

IDEGHÁLÓK ROBOTOK INVERZ KINEMATIKAI VIZSGÁLATÁHOZ: ÁTTEKINTÉS ÉS ALKALMAZÁS

NEURAL NETWORKS FOR LEARNING INVERSE KINEMATICS OF ROBOTS: REVIEW AND APPLICATION

Hazim Nasir Ghafil*, Károly Jármái**

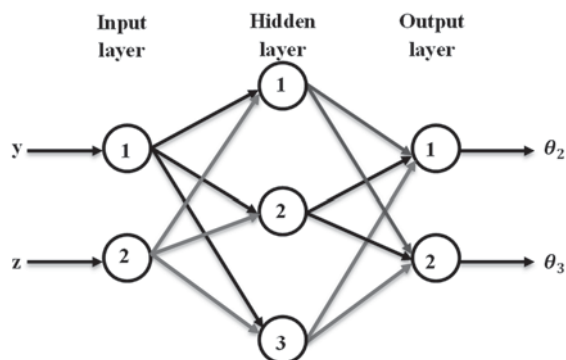
ABSTRACT

Deep learning for Inverse Kinematic problem of robot manipulators was investigated by many works considering the planar case. In this paper, the performance of the three learning algorithms; Levenberg-Marquardt algorithm, Bayesian Regularization algorithm, and Scaled Conjugate Gradient algorithm is studied will learning single hidden layer network how to solve the Inverse Kinematic problem of a three degree of freedom robot manipulator.

1. BEVEZETÉS

Az Inverz Kinematika (IK) megoldás a robot vég-effektorának speciális konfigurációjára, alapvető fontosságú funkció a vezérlőrendszerben. Az olyan optimalizációs algoritmusok, mint a részecske-csoport optimalizálás és a mesterséges méhcsalád [1] garantálhatják a probléma optimális megoldását a célfüggvény differenciálhatósága miatt. Valós ipari alkalmazásokban azonban nem használhatjuk az optimalizációs algoritmusokat, mivel ezek bizonyos időt vesznek igénybe. Az ANN mesterséges ideghálók akkor is tökéletesen alkalmazhatók valós ipari alkalmazásokra, ha megoldásaik közelítőek és nem olyan pontosak, mint az optimalizációs algoritmusok megoldásai. A pontosság és a gyors közelítések között kell választanunk, különösen az érzékelőkkel felszerelt robotok esetében, amelyek gyors megoldásokat igényelnek, hogy a hirtelen eseményekre reagálni tudjanak. Ez ésszerű indok arra, hogy megfontoljuk az ideghálók alkalmazását egy robot inverz kinematikai problémájának megoldásánál. A többrétegű előremenő ideghálót megtanították a három szabadságfokozatú manipulátor csomóponti változóinak a megadására, az end-effektor pozícióra tekintettel [2], a tanulási folyamatot az adott end-effektor pozíciók és a hozzájuk tartozó közös szögek felhasználásával hajtják végre. A többrétegű előremenő idegháló hatékonyabbnak bizonyult, mint az egyrétegű, a funkcionális kapcsolat általánosítására az inverz kinematikai problémára. Betanítás után a hálózat képes

volt visszaadni a közös változókat tetszőleges derékszögű koordináta pozícióhoz. Az 1. ábra bemutatja az IK probléma megoldására használt idegháló architektúráját azáltal, hogy a derékszögű koordináta helyzetet bemenetekként, a csomóponti szögeket pedig kimenetekként jelöli. Az ANN-t a visszaterjesztési algoritmussal alkalmazták a 6R robotmanipulátor inverz kinematikai megoldására a MATLAB Neural Networks Toolbox segítségével [3]. A hálónak 12 bemenete volt a homogén transzformációs mátrix elemeit képviselő és hat kimenete, amelyek hat csomóponti szög képviseltek. E tanulmány szerint az ANN inverz kinematika alkalmazásának hátránya a betanítási folyamathoz szükséges nagy adathalmaz.



