

MODELAREA UNUI SISTEM DE COMANDA NEURONAL CU REACTIE POZITIE PENTRU UN ACTUATOR PNEUMATIC.

Tiberiu VESSELENYI¹, Ioan MOGA¹

¹Universitatea din Oradea

Keywords: artificial neural networks, control modeling, pneumatic actuator.

Abstract: Artificial neural networks ability to simulate nonlinear systems is used in different researches in order to develop automated control systems of industrial processes. In this application of neural networks, there are two important steps: system identification (development of neural process model) and development of control (definition of neural control structure). The paper presents the development of a NARMA-L2 neural controller for a pneumatic actuator using position feedback. The structure had been trained and validated, obtaining very good results. The Matlab[®], environment had been used for model design.

1. INTRODUCERE

Abilitatea rețelelor neuronale de a modela sisteme neliniare, este folosită, în diverse cercetări [4], [5], în vederea realizării unor sisteme de reglare automată a unor procese industriale.

În cazul utilizării rețelelor neuronale în conducerea unor procese, există două faze importante: identificarea sistemului (realizarea modelului neuronal al procesului) și proiectarea comenzii (realizarea sistemului de reglare neuronal);

În faza de identificare a sistemului, se proiectează un model neuronal al procesului condus iar în faza de proiectare a comenzii se folosește modelul neuronal realizat anterior pentru a antrena rețeaua ce va regla procesul.

Antrenarea modelului neuronal al procesului se realizează "offline" (sau "batch processing"), dar pentru optimizarea regulatorului neuronal sunt totuși necesare calcule și în timpul procesului.

2. COMANDA NEURONALA CU LINEARIZAREA REACTIEI.

1. Identificarea.

Mai întâi se va realiza identificarea sistemului ce urmează să fie reglat. Va trebui, deci, antrenată o rețea neuronală pentru a modela dinamica sistemului. Primul pas în procedura de identificare este acela de a găsi o structură convenabilă pentru model. Un model standard pentru reprezentarea sistemelor neliniare discrete este modelul NARMA, (Nonlinear AutoRegressive-Moving Average), dat de relația :

$$y(k+d) = N[y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-n+1)] \quad (1)$$

în care: $u(k)$ este intrarea sistemului și $y(k)$ este ieșirea. În scopul identificării se antrenează rețeaua pentru a aproxima funcția neliniară N . Procedura de antrenare este identică cu cea folosită în cazul comenzii cu model predictiv.

2. Reglarea

Daca se defineste ca scop urmarirea unei traiectorii de referinta, $y(k+d)=y_r(k+d)$, va trebui dezvoltat un regulator neliniar de forma :

$$u(k) = G[y(k), y(k-1), \dots, y(k-n+1), y_r(k+d), u(k-1), \dots, u(k-m+1)] \quad (2)$$

Pentru a genera functia G care minimizeaza eroarea patratica medie, va trebui folosita, ca algoritm de învățare, metoda dinamica de propagare înapoi a erori, care este o metoda dificil de implementat si este foarte lenta (Hagan si De Jesus, 1999). Din aceasta cauza, pentru a reprezenta sistemul, se folosesc modele aproximative (Narendra si Mukhopadhyay, 1997). Un astfel de model aproximativ, este dat de relatia :

$$y(k+d) = f[y(k), y(k-1), \dots, y(k-n+1), u(k-1), \dots, u(k-m+1)] + g[y(k), y(k-1), \dots, y(k-n+1), u(k-1), \dots, u(k-m+1)] \cdot u(k) \quad (3)$$

Acest model este într-o forma, în care intrarea $u(k)$ nu este continuta în termenul neliniar, si daca $y(k+d)=y_r(k+d)$, se poate scrie sub forma :

$$u(k) = \frac{y_r(k+d) - f[y(k), y(k-1), \dots, y(k-n+1), u(k-1), \dots, u(k-n+1)]}{g[y(k), y(k-1), \dots, y(k-n+1), u(k-1), \dots, u(k-n+1)]} \quad (4)$$

care însa ar necesita determinarea valorii intrarii $u(k)$, bazata pe iesire la acelasi pas de calcul $y(k)$, din care cauza se foloseste modelul de forma :

$$y(k+d) = f[y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-n+1)] + g[y(k), \dots, y(k-n+1), u(k), \dots, u(k-n+1)] \cdot u(k+1) \quad (5)$$

în care $d \geq 2$.

