

ZASTOSOWANIE SZTUCZNYCH SIECI NEURONOWYCH W DOBORZE FUNKCJI NAPIĘDOWYCH ŻURAWI NA PODATNYM PODŁOŻU

Andrzej URBAŚ*, Marek SZCZOTKA*

*Katedra Informatyki Stosowanej, Wydział Zarządzania i Informatyki, Akademia Techniczno-Humanistyczna
ul. Willowa 2, 43-309 Bielsko-Biała

aurbas@ath.edu.pl, mszczotka@ath.edu.pl

Streszczenie: W artykule przedstawiono model dynamiczny układu żurawia chwytakowego posadowionego podatnie. Model posłużył do wykonania optymalizacji dynamicznej celem doboru funkcji napędowych ruchu obrotu i wysięgu zapewniających ruch ładunku po zadanej trajektorii (okręgu) oraz stabilizację ładunku po zatrzymaniu, mimo podatności podparcia. Ze względu na długi czas obliczeń optymalizacyjnych, model jest mało przydatny do sterowania układem w czasie rzeczywistym. Zastosowano zatem metodologię sztucznych sieci neuronowych, które generują odpowiedź dla układu sterowania w czasie znacznie krótszym.

1. WPROWADZENIE

Przedstawione w niniejszym artykule rozważania, dotyczące analizy dynamicznej żurawia chwytakowego posadowionego podatnie, są kontynuacją badań prowadzonych przez autorów od kilku lat (Urbaś, 2007), (Urbaś i Wojciech, 2008, 2009). W pierwszych pracach przedstawiono model matematyczny, jego weryfikację pośrednią (MSC. ADAMS; Urbaś, 2007) oraz podjęto próby doświadczalnego zweryfikowania modelu (Urbaś i inni, 2008). W kolejnych pracach (Urbaś i inni, 2009), (Urbaś i Wojciech, 2009), na podstawie doświadczeń (Maczyński i Wojciech, 2003, 2009), zastosowano optymalizację dynamiczną do określenia funkcji napędowych.

Model dynamiczny żurawia opisany został z zastosowaniem formalizmu współrzędnych złączonych i transformacji jednorodnych. Równania ruchu wyprowadzono z równań Lagrange'a II rodzaju. Spośród podstawowych metod całkowania równań, metoda Newmarka cechuje się dobrą stabilnością przy całkowaniu ze stosunkowo dużym krokiem. W zadaniach optymalizacji jest to ważne ze względu na znaczną czasochłonność obliczeń. Zadanie optymalizacyjne polegające na wyznaczeniu napędów żurawia (obrotu i wysięgu), realizujące zadaną trajektorię i stabilizujące ładunek po zatrzymaniu, przy jednoczesnej kompensacji podatności podparcia, rozwiązywano stosując metodę Nelder-Meada.

W przypadku rozpatrywania zagadnień optymalizacyjnych, ze względu na długie czasy obliczeń numerycznych, a tym samym ich niską użyteczność w sterowaniu w czasie rzeczywistym, można zastosować sztuczne sieci neuronowe (Fałat i inni, 2005), (Nakazano K. i inni, 2008).

2. MODEL MATEMATYCZNY UKŁADU

Rozpatrywany model żurawia przedstawiono na Rys. 1. Układ składa się z czterech sztywnych członów, stanowią-

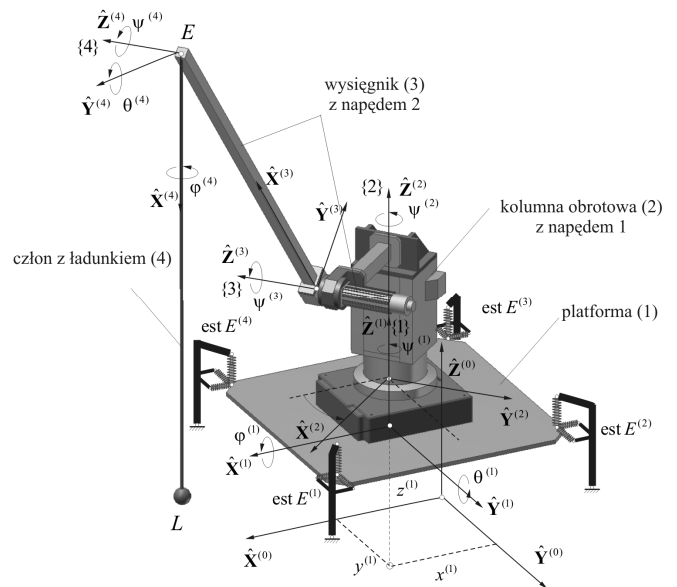
czych strukturę otwartego łańcucha kinematycznego. W opisie ich ruchu zastosowano współrzędne złączone.

Wektory definiujące ruch poszczególnych członów przedstawić można zatem w postaciach:

$$\mathbf{q}^{(p)} = \begin{bmatrix} \mathbf{q}^{(p-1)} \\ \tilde{\mathbf{q}}^{(p)} \end{bmatrix}, \quad p = 1, 2, 3, 4, \quad (1)$$

gdzie: $\tilde{\mathbf{q}}^{(1)} = [x^{(1)} \quad y^{(1)} \quad z^{(1)} \quad \psi^{(1)} \quad \theta^{(1)} \quad \phi^{(1)}]^T$,

$\tilde{\mathbf{q}}^{(2)} = [\psi^{(2)}]$, $\tilde{\mathbf{q}}^{(3)} = [\psi^{(3)}]$, $\tilde{\mathbf{q}}^{(4)} = [\psi^{(4)} \quad \theta^{(4)} \quad \phi^{(4)}]^T$.