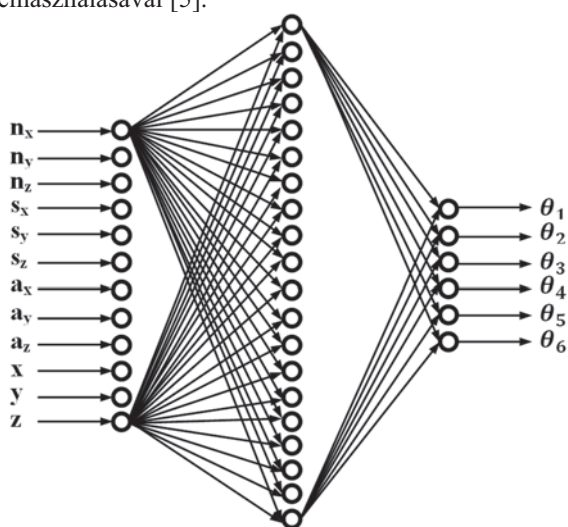
1. ábra: Az ANN [2] által használt architektúrája

Más szavakkal: a képzéshez szükséges nagy mennyiségű adat pontosabb ideghálózathoz vezet. A 2. ábra bemutatja a ANN topológiáját ebben a tanulmányban. 4000 adatot használtak a háló betanítására, amely egyetlen rejtett rétegben 20 neuront tartalmazott. Az ANN rendszermodelleket épít az előre megadott adatok példáinak betanulásával; ez rugalmassá teszi és ezért széles körben használják a robotikában és az automatizálásban. Az ANN egy lineáris vagy nemlineáris függvény, amely megtanulhatja a lineáris és nemlineáris problémák megoldását és osztályozását. Az utóbbi években a különféle kutatások új, megbízható algoritmusokra összpontosítottak a véges adatkészletek

* doktorandusz, Miskolci Egyetem Energetikai és Vegyipari Gépészeti Intézet

** professzor, Miskolci Egyetem Energetikai és Vegyipari Gépészeti Intézet

elemzésére anélkül, hogy növelnék az algoritmusok számítási intenzitását. A robotmanipulátor inverz kinematikai problémájának megoldásakor számos korlátozást figyelembe kell venni az alkalmazott idegháló tanulási folyamata során. Az inverz kinematika egy komplex probléma, ahol a bonyolultság a robot geometriájából és a robot derékszögű koordináta terét és csomóponti terét megadó trigonometriai egyenletekből származik. A kinematikai egyenletek szintén csatolva vannak, és fennáll a szingularitás veszélye, ezek a korlátok növelik a tanulási folyamat nehézségeit. Három csomópontú sík manipulátort használtak próbaként az inverz kinematikai probléma [4] számára egy neurális hálózat betanulásánál; hogyan lehet létrehozni egy függvényt erre a problémára. Nem szokványos és nem testreszabott topológiákat alkalmaztak az IK probléma betanítására a hálózati inverzió fogalmának felhasználásával [5].



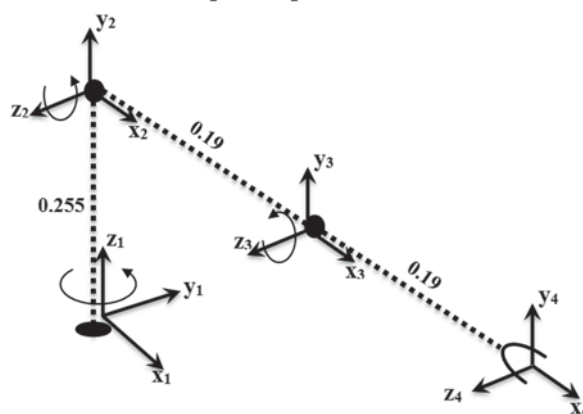
2. ábra: Az IK megtanulásához használt ANN topológia

A Kohonen és a hiba visszatérítési hálózatok a leggyakoribb ideghálók, amelyeket a robotmanipulátorok inverz kinematikájának megoldására alkalmaznak. A visszaterjesztési algoritmus, amely a tanulási mechanizmus egyik eleme, hosszú időt vesz igénybe, hogy megvalósítható leképezést találjanak a derékszög koordináta térről a robot csomópontok helyére vonatkozólag. Az idegháló az összes betanulási minta választ a kimeneti rétegre továbbítja. A kívánt kimenet és a kiszámított kimenet közötti különbség azt a hibát jelöli, amelyet iteratívan a visszacsatolásnál a súlyok beállításánál figyelembe vesznek. Ha a betanulási adatok halmaza túl nagy, a betanulás túl sok időt vesz igénybe, és ez a gyakorlatban nem járható. Alhálózatokat is javasoltak [6], hogy egynél több megoldást találjanak az end-effektor adott derékszögű koordináta helyzetére. Ebben a munkában megvizsgálták a Levenberg-Marquardt tanulási algoritmus, a Bayes-féle Szabályozási algoritmus

