

REPUBLIKA E SHQIPËRISË
UNIVERSITETI POLITEKNIK I TIRANËS
FAKULTETI I INXHINIERISË ELEKTRIKE

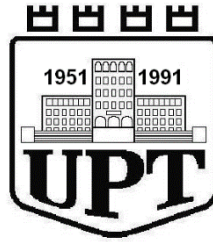
DISERTACION
PËR MBROJTJEN E GRADËS SHKENCORE
“DOKTOR”

*Vlerësim teorik dhe arritjet në rrjetet neurale. Aplikime në
teknikat inxhinierike dhe veçanërisht në automatizimin
industrial*

Përgatitur nga:
Msc. Ing. Ermira BUZI

Udhëheqës Shkencor
Prof.Dr. Petrika MARANGO

Tiranë, 2016



REPUBLIKA E SHQIPËRISË
UNIVERSITETI POLITEKNIK I TIRANËS
FAKULTETI I INXHINIERISË ELEKTRIKE

DISERTACION

i

paraqitur nga

Msc. Ing. Ermira BUZI

për marrjen e gradës shkencore

“DOKTOR”

në Inxhinieri Elektrike, Automatizim i Industrisë

*Vlerësim teorik dhe arritjet në rrjetet neurale. Aplikime në
teknikat inxhinierike dhe veçanërisht në automatizimin
industrial*

Mbrohet më datë 19.12.2016 para Jurisë të miratuar nga Rektori i UPT me
shkresën Nr.1160/1 datë 10.10. 2016.

- | | |
|---------------------------------|------------------|
| 1. Prof. Dr. Aida SPAHIU | Kryetar, Oponent |
| 2. Prof. Dr. Aleksandër XHUVANI | Anëtar |
| 3. Prof. Dr. Orion ZAVALANI | Anëtar |
| 4. Prof. Dr. Zenun MULAJ | Anëtar, Oponent |
| 5. Prof. Asoc. Astrit BARDHI | Anëtar |

Mirënjohje

Gjatë periudhës të studimeve të doktoraturës, kam patur fatin të marr ndihmën, këshillën dhe inkurajimin nga miq, dashamirës e institucione.

Falenderim dhe mirënjohje për udhëheqësin, Prof. Dr. Petrika Marango. Një mentor bujar, ndihma dhe këshillat e të cilit kanë qenë të çmuara dhe “shkallë” që më kanë ndihmuar të ngjitem drejt përfundimit të këtij punimi.

Falemnderim të përzemërt për Profesorët e nderuar brenda dhe jashtë fakultetit si dhe gjithë stafin e fakultetit për ndihmën, komentet, e vlerësimin gjatë punës për dizertacionin dhe fazave të doktoraturës.

Nuk mund të lë pa përmendur të nderuarin Prof. Dr. Kujtim Veisllari, që pa tek unë një studente që mund të arrinte më tepër dhe ushqeui shkëndijën e vijimit të formimit akademik. Brenda katedrës të Automatizimit të Industrisë të Departamentit të Elektronikës, ka qenë fat për mua të mësoja e punoja pranë profesorëve si Ai, Prof. Dr. Gëzim Karapici, dhe Prof. Dr. Petrika Marango.

Dhe gjithë arritjet nuk mund t'i kisha pa mbështetjen dhe dashurinë pa kufi të familjes time: një kushtim për ëndrrën për dije dhe fjalën e fundit të tim Eti, forcës për të përballur vështirësitë të Mëmës, muzës që gjej tek dashuria e motrës e vëllait.

Parathënie

Njeriu, që në gjenezën e vet, ka vëzhguar natyrën dhe gjithçka që e rrethon, dhe njohuritë e përftuara nga vëzhgimi i ka vënë në funksion të krijimit të mjeteve të ndryshme. Për shembull, skicimet e Leonardo da Vinçit për makinat fluturuese, të motivuara nga zogjtë, mund të merren si hapat e parë të një rrugëtimi të gjatë për të arritur në aeroplanët e ditëve të sotme.

Vëzhgimi dhe kurioziteti nuk është kufizuar vetëm në botën e jashtme. Ne gjithmonë i kthehemi vetes për të kuptuar qenësinë tonë, si në kuptimin filozofik ashtu dhe në atë biologjik. Truri dhe sistemi nervor ka qenë objekt i hershëm studimi dhe vazhdon të jetë me interes të kuptuarit e funksionimit të tij në dy drejtime kryesore; në atë mjekësor për kurimin e sëmundjeve të ndryshme të sistemit nervor, dhe në mundësinë e imitimit të funksioneve të tij dhe vënja në zbatim në fusha të ndryshme si sociale, ekonomike, teknike etj.

Rrjetet Neurale Artificiale janë mjete software ose hardware, të motivuara nga funksioni i sistemit nervor biologjik, por që janë larg imitimit të plotë të këtij sistemi. Për shkak të vetive që ofrojnë, këto rrjete kanë gjetur aplikime të shumta në fusha të ndryshme.

Teknologjia e informacionit është zhvilluar me ritme shumë të shpjeta në dy dekadat e fundit. Zhvillime të tilla, si për shembull Inteligjenca Artificiale dhe Rrjetet Neurale si pjesë e saj, kanë shkruar kufijtë midis teknologjisë të informacionit dhe kontrollit industrial. Tashmë është e vështirë të thuhet ku fillon njëra e ku mbaron tjetra.

Kemi zgjedhur këtë temë, nisur nga kurioziteti për të vëzhguar ku po shkojnë zhvillimet e sotme të teknologjisë në fushën e kontrollit industrial, si dhe nga fleksibiliteti për të kryer simulime me mjete software. Punimi është strukturuar në tre pjesë.

Pjesa e parë jep një vështrim të përgjithshëm teorik mbi rrjetet neurale artificiale.

Në pjesën e dytë jepet një përmbledhje e zbatimeve në fusha të ndryshme, me fokus në identifikim dhe kontrollin industrial.

Në pjesën e tretë jepen një sërë simulimesh për përdorimin e rrjeteve neurale në identifikimin e modelit, kontrollin me model neural invers, vëzhgimin për ndikimin e parametrave të rrjetit neural në cilësinë e sistemit të mbyllur, krahasimin me kontrollin me rregullator konvencional PID, shtimin e elementit integruar në skemën e sistemit të mbyllur me model neural, si dhe ndryshimin në linjë të parametrave të modelit neural.

Punimi mbyllet me përfundimet dhe shtojcat përkatëse ku janë pasqyruar rezultatet e simulimeve dhe kodet e Matlab të përdorura për këto simulime.

Përmbajtja

Mirënjohje.....	2
Parathënie.....	3
Përmbajtja	4
Lista e figurave	6
I. Vështrim i përgjithshëm teorik mbi rrjetet neurale artificiale	10
1.1 Hyrje	10
1.2 Historia e Rrjeteve Neurale.....	11
1.3 Vetitë kryesore të Rrjetet Neurale	14
1.4 Analogjia me neuronin biologjik	17
1.4.1 Sistemi nervor biologjik	17
1.4.2 Modeli i Neuronit Artificial	20
1.4.2.1 Funksioni i aktivizimit të një neuroni	21
1.5 Procesi i të mësuarit	23
1.5.1 Trajnimi me përhapje të pasme.....	23
1.5.2 Algoritmi i përhapjes së pasme për rrjetin tre-shtresor.....	27
1.5.2.1 Llogaritja e peshës për neuronet e shtresës së daljes	27
1.5.2.2 Llogaritja e peshës për neuronet e shtresës së fshehur	30
1.6 Arkitektura të Rrjetit Neural Artificial	32
1.7 Projektimi i rrjetit neural.....	37
1.7.1 Përcaktimi i shtresave të fshehura	38
1.7.2 Përcaktimi i numrit të neuroneve në shtresën e fshehur	38
1.7.3 Përcaktimi i numrit të iterimeve të trajnimit.....	39
II. Zbatime të rrjeteve neurale.....	40
2.1 Zbatime të përgjithshme të rrjeteve neurale	40
2.2 Implementimi software dhe hardware i NN	42
2.2.1 Implementimi hardware i rrjeteve neurale.....	43
2.2.2 Implementimi software kundrejt atij hardware.....	49
2.3 Rrjetet neurale në identifikim dhe kontroll.....	49
2.3.1 Rrjetet neurale me vonesë në kohë	51
2.3.2 Modelet e sistemeve dinamike.....	51
2.3.3 Identifikimi me anë të rrjeteve neurale i modeleve dinamike	53
2.3.4 Modeli dinamik invers	55
2.3.5 Kontrolli me bazë rrjetet neurale	56
III. Simulime të rrjeteve neurale në identifikim dhe kontroll	59
3.1 Aplikimi i rrjeteve neurale në identifikimin/modelimin e objektit.....	59
3.1.1 Modeli i aktuatorit	60
3.1.2 Arkitektura e rrjetit neural	61
3.1.3 Trajnimi i rrjetit neural dhe rezultatet e simulimeve	62
3.1.3.1 Identifikimi i modelit të pozicionit	63
3.1.3.2 Identifikimi i modelit të shpejtësisë.....	65
3.1.4 Përfundime.....	67
3.2 Aplikimi i rrjetit neural në sistemin e mbyllur të kontrollit.....	68
3.3 Vëzhgim mbi ndikimin e elementeve të ndërtimit të rrjetit neural në cilësinë e sistemit të mbyllur të kontrollit.....	70
3.3.1 Modeli i motorit të rrymës së vazhduar DC	71
3.3.2 Modeli neural invers	72

3.3.3 Simulime të sistemit të mbyllur me model neural invers	73
3.3.4 Krahasimi i rezultateve të marra nga simulimet	77
3.4 Krahasimi i kontrollit me model neural invers me kontrollin konvencional me rregullator PID	84
3.4.1 Kontrolli me rregullator PID.....	84
3.4.2 Kontrolli me model neural invers	86
3.4.3 Krahasimi i dy metodave të kontrollit	88
3.5 Vëzhgimi i cilësisë të sistemit të mbyllur me model neural invers pas shtimit të elementit integrues.....	90
3.5.1 Kontrolli me model neural invers	90
3.5.2 Kontrolli me model neural invers dhe element integrues në sistemin e mbyllur.....	93
3.5.3 Krahasimi i dy skemave të kontrollit.....	94
3.6 Kontrolli i shpejtësisë të motorit DC me model neural invers dhe ndryshim në linjë të parametrave të rrjetit neural.....	96
3.6.1 Kontrolli me model neural invers të trajnuar jashtë linje	96
3.6.2 Kontrolli me model neural invers të trajnuar në linjë.....	99
3.6.3 Krahasimi i rezultateve	101
3.7 Reagimi ndaj shqetësimeve i sistemit të mbyllur me model neural invers.....	102
Përfundime.....	104
Shtojcë	107
S.1 Rezultatet e paragrafit 3.3.....	107
S.2 Kodet në Matlab të përdorura për simulimet në kapitullin III.....	116
Bibliografia	128

Lista e figurave

Figura 1.1 Diagrama e paraqitjes të sistemit nervor	17
Figura 1.2. Qeliza nervore	17
Figura 1.3. Organizimi strukturor i niveleve në tru	19
Figura 1.4. Harta arkitekturore e korteksit cerebral:.....	20
Figura 1.5 Struktura e një neuroni artificial.....	21
Figura 1.6 Funksionet e aktivizimit të neuronit artificial	22
Figura 1.7 Struktura e një rrjeti neural me lidhje të drejtpërdrejtë (trajnimi me përhapje të pasme)	25
Figura 1.8 Blllokskema për trajnimin me përhapje të pasme të rrjetit neural me lidhje të drejtpërdrejtë	26
Figura 1.9 Rrjeti tre-shtresor i konceptuar për algoritmin e trajnimit me përhapje të pasme	27
Figura 1.10 Minimizimi i gabimit kuadratik ξ_m^2 me anë të metodës të gradientit zbritës.....	30
Figura 1.11 Rrjeti me lidhje të drejtpërdrejtë me një shtresë neuronesh.	33
Figura 1.12 Rrjeti me lidhje të drejtpërdrejtë plotësisht i lidhur me një shtresë të fshehur dhe një shtresë dalje	34
Figura 1.13 Rrjeti rekurent një shtresor	35
Figura 1.14 Rrjeti rekurent me shtresë të fshehur.....	35
Figura 1.15 Struktura e përgjithshme e rrjetit neural rekurent për sistemin dinamik	37
Figura 2.2 Arkitektura e qarkut TrueNorth të IBM [106].....	46
Figura 2.3 Karta IBM Synapse me 16 qarqe [112].....	47
Figura 2.4 Rrjeti neural me vonesë në kohë	51
Figura 2.5 Modeli-1 i sistemi dinamik	52
Figura 2.6 Modeli-2 i sistemi dinamik	53
Figura 2.7 Modeli-3 i sistemi dinamik	53
Figura 2.8 Modeli-4 i sistemi dinamik	53
Figura 2.9 Identifikimi i modelit dinamik me modelin paralel.....	54
Figura 2.10 Identifikimi i modelit dinamik me modelin seri-paralel	55
Figura 2.11 Modeli neural dinamik invers i objektit	56
Figura 2.12 Trajnimi i rregullatorit neural për të emuluar rregullatorin aktual	56
Figura 2.13 Kontrolli adaptiv i objektit me bazë modelin dinamik invers.....	57
Figura 2.14 Kontrolli adaptiv me model reference me rrjet neural (metoda direkte)	57
Figura 2.15 Kontrolli adaptiv me model reference me rrjet neural (metoda indirekte).....	58
Figura. 3.1. a) Manipulatori me dy hallka, b) Blllokskema e sistemit të çiftuesit kinematik.	60
Figura 3.2 Blllokskema e identifikimi të modelit me rrjet neural.....	62
Figura 3.3 Kampionet H-D të përdorura për trajnimin e modelit neural të pozicionit	63
Figura 3.4 Ecuria e peshave dhe spostimeve të shtresës së fshehur	64
Figura 3.5 Ecuria e peshave dhe spostimit të shtresës së daljes	64
Figura 3.6 Gabimi i trajnimit dhe dalja e modelit neural invers të pozicionit.....	64
Figura 3.7 Gabimi i trajnimit dhe dalja e modelit neural të pozicionit.....	65

Figura 3.8 Kampionet H-D të përdorura për trajnimin e modelit neural të shpejtësisë	66
Figura 3.9 Ecuria e peshave dhe spostimeve të shtresës së fshehur	66
Figura 3.10 Ecuria e peshave dhe spostimit të shtresës së daljes	66
Figura 3.11 Gabimi i trajnimit dhe dalja e modelit neural invers të shpejtësisë.....	67
Figura 3.12 Gabimi i trajnimit dhe dalja e modelit neural të shpejtësisë	67
Figura 3.13 Bllokskema e sistemit të mbyllur të kontrollit me model neural invers	68
Figura 3.14 Sinjali i komandës dhe dalja e sistemit të mbyllur për sinjal hyrje shkallë njësi.....	69
Figura 3.15 Sinjali i komandës dhe dalja e sistemit të mbyllur për sinjal hyrje sinusoidal	69
Figura 3.16 Metoda për trajnimin e rrjetit neural dhe vëzhgimin e treguesve të cilësisë të sistemit të mbyllur.....	70
Figura 3.17. a) Skema stukturore e motorit DC b) Përgjigja shkallë njësi e sistemit me lak të hapur.....	72
Figura 3.18 Skema e identifikimi të modelit neural invers të motorit DC me rrjet neural sipas a) arkitekturës së parë b) arkitekturës së dytë.....	73
Figura 3.19 Skema e identifikimi të modelit neural invers të motorit DC me rrjet neural sipas a) arkitekturës së tretë b) arkitekturës së katërt	73
Figura 3.20 Grupi a) i parë dhe b) i dytë i kampioneve hyrje-dalje të përdorura për trajnimin e rrjetit neural	74
Figura 3.21 Grupi i tretë i kampioneve hyrje-dalje të përdorura për trajnimin e rrjetit neural.....	75
Figura 3.22 Dalja e sistemit të mbyllur për RN me A1 dhe trajnim të tij me kampione sipas a) fig. 3.19a dhe b) fig. 3.19b.....	75
Figura 3.23 Dalja e sistemit të mbyllur për a) RN me A1 dhe trajnim të tij me kampione sipas fig. 3.20 dhe b) RN me A2 dhe trajnim të tij me kampione sipas fig. 3.19a	75
Figura 3.24 Dalja e sistemit të mbyllur për RN me A2 dhe trajnim të tij me kampione sipas a) fig. 3.19b dhe b) fig. 3.20.....	76
Figura 3.25 Dalja e sistemit të mbyllur për RN me A3 dhe trajnim të tij me kampione sipas a) fig. 3.19a dhe b) fig. 3.19b.....	76
Figura 3.26 Dalja e sistemit të mbyllur për a) RN me A3 dhe trajnim të tij me kampione sipas fig. 3.20 dhe b) RN me A4 dhe trajnim të tij me kampione sipas fig. 3.19a	76
Figura 3.27 Dalja e sistemit të mbyllur për RN me A4 dhe trajnim të tij me kampione sipas a) fig. 3.19b dhe b) fig. 3.20.....	77
Figura 3.28 Gabimi i sistemit të mbyllur për RN-A1 dhe trajnim të modelit neural me kampione të ndryshme H-D	77
Figura 3.29 Mbirregullimi i sistemit të mbyllur për RN- A1 dhe trajnim të modelit neural me kampione të ndryshme H-D	78
Figura 3.30 Gabimi i sistemit të mbyllur për RN- A2 dhe trajnim të modelit neural me kampione të ndryshme H-D	78
Figura 3.31 Mbirregullimi i sistemit të mbyllur për RN- A2 dhe trajnim të modelit neural me kampione të ndryshme H-D	79
Figura 3.32 Gabimi i sistemit të mbyllur për RN- A3 dhe trajnim të modelit neural me kampione të ndryshme H-D	79