Rys. 1. Model żurawia posadowionego podatnie

Macierze definiujące transformację współrzędnych z układów lokalnych do układu odniesienia przedstawić można zależnościami:

$$\mathbf{B}^{(p)} = \mathbf{B}^{(p-1)} \tilde{\mathbf{B}}^{(p)}, \quad (2)$$

gdzie: $\mathbf{B}^{(0)} = \mathbf{I}$,

$$\tilde{\mathbf{B}}^{(1)} = \begin{bmatrix} 1 & -\psi^{(1)} & \theta^{(1)} & x^{(1)} \\ \psi^{(1)} & 1 & \varphi^{(1)} & y^{(1)} \\ -\theta^{(1)} & \varphi^{(1)} & 1 & z^{(1)} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{przyjęto małe kąty } \psi^{(1)}, \theta^{(1)}, \varphi^{(1)}),$$

$$\tilde{\mathbf{B}}^{(2)} = \begin{bmatrix} c\psi^{(2)} & -s\psi^{(2)} & 0 & x_{2org}^{(1)} \\ s\psi^{(2)} & c\psi^{(2)} & 0 & y_{2org}^{(1)} \\ 0 & 0 & 1 & z_{2org}^{(1)} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\tilde{\mathbf{B}}^{(3)} = \begin{bmatrix} c\psi^{(3)} & -s\psi^{(3)} & 0 & x_{3org}^{(2)} \\ 0 & 0 & -1 & y_{3org}^{(2)} \\ s\psi^{(3)} & c\psi^{(3)} & 0 & z_{3org}^{(2)} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\tilde{\mathbf{B}}^{(4)} = \begin{bmatrix} c\psi^{(4)} & -s\psi^{(4)} & 0 & x_{4org}^{(3)} \\ s\psi^{(4)} & c\psi^{(4)} & 0 & y_{4org}^{(3)} \\ 0 & 0 & 1 & z_{4org}^{(3)} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta^{(4)} & 0 & s\theta^{(4)} & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta^{(4)} & 0 & c\theta^{(4)} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\varphi^{(4)} & -s\varphi^{(4)} & 0 \\ 0 & s\varphi^{(4)} & c\varphi^{(4)} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$x_{porg}^{(p-1)}, y_{porg}^{(p-1)}, z_{porg}^{(p-1)}$ są współrzędnymi początku $\{p\}$ -tego układu współrzędnych w układzie $\{p-1\}$.

Ruch układu opisuje zatem wektor współrzędnych uogólnionych:

$$\mathbf{q} = \mathbf{q}^{(4)} = [\tilde{\mathbf{q}}^{(1)T} \quad \tilde{\mathbf{q}}^{(2)T} \quad \tilde{\mathbf{q}}^{(3)T} \quad \tilde{\mathbf{q}}^{(4)T}]^T. \quad (3)$$

W przypadku założeniu o wymuszeniu kinematycznym ruchu członów 2 i 3 przyjmuje się:

$$\psi^{(k)} = \alpha_0^{(k)}, \quad k = 2, 3 \quad (4)$$

gdzie: $\psi^{(k)}$ – kąty określone w (1), $\alpha_0^{(k)}$ – znane funkcje czasu.

Równania ruchu układu wyprowadzono z równań Lagrange'a II rodzaju:

$$\varepsilon_k(E) + \frac{\partial V}{\partial q_j} + \frac{\partial D}{\partial \dot{q}_j} = Q_j, \quad j = 1, \dots, 11 \quad (5)$$

Gdzie: $\varepsilon_k(E) = \frac{d}{dt} \frac{\partial E}{\partial \dot{q}_j} - \frac{\partial E}{\partial q_j}$ są operatorami Lagrange'a,

E energia kinetyczna układu, V – energia potencjalna układu, D – funkcja dyssypacji energii, Q_j – niepotencjalne siły uogólnione, q_j, \dot{q}_j – współrzędne i prędkości uogólnione z (3).

2.1. Energia kinetyczna i potencjalna sił ciężkości układu

Energię kinetyczną i potencjalną sił ciężkości układu wyrażają zależności:

$$E = \sum_{p=1}^4 \tilde{E}^{(p)}, \quad (6.1)$$

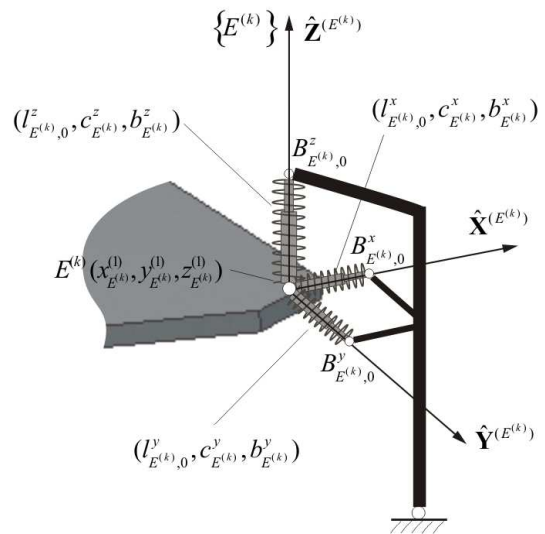
$$V_g = \sum_{p=1}^4 \tilde{V}_g^{(p)}, \quad (6.2)$$

gdzie: $\tilde{E}^{(p)} = \text{tr}\{\dot{\mathbf{B}}^{(p)} \mathbf{H}^{(p)} \dot{\mathbf{B}}^{(p)T}\}$, $\mathbf{H}^{(p)}$ – macierz pseudo-bezwładnościowa, $\tilde{V}_g^{(p)} = m^{(p)} g \boldsymbol{\theta}_3 \mathbf{B}^{(p)} \tilde{\mathbf{r}}_{C^{(p)}}^{(p)}$, $\mathbf{H}^{(p)}$ – macierz pseudo-bezwładnościowa, $\boldsymbol{\theta}_3 = [0 \ 0 \ 1 \ 0]$, g – przyspieszenie ziemskie, $\tilde{\mathbf{r}}_{C^{(p)}}^{(p)}$ – wektor definiujący położenie środka masy członu p , w lokalnym układzie współrzędnych.