(Bayesian Regularization) és a Skálázott Konjugált Gradiens tanulási algoritmus (Scaled Conjugate Gradient) használatának hatását egy bemeneti réteg, egy rejtett réteg és egy kimeneti réteg esetén, hogyan lehet megtalálni az RRR robot kar fordított helyzetét. A neurális hálózat topológiáját részletesen tárgyalták, valamint a fent említett tanulási algoritmusok előnyeit és hátrányait.

2. KÍSÉRLETI MUNKA

Ebben a munkában egy 3D térben történő mozgásképeségű RRR robotmanipulátort vizsgáltak, hogy meghatározzák a rétegek számának hatását egy rögzített nagyságú neuronszám esetén. A 3. ábra a keretmeghatározást és a robot specifikációit mutatja, az 1. táblázat a csomópontok paramétereit.



3. ábra. RRR robotmanipulátor

1. táblázat: A Lab-Volt 5150 manipulátor térbeli paramétereit

Cso mópont ID	Keret	ϕ	α	a	d	θ	Határ
1	$0_1X_1Y_1Z_1-$ $0_2X_2Y_2Z_2$	0	90	0	d_1	θ_1	-185,153
2	$0_2X_2Y_2Z_2-$ $0_3X_3Y_3Z_3$	0	0	a_2	0	θ_2	-32,149
3	$0_3X_3Y_3Z_3-$ $0_4X_4Y_4Z_4$	0	0	a_3	0	θ_3	-147,51

2.1.1 A csigahajtóművek fő veszteségforrásai és azok becslése alapján

Ismert, hogy a manipulátorban lévő három lánc homogén transzformációs mátrixai becsülhetők meg, az egyes összeköttetések négy paraméterének behelyettesítésével a Denavit-Hartenberg mátrixba (Spong et al. 2006). A kinetikai egyenletek halmaza a következő lesz:

3. IDEGHÁLÓ ARCHITEKTÚRA

$$H_2^1 = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$H_3^2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2 \cdot \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2 \cdot \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$H_4^3 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a_3 \cdot \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & a_3 \cdot \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$H_4^1 = H_2^1 \cdot H_3^2 \cdot H_4^3 = \begin{bmatrix} n_x & s_x & a_x & x \\ n_y & s_y & a_y & y \\ n_z & s_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

ahol

$$n_x = \cos(\theta_1) \cos(\theta_{2+3}) \quad (5)$$

$$n_y = \sin(\theta_1) \cos(\theta_{2+3}) \quad (6)$$

$$n_z = \sin(\theta_{2+3}) \quad (7)$$

$$s_x = -\cos(\theta_1) \sin(\theta_{2+3}) \quad (8)$$

$$s_y = -\sin(\theta_1) \sin(\theta_{2+3}) \quad (9)$$

$$s_z = \cos(\theta_{2+3}) \quad (10)$$

$$a_x = \sin(\theta_1) \quad (11)$$

$$a_y = -\cos(\theta_1) \quad (12)$$

$$a_z = 0 \quad (13)$$

$$x = \cos(\theta_1)(a_2 \cos(\theta_2) + a_3 \cos(\theta_{2+3})) \quad (14)$$

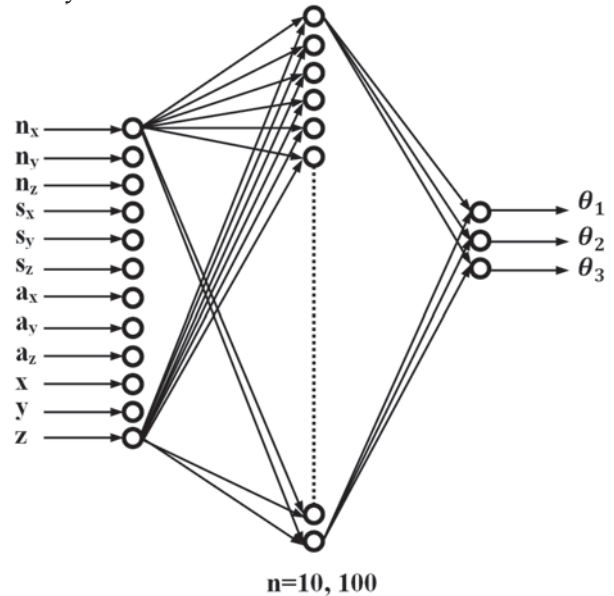
$$y = \sin(\theta_1)(a_2 \cos(\theta_2) + a_3 \cos(\theta_{2+3})) \quad (15)$$