Figura 3.33 Mbirregullimi i sistemit të mbyllur për RN- A3 dhe trajnim të modelit neural me kampione të ndryshme H-D	80
Figura 3.34 Gabimi i sistemit të mbyllur për RN- A4 dhe trajnim të modelit neural me kampione të ndryshme H-D	80
Figura 3.35. Mbirregullimi i sistemit të mbyllur për RN- A4 dhe trajnim të modelit neural me kampione të ndryshme H-D	81
Figura 3.36 Gabimi i sistemit të mbyllur për gjithë arkitekturat dhe amplitudë të kampioneve H-D të trajnimit $a=0.174$	81
Figura 3.37 Mbirregullimi i sistemit të mbyllur për gjithë arkitekturat dhe amplitudë të kampioneve H-D të trajnimit $a=0.174$	82
Figura. 3.38 Bllokskema e sistemit të mbyllur me rregullator PID	84
Figura 3.39 Tarimi i rregullatorit PID me anë të pidtool të Matlab.....	85
Figura 3.40 Përgjigja e sistemit të mbyllur me rregullator PID.....	85
Figura 3.41 a) ecuria e gabimit të trajnimit b) dalja e modelit neural invers të trajnuar kundrejt daljes së dëshiruar	86
Figura 3.42 Ecuria e peshave dhe spostimeve të shtresës së fshehur gjatë trajnimit të modelit neural	87
Figura 3.43 Ecuria e peshave dhe spostimit të shtresës së daljes gjatë trajnimit të modelit neural	87
Figura 3.44 a) dalja e modelit neural invers në sistemin e mbyllur b) përgjigja e sistemit të mbyllur me model neural invers.....	88
Figura 3.45 Krahasimi i përgjigjeve të sistemit të mbyllur për kontrollin me PID dhe me model neural invers	89
Figura 3.46 Skema strukturore e sistemit të mbyllur të kontrollit me model neural invers plus integrim	90
Figura 3.47 kampionet H-D për trajnimin e modelit neural invers	91
Figura 3.48 a) ecuria e gabimit të trajnimit b) dalja e modelit neural invers të trajnuar kundrejt daljes së dëshiruar	91
Figura 3.49 ecuria e ndryshimit të peshave dhe spostimeve të shtresës së fshehur.....	92
Figura 3.50 ecuria e ndryshimit të peshave dhe spostimit të shtresës së daljes.....	92
Figura 3.51 a) ecuria e daljes së modelit neural invers në skemën e mbyllur b) dalja e sistemit të mbyllur të kontrollit.....	92
Figura 3.52 modeli në Simulink i sistemit të mbyllur të kontrollit me model neural invers dhe element integrues.....	93
Figura 3.53 blloqet e shtresës së fshehur dhe të daljes të modelit neural në Simulink	94
Figura 3.54 a) ecuria e daljes së modelit neural invers në skemën e mbyllur b) dalja e sistemit të mbyllur të kontrollit me element integrues.....	94
Figura 3.55 daljet e sistemeve të mbyllur me model neural invers dhe model neural invers plus integrim	95
Figura 3.56 Bllokskema e kontrollit me model neural invers dhe ndryshim në linjë të parametrave të modelit neural.....	96
Figura 3.57 Kampionet e përdorura për trajnimin jashtë linje të rrjetit neural.....	97
Figura 3.58 a) dalja e dëshiruar dhe reale e rrjetit neural b) ecuria e gabimit të trajnimit.....	97
Figura 3.59 ecuria e peshave dhe spostimeve të shtresës së fshehur të rrjetit neural	98
Figura 3.60 ecuria e peshave dhe spostimit të shtresës së daljes të rrjetit neural	98
Figura 3.61 Dalja e modelit neural invers dhe dalja e sistemit të mbyllur	98

Figura 3.62 ecuria e peshave dhe spostimeve të shtresës së fshehur të ndryshuara në linjë	99
Figura 3.63 ecuria e peshave dhe spostimit të shtresës së daljes të ndryshuara në linjë	100
Figura 3.64 Dalja e modelit neural invers dhe dalja e sistemit të mbyllur për përshtatjen në linjë të peshave dhe spostimeve të rrjetit neural.....	100
Figura 3.65 Dalja e sistemit të mbyllur për sinjal reference sinusoidal.....	100
Figura 3.66 Daljet e sistemeve të mbyllura për modelin neural të trajnuar jashtë linje dhe në linjë.....	101
Figura 3.67 Reagimi ndaj shqetësimeve i sistemit të mbyllur me model neural invers për forma të ndryshme sinjali reference.....	102
Figura 3.68 Daljet e sistemeve të mbyllura për modelin neural të trajnuar jashtë linje dhe në linjë, për forma të ndryshme sinjali reference dhe shqetësim në çaste k	103

I. Vështrim i përgjithshëm teorik mbi rrjetet neurale artificiale

1.1 Hyrje

Puna mbi rrjetet neurale artificiale, zakonisht të referuar si “rrjete neurale”, ka qenë motivuar që në fillimet e veta nga të kuptuarit që truri i njeriut llogarit në mënyrë krejt të ndryshme nga kompjuteri dixhital konvencional. Truri është një *kompjuter me kompleksitet të lartë, jolinear dhe paralel* (sistem përpunimi informacioni). Ai ka aftësinë të organizojë përbërësit e tij strukturorë, të njohur si *neurone*, në mënyrë të atillë që të kryejë llogaritje të caktuara (p.sh., njohje të objektit, perceptimin, dhe kontrollin motor) shumë herë më shpejt se kompjuteri më i shpejtë dixhital që ekziston sot.

Në formën e tij më të përgjithshme, rrjeti neural është një makinë që është projektuar të modelojë mënyrën me të cilën truri kryen një detyrë të caktuar ose një funksion me interes; rrjeti zakonisht zbatohet duke përdorur komponente elektronike ose simulohet në software në kompjuter dixhital. Një klasë e rëndësishme e rrjeteve neurale është ajo që kryen përlllogaritjet nëpërmjet procesit të të mësuarit. Për të arritur një performancë të mirë, rrjetet neurale përdorin një ndërlidhje masive të qelizave llogaritëse të thjeshta, të referuara si “neurone” ose “njësi procesimi”. Kështu, mund të jepet përcaktimi i një rrjeti neural, parë si një makinë adaptive [1]:

Një rrjet neural është një procesor i shpërndarë masiv, i përbërë nga njësi të thjeshta përpunimi që kanë një prirje të natyrshme të ruajnë njohuritë eksperimentale dhe t'i bëjnë të disponueshme për përdorim. Rrjeti neural i përngjan trurit në dy aspekte:

- 1. Njohuria kërkohet nga rrjeti në ambientin që e rrethon nëpërmjet procesit të mësuarit*
- 2. Fuqitë e lidhjeve midis neuroneve, njohur si pesha sinaptike, përdoren për të ruajtur njohurinë e përfutur*

Procedura e përdorur për kryerjen e procesit të të mësuarit quhet *algoritmi i të mësuarit*, funksioni i të cilit është të ndryshojë peshat sinaptike të rrjetit për të marrë një model të dëshiruar.

Ndryshimi i peshave sinaptike siguron metodën tradicionale për projektimin e rrjetit neural. Një përafrim i tillë është më afër me teorinë e filtrit adaptiv linear, që është zbatuar në shumë fusha të ndryshme [1], [2]. Megjithatë, është gjithashtu e mundur për një rrjet neural të ndryshojë arkitekturën e vet, që motivohet nga fakti që neuronet në trurin njerëzor mund të vdesin dhe mund të rriten peshat e reja sinaptike.

1.2 Historia e Rrjeteve Neurale

Historia e rrjeteve neurale ndahet në katër faza: Fillimet e rrjeteve neurale, periudha e parë e artë, vitet e qeta dhe rilindja, që tregon ndërveprimin midis eksperimenteve biologjike, modelimit dhe simulimeve kompjuterike, dhe implementimit hardware.

Vitet 1940: Fillea e Rrjeteve Neurale

Fillea e llogaritjeve neuro, referohet te jetë një artikull kërkimi i McCulloch dhe Pitts publikuar në 1943 [3], që tregoi që edhe tipe të thjeshta të rrjeteve neural, në parim, mund të përlllogaritin çdo funksion logjik dhe aritmetik. Artikulli pati influencë dhe përhapje të madhe.

Kërkues të tjerë, si Norbert Wiener dhe von Neumann, shkruan një libër dhe artikull kërkimor ku sugjeronin që projektimi i kompjuterave të ngjashëm dhe frymëzuar nga truri mund të jetë me interes.

Në 1949 Donald O. Hebb formuloi rregullin klasik [4] që në formën e tij më të përgjithësuar paraqet bazën e pothuaj gjithë procedurave të të mësuarit neural. Sipas këtij rregulli, lidhja midis dy neuroneve forcohet kur të dy neuronet janë aktiv në të njëjtën kohë. Ky ndryshim në fortësi është proporcional me produktin e të dy aktivitetëve. Hebb mundi të formulonte rregullin, por për shkak të mungesës të kërkimit neuro nuk ishte në gjendje ta verifikonte.

Edhe pse në vitet 1940 dhe fillimet e hershme të viteve 1950, kishte shumë studiues që ekzaminuan çështjet rreth përlllogaritjeve neuro, puna e tyre kishte më shumë efekt në vendosjen e bazave për zhvillime të mëvonshme sesa sjellja e zhvillimeve aktuale. Tipike në këtë periudhë ishte ndërtimi i kompjuterit të parë neuro (Snark) nga Marvin Minsky në 1951. “Snark” operoi suksesshëm nga pikëpamja teknike, ai ishte në gjendje të përshtaste automatikisht peshat, por ai asnjëherë nuk kreu ndonjë funksion me interes të përpunimit të informacionit.

Vitet 1950 dhe 1960: Periudha e parë e artë e rrjeteve neurale

1957-1958: Në MIT, Frank Rosenblatt, Charles Wightman dhe bashkëpunëtorë të tyre zhvilluan neurokompjuterin e parë të suksesshëm, perceptronin Mark I, që ishte në gjendje të njëjthe vlere të thjeshta numerike të një sensori imazhi prej 20×20 piksel. Neurokompjuteri punonte me 512 potenciometra, çdo potenciometër paraqiste një variabël (peshë) të rrjetit.

1959: Frank Rosenblatt përshkroi versione të ndryshme të perceptronit, formuloi dhe verifikoi teoremën e tij konvergjente të perceptronit. Ai përshkroi shtresat e neuronit duke imituar retinën, prapun e komutimit, dhe një rregull të mësuarit për përshtatjen dhe lidhjen e peshave.

1960: Bernard Widrow dhe Marcian E. Hoff paraqitën ADALINE (ADaptive LInear NEuron) [5], një sistem të mësuarit adaptiv dhe preciz, që ishte rrjeti i parë neural i përdorur komercialisht në masë të gjerë: ai mund të gjendet në pothuaj çdo telefon analog për filtrimin adaptiv të ekos në kohë reale dhe ishte trajnuar sipas rregullit Widrow-Hoff ose rregulli delta. Hoff në atë kohë, më vonë bashkëthemeluesi i Intel Corporation, ishte një student PhD i Widrow, që ishte i njohur si një shpikës i mikroprocesorëve modernë. Një përparësi që rregulli delta kishte kundrejt algoritmit origjinal të të mësuarit të perceptronit ishte përshtatshmëria e tij: Nëse diferenca midis daljes aktuale dhe zgjidhjes së duhur ishte e madhe, peshat lidhëse ndryshonin gjithashtu në hapa më të mëdhenj – më të vegjël hapat, më afër ishte objektivi.

1961: Karl Steinbuch paraqiti realizimin teknik të kujtesës shoqëruese, që mund të shihet si pararendës i kujtesës shoqëruese neurale në ditët e sotme [6]. Ai përshkroi koncepte për teknikat neurale dhe analizoi mundësitë dhe kufijtë e tyre.

1965: Në librin “Learning Machines” (makinat që mësojnë), Nils Nilsson dha një vështrim të përgjithshëm të progresit dhe punës të kësaj periudhe të kërkimit mbi rrjetet neurale. Në këtë libër u mor e mirëqenë që parimet bazë të të vetë-mësuarit, dhe prej këndej sistemet “inteligjente” ishin zbuluar tashmë. Sot kjo hipotezë duket të jetë një mbivlerësim i tepruar, por në atë kohë kishte shumë popullaritet dhe gjeneroi fonde të mjaftueshme kërkimi.

1969: Marvin Minsky dhe Seymour Papert publikuan një analizë matematike të perceptronit [7] për të treguar që modeli i perceptronit nuk ishte në gjendje të paraqiste shumë probleme të rëndësishme. Kjo analizë vendosi mbarimin e mbivlerësimit, popullaritetit dhe fondeve të kërkimit. Si pasojë u nënkuptua që modele më të fuqishme mund të kishin të njëjtat probleme dhe parashikimi që gjithë fusha e kërkimit mund të ishte një fund pa krye, rezultoi në pothuaj një rënie të plotë të fondeve të kërkimit për 15 vitet në vijim, pa marrë parasysh se sa të pasakta ishin këto parashikime nga pikëpamja e ditëve të sotme.

Vitet 1970: Vitet e heshtjes dhe rindërtimit të ngadaltë

Pavarësisht demonstrimit të Minsky dhe Papert për kufizimet e perceptronit, kërkimi për rrjetet neurale vazhdoi. Shumë kërkime ishin nën mbulesën e përpunimit adaptiv të sinjalit, njohjes së objektit dhe modelimit biologjik. Shumë nga kërkuesit në fushë

filluan të publikonin punët gjatë viteve '70. Ata, dhe të tjerë që i pasuan në vitet e tjerë, vendosën fushën e rrjeteve neurale në një bazë të fortë dhe përgatitën rrugën për rilindjen e fushës. Disa prej kërkimeve të kësaj periudhe janë:

1972: Teuvo Kohonen paraqiti një model të një shoqëruesi linear; një model të një kujtesë shoqëruese [8]. Në të njëjtin vit, një model i tillë u paraqit në mënyrë të pavarur nga pikëpamja e një neurofizilogjisti nga James A. Anderson [9].

1973: Christoph von der Malsburg përdori një model neuroni që ishte jolinear dhe biologjikisht më shumë motivues. [10].

1974: Paul Werbos zhvilloi në disertacionin e tij në Harvard, një procedurë të mësuarit të quajtur përhapje e pasme e gabimit [11]. Kjo procedurë arriti vetëm një dekadë më vonë rëndësinë e ditëve të sotme.

1976-1980 dhe më pas: Stephen Grossberg paraqiti shumë artikuj (për shembull [12]) ku analizohen matematikisht shumë modele neurale. Më tej, ai u fokusua në problemin e mbajtjes të një rrjeti neural të aftë për të mësuar, pa prishur lidhjet e mësuara tashmë. Në bashkëpunim me Gail Carpenter, kjo shpuri në teorinë e modeleve ART (adaptive resonance theory – teoria e rezonancës adaptive).

Rilindja

Nëpërmjet influencës të John Hopfield, i cili personalisht bindi shumë kërkues për rëndësinë e fushës, dhe publikimeve të shumta nga Rumelhart, Hinton dhe Williams, fusha e rrjeteve neurale tregoi ngadalë shenja të ngritjes. Për t'u përmendur në këtë periudhë:

1982: Teuvo Kohonen përshkroi hartëzimin SOM (Self-organizing feature maps – Hartat me vetë-organizim) [13], [14], të njohura ndryshe si hartat Kohonen. Ai kërkonte për mekanizimin që përfshin vetë-organizimin në tru. Kohonen e dinte që informacioni për krijimin e një qenie ruhet në gen, që megjithatë, nuk ka kujtesë të mjaftueshme për një strukturë si truri. Si pasojë, truri duhet të organizohet dhe krijohet vetë për shumicën e pjesës të qenies.

John Hopfield shpiku të ashtuquajturit rrjetet Hopfield [15] që janë të frymëzuar nga ligjet e magnetizmit në fizikë. Këto rrjete nuk u përdorën gjerësisht në aplikimet teknike, por fusha e rrjeteve neurale rifitoi rëndësi dalëngadalë.

Po në 1982 në Kyoto, Japoni, u mbajt konferenca USA – Japoni mbi Bashkëpunimin/ Konkurrencën në Rrjetet Neurale. Japonia shpalli Gjeneratën e Pestë në rrjetet neurale dhe periodiku amerikan gjeneroi shqetësimin që USA mund të mbehte prapa në fushë

(Gjenerata e pestë në informatikë përfshin inteligjencën artificiale. Gjenerata e parë përdorte komutuesit (*switch*) dhe linjat (*wire*), gjenerata e dytë përdorte transistorë, e treta përdorte teknologji solide si qarqet e integruar dhe gjuhë programimi të nivelit të lartë, dhe gjenerata e katërt janë gjeneruesit e kodeve).

1983: Fukushima, Miyake dhe Ito paraqitën një model neural të një Neocognitron që mund të njihnte karakteret e shkrimit të dorës [16] dhe ishte një zgjerim i rrjetit Cognitron të zhvilluar tashmë në 1975.

1986: Procedura e të mësuarit nëpërmjet përhapjes të pasme të gabimit si një përgjithësim i rregullit delta u zhvillua veçmas dhe u publikua gjerësisht nga “Parallel Distributed Processing Group” [17]: Problemet jo-linearisht-të ndashme mund të zgjidhen nga perceptronët shumështrësorë, dhe vlerësimet negative të Marvin Minsky u provuan të jenë jo të vërteta.

Nga kjo kohë e më tej, zhvillimet në fushën e kërkimit kanë qenë pothuaj eksplozive dhe nuk mund të detajohen një nga një.

1.3 Vetitë kryesore të Rrjetet Neurale

Fuqia llogaritëse e një rrjeti neural buron së pari nga struktura masive dhe e shpërndarë paralele dhe, së dyti, aftësia për të mësuar, pra për të përgjithësuar. Përgjithësimi i referohet prodhimit nga rrjeti neural të daljeve të pranueshme, për hyrje që nuk janë ndeshur gjatë trajnimit (mësimi). Këto dy aftësi procesimi informacioni i mundësojnë rrjeteve neurale të gjejnë zbatim për probleme me shkallë të lartë kompleksiteti e që janë të pa kontrollueshme. Megjithatë, në praktikë rrjetet neurale nuk mund të ofrojnë zgjidhje nëse punojnë veçmas. Ata duhet të integrohen në një sistem inxhinierik të pandryshueshëm. Specifikisht, një problem kompleks shpërbëhet në një numër detyrash relativisht të thjeshta, dhe rrjeteve neurale u caktohet një nëngrup i detyrave që përputhet me aftësitë e tyre të brendshme. Gjithsesi është e rëndësishme të thuhet që ka një rrugë shumë të gjatë për të kryer (nëse do ndodhë) para se të ndërtohet një arkitekturë kompjuteri që imiton trurin njerëzor.

Rrjetet neurale ofrojnë mundësitë dhe vetitë e mëposhtme [1]:

1. *Jolineariteti*. Një neuron artificial mund të jetë linear ose jolinear. Një rrjet neural, i përbërë nga ndërlidhja e neuroneve, është në vetvete jolinear. Për më tepër, jolineariteti është i një lloji të veçantë në kuptimin që është i shpërndarë përgjatë rrjetit.

2. *Hartëzimi Hyrje-Dalje.* Një model popullor i të mësuarit, i quajtur të mësuarit me mësues, ose të mësuarit e mbikëqyrur, përfshin ndryshimin e peshave sinaptike të një rrjeti neural nëpërmjet aplikimit të një grupi kampionesh trajnimi të etiketuar. Çdo kampion përbëhet nga një sinjal hyrje unik dhe një përgjigje e dëshiruar korresponduese. Rrjeti prezantohet me një kampion të marrë në mënyrë rastësore nga grupi i kampioneve, dhe peshat sinaptike të rrjetit ndryshohen për të minimizuar diferencën midis daljes së dëshiruar dhe asaj aktuale të rrjetit. Trajnimi i rrjetit përsëritet për shumë kampione nga grupi i marrë, gjersa rrjeti arrin një gjendje të qëndrueshme ku nuk ka më ndryshim të ndjeshëm në peshat sinaptike. Kështu, rrjeti mëson nga kampionet duke ndërtuar hartëzimin hyrje-dalje për problemin në shqyrtim.