2.2. Energia odkształcenia sprężystego oraz funkcja dyssypacji energii podpór

Układ podpór modelowano za pomocą elementów sprężysto-tłumiących (est $E^{(k)}$ ($k = 1, 2, 3, 4$)) – Rys. 2.

est $E^{(k)}$ ($k = 1, 2, 3, 4$)



Rys. 2. Model podatnego połączenia

Energię odkształcenia sprężystego oraz funkcję dyssypacji energii można wyrazić w postaciach:

$$V_e = \sum_{k=1}^4 \sum_{\alpha=\{x,y,z\}} V_{E^{(k)}}^\alpha, \quad (7.1)$$

$$D_e = \sum_{k=1}^4 \sum_{\alpha=\{x,y,z\}} D_{E^{(k)}}^\alpha, \quad (7.2)$$

gdzie:

$$V_{E^{(k)}}^\alpha = \frac{1}{2} (d_{E^{(k)}}^\alpha)^T c_{E^{(k)}}^\alpha d_{E^{(k)}}^\alpha, \quad D_{E^{(k)}}^\alpha = \frac{1}{2} (d_{E^{(k)}}^\alpha)^T b_{E^{(k)}}^\alpha d_{E^{(k)}}^\alpha,$$

$d_{E^{(k)}}^\alpha = l_{E^{(k)}}^\alpha - l_{E^{(k)},0}^\alpha$ – odkształcenie sprężyny w est $E^{(k)}$
w kierunku α , $l_{E^{(k)}}^\alpha = \left| \mathbf{U}_{B_{E^{(k)},0}^\alpha}^{(E^{(k)})} \mathbf{q}^{(1)} \right|$ – długość sprężyny napiętej, $l_{E^{(k)},0}^\alpha$ – długość sprężyny nienapiętej, $c_{E^{(k)}}^\alpha, b_{E^{(k)}}^\alpha$ – współczynniki sztywności i tłumienia,

$$\mathbf{U}_{B_{E^{(k)},0}^\alpha}^{(E^{(k)})} = \begin{bmatrix} 1 & 0 & 0 & 0 & z_{B_{E^{(k)},0}^\alpha}^{(E^{(k)})} & -y_{B_{E^{(k)},0}^\alpha}^{(E^{(k)})} \\ 0 & 1 & 0 & -z_{B_{E^{(k)},0}^\alpha}^{(E^{(k)})} & 0 & x_{B_{E^{(k)},0}^\alpha}^{(E^{(k)})} \\ 0 & 0 & 1 & y_{B_{E^{(k)},0}^\alpha}^{(E^{(k)})} & -x_{B_{E^{(k)},0}^\alpha}^{(E^{(k)})} & 0 \end{bmatrix}$$

2.3. Równania ruchu układu

Równania ruchu opisanego powyżej układu można przedstawić w postaci:

$$\mathbf{A}^{(p)} \ddot{\mathbf{q}}^{(p)} = \begin{bmatrix} \mathbf{f}_1^{(p)} & -\frac{\partial V_e}{\partial \mathbf{q}^{(1)}} & -\frac{\partial D_e}{\partial \dot{\mathbf{q}}^{(1)}} \\ \mathbf{f}_2^{(p)} \\ \mathbf{f}_3^{(p)} \\ \mathbf{f}_4^{(p)} \end{bmatrix}, \quad (8)$$

gdzie: $\mathbf{A}_{i,j}^{(p)} = \sum_{l=\max\{i,j\}}^p \tilde{\mathbf{A}}_{i,j}^{(l)}$, $\tilde{\mathbf{A}}_{v,j}^{(p)} = \left(\tilde{a}_{n^{(v-1)}+l, n^{(j-1)}+s}^{(p)} \right)_{l=1, \dots, \tilde{n}^{(v)}; s=1, \dots, \tilde{n}^{(j)}}$,

$v, j = 1, \dots, p$, $\tilde{a}_{k,j}^{(p)} = \text{tr} \left\{ \mathbf{B}_k^{(p)} \mathbf{H}^{(p)} \mathbf{B}_j^{(p)T} \right\}$,

$\mathbf{f}_i^{(p)} = \sum_{l=i}^p [-\tilde{\mathbf{h}}_i^{(l)} - \tilde{\mathbf{g}}_i^{(l)}]$, $\tilde{\mathbf{h}}_v^{(p)} = \left(\tilde{h}_{n^{(v-1)}+l}^{(p)} \right)_{l=1, \dots, \tilde{n}^{(v)}}$, $v = 1, \dots, p$,

$\tilde{\mathbf{h}}_k^{(p)} = \sum_{i=1}^{n^{(p)}} \sum_{j=1}^{n^{(p)}} \text{tr} \left\{ \mathbf{B}_k^{(p)} \mathbf{H}^{(p)} \mathbf{B}_{i,j}^{(p)T} \right\} \dot{q}_i^{(p)} \dot{q}_j^{(p)}$,

$\tilde{\mathbf{g}}_v^{(p)} = \left(\tilde{g}_{n^{(v-1)}+l}^{(p)} \right)_{l=1, \dots, \tilde{n}^{(v)}}$, dla $v = 1, \dots, p$,

$\tilde{\mathbf{g}}_l^{(p)} = m^{(p)} g \mathbf{0}_3 \mathbf{B}_l^{(p)} \tilde{\mathbf{r}}_{C^{(p)}}^{(p)}$,

$\frac{\partial V_e}{\partial \mathbf{q}^{(1)}} = \sum_{k=1}^4 \sum_{\alpha=\{x,y,z\}} c_{E^{(k)}}^\alpha \left(l_{E^{(k)}}^\alpha - l_{E^{(k)},0}^\alpha \right) \frac{1}{l_{E^{(k)}}^\alpha} \left[\left(\mathbf{U}_{B_{E^{(k)},0}^\alpha}^{(E^{(k)})} \right)^T \mathbf{U}_{B_{E^{(k)},0}^\alpha}^{(E^{(k)})} \mathbf{q}^{(1)} \right]$

$\frac{\partial D_e}{\partial \dot{\mathbf{q}}^{(1)}} = \sum_{k=1}^4 \sum_{\alpha=\{x,y,z\}} b_{E^{(k)}}^\alpha \left[\left(\mathbf{U}_{B_{E^{(k)},0}^\alpha}^{(E^{(k)})} \right)^T \mathbf{U}_{B_{E^{(k)},0}^\alpha}^{(E^{(k)})} \dot{\mathbf{q}}^{(1)} \right]$.

Ruch układu opisany jest zatem układem jedenastu nieliniowych równań różniczkowych zwyczajnych drugiego rzędu, które w implementacji komputerowej całkowano metodą Newmarka ze stałym krokiem całkowania.