Structura rețelei neuronale ce reprezinta modelul procesului, este data în figura 1.

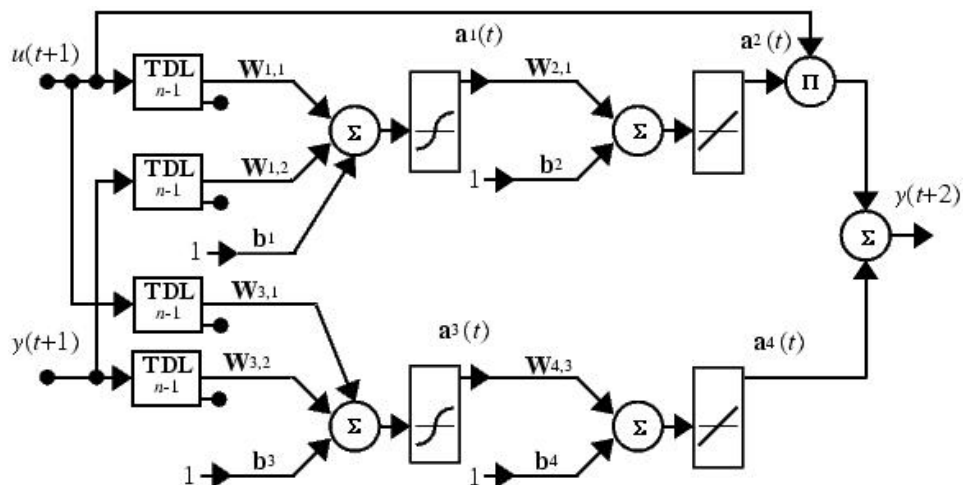


Fig.7.5. Structura modelului neuronal al procesului, construit pe baza relatiei (5)

Din relatia 5, expresia structurii regulatorului va fi :

$$u(k+1) = \frac{y_r(k+d) - f[y(k), \dots, y(k-n+1), u(k), \dots, u(k-n+1)]}{g[y(k), \dots, y(k-n+1), u(k), \dots, u(k-n+1)]} \quad (6)$$

în care $d \geq 2$.

În figura 2 este prezentata schema de principiu a sistemului de reglare, în care marimea y_r este generata de modelul neuronal ("model de referinta").

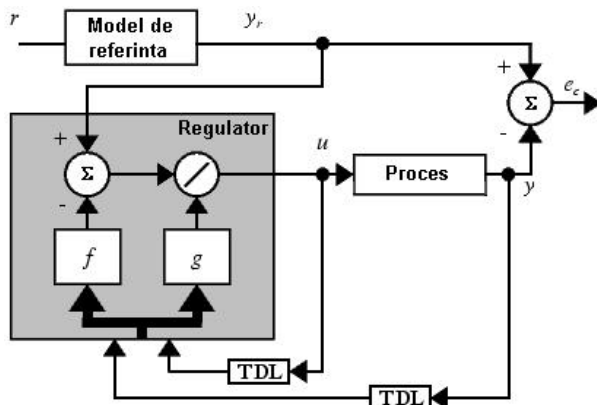


Fig.2. Schema sistemului de reglare, construit pe baza relatiei 6.

Schema structurala a rețelei neuronale ce reprezintă regulatorul prezentat în figura 2, este aratată în figura 3.

În cazul acestei metode, regulatorul execută un număr extrem de mic de calcule (calculul realizat poate fi urmărit pe diagrama din figura 3), iar modelul neuronal este antrenat offline. Din această cauză metoda poate fi folosită cu succes în cadrul comenzii roboților industriali. O aplicație directă poate fi realizarea unor regulatoare neuronale care să înlocuiască regulatoarele clasice pentru comanda unei axe de robot.

Avantajul folosirii unor regulatoare neuronale pentru roboți industriali, pe lângă avantajele date de o stabilire mai simplă a parametrilor, ar putea fi și mai mare dacă acestea ar fi folosite în sisteme în care și generatorul de traiectorii ar funcționa pe baze neuronale (sistem de calcul paralel).