$$z = d_1 + a_2 \sin(\theta_2) + a_3 \sin(\theta_{2+3}) \quad (16)$$

Az (5) - (13) egyenletek az end-effektor tájolását mutatják az alapkerethez viszonyítva, míg a (14), (15) és a (16) egyenletek az end-effektor derékszögű helyzetét ábrázolják. Ezek a csomóponti térről a derékszögű koordináta rendszerbe transzformálják az end-effektor helyzetét, adott csomóponti szögek esetén. Ezekből az egyenletekből egy 2000 bemeneti/kimeneti képzési adatkészlet generálódik, hogy megtanulja az idegháló a javasolt topológiáit, hogyan lehet az inverz helyzetét meghatározni a tanult adatok alapján.

A 4. ábra bemutatja a tanulmányban alkalmazott mesterséges idegháló topológiáját, amely 12 bemenettel, egy rejtett réteggel és három kimenettel rendelkezik. A bemeneti réteg a homogén transzformációs mátrix elemei, amelyek az end-effektort az alapkerethez kapcsolják, és ezeket az (5-16) egyenletek magyarázzák. Ebben a tanulmányban egy rejtett réteget használtunk $n=10$ és $n=100$ neuronsszámmal. A kimeneti réteg három neuronból áll, amelyek képviselik a robotkar három csuklóját.

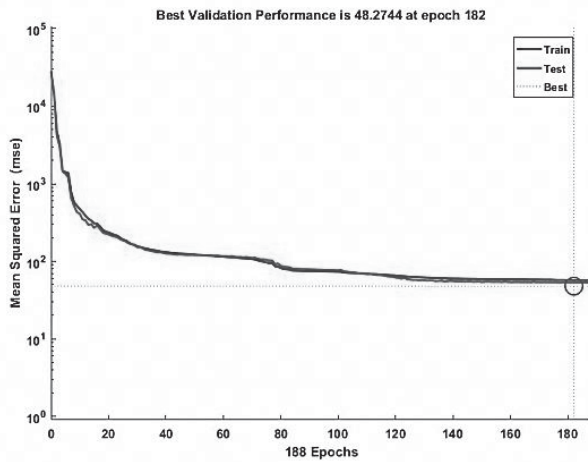
Három tanulási algoritmus áll rendelkezésre a MATLAB neurális hálózati eszközkészletében, amelyeket alkalmaztunk és összehasonlítottuk a kapott eredményeket $n=10$ és $n=100$ esetén. A javasolt neurális hálózatot három tanulási algoritmus segítségével lett betanítva; Levenberg-Marquardt (LM) algoritmus, a Bayes-féle Szabályozási algoritmus (BR) és Skálázott Konjugált Gradiens algoritmus (SCG), ahol ezen algoritmusok mindegyikének megvan a maga előnye és hátránya.