3. *Përshtatshmëria.* Rrjetet neurale kanë aftësi të brendshme të ndryshojnë peshat e tyre sinaptike me ndryshimin e ambientit rrethues. Në veçanti, një rrjet neural i trajnuar të veprojë në një ambient specifik mund të ri-trajnohet lehtësisht që të përshtatet me ndryshime të vogla në kushtet e ambientit operacional. Për më tepër, kur rrjeti operon në një ambient të ndryshueshëm (p.sh., ku statika ndryshon në kohë), ai mund të projektohet të ndryshojë peshat sinaptike në kohë reale. Arkitektura e rrjetit neural për aplikime të klasifikimit të objektit, procesimit të sinjalit, dhe kontroll, shoqëruar me aftësitë përshtatëse të rrjetit, e bën atë një mjet të fuqishëm në klasifikim adaptiv të objektit, procesimin adaptiv të sinjalit dhe kontrollin adaptiv. Si rregull i përgjithshëm, mund të thuhet që sa më shumë adaptiv e bëjmë një sistem, duke siguruar që gjithë kohës sistemi është i qëndrueshëm, aq më robuste ka gjasa të jetë performanca e tij kur sistemi kërkohet të operojë në ambient jo stacionar (të ndryshueshëm). Megjithatë duhet theksuar, që përshtatshmëria jo gjithmonë shpie në robustshmëri; në të vërtetë, mund të bëjë të kundërtën. Për shembull, një sistem adaptiv me konstante kohe të vogël mund të ndryshojë shumë shpejt dhe ndaj tenton të përgjigjet ndaj shqetësimeve të rreme, duke shkaktuar një degradim drastik të performancës të sistemit. Për të realizuar përfitimet e plota të përshtatshmërisë, konstantet primare të kohës të sistemit duhet të jenë mjaftueshëm të mëdha që sistemi të injorojë shqetësimet e rreme, dhe përsëri mjaftueshëm të vogla për t'u përgjigjur ndryshimeve me vlerë në ambient; ky problem referohet si dilema qëndrueshmëri-plasticitet [18].

4. *Përgjigja evidentuese.* Në kontekstin e klasifikimit të objektit, një rrjet neural mund të projektohet të japë informacion jo vetëm për objektin e veçantë që duhet të zgjedhë, por gjithashtu edhe për besueshmërinë në vendimin e marrë. Ky informacion mund të përdoret për të përjashtuar objekte të paqarta, nëse dalin, dhe kështu përmirëson performancën e klasifikimit të rrjetit

5. *Informacioni kontekstual.* Njohuria paraqitet nga struktura dhe gjendja e aktivizimit e rrjetit neural. Çdo neuron në rrjet ndikohet potencialisht nga aktiviteti global i gjithë neuroneve në rrjet. Rrjedhimisht, informacioni kontekstual shpërndahet në mënyrë natyrale nga rrjeti neural.

6. *Toleranca ndaj dëmtimit.* Një rrjet neural i zbatuar në hardware, ka potencialin për të qenë në thelb tolerant ndaj dëmtimeve, ose i aftë për llogaritje robuste, në kuptimin që performanca e tij degradon në formë të atillë që rrjeti operon me funksione të kufizuara nën kushte operimi jo të favorshme. Për shembull, nëse një neuron ose lidhjet e tij dëmtohen, rikthimi i një kampioni të ruajtur përkeqësohet në cilësi. Megjithatë, për shkak të natyrës të shpërndarë të informacionit të ruajtur në rrjet, dëmtimi duhet të jetë shumë i përhapur para se përgjigja e përgjithshme e sistemit të degradohet seriozisht. Kështu, në parim, një rrjet neural shfaq një degradim të lehtë në performancë dhe jo një dëmtim katastrofik. Për t'u siguruar që rrjeti neural është në fakt tolerant ndaj dëmtimit, mund të jetë e nevojshme të merren masa korrektuese në projektimin e algoritmit të përdorur për trajnimin e rrjetit [19].

7. *Njëtrajtshmëria e Analizës dhe Projektimit.* Rrjetet neurale gëzojnë universalitetin si procesorë informacioni. Kjo në kuptimin që i njëjti simbol/kuptim përdoret në të gjithë fushat që përfshijnë aplikimin e rrjeteve neurale. Kjo veti manifestohet në mënyra të ndryshme:

- Neuronet, në një formë ose në tjetrën, paraqesin një përbërës të përgjithshëm në gjithë rrjetet neurale.
- Përgjithshmëria bën të mundur përdorimin e të njëjta teorive dhe algoritmeve të të mësuarit në aplikime të ndryshme të rrjeteve neurale.
- Rrjetet modulare mund të ndërtohen lehtësisht nga integrimi i moduleve me njëri-tjetrin.

8. *Analogjia Neurobiologjike.* Projektimi i rrjeteve neurale është motivuar nga analogjia me trurin, që është një provë jetësore që procesimi paralel dhe me tolerancë ndaj dëmtimit jo vetëm që është fizikisht i mundur, por edhe i shpejtë dhe i fuqishëm.

- Neurobiologët i shohin rrjetet neurale (artificiale) si një mjet kërkimi për interpretimin e fenomeneve neurobiologjike.
- Ndërsa inxhinierët e shohin neurobiologjinë për ide të reja për të zgjidhur probleme më komplekse sesa ato të bazuara në teknikat konvencionale të projektimit.

1.4 Analogjia me neuronin biologjik

1.4.1 Sistemi nervor biologjik

Sistemi nervor human mund të shihet si një sistem me tre faza, siç jepet në Figura 1.1 [20]. Qendra e sistemit është truri, i paraqitur nga rrjeti neural që vazhdimisht merr informacion, e percepton, dhe merr vendime të duhura. Në figurë tregohen dy grupe shigjetash. Ato me drejtim nga e majta në të djathtë tregojnë transmetimin e drejtpërdrejtë të informacionit, duke mbartur sinjalet përgjatë sistemit. Shigjetat nga e djathta në të majtë tregojnë praninë e lidhjes së kundërt në sistem. Marrësit shndërrojnë nxitjet nga trupi human ose ambienti i jashtëm në impulse elektrike që përçojnë informacionin në rrjetin neural (tru). Ndikuesit shndërrojnë impulset elektrike të gjeneruara nga rrjeti neural në përgjigje ose dalje të sistemit.

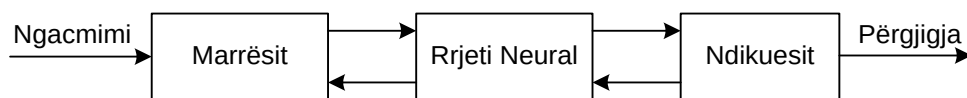


Figura 1.1 Diagrama e paraqitjes të sistemit nervor

Vështirësia për të kuptuar trurin është bërë më e lehtë në sajë të punës pioniere të Ramon Cajal [21], që shtroi idenë e neuroneve si struktura përbërëse e trurit. Neuronet biologjike janë pesë ose gjashtë rinde madhësie më të ngadaltë se portat logjike të silikonit; veprimet në një çip silikon ndodhin në rinde nanosekonde, ndërsa në neuron ndodh në rinde të milisekondës. Megjithatë, neuronet që përbëjnë trurin, edhe pse të ngadaltë në veprime, kanë një numër masiv ndërlidhjesh midis tyre. Është vlerësuar që ka afërsisht 10 bilion neurone në korteksin human, dhe 60 trilion sinapse ose lidhje [22]. Rezultat i këtij rrjeti është që truri është një strukturë jashtëzakonisht efiçente.

Neuronet kanë një gamë të gjerë formash dhe madhësish në pjesë të ndryshme të trurit. Figura 1.2 ilustron formën e një qelize neurale.

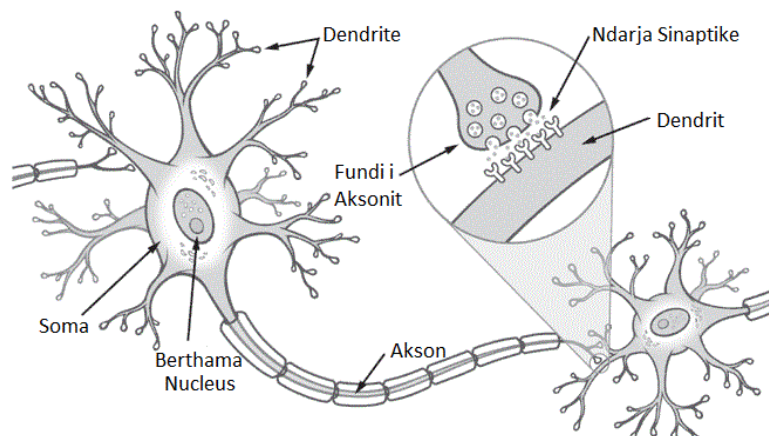


Figura 1.2. Qeliza nervore

Sinapset, ose nervat fundore, janë struktura elementare dhe njësi funksionale që ndërmjetësojnë komunikimin midis neuroneve. Tipi më i zakonshëm i sinapsit është një sinaps kimik që vepron si vijon: Një proces parasinaptik çliron një substancë që përhapet përgjatë bashkimit sinaptik midis neuroneve dhe më pas vepron në procesin pas-sinaptik. Kështu një sinaps shndërron një sinjal elektrik parasinaptik në një sinjal kimik, dhe më pas në një sinjal elektrik pas-sinaptik [22]. Në terminologjinë elektrike, thuhet të jetë një pajisje dy-portëshe jo-reciproke. Në përshkrimet tradicionale të organizimit neural, është supozuar që një sinaps është një lidhje e thjeshtë që mund të imponojë në neuronin marrës ngacmim ose frenim, por jo të dyja.

Plasticiteti lejon zhvillimin e sistemit nervor për t'u përshtatur me ambientin rrethues [23]–[25]. Në një tru të rrituri, plasticiteti mund të vlerësohet nga dy mekanizma: krijimi i lidhjeve të reja sinaptike midis neuroneve, dhe ndryshimi i sinapseve ekzistuese.

Aksonet, linjat transmetuese, dhe dendritet, zonat marrëse, përbëjnë dy tipet e filamenteve qelizore që dallohen nga ana morfologjike; një akson ka sipërfaqe më të lëmuar, më pak degë, gjatësi më të madhe, ndërsa një dendrit ka një sipërfaqe të parregullt dhe shumë degë [26].

Shumica e neuroneve kodojnë daljet e tyre si një seri pulsesh të shkurtër tensioni. Këto pulse, zakonisht të njohur si potenciale veprimi, fillojnë në/ose pranë trupit të qelizës të neuronit dhe më pas përhapen përgjatë neuroneve individuale me amplitudë dhe shpejtësi konstante. Arsyeja për përdorimin e potencialeve të veprimit për komunikimin midis neuroneve bazohet në fizikën e aksoneve. Aksoni i një neuroni është shumë i gjatë dhe i hollë dhe karakterizohet nga rezistencë e lartë elektrike (R) dhe kapacitet të madh (C). Të dy këto elemente janë të shpërndarë përgjatë aksonit. Ndaj, aksoni mund të modelohet si një linjë transmetuese RC. Mekanizmi i këtij mekanizmi përhapje, tregon që kur një tension aplikohet në një nga fundet e aksonit, ai dobësohet eksponencialisht me distancën, duke rënë në një nivel të papërfillshëm në kohën që arrin në fundin tjetër. Potencialet e veprimit sigurojnë një mënyrë për të shmangur këtë problem transmetimi [27].

Në tru, janë të dyja organizimet atomike, ato në shkallë të vogël dhe ato në shkallë të madhe, dhe funksionet e ndryshme ndodhin në nivele më të ulëta ose më të larta. Figura 1.3 tregon një hierarki të niveleve të organizimit, që ka dalë nga puna e bërë në analizën e rajoneve lokale të trurit [22], [24]. Sinapset paraqesin nivelin më thelbësor, dhe janë të varur për veprimet e tyre nga qelizat dhe jonet. Në nivelet e tjerë, janë mikroqarqet neurale, pemët e dendriteve, dhe pastaj neuronet. Një mikroqark neural i referohet një grumbullimi sinapsesh të organizuar në mostra të lidhjes për të prodhuar një operacion funksional në interes. Një mikroqark neural mund të krahasohet me një çip silikoni të përbërë nga assemblimi i disa tranzistorëve. Madhësia më e vogël e mikroqarqeve matet në mikrometër (μm), dhe shpejtësia më

e madhe e tyre e operimit matet në milisekonda. Mikroqarqet neurale janë grupuar për të formuar nën-njësi dendritike brenda pemëve dendritike të neuroneve individualë. Neuroni si i tërë, me madhësi rreth 100 μm , ka nën-njësi të shumta dendritike. Në nivelin tjetër të kompleksitetit, janë qarqet lokale (rreth 1mm në madhësi) të përbërë nga neurone me veti të njëjta ose të ndryshme; këto grupime neuronesh kryejnë veprimeve të një zone të lokalizuar të trurit. Ato ndiqen nga qarqet ndër-rajonale të përbërë nga rrugë, kolona, dhe harta topografike, që përfshijnë rajone të shumtë të lokalizuar në pjesë të ndryshme të trurit.

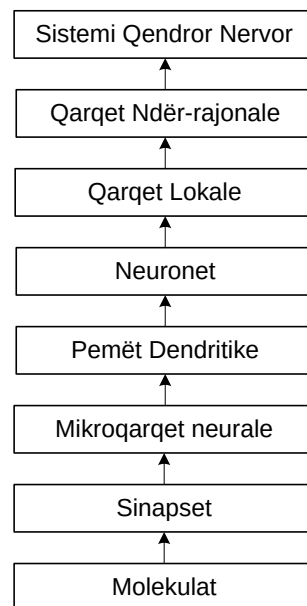


Figura 1.3. Organizimi strukturor i niveleve në tru

Hartat topografike janë organizuar për t'iu përgjigjur informacionit ndijor hyrës. Këto harta shpesh organizohen në shtresa, ku hartat e të parit, të dëgjuarit, dhe somatosensore janë vendosur në shtresat ndihmëse në mënyrë të tillë që ngacmimi nga pikat korresponduese në hapësirë qëndron sipër ose poshtë tyre. Figura 1.4 paraqet një hartë arkitekturore të korteksit cerebral [28]. Kjo figurë tregon që hyrjet e ndryshme sensorë (motore, somatosensore, pamore, dëgjimore, etj.) janë hartëzuar në mënyrë të rregullt në zona korresponduese të korteksit cerebral. Në nivelin final të kompleksitetit, hartat topografike dhe qarqet ndër-rajonale përçojnë tipe specifike të sjelljes në sistemin qendror nervor.

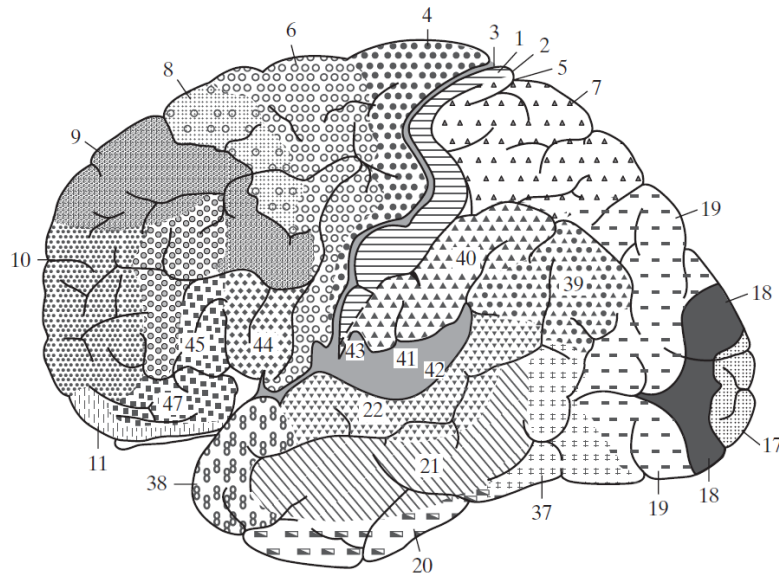


Figura 1.4. Harta arkitekturore e korteksit cerebral:

Zonat e ndryshme identifikohen nga trashësia e shtresave të tyre dhe tipi i qelizave që kanë. Disa nga zonat kryesore sensore janë: korteksi motor: brezi i ngushtë motor, zona 4; zona paramotore, zona 6; zona pamore ballore, zona 8; korteksi somatosensor: zonat 3, 1, dhe 2; korteksi vizual cortex: zonat 17, 18, dhe 19; Korteksi dëgjimor: zonat 41 dhe 42. (Nga A. Brodal, 1981[28]).

Është e rëndësishme të thuhet që nivelet strukturore të organizimit të përshkruar janë një karakteristike unike e trurit. Ato nuk mund të gjenden asgjëkundi në kompjuterat dixhitalë, dhe nuk ka ndonjë arritje për ri-krijimin e tyre në rrjetet neurale artificiale. Megjithatë, ka përparime drejt një hierarkie të niveleve llogaritëse të ngjashme me atë të përshkruar në Figurën 1.3. Neuronet artificiale të përdorur për ndërtimin e rrjeteve neurale janë shumë primitive në krahasim me ata të gjendur në tru. Rrjetet neurale që mund të projektohen për momentin janë shumë primitive në krahasim me qarqet lokale dhe qarqet ndër-rajonale në tru. Megjithatë, çfarë është vërtet e kënaqshme, është progresi i jashtëzakonshëm bërë në shumë drejtime. Me analogjinë neurobiologjike si burim frymëzimi, dhe me bollëkun e mjeteve llogaritëse dhe teorike, është e sigurtë që të kuptuarit e rrjeteve neurale artificiale dhe aplikimeve të tyre do vazhdojë të rritet në shkallë të gjerë vit pas viti.

1.4.2 Modeli i Neuronit Artificial

Një neuron artificial është një koncept, komponentët e të cilit kanë një analogji direkte me neuronin biologjik. Figura 1.5 tregon strukturën e një neuroni artificial, që të kujton një llogaritje të ngjashme me shumën. Ai gjithashtu quhet një neuron, element përpunues (EP), neurodë, nyje, ose qelizë. Sinjalet hyrës, $X_1, X_2, X_3, \dots, X_N$, janë normalisht variable të vijuar por mund të jenë edhe impulse diskrete. Çdo sinjal

(1.1)

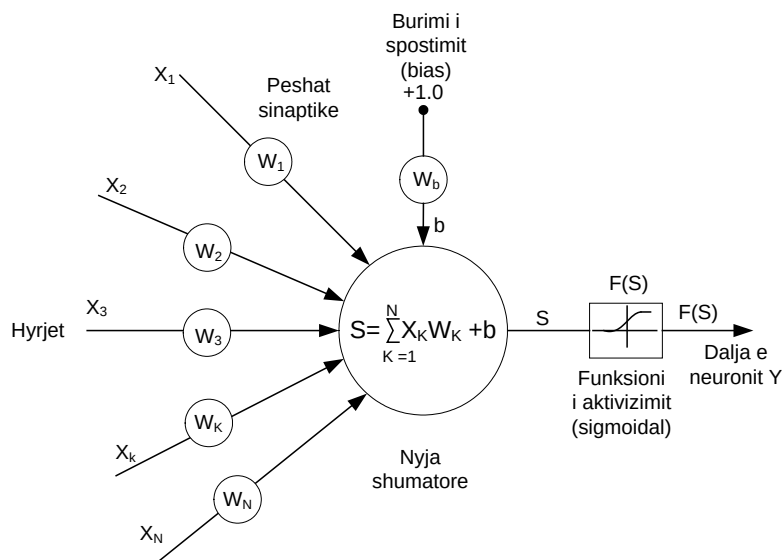


Figura 1.5 Struktura e një neuroni artificial

1.4.2.1 Funksioni i aktivizimit të një neuroni

Figura 1.6 tregon një numër të mundshëm funksionesh aktivizimi (të quajtur gjithashtu funksion transmetimi) në një neuron. Më i thjeshti nga të gjithë është funksioni linear i aktivizimit, ku dalja ndryshon linearisht me hyrjen por saturohet në ± 1 nga një amplitudë e madhe e hyrjes. Mund të jetë gjithashtu një funksion i tipit njëdrejtëmsh. Funksioni i aktivizimit mund të jetë shkallë ose i tipit kufizues që kalon 1-in logjik nëse $S > 0$, ose -1 nëse $S < 0$. Një zhvendosje pozitive ose negative mund të paraqitet në S për të ndryshuar vlerën e kufizimit. Në funksionin e aktivizimit *signum* të treguar në (c) me zhvendosje zero, dalja është +1 nëse $S > 0$, ose -1 nëse $S < 0$. Një neuron i vetëm me një funksion aktivizimi kufizues njihet si një Perceptron një-shtresor, ndërsa i njëjti neuron me një funksion aktivizimi *signum* njihet si Adaline.

