

Badania wrażliwości działania algorytmu KAE potwierdziły wysoką jego krzepkość (skuteczność i efektywność) w procesie poprawy dokładności wartości wag, co przełożyło się na poprawę jakości modelu neuralnego ruchu końca ramienia robota PR-02. Algorytm był przy tym wrażliwy na zmianę prawdopodobieństwa wystąpienia mutacji, co zestawiono w tabeli 3. Najlepsze rezultaty uzyskano przy zastosowaniu wartości prawdopodobieństwa mutacji 0.001 oraz dokładności  $5,0000055 \cdot 10^{-10}$ , dla której średnia arytmetyczna błędów chromosomów najlepiej przystosowanych dla 100 pokoleń wyniosła 0.09.

Tabela 3. Wyniki badania wrażliwości KAE na zmianę prawdopodobieństwa mutacji [5, 10]

Liczba przebiegów algorytmu KAE dla 100 epok	Prawdopodobieństwo mutacji	Średnia arytmetyczna z uzyskanych błędów dla najlepszych osobników (Elita)
50	0,5	2.34
	0,1	1.66
	0.01	0.23
	0.001	0.09
	0.0001	0.27
	0.00001	0.65
	0.000001	0.76
	0.0000001	0.92

Algorytm KAE dawał lepsze wyniki niż dla tych samych parametrów klasyczny AE. Wartość błędu dla elity KAE wyniosła  $0,17575 \cdot 10^{-11}$  (rys. 4), a dla klasycznego AE wyniosła  $0,1757286 \cdot 10^{-8}$  (zwiększenie dokładności o trzy rzędy).



Rys. 4. Wyniki działania KAE. Oznaczenia w tekście, [10]

#### 4. WNIOSKI

Niniejszy artykuł jest kontynuacją pracy pod tym samym tytułem głównym oraz pod tytułem Część 1. Kwantowy algorytm ewolucyjny [8].

Uzyskane wyniki badań na KAE zaimplementowanym w j. Matlab potwierdziły możliwość wykorzystania informatyki kwantowej do praktycznego zbu-



dowania modelu kwantowego algorytmu ewolucyjnego i wykorzystania go do poprawy parametrów ruchu końca ramienia robota PR-02.

Badania wrażliwości wykazały, że kwantowy algorytm ewolucyjny poprawił dokładność modelu neuralnego ruchu ramienia robota PR-02 o trzy rzędy dokładności.

### LITERATURA

- [1] Albert F.Y.C., Koh S.P., Chen C.P., Tiong S.K., Edwin S.Y.S., Optimizing Joint Angles of Robotic Manipulator Using Genetic Algorithm. IACSIT Press, wyd. 2, Singapur 2011.
- [2] Hendzel Z., Żylski W.: Krzepki obserwator uogólnionych prędkości mobilnego robota kołowego, Materiały VI Krajowej Konferencji Robotyki, Wrocław 1998.
- [3] Hirvensalo M., Algorytmy kwantowe, WSiP, Warszawa 2004.
- [4] Kaczorek T., Dzieliński A., Dąbrowski W., Łopatka R., Podstawy teorii sterowania. WNT, Warszawa 2007.
- [5] Michalewicz Z., Algorytmy genetyczne + struktury danych = programy ewolucyjne. WNT, Warszawa 1999.
- [6] Osowski S., Sieci neuronowe do przetwarzania informacji. OWPW, Warszawa 2013.
- [7] Szkodny T.: Kinematyka robotów przemysłowych. Wydawnictwo Politechniki Śląskiej, Gliwice 2013.
- [8] Tchórzewski J., Możliwości informatyki kwantowej do poprawy dokładności modelowania. Część 1. Kwantowy algorytm ewolucyjny. Poznan University of Technology. Academic Journal. Wyd. PP, Poznań 2016 (w procesie wydawniczym).
- [9] Tchórzewski J., Chyży E., Researching the Development of the Electrical Power System Using Systemically Evolutionary Algorithm. International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 6, No. 1, pp. 1-10, 2016.
- [10] Wołynka Ł., Model i implementacja kwantowego algorytmu ewolucyjnego na przykładzie ruchu robota PR-02. Praca magisterska napisana pod kierunkiem prof. nzw. dr hab. inż. Jerzego Tchórzewskiego w Zakładzie Modelowania i Projektowania Systemów Informatycznych Instytutu Informatyki na Wydziale Nauk Ścisłych, UPH, Siedlce 2015.

### POSSIBILITY OF QUANTUM COMPUTER TO IMPROVE ACCURACY OF MODELING. PART 2. KAE ON EXAMPLE ON MOTION ROBOT PR-02

The article contains selected results of research on trying to develop a quantum evolutionary algorithm and its implementation in Matlab to improve the parameters of the model of neural movement of the robot arm PR-02. The initial population is constructed on the basis of the matrix weights artificial neural network. The drawn values of individual initial population of chromosomes have been converted to binary values, and the latter value using quantum developed for this purpose function of *quatization()*. The

value of the quantum of the gene was determined on the basis of a stronger state of pure represented by subchromosomes, what was used a function of dequantization(). Selection of individuals was carried out based on the model of neural traffic robot PR-02 implemented as a function of *calculationsNeuralNetworks()*.

*(Received: 14. 02. 2016, revised: 3. 03. 2016)*