4. ábra: ANN architektúra az RRR robot manipulátorhoz

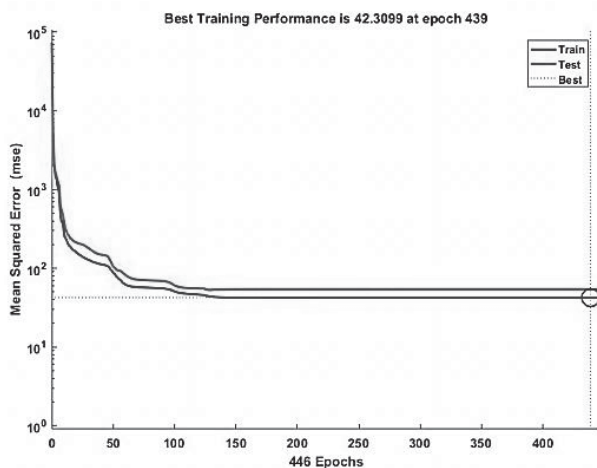
Az 5–7. ábra a Levenberg-Marquardt, a Bayes-féle szabályozó és a skálázott konjugált gradiens teljesítményét mutatja a tanulási folyamatban, ha a neuronok száma 10-es. A Levenberg-Marquardt algoritmusnak kevesebb időre van szüksége, mint a Bayes-féle Szabályozási algoritmusnak és több időre és több memóriára van szüksége, mint a Skálázott Konjugált Gradiens algoritmusnak. Az LM pontosabb, mint az SCG, de kevésbé pontos, mint a BR. Ugyanazon összehasonlítás $n=100$ esetén történő megismétlésével (lásd a 8. és 10. ábrát) könnyű felismerni, hogy a teszt során a négyzetes eltérés jelentős csökkenését

összehasonlítva azzal az esettel, amikor $n=10$. A 2000 tételből álló adatkészletet használták az algoritmusok a tanulási folyamathoz, ahol a felhasznált elemek 70% -a tanulásra, míg 30% -a tesztelésre szolgált.

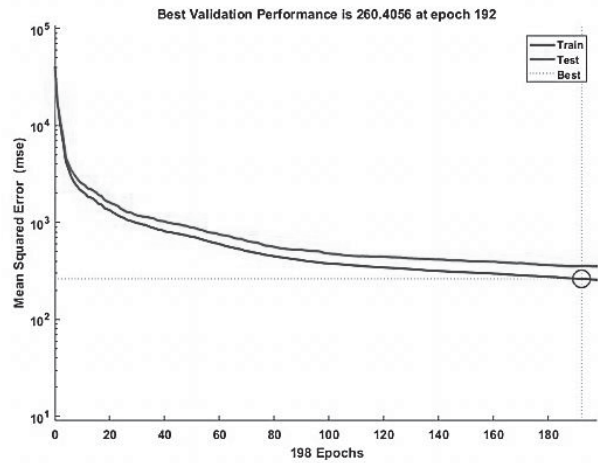


5. ábra. A Levenberg-Marquardt algoritmus átlagos négyzetes hibája, $n=10$

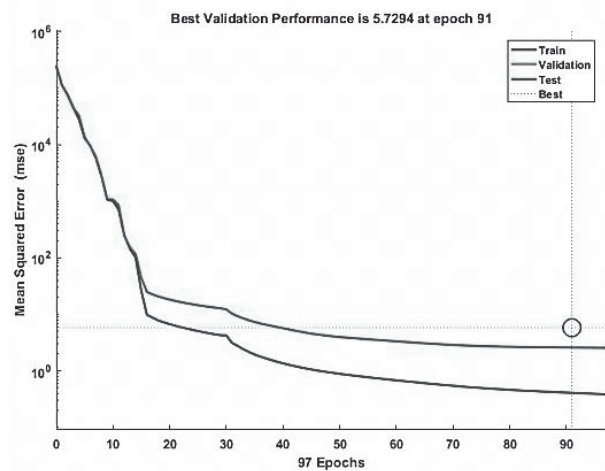
Axiomatikusan, a növekvő idegsejtek száma a rejtett rétegben a legjobb eredményt adhatja kevesebb hibával, de itt ki kell értékelnünk a három tanulási algoritmus teljesítményét. A 2. táblázat bemutatja az eltelt időt a hálózat tanulására, hogyan oldható meg a robot inverz kinematikája, ahol a BR-nek több időre van szüksége, mint az LM-re, míg az SCG nagyon rövid időt igényelt. Visszatérve az 5. ábrára, a 10. ábrára, a neuronok számának növekedése 100-hoz az LM és BR tanulási algoritmusok esetében az átlagos négyzeteshiba (MSE) drámai csökkenéséhez vezetett, ám az SCG algoritmus még mindig nem tudta megtanulni az inverz kinematika hálózati megoldását. A 3. táblázat bemutatja a javasolt problémára a három említett tanulási algoritmus felhasználásával indukált MSE-t.