Funksionet e aktivizimit më të përdorura zakonisht janë jolineare, tipe që ndryshojnë vazhdueshmërisht midis dy vlerave asimptotike 0 dhe 1 ose -1 dhe +1. Këto janë, respektivisht, funksioni sigmoidal (i quajtur gjithashtu funksioni log-sigmoid ose logjistik) i treguar në (d) dhe funksioni tangent hiperbolik (i quajtur gjithashtu tan-sigmoid) treguar në (e) të figurës. Shprehjet matematike të këtyre funksioneve jepen në figurë, ku α është amplifikimi ose koeficienti që përshtat pjerrësinë ose sensitivitetin. Këto funksione janë të diferencueshëm, dhe derivati $(F(S))/dS$ ka maksimum në $S = 0$, dhe zvogëlohet gradualisht me rritjen e vlerës të S në çfarëdo drejtimi. Me një vlerë amplifikimi shumë të lartë, funksioni linear (njëdrejtues) dhe ai sigmoidal i përafrohen funksionit kufizues, dhe funksionet dy-drejtimësh linear dhe tangent hiperbolik i përafrohen funksionit signum. Funksioni i aktivizimit mund të jetë gjithashtu edhe i tipit Gaussian. Gjithë këta funksione janë karakterizuar si funksione shtypës sepse ata shtypin ose kufizojnë midis vlerave asimptotike operacionin dinamik të një neuroni. Shënohet që një funksion aktivizimi jolinear kontribuon në karakteristikën jolineare transferuese të një neuroni, që lejon hartëzimin jolinear hyrje/dalje të një rrjeti neural. Kjo veti jolineare humb nëse zgjidhet një funksion linear aktivizimi.[29]–[31], [34]

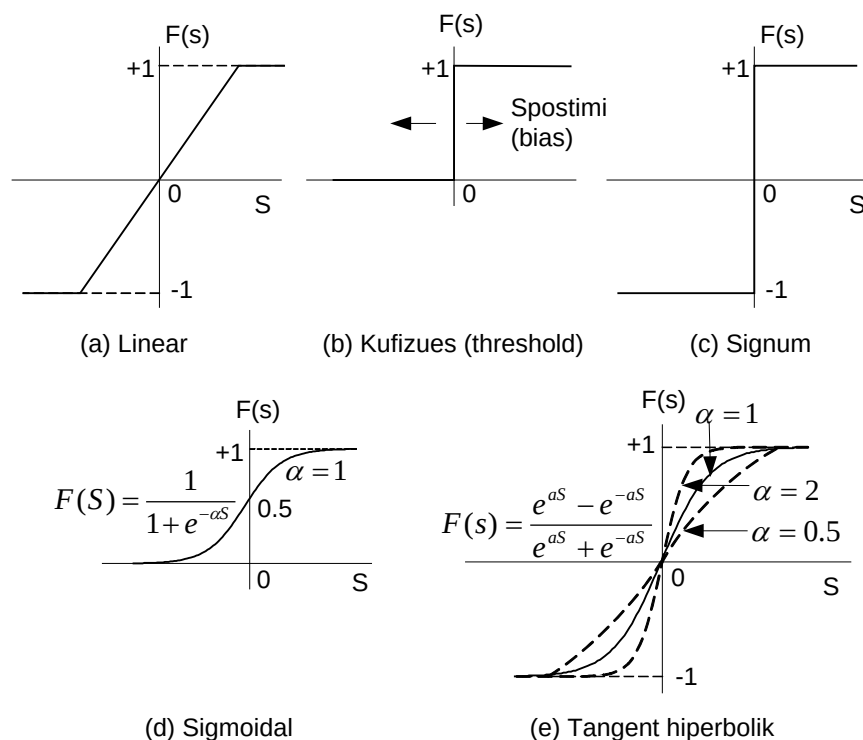


Figura 1.6 Funksionet e aktivizimit të neuronit artificial

1.5 Procesi i të mësuarit

Të mësuarit është një metodë përshtatje me anë të të cilës një rrjet neural vetë organizohet për të dhënë sjellje të dëshiruar.

E thënë ndryshe, të mësuarit në kontekstin e një rrjeti neural është procesi i të përshtaturit të peshave dhe spostimeve në mënyrë të tillë që për hyrje të dhëna, arrihen përgjigje ose dalje të sakta.

Tre metodat kryesore të të mësuarit të rrjetit neural janë [1], [29], [31], [34]:

Të mësuarit e pambikëqyrur ose me vetë-mësim, e quajtur ndryshe edhe të mësuarit pa mësues. Kjo metodë është më afër mënyrës të të mësuarit të rrjetit neural biologjik, por nuk është e përshtatshme për gjithë problemet. Në këtë metodë grupi i të dhënave të trajnimit konsiston vetëm në kampionet e hyrjes, dalja që duhet të japë rrjeti nuk njihet; rrjeti përpiket të identifikojë kampione të ngjashme dhe t'i klasifikojë në kategori të ngjashme. Shembull tipik i kësaj metode janë hartat Kohonen me vetë-organizim.

Të mësuarit e përforcuar: Grupi i të dhënave të trajnimit konsiston vetëm në kampionet e hyrjes. Pas plotësimit të një sekuence hyrje-dalje, dihet vetëm nëse rrjeti jep rezultatin e duhur ose jo. Peshat rifreskohen bazuar vetëm në këtë informacion, që është vlerë "1" ose "0" ("true" ose "false").

Të mësuarit e mbikëqyrur, i quajtur ndryshe të mësuarit me mësues. Në këtë metodë, grupi i të dhënave të trajnimit konsiston në kampionet hyrje-dalje; njihen jo vetëm kampionet e hyrjes por edhe daljet korresponduese. Kështu, që pas çdo grupi trajnimi, rrjeti mund të marrë një vektor gabimi që paraqet diferencën e daljes së rrjetit neural me daljen reale. Rrjeti rifreskon peshat dhe spostimet në varësi të këtij vektori gabimi. Objektivi është të ndryshohen peshat e rrjetit në atë shkallë që rrjeti jo vetëm të arrijë të hartëzojë kampionet e njohura hyrje-dalje, por të arrijë të japë rezultate të besueshme edhe kundrejt hyrjeve të panjohura.

Të mësuarit e mbikëqyrur është më i përhapuri ndër metodat e të mësuarit.

1.5.1 Trajnimi me përhapje të pasme

Trajnimi me përhapje të pasme (*Back Propagation training*) është një metodë e mbikëqyrur të mësuarit për trajnimin e rrjeteve neurale artificiale. Ajo është një nga metodat më popullore për një rrjet me lidhje të drejtpërdrejtë shumë-shtresor; ndaj rrjeti standart (Figura 1.7) i trajnuar nga ky algoritëm shpesh quhet rrjeti me përhapje të pasme. Rumelhart, Hinton, dhe Williams [17] propozuan metodën e trajnimit me përhapje të pasme, ndonëse të tjerë dhanë kontributin e tyre në mënyrë të pavarur. Në parim, ky tip trajnimi është një përgjithësim i rregullit të të mësuarit delta të

zhvilluar nga Widrow dhe Hoff [5] për trajnimin Adaline. Figura 1.8 tregon bllokskemën për trajnimin me përhapje të pasme të një rrjeti. Fillimisht, zgjidhet topologjia e rrjetit me numrin e shtresave, numrin e neuroneve në shtresat e fshehura, dhe funksioni i aktivizimit. Kampione shembull të dhënash hyrje/dalje mund të mblidhen nga eksperimenti, ose nga sistemi i simuluar nëse është i disponueshëm modeli i sistemit. Para se të fillojë trajnimi rrjeti inicializohet me peshë pozitive ose negative të rastësishme. Me një kampion hyrje, dalja llogaritet (përcaktuar si kalim i përpamë/drejtëpërdrejtë) dhe krahasohet me kampionin e dëshiruar të daljes. Më pas, peshat ndryshohen derisa gabimi midis kampionit të llogaritur dhe atij të dëshiruar është shumë i vogël dhe i pranueshëm. Trajnim i ngjashëm kryhet me të gjithë kampionet në mënyrë që përputhja të jetë për gjithë daljet. Në këtë pikë, rrjeti thuhet të jetë trajnuar në mënyrë të kënaqshme për të kryer funksione të vlefshme. Nëse gabimi nuk konvergjon mjaftueshëm, mund të jetë e nevojshme të alternohet numri i neuroneve të shtresës së fshehur ose të shtohen shtresa. Në vend të zgjedhjes të një kampioni në një kohë në sekuencë, mund të përdoret trajnimi në tufë, ku gjithë kampionet paraqiten njëherësh dhe rifreskimi i peshave përfundimtare bëhet pas përpunimit të gjithë kampioneve.

Përshtatja e peshave për reduktimin e gabimit (zakonisht një funksion kosto i dhënë nga gabimi kuadratik) kryhet nga një teknikë gradienti standarte zbritëse, ku peshat iterohen një në një kohë duke filluar mbrapsht nga shtresa e daljes. Një ecje e tillë mbrapsht për llogaritje të tilla është njohur si kalim i kundërt (revers). Prej kësaj ka rrjedhur edhe origjina e emrit “përhapje e pasme – back propagation”. Një kalim-i plotë i llogaritjeve, që konsiston në një çift të një kalimi të drejtëpërdrejtë dhe një kalimi të kundërt (revers), është përcaktuar si një periodë. Algoritmi është kompleks dhe zakonisht e implementon një program kompjuteri. Baza matematikore e algoritmit përshkruhet më poshtë. [1], [29]–[34]

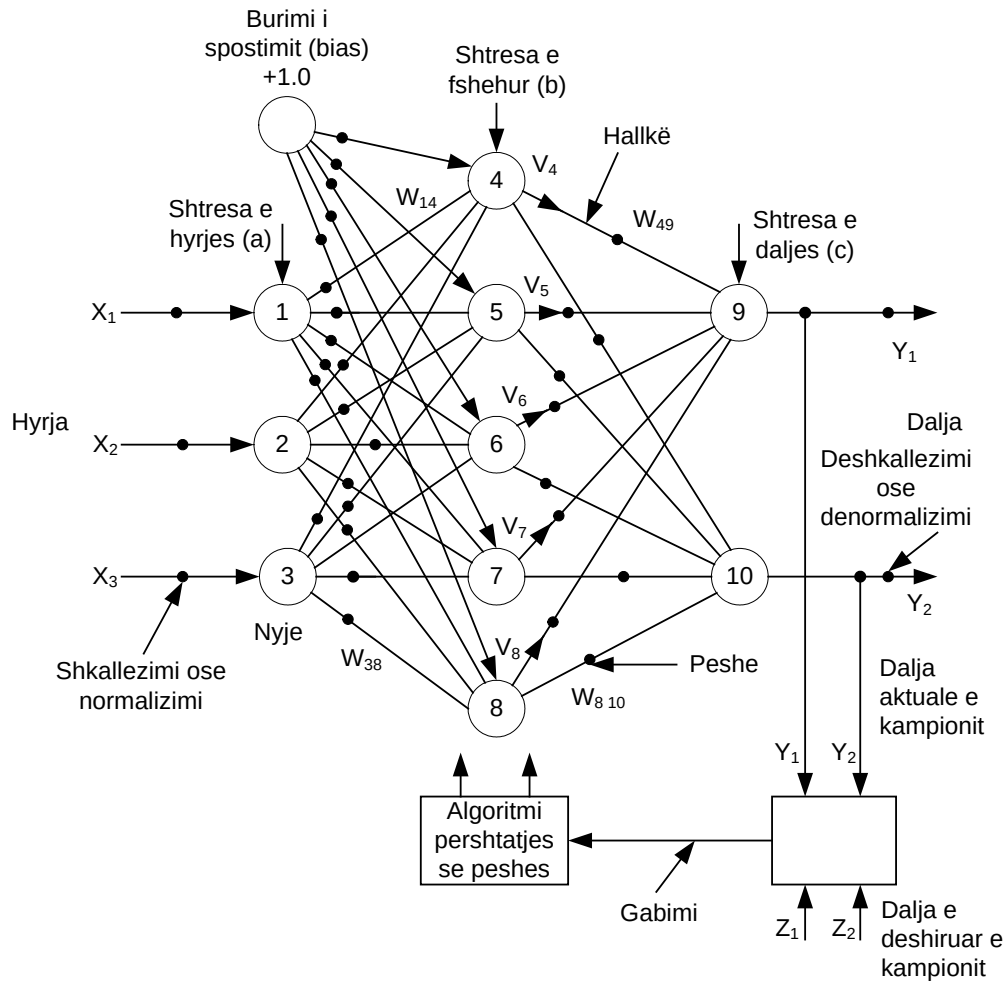


Figura 1.7 Struktura e një rrjeti neural me lidhje të drejtpërdrejtë (trajnimi me përhapje të pasme)

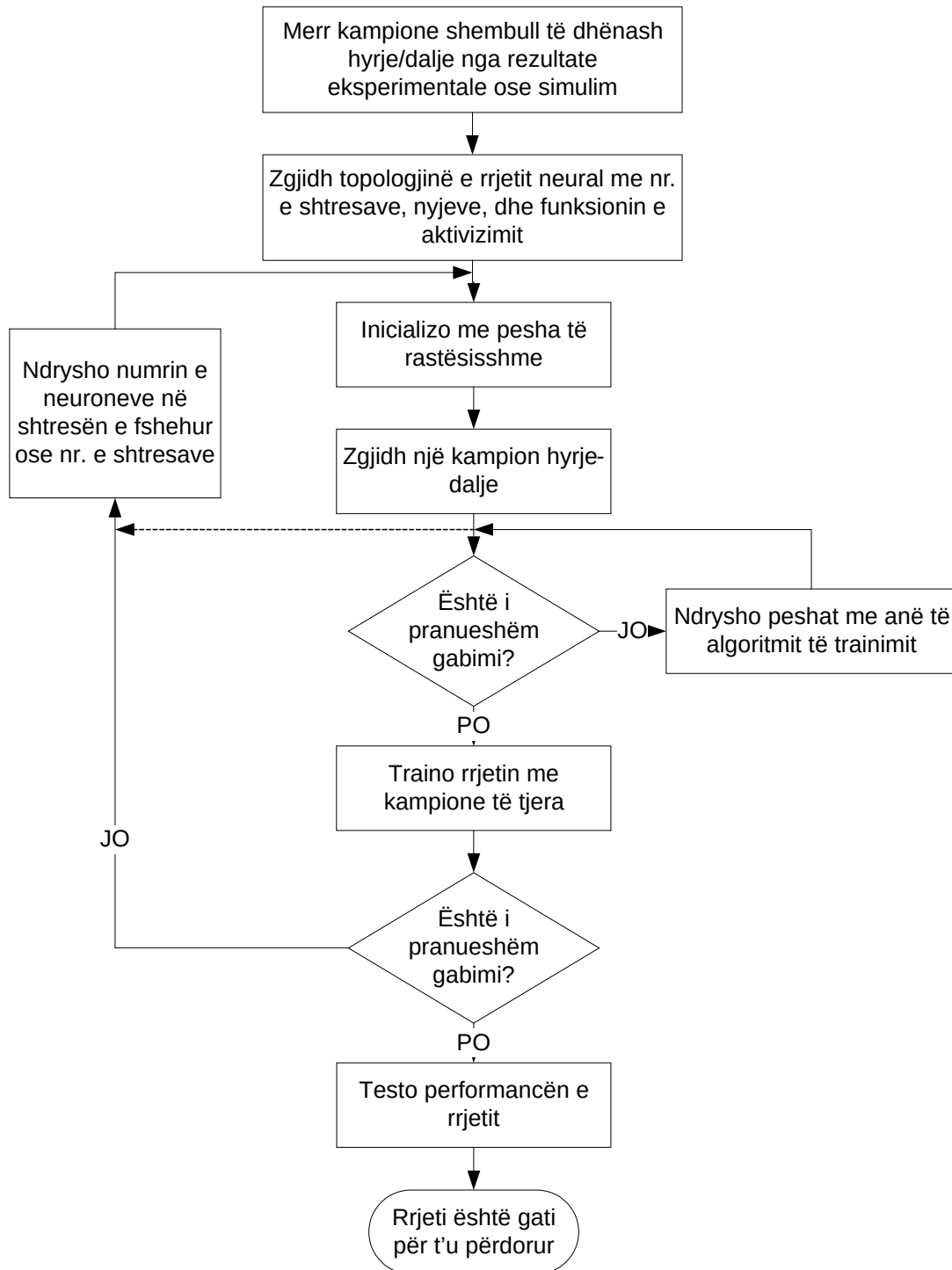


Figura 1.8 Blllokskema për trajnimin me përhapje të pasme të rrjetit neural me lidhje të drejtpërdrejtë

1.5.2 Algoritmi i përhapjes së pasme për rrjetin tre-shtresor.

Shqyrtojmë rrjetin tre-shtresor të përgjithësuar, treguar në Figurën 1.9, ku shtresa e parë ka n neurone (ose nyje), shtresa e fshehur ka p neurone, dhe shtresa dalëse ka q neurone. Spostimi (*bias*) i rrjetit nuk është treguar për thjeshtësi. Pranohet si funksion aktivizimi ai i tipit sigmoidal dhe për çdo neuron është përcaktuar nga blloku F . Për shtresën e parë nuk ka funksion aktivizimi. Vektorët e hyrjes dhe daljes në rrjet janë $X = [X_1 \dots X_k \dots X_n]$ dhe $Y = [Y_1 \dots Y_m \dots Y_q]$, ku $Y_1 = F_{1,3}$, $Y_m = F_{m,3}$ dhe $Y_q = F_{q,3}$.