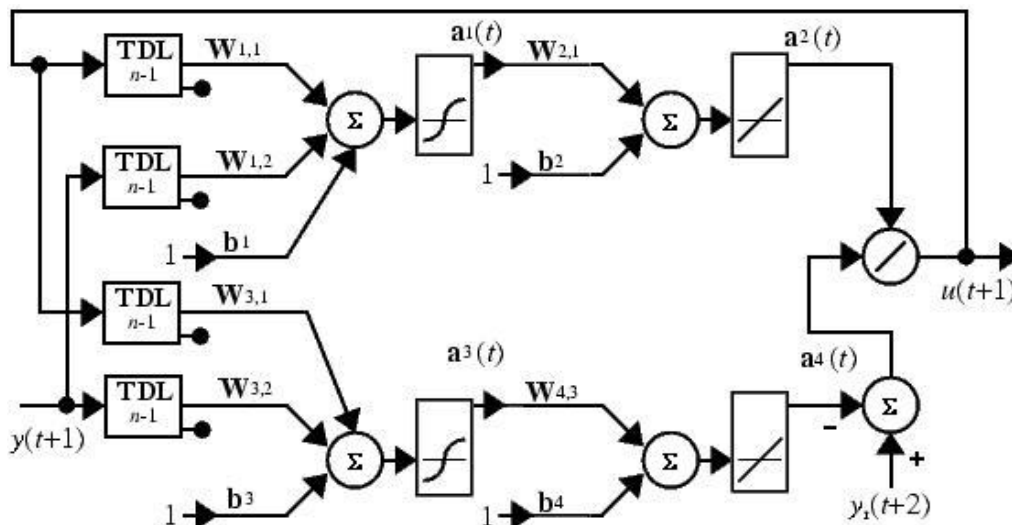


Fig.3. Structura rețelei neuronale reprezentând regulatorul, bazat pe relația (7.15)

Ca suport al realizării modelului și a simulării s-a folosit modulul SIMULINK® al mediului de programare MATLAB® R12. SIMULINK, conține blocuri predefinite pentru generarea modelului neuronal cât și o interfață grafică ce permite setarea parametrilor, antrenarea, și validarea regulatorului neuronal.

În cele ce urmează este prezentată pentru comparație și modelarea variantei clasice de reglare on-off [1].

3. MODELAREA AXEI DE TRANSLATIE PNEUMATICE CU REGULATOR NEURONAL.

Pentru sistemul cu reglare neuronală s-a folosit un singur ventil regulator de presiune și s-au utilizat două ventilele de cale. Schema astfel obținută este arată în figura 4.

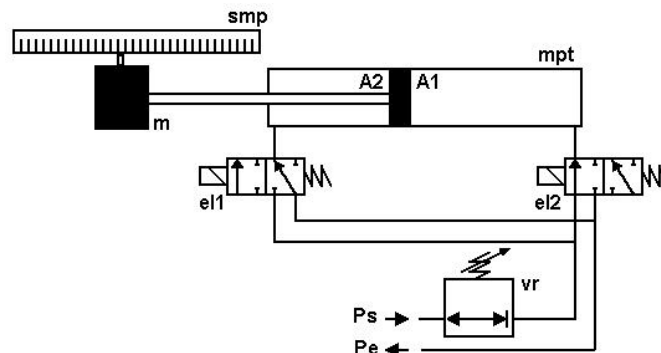


Fig.4. Schema sistemului pneumatic cu ventil regulator de presiune proportional.

Modelul sistemului este un model complex și deoarece simularea se realizează pe un calculator PC echipat cu un procesor PIII- 800MHz, ciclurile de antrenare a rețelelor neuronale durează foarte mult. Din această cauză, pentru simularea reglării neuronale s-a folosit un model simplificat al procesului, arătat în figura 5. Simplificarea constă în neglijarea factorilor ce țin de compresibilitatea aerului, și eliminarea aerului din camera opusă celei acționate.