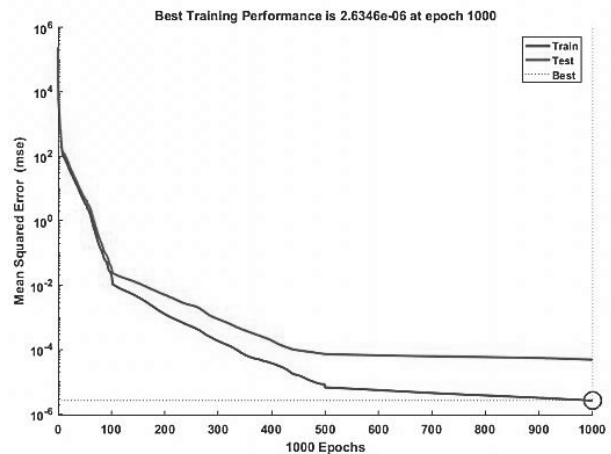
6. ábra. Átlagos négyzetes hiba (MSE) a Bayes-féle Szabályozási algoritmusnál $n=10$ esetén



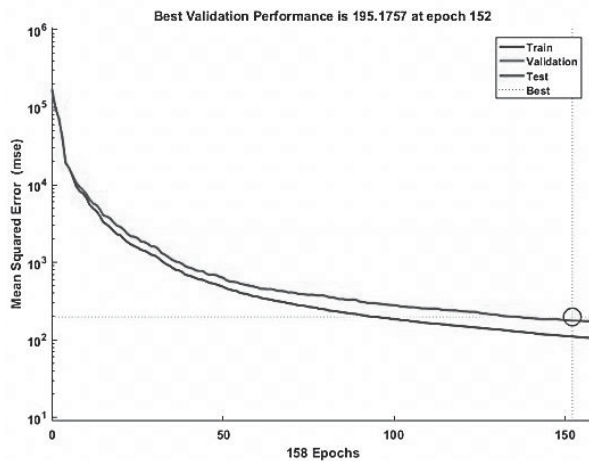
7. ábra. Átlagos négyzetes hiba (MSE) a Skálázott Konjugált Gradiens algoritmusnál $n=10$ esetén



8. ábra. Átlagos négyzetes hiba (MSE) a Levenberg-Marquardt algoritmusnál $n=100$ esetén



9. ábra. Átlagos négyzetes hiba (MSE) a Bayes-féle Szabályozási algoritmusnál $n=100$ esetén



10. ábra. Átlagos négyzetes hiba (MSE) a Skálázott Konjugált Gradiens algoritmusnál $n=100$ esetén

2. táblázat. Időszükséglet a Levenberg-Marquardt, a Bayes-féle Szabályozási és a Skálázott Konjugált Gradiens algoritmusnál

tanuló algoritmus	idő (sec)	
	$n=10$	$n=100$
Levenberg-Marquardt	4	171
Bayes-féle Szabályozási	8	1840
Skálázott Konjugált Gradiens	0.01	1

3. táblázat. Átlagos négyzetes hiba (MSE) a Levenberg-Marquardt, a Bayes-féle Szabályozási és a Skálázott Konjugált Gradiens algoritmusnál

tanuló algoritmus	Átlagos négyzetes hiba (MSE)	
	$n=10$	$n=100$
Levenberg-Marquardt	48.2744	5.7294
Bayes-féle Szabályozási	42.3099	2.6346E-06
Skálázott Konjugált Gradiens	260.4056	195.1757

4. KÖVETKEZTETÉSEK

A 3. ábrán bemutatott manipulátor egyszerű architektúrájához 2000 egyedi megoldásra van szükségünk ahhoz, hogy ésszerű összehasonlítást kapjunk ebben a munkában. Más szavakkal, hatalmas mennyiségű képzési adatra van szükségünk, hogy megtanuljuk az inverz problémát olyan komplex architektúrákban, mint a 7 DOF. Ezenkívül a hiba megnövekszik, miközben a robot szabadságfoka növekszik és további idő szükséges az elfogadható eredmények elérése érdekében. Így elmondhatjuk, hogy az idegháló használata az inverz kinematikára a robot DOF-jére korlátozódik. A másik oldalról a Bayes-féle Szabályozási algoritmus pontos eredményeket ad, ám ennek a módszernek a használata valós idejű alkalmazásoknál kritikus, mivel viszonylag hosszú időbe telik, hogy a megvalósítható megoldást elérje. A valósidejű alkalmazáshoz a Levenberg-Marquardt módszer jó választás lehet, különösen, ha a közelített megoldások megfelelőek (kevésbé pontosak, de gyorsabbak, mint a Bayes-féle Szabályozási). Noha a Skálázott Konjugált Gradiens módszernek nem sikerült ehhez az elfogadható megoldáshoz konvergálni ezen az

egyszerű robotnál, semmilyen típusú robotnál nem használható.