1.5.2.1 Llogaritja e peshës për neuronet e shtresës së daljes

Fillimisht shqyrtohet problemi i llogaritjes të peshave për shtresën e daljes. Le të marrim një neuron m dhe derivojmë një shprehje matematikore për iterimin e peshës së tij hyrëse $W_{lm,3}$, që merr sinjalin $F_{l,2}$ nga neuroni l . Dalja e neuronit m , $F_{m,3}$, krahasohet me vlerën e dëshiruar ose objektivin D_m për të llogaritur gabimin ξ_m si

$$\xi_m = D_m - Y_m \quad (1.2)$$

ku $Y_m = F_{m,3}$. Funksioni objektiv për t'u minimizuar është përcaktuar si

$$\xi_m^2 = (D_m - F_{m,3})^2 \quad (1.3)$$

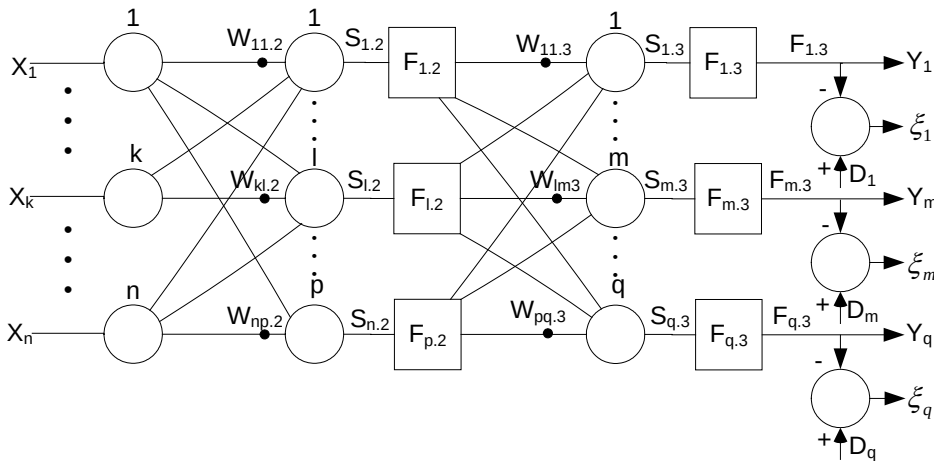


Figura 1.9 Rrjeti tre-shtresor i konceptuar për algoritmin e trajnimit me përhapje të pasme

Dalja $F_{m,3}$, dhe korresponduesja ξ_m^2 do të variojë me ndryshimin e peshës $W_{lm,3}$, me gjithë sinjalet e hyrjes që mbasin konstantë. Për të minimizuar ξ_m^2 me anë të metodës të gradientit, ndryshimi në peshë duhet të jetë proporcional me shkallën (raportin) e ndryshimit të katrorit të gabimit në lidhje me këtë peshë, domethënë,

$$\Delta W_{lm,3} = -\eta \frac{\partial \xi_m^2}{\partial W_{lm,3}} \quad (1.4)$$

ku η është konstantja e proporcionalitetit e përcaktuar si raporti (shkalla) i të mësuarit. Ndaj, pesha e re është dhënë si

$$W_{lm,3}(k+1) = W_{lm,3}(k) + \Delta W_{lm,3} \quad (1.5)$$

$$= W_{lm,3}(k) - \eta \frac{\partial \xi_m^2}{\partial W_{lm,3}} \quad (1.6)$$

ku $W_{lm,3}$ është pesha e vjetër dhe k është numri i iterimeve të përfshira. Derivati i pjesshëm në ekuacionin (1.4) mund të vlerësohet si vijon nga rregulli zinxhir i diferencimit:

$$\frac{\partial \xi_m^2}{\partial W_{lm,3}} = \frac{\partial \xi_m^2}{\partial F_{m,3}} \cdot \frac{\partial F_{m,3}}{\partial S_{m,3}} \cdot \frac{\partial S_{m,3}}{\partial W_{lm,3}} \quad (1.7)$$

ku çdo term mund të vlerësohet individualisht. Termi i parë ndiqet nga ekuacioni (1.3) si

$$\frac{\partial \xi_m^2}{\partial F_{m,3}} = -2(D_m - F_{m,3}) = -2\xi_m \quad (1.8)$$

Termi i dytë ndjek diferencimin e funksionit sigmoidal (si në Figurën 1.3d), që mund të derivohet në formën

$$\frac{\partial F_{m,3}}{\partial S_{m,3}} = \frac{\partial}{\partial S_{m,3}} \left(\frac{1}{1 + e^{-\alpha S_{m,3}}} \right) = \alpha F_{m,3}(1 - F_{m,3}) \quad (1.9)$$

Meqë në sinjalin $S_{m,3}$ kontribuojnë gjithë sinjalet e hyrjes të neuronit m , mund të shkruajmë

$$S_{m,3} = \sum_{l=1}^p W_{lm,3} \cdot F_{l,2} \quad (1.10)$$

ku p është numri i neuroneve në shtresën e mesit. Marrja e derivatit të tij të pjesshëm në lidhje me $W_{lm,3}$ jep

$$\frac{\partial S_{m.3}}{\partial W_{lm.3}} = F_{l.2} \quad (1.11)$$

që tregon që vetëm $F_{l.2}$ kontribuon në dalje kur ndryshon $W_{lm.3}$. Nga kombinimi i ekuacioneve 1.7 – 1.11 marrim

$$\frac{\partial \xi_m^2}{\partial W_{lm.3}} = -2\alpha(D_m - F_{m.3})F_{m.3}(1 - F_{m.3})F_{l.2} = -\delta_{lm.2} f_{l.2} \quad (1.12)$$

ose

$$\Delta W_{lm.2} = \eta \delta_{lm.3} F_{l.2} \quad (1.13)$$

ku

$$\delta_{lm.3} = 2\alpha(D_m - F_{m.3})F_{m.3}(1 - F_{m.3}) = 2\xi_m \frac{\partial F_{m.3}}{\partial S_{m.3}} \quad (1.14)$$

Pesha përshtatet në hapa në ekuacionin 1.5 me ndihmën e ekuacioneve 1.13 dhe 1.14, ku vlera e $F_{l.2}$ është e njohur dhe janë përcaktuar vlerat e η dhe α . Një procedurë e ngjashme është adaptuar për përshtatjen e gjithë peshave të shtresës së daljes. Figura 1.10 ilustron minimizimin sipas gradientit zbritës të ξ_m^2 nëpërmjet përshtatjes të peshës $W_{lm.3}$. Pika fillestare e operimit është 1, që korrespondon me peshën W_0 . Pesha rritet në hapa derisa pika e operimit shkon në 0, ku ξ_m^2 është në minimumin e tij. Koeficienti i të mësuarit η përcakton shpejtësinë e konvergjencës, ose, sa shpejt arrihet pika optimale. Normalisht, në fillim merret një vlerë e madhe e η , që më pas reduktohet gradualisht (shkalla adaptive e të mësuarit). Një nga problemet në optimizimin e peshës sipas gradientit zbritës është minimumi lokal (ose fals) i treguar në Figurën 1.10 nga kurba e pikëzuar U në vend të pikës së minimumit global. Për të kapërcyer këtë hendek, në të djathtë të ekuacionit (1.6) shtohet një term, që jepet si vijon:

$$W_{lm.3}(k+1) = W_{lm.3}(k) - \eta \frac{\partial \xi_m^2}{\partial W_{lm.3}} + \mu [W_{lm.3}(k) - W_{lm.3}(k-1)] \quad (1.15)$$

ku μ është faktori i shpejtësisë.

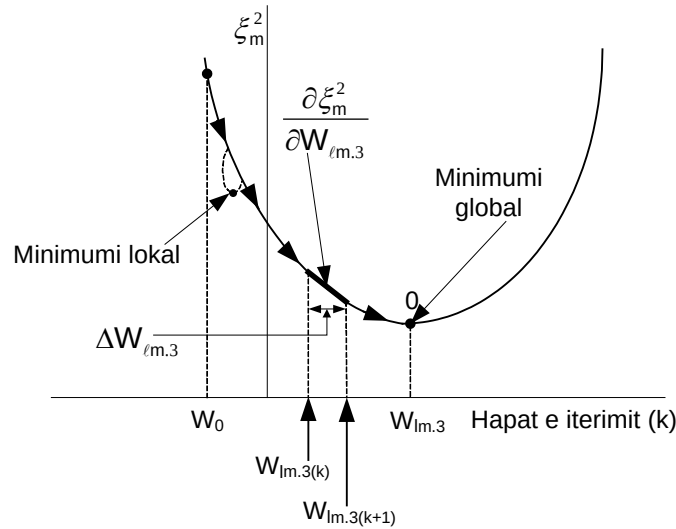


Figura 1.10 Minimizimi i gabimit kuadratik ξ_m^2 me anë të metodës të gradientit zbritës

1.5.2.2 Llogaritja e peshës për neuronet e shtresës së fshehur

Llogaritja e peshave për neuronet e shtresës së fshehur është më kompleks sepse nuk ka ndonjë vlerë direkte referuese (drejt të cilës synohet) për të llogaritur gabimet. Gabimi nga dalja duhet të përhapet mbrapsht, shtresë pas shtrese, duke përshtatur peshat për çdo shtresë në sekuencë. Procedura është në thelb si më parë, por më komplekse. Konsiderojmë problemin e llogaritjes të peshës $W_{kl.2}$ në shtresën e fshehur të Figurës 1.9. Shënojmë që kur ndryshohet $W_{kl.2}$, ndryshon dalja e gjithë neuroneve të shtresës dalëse. Prandaj, funksioni objektiv duhet të marrë në konsideratë shumën e gjithë gabimeve kuadratike, domethënë,

$$\xi^2 = \sum_{m=1}^q \xi_m^2 = \sum_{m=1}^q [D_m - F_{m.3}]^2 \quad (1.16)$$

Ekuacioni i rifreskimit të peshës mund të jepet si

$$W_{kl.2}(k+1) = W_{kl.2}(k) + \Delta W_{kl.2} \quad (1.17)$$

ku

$$\Delta W_{kl.2} = -\eta \frac{\partial \xi^2}{\partial W_{kl.2}} \quad (1.18)$$

ku η është shkalla e të mësuarit. Përsëri, nga diferencimi zinxhir, mund të shkruajmë

$$\frac{\partial \xi^2}{\partial W_{kl,2}} = \sum_{m=1}^q \frac{\partial \xi_m^2}{\partial F_{m,3}} \cdot \frac{\partial F_{m,3}}{\partial S_{m,3}} \cdot \frac{\partial S_{m,3}}{\partial F_{l,2}} \cdot \frac{\partial F_{l,2}}{\partial S_{l,2}} \cdot \frac{\partial S_{l,2}}{\partial W_{kl,2}} \quad (1.19)$$

Dy termat e parë të këtij ekuacioni janë respektivisht të njëjtë si ekuacionet 1.8 dhe 1.9. Meqë $S_{m,3}$ kontribuohet nga gjithë sinjalet hyrëse, ekuacionin 1.10 mund ta shkruajmë:

$$S_{m,3} = \sum_{l=1}^q W_{lm,3} \cdot F_{l,2}$$

ndaj,

$$\frac{\partial S_{m,3}}{\partial F_{l,2}} = W_{lm,3} \quad (1.20)$$

Dy diferencialët e fundit mund të shprehen, respektivisht, si

$$\frac{\partial F_{l,2}}{\partial S_{l,2}} = \alpha F_{l,2} (1 - F_{l,2}) \quad (1.21)$$

$$\frac{\partial S_{l,2}}{\partial W_{kl,2}} = X_k \quad (1.22)$$

Duke zëvendësuar ekuacionet 1.8, 1.9, 1.20, 1.21 dhe 1.22 në 1.19, marrim

$$\frac{\partial \xi^2}{\partial W_{kl,2}} = \sum_{m=1}^q (-2)\alpha (D_m - F_{m,3}) F_{m,3} (1 - F_{m,3}) W_{lm,3} \alpha F_{l,2} (1 - F_{l,2}) X_k \quad (1.23)$$

Ekuacioni 1.23 mund të shkruhet në formën

$$\frac{\partial \xi^2}{\partial W_{kl,2}} = - \sum_{m=1}^q \delta_{lm,3} W_{lm,3} \frac{\partial F_{l,2}}{\partial S_{l,2}} \cdot X_k \quad (1.24)$$

ku

$$\delta_{lm,3} = -2\alpha \xi_m F_{m,3} (1 - F_{m,3}) \quad (1.25)$$

Përfundimisht, nga ekuacionet 1.18 dhe 1.24, shprehja për ndryshimin e $W_{kl,2}$ jepet si

$$\Delta W_{kl,2} = -\eta X_k \sum_{m=1}^q \delta_{lm,3} W_{lm,3} \cdot \frac{\partial F_{l,2}}{\partial S_{l,2}} \quad (1.26)$$

ku vlerat e parametrave janë të njohura. Përsëri, gjithë peshat e shtresës së fshehur duhet të përshtaten nga një proces i ngjashëm. Kur trajnimi është plotësuar për një kampion hyrjeje, një procedurë e ngjashme duhet të përsëritet për gjithë kampionet në një mënyrë poshtë e lartë. Funkcioni objektiv grumbullues total për kampionet P , i cili duhet të minimizohet, jepet si:

$$\xi^2 = \sum_{p=1}^P \sum_{m=1}^q \xi_m^2$$

(1.27)

Kur ξ^2 është minimizuar në vlerën e dëshiruar, trajnimi i rrjetit neural është plotësuar dhe peshat mund të transferohen për zbatim praktik. Siç duket, trajnimi është një proces i vështirë dhe është e nevojshme ndihma e një kompjuteri. Vitet e fundit janë propozuar shumë versione të përmirësuara të algoritmit me përhapje të pasme që tentojnë të reduktojnë madhësinë e memories dhe të përshpejtojnë llogaritjen në kohë. Një nga ato është algoritmi Levenberg-Marquard (L-M), i cili përdoret në ambientin e Matlab-it në “Neural Network Toolbox”.

1.6 Arkitektura të Rrjetit Neural Artificial

Mënyra se si janë strukturuar neuronet e një rrjeti neural lidhet ngushtë me algoritmin e të mësuarit të përdorur për trajnimin e rrjeti. Në përgjithësi, mund të identifikohen tre klasa të ndryshme themelore të arkitekturave të rrjetit [1], [29], [31], [34], [35]:

1. Rrjetet një-shtresore me lidhje të drejtpërdrejtë

Në një rrjet neural të shtresëzuar, neuronet janë organizuar në formë shtresash. Në formën më të thjeshtë të një rrjeti të shtresëzuar, ka një shtresë hyrje të nyjeve burim që projektohet direkt në shtresën e daljes të neuroneve (nyjet llogaritëse), por jo e anasjellta. Me fjalë të tjera, ky rrjet është në mënyrë strikte një tip me lidhje të drejtpërdrejtë. Figura 1.11 ilustron rastin e katër nyjeve në të dy shtresat, të hyrjes dhe të daljes. Një rrjet i tillë quhet rrjet një-shtresor, me përcaktimin “një-shtresor” që i referohet shtresës së daljes të nyjeve llogaritëse (neuroneve). Shtresa e hyrjes e nyjeve burim nuk numërohet sepse në ‘të nuk kryhet ndonjë llogaritje.

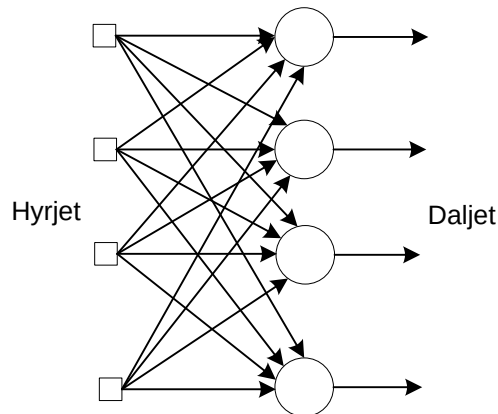


Figura 1.11 Rrjeti me lidhje të drejtpërdrejta me një shtresë neuronesh.

2. Rrjetet shumë-shtresorë me lidhje të drejtpërdrejta

Klasa e dytë e rrjeteve neurale me lidhje të drejtpërdrejta dallohet nga prania e një ose më shumë shtresave të fshehura, nyjet llogaritëse të cilave quhen neurone të fshehura ose njësi të fshehura; termi “i fshehur” i referohet faktit që kjo pjesë e rrjetit neural nuk shihet direkt nga hyrjet dhe daljet e rrjetit. Funkcioni i neuroneve të fshehur është të ndërhyjnë midis hyrjeve të jashtme dhe daljeve të rrjetit në mënyrë të dobishme. Duke shtuar një ose më shumë shtresa të fshehura, rrjeti mund të ekstraktojë statistika të rendit në të lartë nga hyrjet e tij. Në kuptim të përciptë, rrjeti kërkon një perspektivë globale pavarësisht nga lidhjet lokale, për shkak të grupeve shtesë të lidhjeve sinaptike dhe dimensionit shtesë të ndërveprimeve neurale [24].

Nyjet burim në shtresën e hyrjes të rrjetit furnizojnë elementet respektive të objektit të aktivizimit (vektori i hyrjes), që zëvendëson sinjalet e hyrjes të aplikuar në neuronet (nyjet llogaritëse) e shtresës së dytë (p.sh., shtresa e parë e fshehur). Sinjalet e daljes të shtresës së dytë janë sinjalet e hyrjes të shtresës së tretë, e kështu me radhë për pjesën tjetër të rrjetit. Neuronet e çdo shtrese të rrjetit kanë si hyrje të tyre vetëm sinjalet e daljes të shtresës pararendëse. Grupi i sinjaleve të daljes të neuroneve në shtresën e daljes të rrjetit jep përgjigjen e përgjithshme të rrjetit kundrejt kampionit të aktivizimit të furnizuar nga nyjet burim në shtresën e hyrjes (të parë). Figura 1.12 ilustron një rrjet neural shumë-shtresor me lidhje të drejtpërdrejta për rastin e një shtrese të fshehur. Për efekt paraqitje, rrjeti në Figurën 1.12 është referuar si rrjet 6–4–2, sepse ka 6 nyje burim (hyrje), 4 neurone të fshehur, dhe 2 neurone dalje. Një rrjet me lidhje të drejtpërdrejta me m nyje në hyrje, h_1 neurone në shtresën e parë të fshehur, h_2 neurone në shtresën e dytë të fshehur, dhe o neurone në shtresën e daljes referohet si një rrjet $m-h_1-h_2-o$.

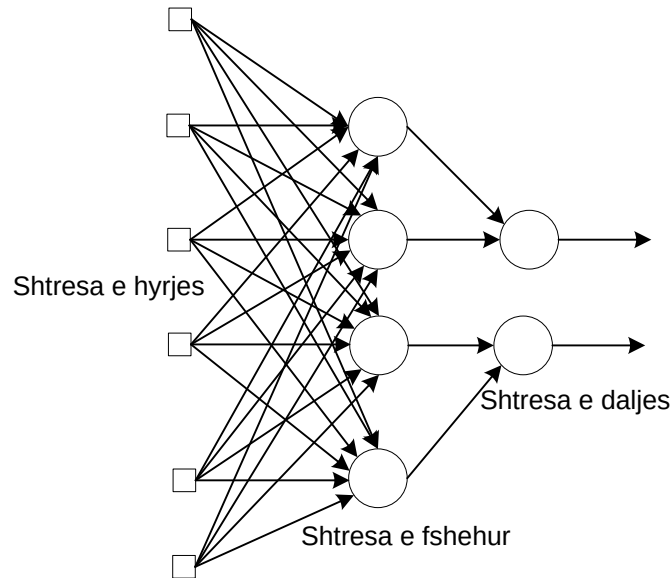


Figura 1.12 Rrjeti me lidhje të drejtpërdrejta plotësisht i lidhur me një shtresë të fshehur dhe një shtresë dalje

Rrjeti neural në Figurën 1.12 thuhet të jetë plotësisht i lidhur në kuptimin që çdo nyje në çdo shtresë është e lidhur me çdo nyje tjetër në shtresën pasardhëse. Thuhet që rrjeti është pjesërisht i lidhur nëse disa nga lidhjet e komunikimit (lidhjet sinaptike) mungojnë.

3. Rrjetet Rekurente

Një rrjet rekurent dallohet nga një rrjet me lidhje të drejtpërdrejta sepse ai ka të paktën një lak me lidhje të kundërt. Për shembull, një rrjet rekurent mund të përbëhet nga një shtresë e vetme neuronesh me çdo neuron që furnizon sinjalin e tij të daljes në hyrjen e gjithë neuroneve të tjerë, siç ilustrohet në arkitekturën në Figurën 1.13. [1]

Figura 1.14 ilustron një klasë tjetër të rrjeteve rekurentë me neurone të fshehur. Lidhjet e kundërta të treguara në Figurën 1.14 nisin nga neuronet e fshehur si dhe neuronet e daljes.