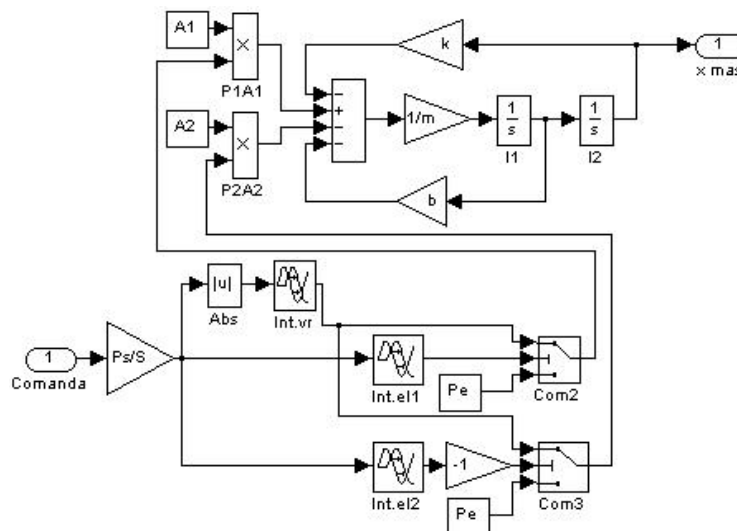


Fig.5. Schema simplificată a procesului („motor pneumatic liniar”).

Bazele teoretice ale metodei de reglare neuronală cu linearizarea reacției (NARMA) au fost prezentate în paragraful anterior.

Modelul procesului este integrat apoi în modelul sistemului cu regulator neuronal prezentat în figura 6.

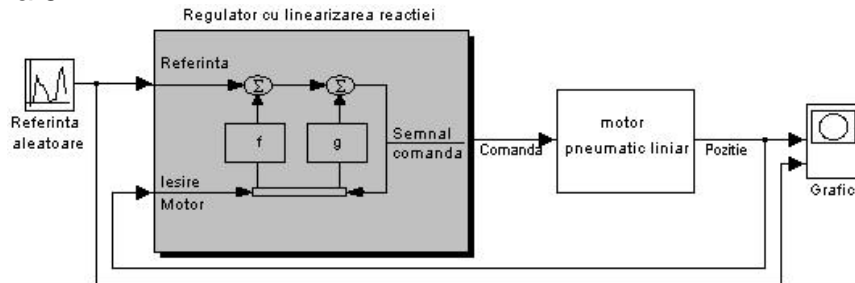


Fig.6. Schema sistemului pneumatic cu regulator neuronal.

Dupa realizarea modelelor, s-au generat seturile de antrenare, validare și testare cu ajutorul modelului procesului. Pe baza acestora a fost generat modelul neuronal al procesului, regulatorul obținându-se în mod automat pe baza acestui model, după cum s-a descris în paragraful anterior. În figura 7 sunt prezentate seturile de intrări-ieșiri generate cu ajutorul modelului procesului din figura 6, ieșirile modelului neuronal cât și erorile dintre ieșirile celor două modele, pentru seturile de date de antrenare.

În figura 8 se prezintă evoluția performanței rețelei în funcție de numărul de epoci de antrenare. Pentru antrenare s-au folosit seturi de câte 3000 de date în 300 de epoci.

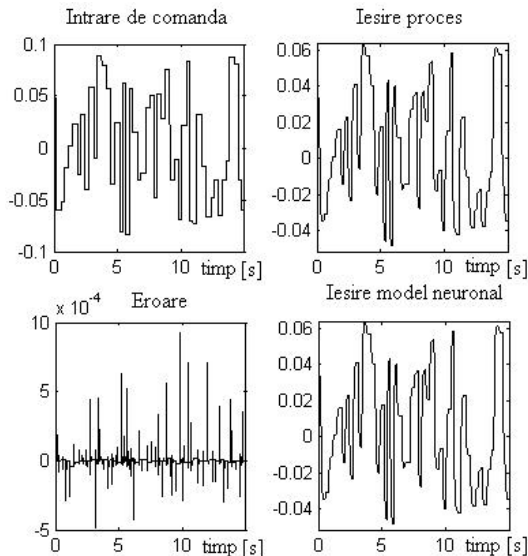


Fig.7. Reprezentarea grafică a datelor de antrenare.