5. ÖSSZEFOGLALÁS

Az idegháló hatékony eszköz a nemlineáris problémákhoz, például az inverz kinematikai problémákhoz. Az idegháló a bemenetek és a kimenetek közötti funkciót képviseli, amelyet az idegsejtek közötti kapcsolatok elosztott súlyai képviselnek. Ebben a cikkben három különféle tanulási algoritmus, nevezetesen Levenberg-Marquardt algoritmus, Bayes-féle Szabályozási algoritmus és Skálázott Konjugált Gradiens algoritmus használatát vizsgáltuk meg, hogy megismerjünk egy neurális hálózatot, hogyan lehet megoldani egy három csuklós robotmanipulátor inverz kinematikáját. A hálózat topológiája egy bemeneti rétegből áll, tizenkét neuronnal, egy rejtett rétegből, amelynek neuron száma = 10, 100, és egy kimeneti rétegből áll, három neuronnal. A bemeneti rétegben lévő 12 neuron a homogén transzformációs mátrix adott elemeit képviseli, míg a kimeneti rétegben található három neuron a robot kívánt csukló szögeit adja meg. A tanulmány során azt találtuk, hogy a Skálázott

Konjugált Gradiens tanulási algoritmus nem képes megtanulni a hálózatot a javasolt problémára még akkor sem, ha a rejtett rétegben nagy számú neuron található. A Bayes-féle Szabályozási tanulási algoritmus a legjobb eredményeket hozta, de az eltelt idő nagyobb, mint ami a Levenberg-Marquardt algoritmus számításához szükséges. Célunk más szerkezeteknél is megvizsgálni az egyes algoritmusok hatékonyságát, hogy általánosabb következtetéseket tudjunk levonni.

6. KÖSZÖNETNYILVÁNÍTÁS

"A cikkben/előadásban/tanulmányban ismertetett kutató munka az EFOP-3.6.1-16-2016-00011 jelű „Fiatalodó és Megújuló Egyetem – Innovatív Tudásváros – a Miskolci Egyetem intelligens szakosodást szolgáló intézményi fejlesztése” projekt részeként – a Széchenyi 2020 keretében – az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg"

7. IRODALOM

1. Ghafil H. N., Jármái K.: Comparative study of particle swarm optimization and artificial bee colony algorithms, MultiScience-XXXI. microCAD International Multidisciplinary Scientific Conference, 2018.09.05-2018.09.06. Miskolc: Miskolci Egyetem, 2018. Proceedings on CD. Paper D1_Hazim_Nasir_Ghafil.pdf. 6 p. ISBN 978-963-358-162-9, DOI: 10.26649/musci.2018.030
2. Choi BB, Lawrence C (1992) Inverse kinematics problem in robotics using neural networks. NASA Technical Memorandum 105869, 27 p.
3. Bingul Z, Ertunc H, Oysu C (2005) Applying neural network to inverse kinematic problem for 6R robot manipulator with offset wrist. In: Adaptive and Natural Computing Algorithms. Springer, pp 112-115.
4. Duka A-V (2014) Neural network based inverse kinematics solution for trajectory tracking of a robotic arm. Procedia Technology 12:20-27.
5. Tejomurtula S, Kak S (1999) Inverse kinematics in robotics using neural networks. Information Sciences 116 (2-4):147-164.
6. Lu B-L, Ito K Regularization of inverse kinematics for redundant manipulators using neural network inversions. In: Proceedings of ICNN'95-International Conference on Neural Networks, 1995. IEEE, pp 2726-2731.
7. Spong MW, Hutchinson S, Vidyasagar M (2006) Robot modeling and control. Emerald Group Publishing Limited, 303 p. ISBN-13: 978-0471649908.