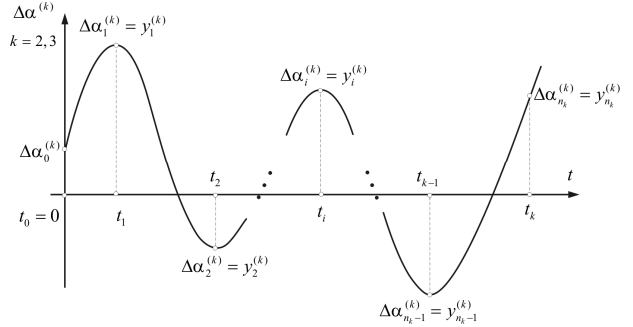
3. ZADANIE OPTYMALIZACJI DYNAMICZNEJ

Celem optymalizacji jest znalezienie korekt $\Delta \alpha^{(k)}$ ($k=2,3$) jakie należy dodać do początkowych (założonych)

przebiegów funkcji napędowych $\alpha_0^{(k)} = \psi^{(k)}$, aby zapewnić realizację założonej trajektorii (ruch po okręgu) i pozycjonowanie ładunku po zatrzymaniu, kompensując jednocześnie podatność podparcia. Przyjmuje się, że:

$$\alpha^{(k)} = \alpha_0^{(k)} + \Delta \alpha^{(k)}. \quad (9)$$

Korekty $\Delta \alpha^{(k)}$ wyznaczono stosując interpolację funkcjami sklejonymi trzeciego rzędu zdefiniowanymi przez n_k-1 wartości – Rys. 3.



Rys. 3. Interpolacja funkcjami trzeciego stopnia

Celem pojedynczego zadania optymalizacji dynamicznej jest minimalizacja funkcjonau:

$$F(\Delta \alpha^{(k)}) = c_1 \frac{1}{t_k} \sqrt{\int_0^{t_k} [r_L - r_0]^2 dt} + c_2 E(t_k) + c_3 \Delta(t_k), \quad (10)$$

gdzie: c_1, c_2, c_3 – współczynniki, $r_L^2 = x_L^2 + y_L^2$ – współrzędne ładunku, r_0 – promień założonej trajektorii (okręgu), $E(t_k)$ – energia kinetyczna po zatrzymaniu, $r_E = [x_E \ y_E]^T$ – wymagana pozycja końcowa ładunku,

$$\Delta(t_k) = \sqrt{(x_L - x_E)^2 + (y_L - y_E)^2}$$

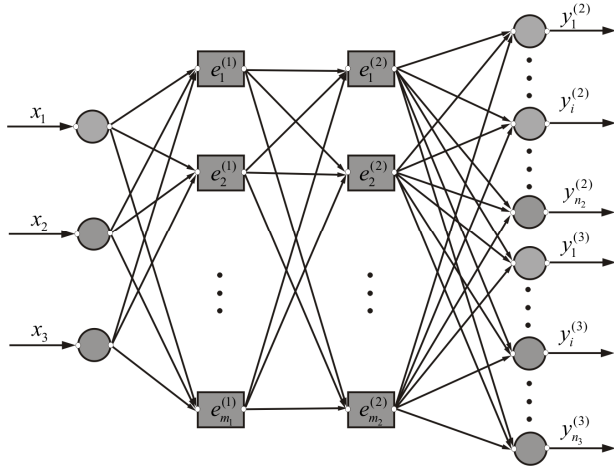
4. SZTUCZNA SIĘĆ NEURONOWA

Model wielowarstwowej, sigmoidalnej sztucznej sieci neuronowej przedstawiono na Rys. 4. Na warstwie wejściowej założono trzy neurony, na które podaje się wartości wejściowe x_1, x_2, x_3 . Sieć posiada dwie warstwy ukryte, posiadające odpowiednio 20 i 50 neuronów. Liczba neuronów warstwy wyjściowej ustalana jest arbitralnie, w zależności od liczby punktów dyskretnych określających przebieg funkcji napędowych. Wartościami wejściowymi, przekazywanymi do wejść sieci, są: $x_1 = m^{(L)}$ – masa ładunku, $x_2 = c_{E^z}^z$ – sztywność podpór w kierunku pionowym, $x_3 = \alpha_{\max}^{(2)}$ – kąt obrotu żurawia

Natomiast wektor wyjściowy \mathbf{y} zawiera dyskretne wartości obu funkcji napędowych $\Delta \alpha^{(k)}$:

$$\mathbf{y} = [y_1^{(2)} \ \dots \ y_{n_2}^{(2)} \ | \ y_1^{(3)} \ \dots \ y_{n_3}^{(3)}]^T, \quad (11)$$

gdzie: n_2, n_3 – liczba punktów definiujących funkcje $\Delta \alpha^{(2)}$ i $\Delta \alpha^{(3)}$.



Rys. 4. Architektura sieci neuronowej

Odpowiedź sieci neuronowej na zadany wektor wejściowy \mathbf{x} może być obliczony jako:

$$\mathbf{y} = \mathbf{y}(\mathbf{x}, \mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}) = \mathbf{e}^{(3)} \left\{ \mathbf{W}^{(3)} \mathbf{e}^{(2)} \left[\mathbf{W}^{(2)} \mathbf{e}^{(1)} (\mathbf{W}^{(1)} \mathbf{x}) \right] \right\} \quad (12)$$

gdzie: $\mathbf{e}^{(k)} = [e_1^{(k)} \dots e_{m_k}^{(k)}]^T$,

$$e_i^{(k)}(x) = \begin{cases} \frac{1}{1 + e^{-\lambda x}} & \text{dla } k = 1, 2, \\ x & \text{dla } k = 3 \end{cases}$$

$\mathbf{W}^{(k)} = (w_{i,j}^{(k)})_{\substack{i=1,2,3, \\ j=1,\dots,m_k}}^{k=1,2,3}$ – macierz współczynników warstwy k ,

$w_{i,j}^{(k)}$ – waga połączenia neuronu i z wejściem j w k -tej warstwie, J , m_k – odpowiednio, liczba wejść i liczba neuronów (wyjść) warstwy k .