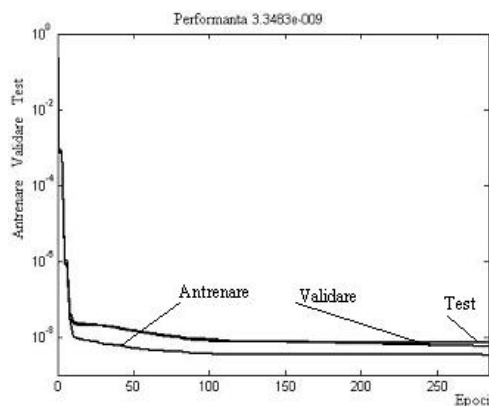


Fig.8. Evoluția performanței rețelei neuronale a modelului procesului în 300 de epoci.

4. CONCLUZII SI REZULTATE.

Dupa antrenarea rețelor neuronale, s-a efectuat simularea folosind un set aleator de comanda (sc) ce au generat semnalele de raspuns (ps) (figura 9.). Pentru a scoate în evidenta marimea erori s-a efectuat o simulare cu o referinta 0,04 (semnal tip treapta, figura 10.), pentru care raspunsul sistemului este de 0.042, rezultând o eroare de 0.002 (5%).

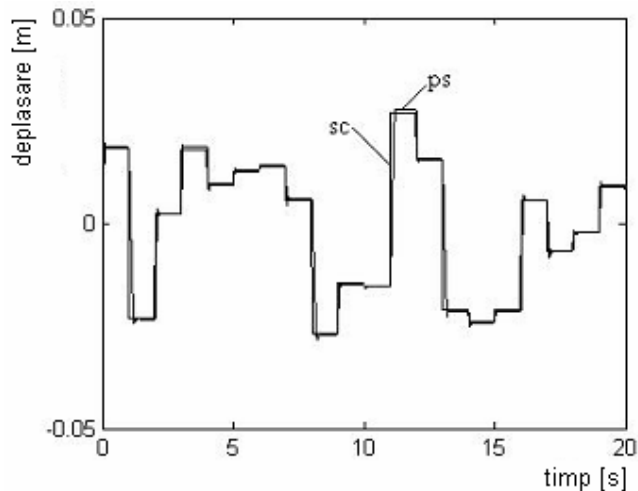


Fig.9. Referinta aleatoare (sc), raspunsul sistemului (ps).

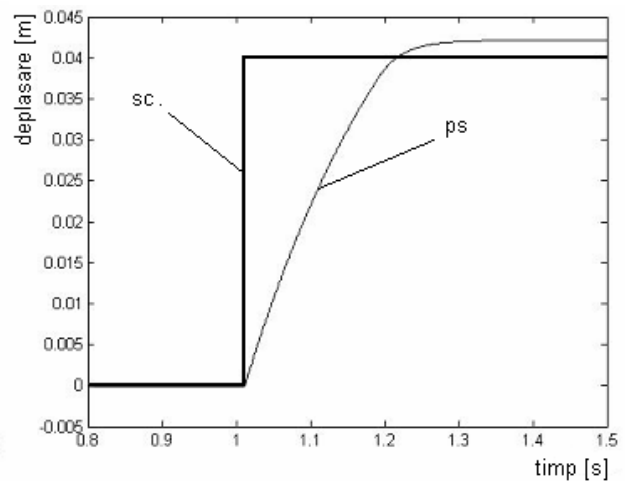


Fig.10. Raspuns la un semnal treapta, al sistemului.

BIBLIOGRAFIE

- [1] M.M., Chen, J.A., Fairwater, S.A., Green, EDUMECH. Mechatronic Instructional Systems. Case Study: Pneumatics Systems, Production of Shandor Motion Systems, Inc, 1999.
- [2] Harbick K., Sukhatme S., Speed Control of a Pneumatic Monopod using a Neural Network, www.harbick-ann.com (2002).
- [3] Vesselenyi T. – Teza de doctorat, Universitate “Politehnica” Timisoara, 2005.
- [4] Wenmei H., Yong Y., Yali T., Adaptive neuron control based on predictive model in pneumatic servo system. INTERNET (2002).
- [5] Zhang J., Knoll A., Schmidt R., A neuro-fuzzy control model for fine-positioning of manipulators, Robotics and Autonomous Systems 32 (2000) 101–113.