Prania e leqeve të lidhjes së kundërt, qoftë në strukturën rekurentë të Figurës 1.13 ose në atë të Figurës 1.14, ka një impakt të thellë në aftësinë e të mësuarit të rrjetit dhe në performancën e tij. Për më tepër, leqet e lidhjes së kundërt përfshijnë përdorimin e degëve të veçanta të përbëra nga elemente me vonësë në kohë (të përcaktuar nga z^{-1}), që rezultojnë në një sjellje dinamike jolineare, duke supozuar që rrjeti neural përmban njësi jolineare.

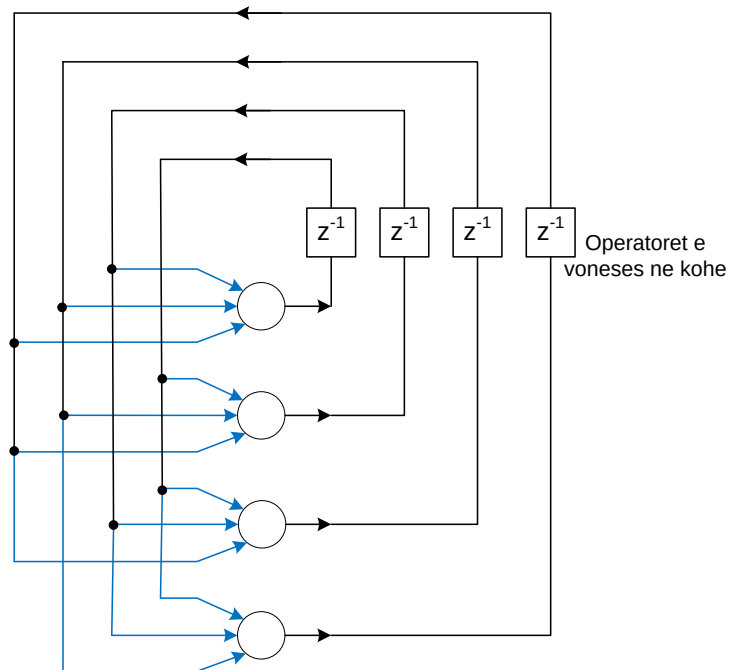


Figura 1.13 Rrjeti rekurent një shtresor

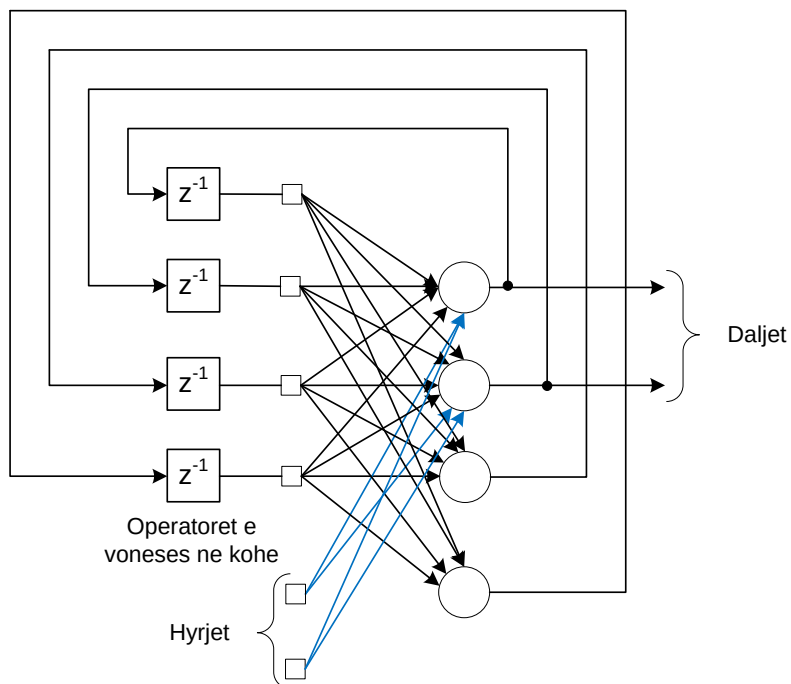


Figura 1.14 Rrjeti rekurent me shtresë të fshehur

Figura 1.15 tregon një strukturë të përgjithshme të një rrjeti rekurent dy-shtresor për një sistem dinamik ku sinjalet e daljes janë kthyer mbrapsht në shtresën e hyrjes me një vonesë në kohë. Në këtë rast, dalja e një neuroni varet jo vetëm nga sinjalet korrent të hyrjes, por gjithashtu edhe nga hyrjet e mëparshme, duke dhënë kështu sjellje të përkohshme të rrjetit. Kjo do të thotë që nëse, për shembull, një bashkësi e

sinjaleve funksion shkallë jepet në hyrje, përgjigja do pasqyrohet në rrafshin e kohës derisa arrihet një kusht në gjendje të vendosur. Kjo është e ngjashme me përgjigjen e një qarku R-L ose R-C. Shpesh ky rrjet është përcaktuar si një rrjet rekurent në kohë reale. Rrjeti i treguar në Figurën 1.15, ka një total prej N neuronesh dalëse dhe M lidhje të jashtme hyrjesh. Daljet janë lidhur mbrapsht si hyrje me një vonesë njësi në kohë. Kështu, gjithsej janë $M+N$ hyrje. $X(k)$ përcakton vektorin e hyrjeve të jashtme të aplikuar në rrjet në çastin e k -të diskret., dhe $Y(k+1)$ përcakton vektorin korrespondues të neuronit të daljes të krijuar një hap më vonë në çastin diskret $(k+1)$. Hyrja $X(k)$ dhe vektori i daljes $Y(k)$ me një shkallë vonesë përbëjnë së bashku vektorin total të hyrjes të dhënë si

$$\begin{aligned} U(k) &= [U_1(k) \dots U_{N+M}(k)]^T \\ &= [Y_1(k) \dots Y_N(k) \ X_1(k) \dots X_M(k)]^T \end{aligned} \quad (1.29)$$

Dalja e neuronit të j -të në çastin k jepet nga

$$Q_j(k) = \sum_{i=1}^{N+M} W_{ij}(k) U_i(k) \quad (1.30)$$

ku W_{ij} është pesha lidhëse ndërmjet hyrjes së i -të të neuronit të j -të dhe $U_i(k)$ është hyrja e i -të. Në formë matricore dalja është:

$$\begin{bmatrix} Q_1 \\ \vdots \\ Q_N \end{bmatrix} = \begin{bmatrix} W_{11} & \dots & W_{(N+M)1} \\ \vdots & \dots & \vdots \\ W_{1N} & \dots & W_{(N+M)N} \end{bmatrix} \begin{bmatrix} U_1 \\ \vdots \\ U_{N+M} \end{bmatrix} \quad (1.31)$$

Në hapin tjetër, dalja e neuronit të j -të është llogaritur duke kaluar sinjalin nga funksioni i aktivizimit $F(\cdot)$ si

$$Y_j(k+1) = F(Q_j(k)) \quad (1.32)$$

Ekuacionet 1.31 dhe 1.32 përbëjnë modelin dinamik të rrjetit.

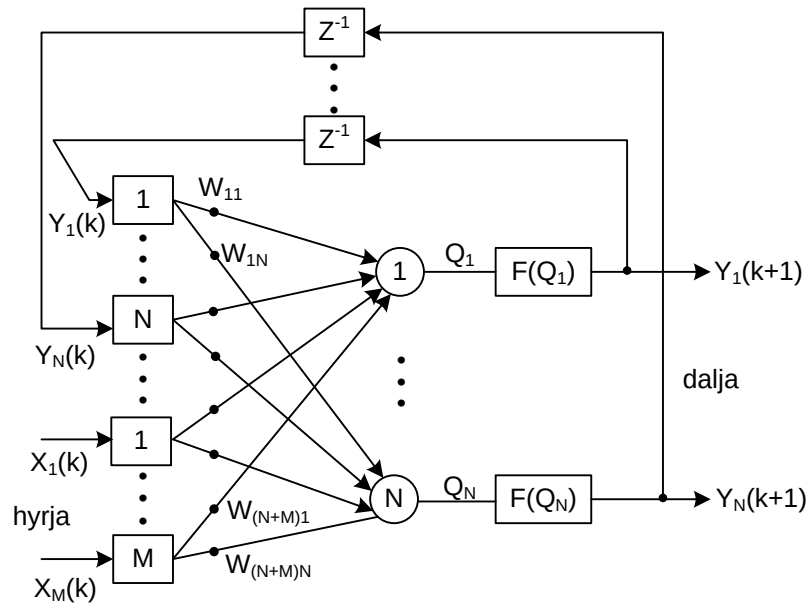


Figura 1.15 Struktura e përgjithshme e rrjetit neural rekurent për sistemin dinamik

1.7 Projektimi i rrjetit neural

Projektimi i strukturës të rrjetit neural përfshin përcaktimin e shtresave dhe neuroneve në çdo shtresë, dhe zgjedhjen e algoritmit të trajnimit.

Zgjedhja e parametrave të hyrjes të dobishëm për rrjetin neural është nga proceset më të vështira sepse [32], [36]:

- Mund të ketë ndërvarësi dhe dublikim midis parametrave
- Shpesh është më mirë të lëmë mënjatë disa parametra për të zvogëluar numrin e përgjithshëm të parametrave të hyrjes, dhe kështu të zvogëlojmë kompleksitetin llogaritës të problemit dhe arkitekturën e rrjetit
- Rrjeti neural zakonisht aplikohet në probleme ku nuk ka njohje të mirë të marrëdhënieve midis hyrjes dhe daljes, ndaj nuk është e qartë se cili prej parametrave të hyrjes mund të jetë më i vlefshmi.

Projektimi i parametrave të tjerë të arkitekturës të rrjetit neural, siç janë numri i neuroneve në shtresën e hyrjes, numri i shtresave të fshehura, numri i neuroneve në shtresat e fshehura dhe numri i neuroneve në shtresën e daljes, gjenden duke përsëritur disa cikle trajnimi të rrjetit, bazuar në metodën me prova dhe dështime. Kjo sepse nuk ka ndonjë metodë të qartë për zgjedhjen e arkitekturës dhe parametrave optimalë të rrjeteve neurale Megjithatë, disa punë kërkimore kanë kontribuar në përcaktimin e numrit të shtresave të fshehura, numrin e neuroneve në çdo shtresë dhe parametrave të tjerë. [37]–[39].

Numri i neuroneve në shtresën e hyrjes zgjidhet në varësi të dinamikës të objektit dhe të dhënave në hyrje. Këto variable zgjidhen në mënyrë të atillë që të dhënat e zgjedhura të përmbajnë informacion sa më të mirë për gjendjet e objektit. [31], [32], [40], [41].

Numri i neuroneve në shtresën daljes është i njëjtë me numrin e variableve që dëshirojmë të marrim si dalje të modelit [31], [32].

1.7.1 Përcaktimi i shtresave të fshehura

Përcaktimi i numrit të neuroneve në shtresat e fshehura dhe numri i neuroneve në secilës shtresë është një punë e konsiderueshme. Numri i shtresave të fshehura përcaktohet në fillim dhe është një hap kritik. Numri i shtresave të fshehura që nevojiten varet nga kompleksiteti i marrëdhënieve midis parametrave të hyrjes dhe daljes. Shumica e problemeve kërkojnë një shtresë të fshehur, dhe nëse marrëdhënia midis hyrjeve dhe daljes është lineare, rrjeti nuk ka nevojë për shtresë të fshehur. Vështirë se ndonjë problem praktik do kërkojë më shumë se dy shtresa të fshehura. Cybenko [42] dhe Bounds [43] sugjeruan që një shtresë e fshehur është e mjaftueshme për klasifikimin e kampioneve të hyrjes në grupe të ndryshme.

Chester [44] argumentoi që rrjeti me dy shtresa të fshehura duhet të funksionojë më mirë se një rrjet me një shtresë të fshehur. Më shumë se një shtresë e fshehur mund të jetë e vlefshme në arkitektura të caktuara, të tilla si korrelimi kaskadë [45] dhe të tjera. Një shpjegim i thjeshtë se përse rrjetet e mëdhenj mund të japin shpesh gabim të përgjithshëm më të vogël është që shkallët shtesë të lirisë mund të ndihmojnë konvergjencën; shtimi i parametrave shtesë mund të ulë shansin për të ngecur në një minimum lokal. Metodatat më të përdorura të trajnimit për rrjetet me përhapje të pasme bazohen në zvogëlimin e gradientit; domethënë gabimi zvogëlohet gjersa arrihet një minimum, që mund të jetë një minimum global ose lokal. Megjithatë, nuk ka ndonjë teori të qartë për të treguar numrin e shtresave të fshehura të nevojshme për të përafruar çdo funksion të dhënë. Rregulli i përgjithshëm për përcaktimin e numrit të këtyre shtresave është normalisht të fillohet me një shtresë të fshehur [46]. Nëse arkitektura me një shtresë të fshehur nuk jep rezultat të mirë, atëherë të provohet të rritet numri i neuroneve në shtresën e fshehur. Shtimi i shtresave të fshehura duhet të jetë mundësia e fundit.

1.7.2 Përcaktimi i numrit të neuroneve në shtresën e fshehur

Zgjedhja e numrit të neuroneve në shtresën e fshehur varet nga problemi. Për shembull çdo rrjet që kërkon ngjeshje të të dhënave duhet të ketë një shtresë të fshehur më të vogël se shtresa e hyrjes [47]. Një përafrim konservator është zgjedhja

e numrit të neuroneve të shtresës së fshehur midis numrit të neuroneve të shtresës së hyrjes dhe numrit të neuroneve të daljes. Gjithsesi ky përafrim është disi kontradiktor. Një rregull i pranueshëm është të fillohet me një numër sa gjysma e neuroneve të shtresës hyrjes dhe më pas ose rritet numri i neuroneve nëse gabimi i trajnimit është më i madh se gabimi i lejuar, ose zvogëlohet nëse gabimi i trajnimit bie shpejt nën gabimin e lejuar të trajnimit. Tabela më poshtë jep disa sugjerime për zgjedhjen fillestare të numrit të neuroneve në shtresën e fshehur.

Formula	Përshkrimi
$h = 2i + 1$	Hecht-Nelson [48] përdori teoremën e Kolmogorov sipas së cilës çdo funksion i I variableve mund të paraqitet nga superpozimi i një grupi prej $2i+1$ funksionesh me një variabël – për të derivuar kufirin e sipërm për numrin e
$h = (i + o) / 2$ $\frac{N}{10} - i - o \leq h \leq \frac{N}{2} - i - o$	Lawrence dhe Fredrickson [49] sugjeruan që vlerësimi më i mirë për numrin e neuroneve në shtresën e fshehur është gjysma e shumës të neuroneve të shtresës së hyrjes dhe të daljes. Më tej ata propozuan gamën e numrit të neuroneve shtresës të fshehur
$h = i \log_2 P$	Marchandani dhe Cao [50] propozuan këtë ekuacion për përcaktimin e numrit të neuroneve të shtresës së fshehur

Tabela 1.1 Disa rregulla për zgjedhjen e numrit të neuroneve të shtresës së fshehur
 h – numri i neuroneve të shtresës së fshehur,
 i – numri i neuroneve të shtresës së hyrjes,
 o – numri i neuroneve të shtresës së daljes.

1.7.3 Përcaktimi i numrit të iterimeve të trajnimit

Numri i grupit të të dhënave të trajnimit duhet zgjedhur me kujdes në mënyrë që të kemi një rrjet neural të trajnuar mirë.

Nuk ka ndonjë rregull të përgjithshëm për përcaktimin e numrit të iterimeve të trajnimit të një modeli neural. Megjithatë Lawrence dhe Fredrickson [49] sugjeruan rregullin e mëposhtëm:

$$2(i + h + o) \leq N \leq 10(i + h + o) \quad (1.33)$$

II. Zbatime të rrjeteve neurale

2.1 Zbatime të përgjithshme të rrjeteve neurale

Rrjetet neurale artificiale janë një nga teknologjitë e suksesshme në dy dekadat e fundit, zbatuar në një gamë të gjerë aplikimesh në fusha të ndryshme. Kjo ka ardhur si nevojë e përdorimit të metodave inteligjente për t'u përballur me faktorë të tillë si:

- rritja në mënyrë eksponenciale e sasisë të informacionit
- rritja e kompleksitetit
- kërkesat në rritje të konsumatorëve
- zhvillimet e teknologjisë
- kërkesat ndaj prodhuesve të të mirave materiale për respektimin e ligjeve dhe rregullave të vendosura nga entet e ndryshme rregullatore
- nevoja për mbajtjen optimale të kostove

Shtrirja e gjerë e aplikimeve ka qenë për shkak të vetive tërheqëse të rrjeteve neurale të cituara në kapitullin 1 si:

- Aftësia për të përafruar arbitrarisht funksione jolineare
- Ata mund të trajnohen lehtësisht duke përdorur të dhëna nga regjistrime të mëparshme të objektit nën studim
- Janë të aplikueshëm në sisteme me shumë variable
- Nuk kërkojnë specifikimet e marrëdhënieve strukturore midis të dhënave hyrje – dalje.

Duke konsideruar larminë e madhe të zbatimeve të rrjeteve neural, më poshtë tentohet të jepen disa prej tyre.

Inxhinieri kontrolli, energjitikë dhe robotikë: Këtu përfshihen një shumëllojshmëri aplikimesh si kontrolli jolinear, kontrolli i motorëve elektrikë, identifikim, monitorim dhe diagnostikim të defekteve të sistemeve, kontrolli i proceseve kimike, kontrolli i proceseve në industrinë ushqimore, etj. [51]–[56]. Siemens ka aplikime për diagnostikimin e hershëm të defekteve në sistemet energjitike dhe parashikimin e fluktuacionit të energjisë [57], [58]. Nasa ka raportuar aplikimin e rrjeteve neurale në kontrollin e aeroplanëve, aplikime në sistemet spaciale, dhe ka një patentë për projektimin aerodinamik me rrjete neurale [59]–[62]. Përdorimi i rrjeteve neurale është raportuar në teknologjitë e përdorura në epokën e re të automatizimit të industrisë “Industry 4.0” [63]–[71]. Në robotikë ka aplikime në kontrollin e pozicionit, orientim dhe gjetje drejtimi, njohje të objektit, të mësuarit e sjelljes, etj. [36], [72]–[74]

Industri Automobilistike: Prodhuesit e makinave janë ndodhur para kërkesave në rritje për uljen e ndotjes së ambientit dhe konsumit të karburantit për kilometër. Nga ana tjetër, është nevoja për të kënaqur kërkesat e konsumatorit për karakteristikat e makinës si lidhja me internet, GPS, dhe funksione të tjerat të avancuara. Kënaqja e këtyre kërkesave të ndryshme kërkon një kontroll preciz të motorit dhe monitorim të mirë të parametrave operacionalë të njësisë së fuqisë. Mbajtja e kostos optimale ndaj kërkesave të zgjeruara të kontrollit, si rrjedhim shtimin e nivelit të matjeve, ka rritur interesin për aplikimin e sensorëve virtualë pa qenë nevoja e shtimit të atyre fizikë. Aplikimet e rrjeteve neurale në këtë fushë janë me interes dhe janë aplikuar komercialisht në sistemin e kontrollit të motorit EMS (Engine Management System) dhe ECU (Engine Control Unit) [53], [75], [76]. Një aplikim tjetër në kontrollin e makinave është edhe sistemi i ndihmës së shoferit ADAS (advanced driver assistance systems) [77].