Wyznaczanie składowych macierzy współczynników wagowych $w_{i,j}^{(k)}$ dla wszystkich warstw odbywa się w procesie uczenia sieci neuronowej. Podobnie jak w przypadku klasycznej optymalizacji, iteracyjnie poszukuje się minimum dla następującego funkcjonau:

$$\Psi(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}) = \frac{1}{2} \sum_{s=1}^{m_p} [\mathbf{y}_s - \mathbf{p}_s]^2, \quad (13)$$

gdzie: $\mathbf{y}_s = \mathbf{y}_s(\mathbf{x}_s)$ jest odpowiedzią sieci na s -ty wektor wejściowy, $\mathbf{p}_s = [p_1^{(s)}, \dots, p_n^{(s)}]^T$ jest żądanym wyjściem dla zestawu wejściowego \mathbf{x}_s , m_p – liczba par wzorcowych w procesie uczenia

Spśród wielu dostępnych metod uczenia sieci, często stosowanym jest algorytm propagacji wstecznej (Ossowski, 1996), (Żurada i inni, 1996), używany również w tej pracy. Szczególnie efektywną metodą (dla problemu przedstawionego w pracy), okazał się algorytm Levenberg'a-Marquardt'a.

Dane uczące zostały wygenerowane, poprzez wykonanie obliczeń optymalizacyjnych dla ustalonych kombinacji poniższych parametrów, składowych wektora \mathbf{x} :

$$x_1 = m^{(L)} \in \{1.0, 1.5, 2.0, 2.5, 3.0\} \text{ kg},$$

$$x_2 = c_{E^{(k)}}^z \in \{0.3, 0.7, 1.1, 1.5, 1.9\} \times 10^4 \frac{\text{N}}{\text{m}}, \quad k = 1, \dots, 4, \quad (14)$$

$$x_3 = \alpha_{\max}^{(2)} \in \{80, 90, 100, 110, 120\} \text{ deg},$$

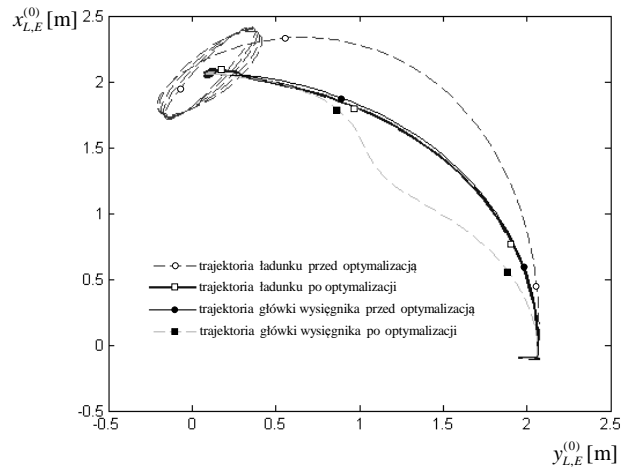
co daje 125 kombinacji.

4. WYNIKI OBLICZEŃ Z WYKORZYSTANIEM SIECI NEURONOWEJ

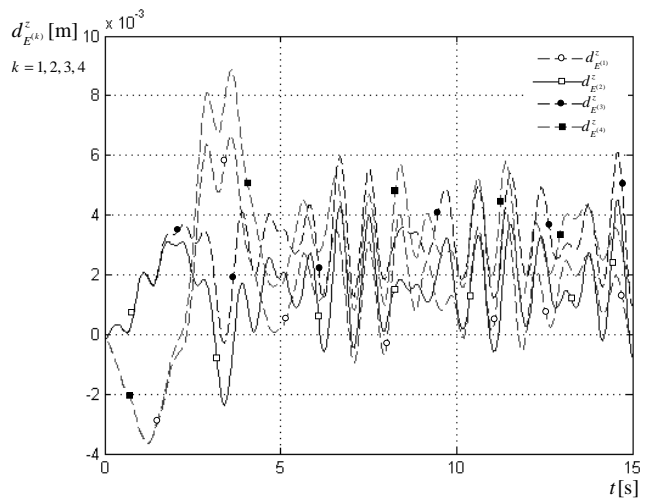
W tabeli 1 podano parametry układu przyjęte w prezentowanych dalej obliczeniach. Odpowiadają one własnościom rzeczywistego obiektu.

Na Rys. 5. przedstawiono przykładowe trajektorie ładunku oraz główki wysięgnika, dla obrotu żurawia o kąt 90deg.

Odształcenie podatnego podparcia przedstawiono na Rys. 6.

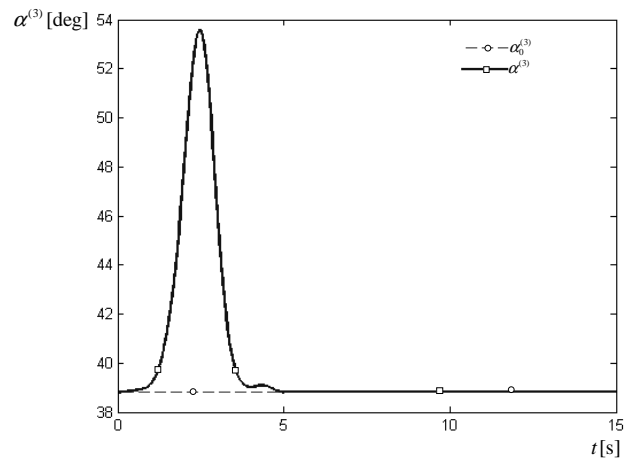
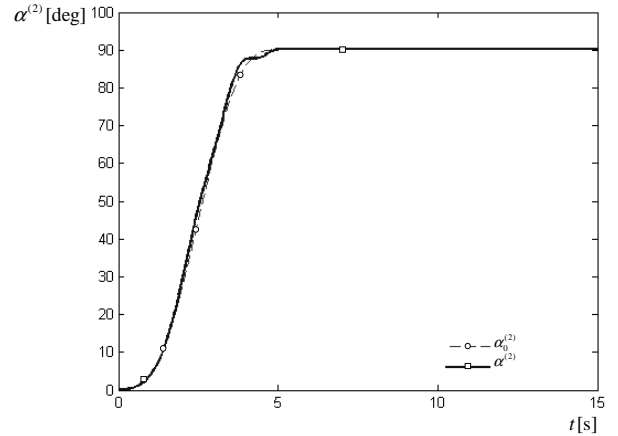
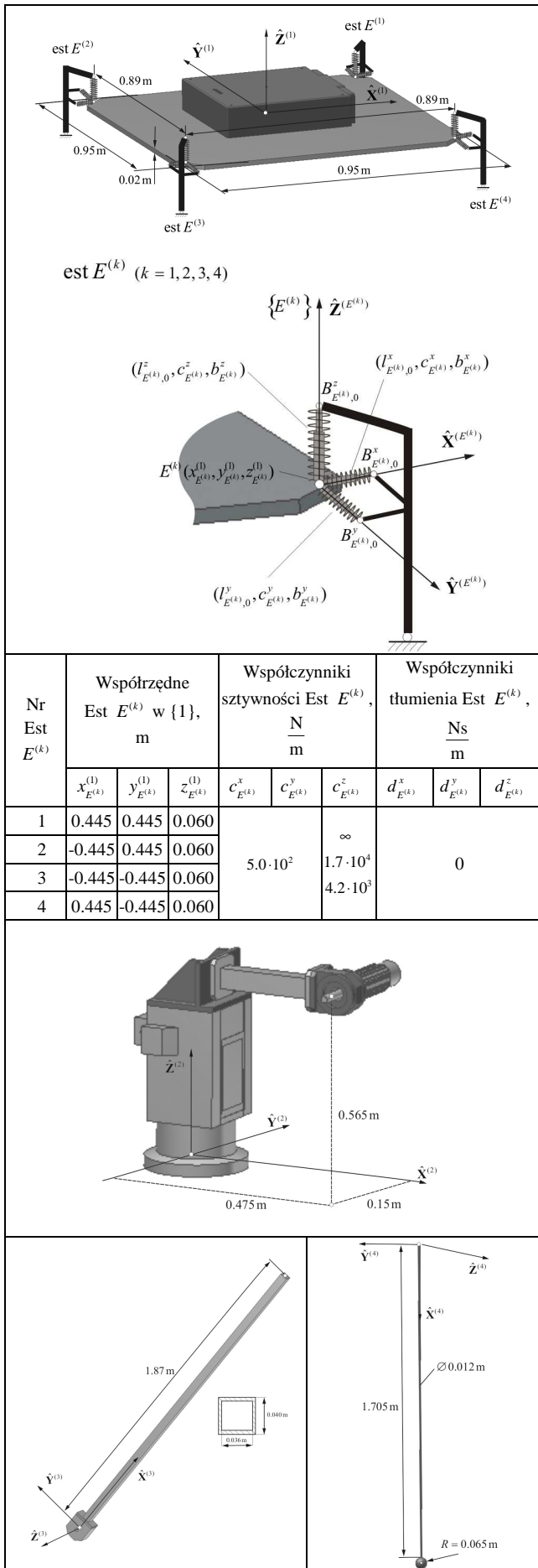


Rys. 5. Trajektoria ładunku i główki wysięgnika przed oraz po zastosowaniu sieci neuronowej



Rys. 6. Pionowe odształcenie sprężyn modelujących podpory

Tab. 1. Parametry układu



Rys. 7. Założone (początkowe) i optymalne przebiegi funkcji napędowych a) kąt obrotu, b) kąt pochylenia wysięgnika

Dla założonych domyślnie funkcji napędowych (obrotu i zmiany wysięgu), otrzymano znaczne oscylacje ładunku po zakończeniu obrotu. Po zastosowaniu sieci neuronowej do doboru funkcji napędowych, zarówno wahania ładunku po zakończeniu cyklu roboczego, jak i trajektoria w czasie obrotu, są bliskie założonym wartościom. Zmiana trajektorii główki wysięgnika po zastosowaniu optymalnych funkcji napędowych, wynika z realizacji zadanego kąta podniesienia w trakcie obrotu.

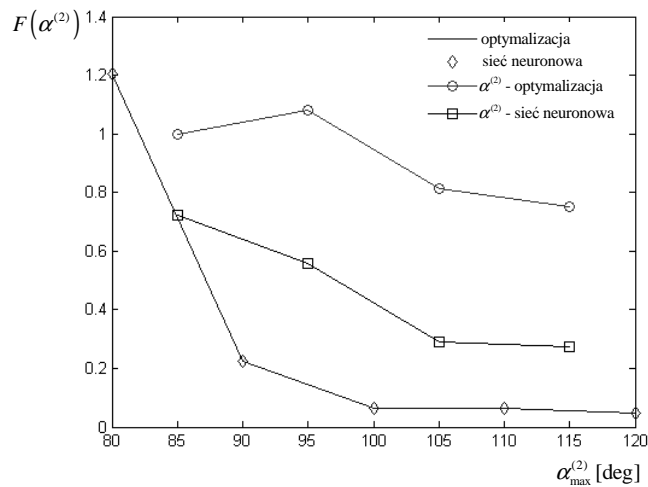
Na Rys. 7 przedstawiono przebiegi funkcji napędowych wyznaczonych w zadaniu optymalizacji.

Wykresy na Rys. 5 – Rys. 7. otrzymano dla danych:

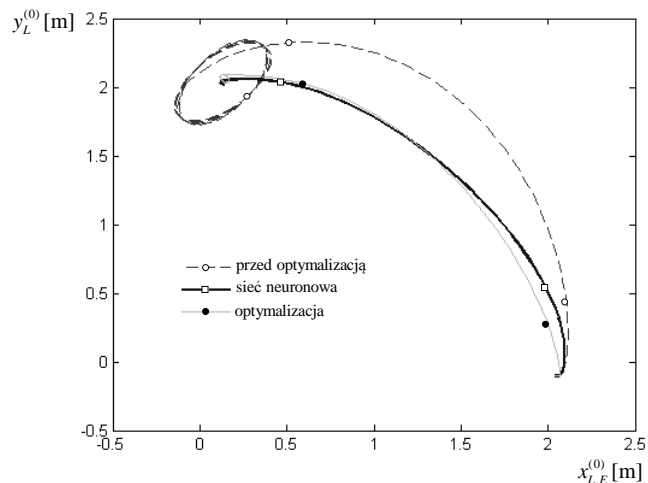
$$m^{(L)} = 1\text{kg}, c_{E^{(k)}}^z = 0.3 \cdot 10^4 \frac{N}{m}, \alpha_{\max}^{(2)} = 90\text{deg}.$$

Na Rys. 8. przedstawiono wartości funkcji celu określonej w (10) dla masy i sztywności pionowej jak powyżej. Funkcja celu wyznaczona w ostatnim kroku optymalizacji (obliczana dla wartości $\alpha_{\max}^{(2)}$ jak podano w (13)) oraz wartości funkcji celu uzyskane stosując sieć neuronową, pokrywają się. Różnica nie jest większa niż 0.5%, dla zestawu parametrów wejściowych będącymi składowymi zbioru uczącego sieci. Przyjmując jednak inne wartości kąta $\alpha_{\max}^{(2)} \in \{85,95,105,115\}\text{deg}$, oraz stosując uśrednianie funkcji uzyskanych w optymalizacji dla sąsiednich wartości kąta

$\alpha_{\max}^{(2)}$, wartości funkcji celu są wyższe, niż w przypadku stosowania funkcji napędowych uzyskanych z sieci neuronowej.