Industri tekstile dhe shkencë materiale: Këtu mund të përmenden aplikime në vlerësim të vetive mekanike të materialit, në gjeologji, kërkim nafte, mineralogji, analizë gjeokimike, sizmologji [41], [53], [78]–[80].

Mjekësi: Aplikime komerciale të rrjeteve neural në mjekësi kanë ndihmuar në përcaktimin e diagnozave të sëmundjeve të tilla si për shembull kancer, sëmundje të zemrës, etj. Kjo arrihet me anë të përpunimit, analizës dhe klasifikimit të të dhënave mbi pacientin, simptomat, analizave biokimike, imazheve radiografike, sinjaleve të marra nga EKG etj. [81]–[85], [85]–[88]

Ekonomi dhe financë: rrjetet neurale kanë gjetur zbatim të gjerë dhe në këto fusha. Mund të përmenden parashikimet në luhatjet e tregut të aksioneve (*stock market*); vlerësimet në dhënien e huave; parashikimin e falimentimeve të kompanive; vlerësim të pronave; gjetje të mashtrimeve financiare në taksa, transaksione, deklarime të sigurimeve; gjetje të mashtrimeve të kartave të kreditit; parashikimi i çmimeve të materialeve, komoditeteve, produkteve; parashikimi i treguesve ekonomikë. [52], [89]–[91]

Kërkim të dhënash, njohje dhe klasifikim objekti: njohja e fytyrës, njohja e shkrimit; njohja e zërit; njohja e objektit; kthimi i fjalëve të folura në të shkruara; janë një pjesë e vogël e shembujve në këtë kategori. Facebook për shembull ka aplikuar rrjetet neurale për njohjen e figurës në foto dhe klasifikimin e tyre në grupe të ndryshme, duke e përdorur këtë si mjet për sugjerimin e lidhjeve të ndryshme midis personave që kanë llogari në facebook [92]. Google i ka aplikuar rrjetet neurale në pakësimin e spam-ve në email-e, klasifikimin e të dhënave në motorin e kërkimit, dhe në një veti të Gmail për ofrimin e fjalive të gatshme gjatë kthimit të përgjigjes të një email-i. Google gjithashtu është aktiv në kërkimet për aplikimin e rrjeteve neurale me anë të

programit të kërkimeve të kompanisë DeepMind, që u ble nga Google në 2014 [93], [94].

2.2 Implementimi software dhe hardware i NN

Implementimi i rrjeteve neurale dhe mënyrat e zbatimit më së shumti bazohet në motivimet dhe pyetjet:

- A mundet që një rrjet neural të zëvendësojë teknologjitë ekzistuese në një fushë ku përmirësime të vogla të performancës mund të rezultojnë me ndikim të madh ekonomik
- A mundet që një rrjet neural të përdoret në një fushë ku teknologjitë ekzistuese kanë treguar të jenë të papërshtatshme për të bërë një sistem të shfrytëzueshëm
- Cilat janë arkitekturat më të përdorshme për implementimin e rrjeteve neurale në kohë reale
- Cili është tipi dhe kompleksiteti i rrjetit
- Cilat janë kushtëzimet kohore (vonesa, frekuenca, etj.)
- A nevojiten veti shtesë (të mësuarit në-linjë etj.)
- A duhet të implementohet rrjeti neural në një mjedis të veçantë (afër sensoreve, aplikime të integruara që kërkojnë pak konsum, etj.)
- Kur është nevoja e qarkut neural

Zgjidhjet që ekzistojnë janë:

- Arkitektura të përgjithshme
- Neuro-Hardware Specifik
- Qarqe të dedikuara

Më tej, implementimi i rrjeteve neurale mund të ndahet në tre kategori:

- Simulim software,
- Emulim (hardware),
- Implementim hardware.

Ndryshimi midis dy kategorive të fundit nuk është i ndarë qartësisht.

Do të përdorim termin simulim për të përshkruar paketat software që mund të ekzekutohen në makina mbartëse të ndryshme. Software-t neurale përdoren për të simuluar, kërkuar, zhvilluar dhe aplikuar rrjetet neurale artificiale, rrjetet neurale biologjike dhe në disa raste një fushë më të gjerë të sistemeve adaptive.

Ndër paketat software mund të përmenden MATLAB Neural Net Toolbox, NeuralWare, NeuroSolutions, Alyuda NeuroIntelligence, DTREG, SPSS Neural Connection, NeuroXL, Synapse, Nestor, etj.

Implementimit të rrjeteve neurale në makina multi-procesore me qëllim të përgjithshëm do t'i referohemi si emulim. Shembuj të arkitekturave hardware të përgjithshme janë mikroprocesorët konvencionalë si p.sh., Sun Microsystems.

Termin implementim hardware do ta përdorim për neuro-çipet dhe për pajisjet që janë projektuar posaçërisht për rrjetet neurale.

2.2.1 Implementimi hardware i rrjeteve neurale

Hardware-i neural është zhvilluar në mënyrë shumë të shpejtë në dy dekadat e fundit. Në ndryshim nga arkitektura konvencionale von-Neumann që është sekuenciale, rrjetet neurale artificiale kanë natyrë procesimi paralele. Janë projektuar një larmi hardware-sh për të përdorur natyrën paralele të modeleve të rrjeteve neurale. Pavarësisht rritjes të jashtëzakonshme të fuqisë llogaritëse të procesorëve me qëllim të përgjithshëm, hardware-i neural është premtues në disa aplikime specifike, të tilla si njohja e objektit, procesimi i imazhit, etj.

Hardware-t neurale janë përcaktuar si pajisje të projektuara sipas natyrës paralele të rrjeteve neurale artificiale, pra të projektuara për të implementuar arkitekturat neurale dhe algoritmet e të mësuarit.

Për klasifikimin e neurokompjuterave merren në konsideratë këta faktorë:

- Tipi i sinjalit të përdorur në rrjet
- Implementimi i peshave
- Integrimi dhe funksionet e daljes të njësive

Sinjalet e transmetuara në rrjet mund të jenë analogë ose dixhitalë. Në përafrimin analog, një sinjal paraqitet nga madhësia e rrymës ose diferenca në tension. Në përafrimin dixhital, ruhen dhe transmetohen vlera diskrete. Neurokompjuterat hibridë janë kombinim i qarqeve analoge dhe dixhitale. Përmbledhtazi, klasifikimi e neuro-hardware sipas tipit të sinjalit do ishte [34], [95]:

- Analog
 - o Komponente elektronike
 - o Komponente Optike
- Dixhital
 - o Multiprocesor von-Neumann
 - o Procesorë vektorialë

- o Matrica sistolike
- o Projektme speciale
- Hibridë
 - o Kombinim i qarqeve analoge dhe dixhital

Klasifikime të tjera të tjera të neuro-hardware janë: [95]–[97]

- Klasifikimi sipas modeli të komunikimit ndër-neuron
 - o Arkitekturë e shtresëzuar
 - o Arkitekturë e rikonfigurueshme
 - o Rrjet plotësisht i lidhur
 - o Rrjet i lidhur lokalisht
- Klasifikimi sipas tipi të arkitekturës
 - o Qark me kontrollor në-qark
 - o Neurokompjuter stand-alone
 - o Karta përshpejtuese
- Klasifikimi sipas shkallës të paralelizmit
 - o Paralelizëm i ulët (më pak se 10-15 procesorë)
 - o Paralelizëm i mesëm ($10-10^2$ procesorë)
 - o Paralelizëm i lartë (10^2-10^3 procesorë)
 - o Paralelizëm masiv (më shumë se 10^3 procesorë)
- Klasifikimi sipas tipit të të mësuarit
 - o *Të mësuarit në-qark*: përdor vetëm qarkun fizik për të kryer të mësuarit. Megjithëse është më e ngadaltë dhe më pak e saktë se dy metodat e tjera për llogaritjet e peshave, kjo metodë nuk përfshin manipulime të jashtme në fazën e të mësuarit. Kjo e bën më shumë realiste për implementimet hardware, megjithatë projektimi me këtë metodë është më shumë kompleks dhe më pak fleksibël, sepse algoritmi i të mësuarit duhet të implementohet në hardware.
 - o *Të mësuarit jashtë-qarku*: faza e të mësuarit kryhet në një rrjet të simuluar në software. Kjo lejon llogaritjet e trajnimit të kryhen më shpejtë dhe më shumë saktësi mund të arrihet duke përdorur rrjetin neural në hardware. Megjithatë, ndryshesat gjatë fabrikimi të hardware nuk merren parasysh.
 - o *Të mësuarit me-qark-në-lak*: përfshin të dyja rrjetin hardware dhe llogaritjet në një software të jashtëm. Kjo metodë mbështetet në software për algoritmin e të mësuarit por përdor hardware për të kryer llogaritjet. Për shembull, në rastin e algoritmit me përhapje të pasme, hapi i kalimit të drejtpërdrejtë kryhet në hardware kurse përditësimi i peshave në software. Kështu, saktësia e llogaritjeve nuk kufizohet nga cilësitë e hardware-it, ndërsa merret parasysh sjellja aktuale e rrjetit.

Sfidat për implementimin hardware të rrjeteve neurale rrjedhin nga elementet bazë të këtyre rrjeteve [95]–[97]:

- Lidhjet sinaptike
 - o Lidhjet e rrjetit: numri i lidhjeve sinaptike shkallëzohet në mënyrë kuadratike në lidhje me numrin e neuroneve
 - o Peshat sinaptike: peshat duhet të përcaktohen me saktësi të madhe në mënyrë që të sigurohet konvergencë e duhur e algoritmit të të mësuarit.
 - o Të mësuarit: peshat sinaptike duhet të jenë të rifreskueshme
- Neuroni
 - o Gjendja e neuronit: duhet kryer shuma e peshave të hyrjes
 - o Funkzioni i aktivizimit: llogaritje me shkallë të lartë jolineariteti

Me rritjen e madhësisë të rrjetit neural artificial, numri i lidhjeve sinaptike rritet në mënyrë kuadratike për një rrjet plotësisht të lidhur, që praktikisht bëhet e pamundur për të kryer gjithë lidhjet. Teknikat e sotme të fabrikimit të qarqeve të integruara janë në thelb struktura dy dimensionale, që kufizojnë mundësinë e një ndërlidhje të plotë. Ndaj shumica e implementimeve kufizohet në lidhjen e disa fqinjëve të çdo neuroni. [97]

Bloqet përbërës të një neuro-hardware ose neurokompjuter mund jepen të skematikisht si në figurën 2.1

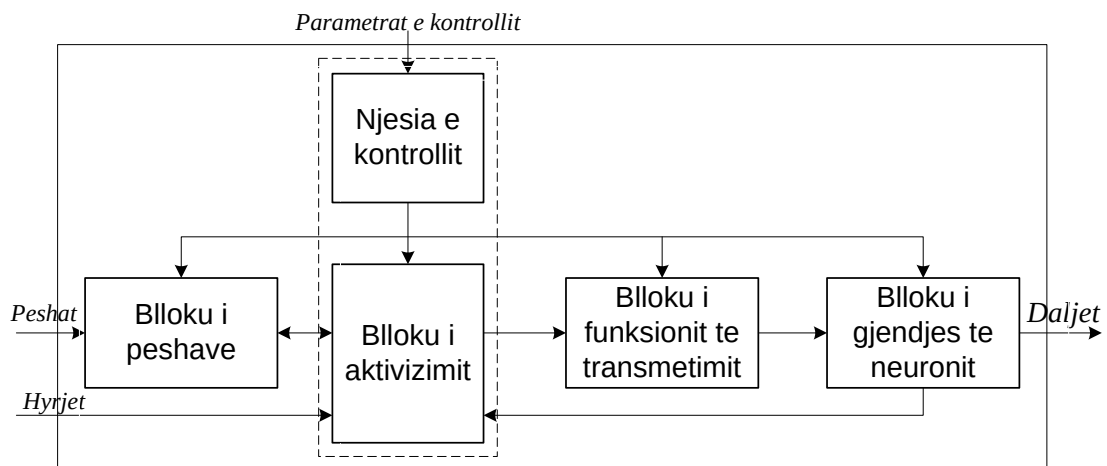


Figura 2.1 bllokskema e paraqitjes të elementeve të qarqeve neuro dhe neurokompjuterave [96], [98]

Referuar klasifikimit sipas tipit të sinjalit mund të përmenden këto veçori:

Qarqet dixhital: Rrjetet neurale artificiale mund të implementohen duke përdorur qarqet dixhitale CMOS [77], [99]–[111], që kanë avantazhin e të qenit të lehtë për t’u projektuar dhe ndërtuar. Kjo mbështetet në elementet logjike ekzistues

dhe shfrytëzon avantazhin e njohurive të fituara në dekadat e zhvillimit të qarqeve dixhitale.

Peshat sinaptike mund të implementohen duke përdorur qelizat dixhitale të memories ose flip-flop. Numri i bit-eve të përdorur për ruajtjen e peshave sinaptike është kritik në saktësinë e algoritmit, veçanërisht gjatë fazës së mësim. Kjo mund të përmirësohet në disa raste duke përdorur të mësuarit jashtë-qarku [34], [96], [97] Shuma e gjendjes të neuronit mund të implementohet duke përdorur pragjet shumatore dhe shumëzuese. Megjithatë, për numër të madh të neuroneve që japin sinjale në hyrje të secilit neuron, numri total i këtyre elementeve mund të rritet e të jetë jo i neglizhueshëm.

Implementimi i funksionit të aktivizimit është më i komplikuar për t'u implementuar, duke konsideruar që është jolinear. Një funksion i vetëm kufizues mund të implementohet lehtë, por ajo çfarë ky funksion ofron është shumë e kufizuar. Nga ana tjetër, një funksion aktivizimi më kompleks, si p.sh., funksioni sigmoidal, ul ndjeshëm fuqinë llogaritëse dhe kërkon fuqi dhe hapësirë të konsiderueshme nëse dëshirohet saktësi e mirë.

Megjithëse adaptimi i logjikës dixhitale CMOS shpie në projekte të thjeshta, rezultati është në thelb jo i optimizuar për fuqinë që përdoret dhe sipërfaqen që zë qarku. Megjithatë, një avantazh i implementimeve CMOS është që këto mund të integrohen lehtësisht me qarqet standarte. Një implementim i tillë jepet në figurën 2.2.

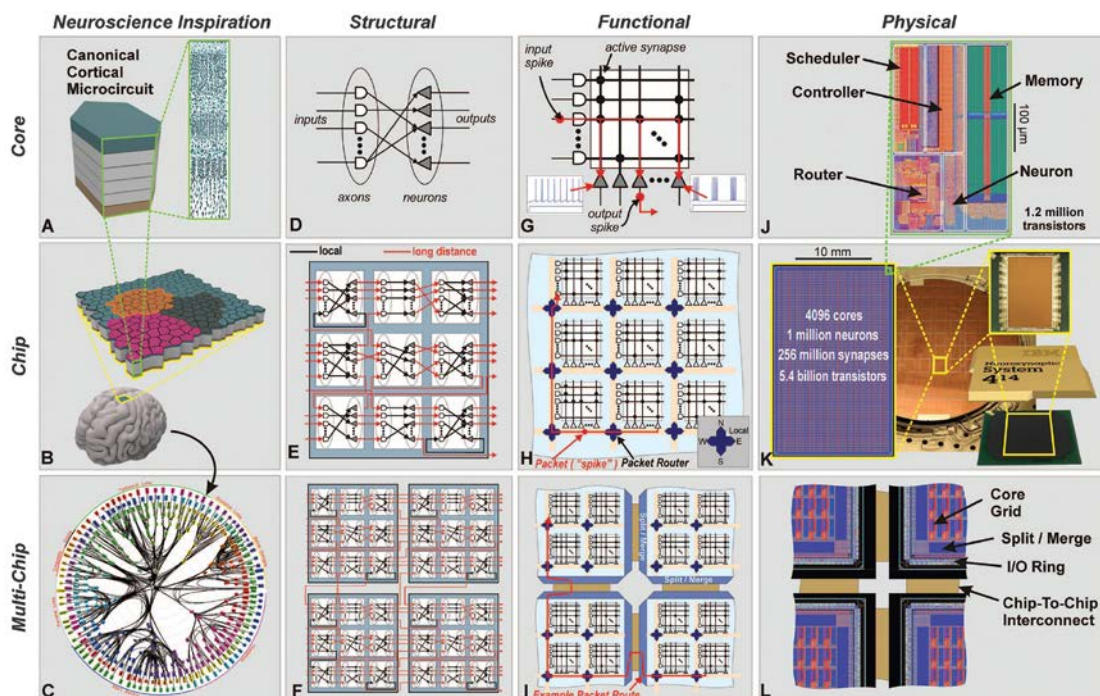


Figura 2.2 Arkitektura e qarkut TrueNorth të IBM [106]

Llogaritjet në qarkun TrueNorth kryhen në bërthamën lokale, por vlerat e daljes të një neuroni mund shpërndahen në çfarëdo neuron në qark. Për të ulur konsumin e energjisë, vetëm vlerat e daljes të çdo neuroni komunikohen; kjo rezulton në fuqinë aktive proporcionale me aktivitetin e secilit neuron. [106]

Karta IBM e prodhuar me këto qarqe jepet në figurën 2.3. Kjo kartë ka 16 milion neurone dhe 4 bilion sinapse. [112]

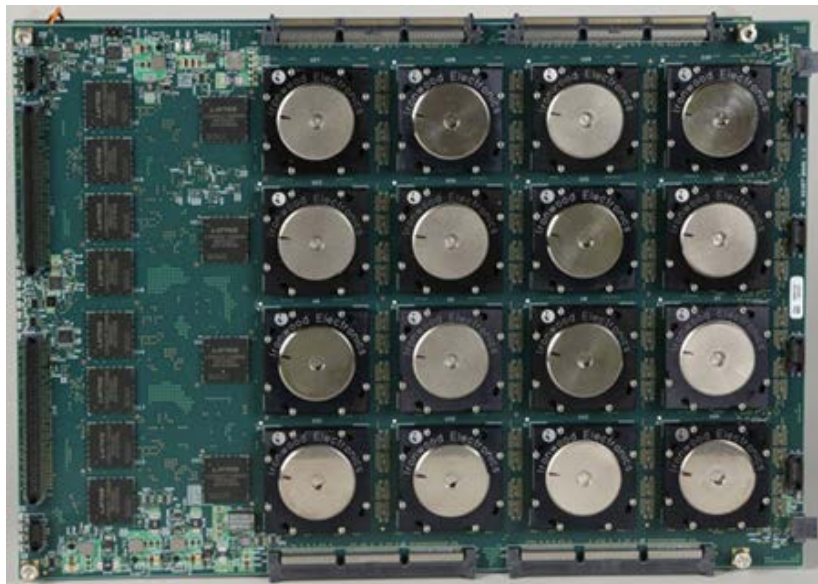


Figura 2.3 Karta IBM Synapse me 16 qarqe [112]

Qarqet analoge: implementimet analoge të rrjeteve neurale janë më së shumti ato në teknologjinë VLSI/CMOS.[34], [95]–[98], [109], [113, pp. 101–105] Projektimi i qarqeve analoge është më i komplikuar se ai i qarqeve dixhitale. Këto qarqe konsiderohen kur kërkohet konsum i ulët energjie, madhësi e vogël qarku dhe/ose shpejtësi e madhe; për aplikime të caktuara qarqet analoge mund të jenë shumë më eficientë se qarqet dixhitale. [114], [115]

Peshat sinaptike mund të implementohen duke ruajtur një peshë dixhitale, siç bëhet për sinapsët dixhitale. Në këtë rast, kryerja e llogaritjeve do kërkojë elementë konvertues analog/dixhital (A/D) dhe dixhital/analog (D/A). Kjo jo vetëm që do paraqesë vonesë të padëshiruar në llogaritje, por fuqia dhe përshkallëzimi i sipërfaqes do varen nga saktësia në peshat sinaptike. Mënyrë alternative është të ruhen peshat duke përdorur elemente analogë, si rezistorë ose kondensatorë. Mënyrë tjetër është të ruhen peshat e programueshme në mënyrë dixhitale në elemente të ri-konfigurueshëm analogë. Kjo do kërkonte shndërrues A/D dhe D/A për hapin e të mësuarit (për shembull rifreskimi i peshave).