Rys. 8. Wartości funkcji celu dla parametrów wejściowych ze zbioru uczącego oraz wartości pośrednich



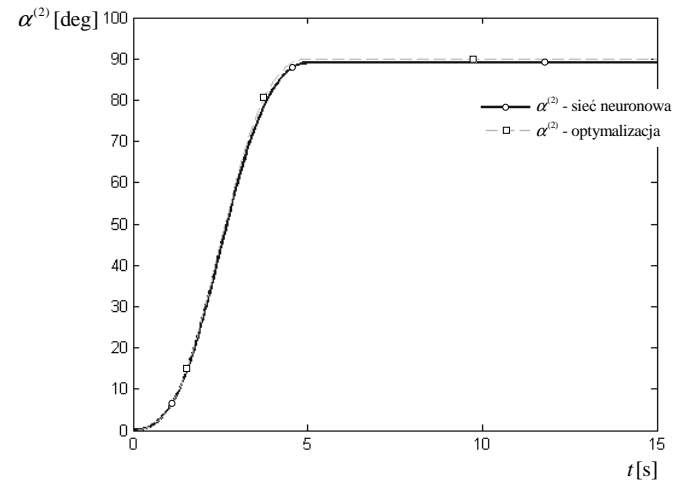
Rys. 9. Trajektorie ładunku dla parametrów wejściowych niezawartych w zbiorze uczącym oraz wynik po zastosowaniu optymalizacji

Zdolność uogólnienia sieci można wykorzystać do generowania odpowiednich funkcji napędowych w pewnym ciągłym przedziale, przyjmując dowolne wartości zmiennych wektora \mathbf{x} . Poprawnie skonstruowana i wytrenowana sieć może być stosowana do wyznaczania charakterystyk napędowych w czasie znacznie krótszym niż rzeczywisty. Trajektorja ładunku na Rys. 9 została wygenerowana dla wartości:

$$\begin{aligned} x_1 &= m^{(L)} = 2.75 \text{ kg} , \\ x_2 &= c_{E^{(k)}}^z = 1.7 \cdot 10^4 \frac{\text{N}}{\text{m}} , \\ x_3 &= \alpha_{\max}^{(2)} = 90 \text{ deg} . \end{aligned} \quad (15)$$

Wartości z (15) nie były uwzględnione w zbiorze uczącym. Zastosowanie przebiegu wyznaczonego przez sieć pozwala uzyskać znaczącą poprawę trajektorii ładunku.

Linia oznaczona jako „optymalizacja” odpowiada zoptymalizowanym funkcjom napędowym dla danych wejściowych jak w (15).

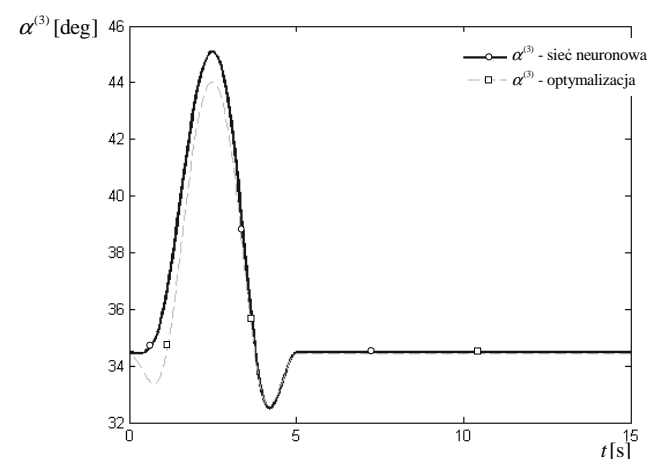


Rys. 10. Funkcja obrotu podstawy żurawia wygenerowana przez sieć oraz uzyskana w optymalizacji

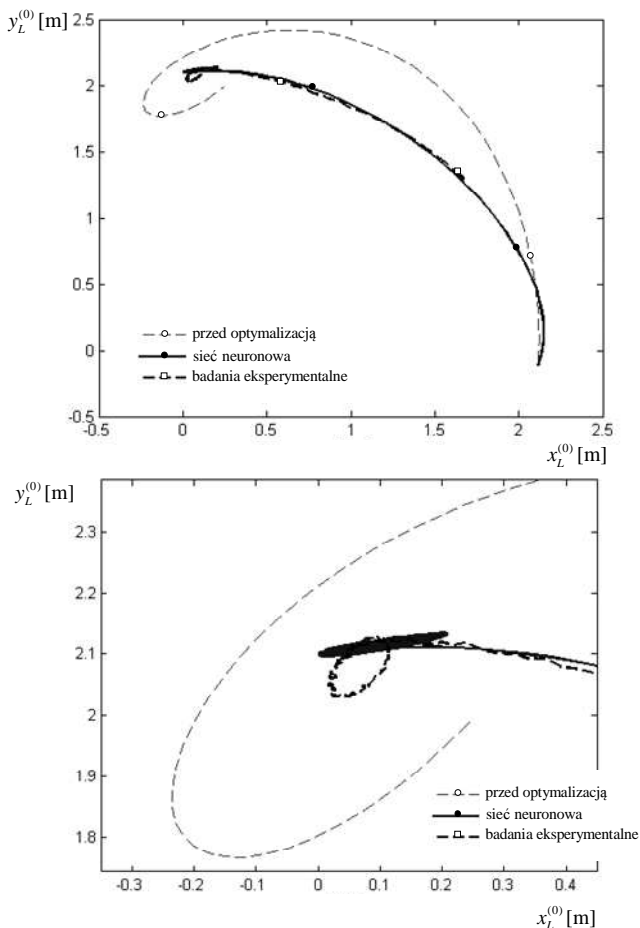
Jak przedstawiono na Rys. 10 i 11, funkcje napędowe wygenerowane przez sieć są zbliżone do optymalnych. Zatem sieć może być zastosowana do sterowania ruchem członów żurawia również przy dowolnych, innych kombinacjach parametrów wejściowych.

Wyniki obliczeń numerycznych zweryfikowano na stanowisku laboratoryjnym. Na Rys. 12 przedstawiono trajektorię ładunku. Przyjęto następujące dane:

$$\begin{aligned} x_1 &= m^{(L)} = 1.495 \text{ kg} , \\ x_2 &= c_{E^{(k)}}^z = 1.7 \cdot 10^4 \frac{\text{N}}{\text{m}} , \\ x_3 &= \alpha_{\max}^{(2)} = 90 \text{ deg} . \end{aligned} \quad (16)$$



Rys. 11. Funkcja przedstawiająca przebieg kąta pochylenia wsięgnika wygenerowana przez sieć i uzyskana w optymalizacji



Rys. 12. Weryfikacja pomiarowa trajektorii ładunku

3. PODSUMOWANIE

Głównymi zaletami stosowania sztucznych sieci neuronowych jest ich możliwość uogólnienia oraz, w przypadku zagadnień sterowania, szybkość obliczeń. W praktycznych zastosowaniach sterowania, wszędzie tam gdzie funkcja określająca napęd urządzenia musi być znana w czasie rzeczywistym, sieć neuronowa może być dobrym rozwiązaniem. Długi czas obliczeń optymalizacyjnych dla jednego zestawu danych wejściowych, praktycznie eliminuje je z zastosowania w układach rzeczywistych, także ze względu na zmienną liczbę iteracji dla każdego zestawu parametrów. Oczywistym mankamentem metody sztucznych sieci neuronowych wydaje się być znaczny nakład czasowy związany z generowaniem niezbędnych danych uczących (wykonanie szeregu obliczeń optymalizacyjnych), oraz uczeniem sieci i doбором jej architektury. Dobór odpowiedniej liczby warstw i liczby neuronów nie zawsze jest oczywisty i wymaga eksperymentów. Jednak po pozytywnym zakończeniu uczenia i weryfikacji wyników, sieć może być zapamiętana i użyta dla dowolnych, innych parametrów.

Sterowania w obecnej postaci jest układem otwartym, realizującym dobrane przebiegi funkcji napędowych. Jego zastosowanie jest możliwe przy braku istotnych zakłóceń, które wpłynęłyby na jakość otrzymywanych trajektorii.

W dalszych pracach przewiduje się połączenie sieci neuronowej z układem regulacji automatycznej opartym

na regulatorach PID, w postaci układu zamkniętego. Podejście takie, choć bardziej skomplikowane, pozwoliłoby na korygowanie błędów pozycjonowania spowodowanych zakłóceniami lub niedokładną odpowiedzią, generowaną przez sieć neuronową (np. gdy parametry wejściowe są spoza zakresu wartości danych uczących).

LITERATURA

1. **Maczyński A., Wojciech S.**, (2003), Dynamics of a mobile Crane and optimization of the slewing motion of its upper structure, *Nonlinear Dynamics*, No. 32, 259-290.
2. **Maczyński A., Wojciech S.**, (2009), Optymalizacja w planowaniu ruchu urządzeń dźwigowych, *Pomiary, Automatyka, Kontrola*, 2009 Vol.55 Nr 6, 375-379.
3. **Fałat P., Brzozowska L., Brzozowski K.**, (2005), Application of object oriented neural network to control motion of the load of a sea crane, *International Scientific Journal Computing*, Vol.4 Iss.3, 30-36.
4. **Nakazano K., Ohnishi K., Kinjo H., Yamamoto T.**, Vibration control of load for rotary crane system using neural Network with GA-based training, *Artif Life Robotics* (2008), 13, 98-101
5. **Ossowski S.**, (1996), *Sieci neuronowe w ujęciu algorytmicznym*, WNT, 1996.
6. **Urbaś A.**, (2007), Analiza dynamiczna maszyn roboczych posadowionych podatnie, *Materiały I Kongresu Mechaniki Polskiej*, Warszawa, 2007.
7. **Urbaś A., Adamiec-Wójcik I., Wojciech S.**, (2009), Dynamics of a manipulator fixed on a flexibly supported base, *Proc. of Multibody Dynamics Conference*, Warsaw, 2009.
8. **Urbaś A., Augustynek K., Janusz J., Sidzina M.**, (2008), Badania doświadczalne uproszczonego modelu żurawia chwytakowego, *Materiały XXI Konferencji Teorii Maszyn i Mechanizmów*, Bielsko-Biała, 2008, 255-262.
9. **Urbaś A., Wojciech S.**, (2008), Analiza dynamiczna żurawia chwytakowego posadowionego podatnie, *Zeszyty Naukowe Ośrodka Badawczo-Rozwojowego „BOSMAL”*, Nr 39, Bielsko-Biała, 2008.
10. **Urbaś A., Wojciech S.**, (2009), Mathematical model of the crane fixed on a flexibly supported base, *Proc. of 10th Conference – Dynamical Systems Theory and Applications*, Łódź, 2009, 389-396.
11. **Żurada J., Barski M., Jędruch W.**, (1996), *Sztuczne sieci neuronowe*, PWN, 1996.

APPLICATION OF THE NEURAL NETWORK IN CONTROL OF A FLEXIBLY SUPPORTED CRANE

Abstract: The paper presents the application of the neural network to real-time control of drive characteristics. The mathematical model of the crane, which is flexibly supported, is considered. The problem of finding an optimal drive function can be solved by mean of the classic optimisation methods. However, due to long calculation time, this approach can not be applied in the crane control tasks in practice. Utilising the advantage of neural networks (fast response, ability to generalisation), we are able to obtain the courses of drive functions in the real-time. A well trained network can be then used also for other inputs, than those used during the training task. Some example of simulations have been presented in the article.

Pracę wykonano w ramach realizacji projektu badawczego nr N N502 464934 finansowanego ze środków Ministerstwa Nauki i Szkolnictwa Wyższego.