Shumatorja e hyrjeve të neuronit zakonisht nuk është problem për qarkun analog, sepse hyrjet dhe daljet e neuronit do të paraqiten ose si rrymë ose si tension. Rrymat mbledhen duke bashkuar degët në paralel, ndërsa tensionet duke bashkuar degët në seri; në të dy rastet nuk kërkohen elemente shtesë për të kryer llogaritjet.

Funksioni i aktivizimi zakonisht implementohet duke përdorur një amplifikator, që paraqet jolinearitete në regjim të ngopur. Funksionet kryesore të aktivizimi mund të ndërtohen duke përdorur qarqe relativisht të thjeshtë.

Variacionet e prodhimit të qarqeve analoge kufizon arritjen e saktësisë të rrjeteve neurale analoge. Kjo do të thotë që për të arritur një vlerë të caktuar të saktësisë, qarqet duhet të bëhen mjaftueshëm të mëdhenj për të kufizuar efektin e tolerancës të prodhimit. [34], [96], [97]

Teknologji të tjera hardware: implementimi i rrjeteve neurale artificiale duke përdorur proceset CMOS në të dy rastet, dixhitale dhe analoge, ka të meta të konsiderueshme, veçanërisht kur rrjetet tentohen të shkallëzohen në numër të madh neuronesh dhe sinapsesh. [97] Siç u përmend më sipër, rritja e numrit të neuroneve rrit numrin e lidhjeve sinaptike dhe bën të pamundur lidhjen e gjithë neurone në rrjet. Zhvillimet e reja të teknologjisë janë premtuese për implementimin hardware të rrjeteve neurale me numër shumë të madh elementesh. Ndër këto është memristori, që është një nanodevice. [116]–[118] Për implementimin në hardware të një sinapsi të vetëm nevojiten shumë transistorë në teknologjinë CMOS. Në kontrast, një sinaps mund të paraqitet nga një memristor i vetëm [119]. Rrjedhimisht, numri i neuroneve dhe lidhjeve sinaptike që mund të implementohen në hardware me memristorë është shumë më i madh sesa në teknologjitë ekzistuese në silikon.

Në 1975, Gordon Moore, një nga themeluesit e Intel, formuloi ligjin Moor [120], [121], sipas të cilit numri i transistorëve në një qark elektronik dyfishohet afërsisht çdo dy vjet. Gjer më tani ky ligj është provuar të jetë i saktë dhe ka vendosur shpejtësinë për rritje konstante të fuqisë llogaritëse. Por kjo prirje mund të mbarojë shpjet, sepse ka një kufi të numrit të transistorëve që mund të pakëtohen në një qark. Ndaj është e pashmangshme që një ditë do ketë zhvendosje nga teknologjia e silikonit. Memristori, që është pothuaj gati për daljen në masë në treg, është një kandidat për këtë hap kritik. [122]–[124]

Termi memristor (memory resistor – rezistencë memorie) është një term i përdorur për herë të parë në 1971 nga Leon Chua [125], si një element i ri qarku dy-terminalësh, i karakterizuar nga një marrëdhënie midis ngarkesës elektrike dhe fluksit magnetik.

Sipas marrëdhënieve matematikore, memristori do operonte në këtë mënyrë: rezistenca elektrike e tij nuk është konstante por varet nga historia e rrymës që ka kaluar më parë në qark, psh rezistenca aktuale varet sa ngarkesë elektrike ka kaluar në një drejtim të caktuar në të shkuarën. Qarku kujton historinë e tij: kur burimi i energjisë elektrike fiket, memristori kujton rezistencën e tij të fundit derisa burimi i energjisë ndizet përsëri.

2.2.2 Implementimi software kundrejt atij hardware

Rritja e shpejtësisë dhe kapacitetit të hardware konvencional me procesorë von-Neuman, ka sjellë dhe zhvillimin e madh të paketave software për aplikimin e rrjeteve neurale. Ka një sasi të konsiderueshme programesh dhe simulatorësh, ku përfshihen programe për demonstrim dhe studim, paketa të specializuara për modelim neuro, mjete të përgjithshme modelimi për analizë të detajuar, programe specifike për aplikime të ndryshme për shembull në industri. Si shembull, [113, pp. 776–780] ka një listë dhe vështrim të përgjithshëm në programet neuro.

Përdorimi i GPU në paralel me CPU është një ambient hardware me qëllim të përgjithshëm, që ka gjetur zbatim në përmirësimin e shpejtësisë të implementimit të paketave neuro-software. [92]

Shumica e projektuesve të zgjidhjeve me rrjete neurale, gjejnë implementimin software në makina me qëllim të përgjithshëm, më të përshtatshëm kur nuk kërkohet shpejtësi jashtëzakonisht e madhe. Implementimi software ka fleksibilitetin e ndryshimit të shpejtë të parametrave dhe është i një qëllimi më të përgjithshëm.

Megjithatë, procesorët sekuencialë nuk mund të japin përgjigje dhe trajnim shumë të shpejtë në kohë reale për rrjetet me numër shumë të madh neuronesh dhe sinapsesh. Procesimi paralel me elemente paralele procesimi, mund të sigurojë shpejtësi shumë të madhe. Disa aplikime të specializuara, si për shembull ato për njohjen e të folurit, përdorin hardware neuro për këtë veti të tyre. [96]

Pra projektuesit zgjedhin dy forma të implementimit:

- implementim software në hardware të përgjithshëm
- implementim hardware të rrjetit neural nëse ka kërkesa specifike për shpejtësinë e përpunimit. Implementimi hardware mund të jetë i pjesshëm ose i plotë; në varësi të metodës të zgjedhur për trajnimin e rrjetit, trajnim në-qark ose jashtë-qarku, siç u përmend më sipër në 2.2.1

2.3 Rrjetet neurale në identifikim dhe kontroll

Nevoja për të kënaqur kërkesat e kontrollit në sistemet e kontrollit, dinamike dhe me kompleksitet në rritje, nën të panjohura të konsiderueshme, bën përdorimin e rrjeteve neurale në këto sisteme shumë tërheqës. Arsyeja për këtë janë vetitë e tyre të përmendura më parë, si aftësia e tyre për të mësuar të përafrojnë funksione, klasifikojnë objekte dhe potenciali për t'u implementuar masivisht në hardware në mënyrë paralele. Me fjalë të tjera, rrjetet neurale mund të implementojnë, si në software ashtu dhe në hardware, shumë funksione thelbësore të kontrollit të sistemeve.

Rrjetet neurale janë përdorur në kontrollin e proceseve jolineare dhe/ose Shumë-Hyrje-Shumë-Dalje për shkak të aftësisë së tyre për të mësuar marrëdhëniet

funksionale komplekse jolineare. Aplikimi i rrjeteve neurale në identifikim dhe kontroll është rritur në mënyrë eksponenciale gjatë viteve të fundit. [30], [31], [36], [51]–[54], [76], [80], [126]

Zakonisht, në shumë probleme të kontrollit të procesit, modelet e sistemit nuk janë të mirëpërcaktuar; ato ose mungojnë ose parametrat e sistemit mund të ndryshojnë me kohën. Rrjetet neurale janë të volitshëm për të përfutur modelet hyrje-dalje të sistemeve sepse ata mund të imitojnë sjelljen e sistemit pasi janë trajnuar. Edhe nëse modeli ose identifikimi neural mund të ketë mospërputhje në fillim me objektin, rrjeti neural përmirësohet gjithnjë e më shumë me proceset e trajnimit në-linjë. Për më tepër, trajnimi në-linjë bën që modeli neural të reagojë direkt ndaj parametrave të ndryshueshëm në kohë të objektit. [36]

Meqë proceset janë zakonisht shumë komplekse, rrjetet neurale përdoren së bashku me teknikat konvencionale të kontrollit siç janë PI dhe PID, ose me teknikat më të fundit bazuar në sistemet eksperte, logjikën fuzzy. Përdorimi në mënyrë hibride përmirëson performancën e përgjithshme të sistemit të kontrollit.

Më të përdorurit në aplikimet e sistemeve të kontrollit janë rrjetet neurale shumëshresorë me lidhje të drejtpërdrejta dhe trajnim të mbikëqyrur. Vetia kryesore e këtyre rrjeteve është që mund të gjenerojnë hartëzim hyrje-dalje dhe mund të përafrojnë çdo funksion me një saktësi të dëshiruar. Rrjetet neurale janë përdorur përgjithësisht në sistemet e kontrollit për identifikim dhe kontroll. [31], [36], [127]

Në identifikimin e sistemit, për të modeluar sjelljen hyrje-dalje të një sistemi dinamik, rrjeti trajnohet duke përdorur të dhënat hyrje-dalje dhe peshat e rrjetit përshtaten zakonisht duke përdorur algoritmin me përhapje të pasme. Supozimi i vetëm është që hartëzimi statik jolinear i gjeneruar nga rrjeti mund të prezantojë në mënyrë të kënaqshme sjelljen dinamike të sistemit për një aplikim të caktuar. Rrjeti neural duhet të prezantohet me informacionin për historinë e sistemit: hyrjet dhe daljet e mëparshme. Se sa informacion kërkohet për këtë varet nga saktësia e kërkuar dhe aplikimi përkatës.

Kur një rrjet shumë-shtresor trajnohet si një rregullator, në lak të hapur ose të mbyllur, shumica e problematikave është e ngjashme me rastin e identifikimit. Diferenca kryesore është që dalja e dëshiruar e rrjetit, që është sinjali i kontrollit që duhet të jepet në hyrje të objektit, nuk është e disponueshme por duhet të nxirret nga dalja e njohur, e dëshiruar e objektit. Për të arritur këtë, mund të përdoren përafrimet e bazuara në modelin matematikor të objektit (nëse është i disponueshëm), ose një model neural i dinamikës të objektit, ose modeli neural i dinamikës inverse të

objektit. Rrjetet neurale mund të kombinohen për identifikimin dhe kontrollin e objektit, duke krijuar kështu një strukturë kontrolli adaptive. [31], [36], [127]

Në paragrafët më poshtë jepet një përmbledhje e disa prej modeleve të përdorura në identifikim dhe kontroll.

2.3.1 Rrjetet neurale me vonesë në kohë

Identifikimi dhe kontrolli i një sistemi dinamik jolinear mund të kryhet nga një rrjet neural dinamik, i cili mund të ketë ose një arkitekturë rekurente ose një arkitekturë me vonesë në kohë. Figura 2.4 tregon një rrjet neural me vonesë në kohë, i cili përdor një sinjal të vetëm hyrje $x(k)$, por disa hyrje paralele të rrjetit neural artificial realizuar nga vonesat në kohë. Sekuencat e kohës korrente të sinjaleve dhe ato të kaluara propagandohen në rrjet dhe dalja $y(k)$ jepet nga shprehja

$$y(k) = F \left[\sum_{n=0}^N W_{nk} x(k-n) \right] \quad (1.33)$$

ku $W_{nk} = [W_{0k}, W_{1k} \dots W_{nk}]$ është vektori i peshave të rrjetit dhe F përcakton funksionin e aktivizimit. Rrjeti mund të trajnohet nga një algoritëm me propagim të pasëm statik të diskutuar më parë.

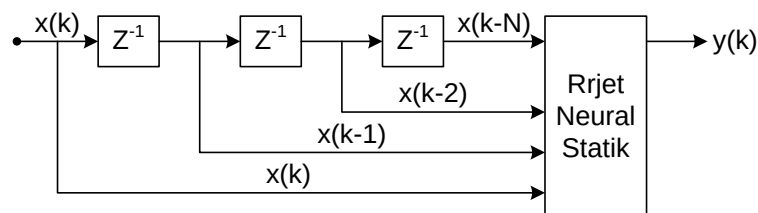


Figura 2.4 Rrjeti neural me vonesë në kohë

2.3.2 Modelet e sistemeve dinamike

Sistemet dinamike jolineare mund të modelohen me anë të ekuacioneve diferenciale jolineare. Këto ekuacione mund të paraqiten në formën me diferenca të fundme për t'u emuluar nga një rrjet neural.

Në literaturë [31], [36] është treguar që sistemet jolineare me një hyrje dhe një dalje mund të modelohen në një nga format e mëposhtme:

Modeli 1 – Funksioni i hyrjes me vonesë në kohë me dalje të vonuar në kohë:

$$y(k+1) = F[x(k), x(k-1), \dots, x(k-M)] + \sum_{i=0}^N \alpha_i y(k-i) \quad (1.34)$$

Modeli 2 – Funkzioni i dalje me vonesë në kohë me hyrje të vonuar në kohë:

$$y(k+1) = \sum_{i=0}^M \beta_i x(k-i) + F[y(k), y(k-1), \dots, y(k-N)] \quad (1.35)$$

Modeli 3 – Funkzioni i hyrjes me vonesë në kohë me funksioni dalje me vonesë në kohë:

$$y(k+1) = F[x(k), x(k-1), \dots, x(k-M)] + G[y(k), y(k-1), \dots, y(k-N)] \quad (1.36)$$

Modeli 4 – Funkzioni me hyrje me vonesë në kohë dhe lidhje të kundërt me vonesë në kohë:

$$y(k+1) = F[x(k), x(k-1), \dots, x(k-M); y(k), y(k-1), \dots, y(k-N)] \quad (1.37)$$

ku $[x(k) \ y(k)]$ është çifti i të dhënave hyrje/dalje të objektit në kohën k , $F(\cdot)$ dhe $G(\cdot)$ funksione jolineare, α_i është faktori i amplifikimit të daljes, dhe β_i është faktori i amplifikimit të hyrjes. Bllokskemat e të katër modeleve tregohen respektivisht në figurat 2.5, 2.6, 2.7 dhe 2.8.

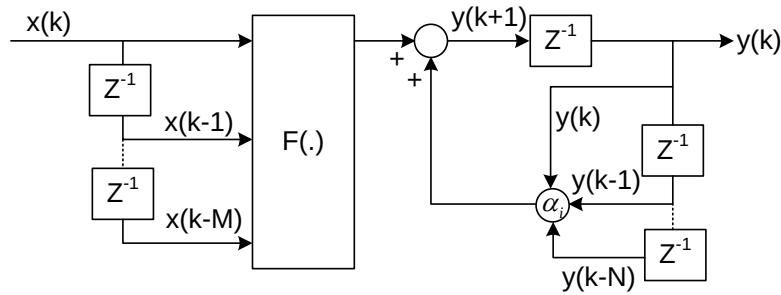


Figura 2.5 Modeli-1 i sistemi dinamik

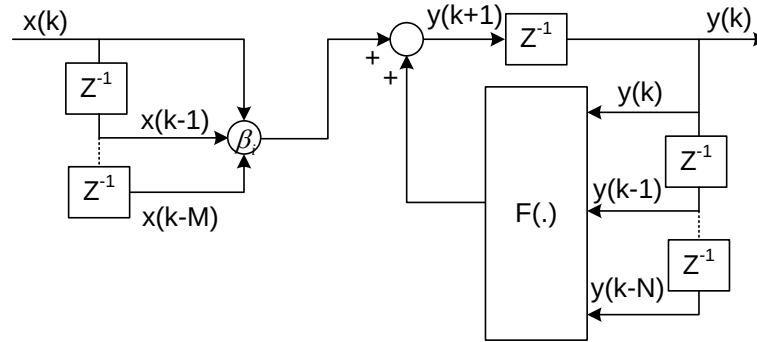


Figura 2.6 Modeli-2 i sistemi dinamik

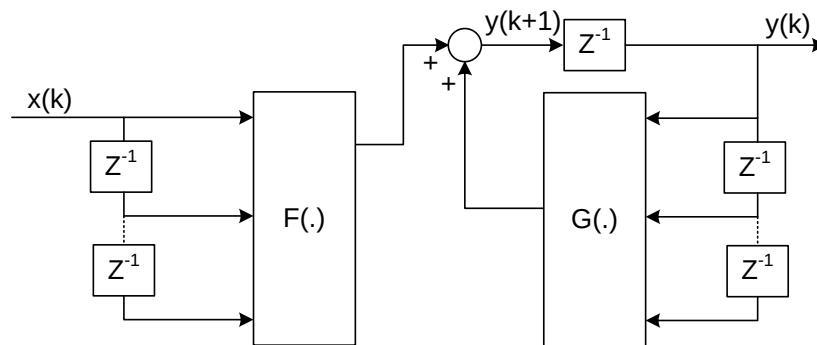


Figura 2.7 Modeli-3 i sistemi dinamik

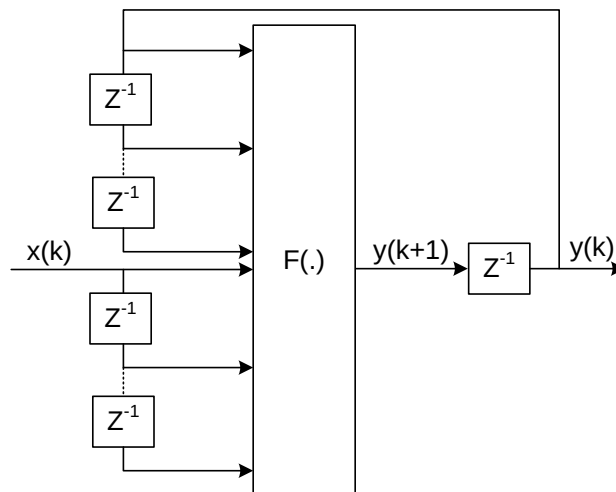


Figura 2.8 Modeli-4 i sistemi dinamik

2.3.3 Identifikimi me anë të rrjeteve neurale i modeleve dinamike

Si mund të trajnohen rrjetet neurale për emulimin e këtyre modeleve dinamike në 2.3.2? Shënohet që këto bllokskema paraqesin gjithashtu rrjete neurale që kanë të

përfshirë një strukturë neurale me vonesë në kohë, si tregohet në Figurën 2.4. Për trajnimin e rrjetit neural artificial, nuk kërkohet modeli eksplicit i objektit, por duhet të njihet struktura e modelit me rendin e sistemit, dhe duhet të ketë të dhëna hyrje/dalje të mjaftueshme nga sistemi eksperimental ose modeli i sistemit të simuluar (nëse është në dispozicion një model matematik).

Trajnimi i rrjetit neural artificial mund të kryhet ose në mënyrën paralele, ose në mënyrën seri-paralele, siç tregohet respektivisht në Figurat 2.9 dhe 2.10 (të treguara vetëm për Modelin-4). Në të dyja këto mënyra të dhënat hyrje/dalje janë funksione të kohës, duke kënaqur modelin dinamik të objektit, në ndryshim nga të dhënat statike që kërkohen për të trajnuar një rrjet statik. Trajnimi në mënyrën seri-paralele mund të kryhet duke përdorur algoritmin me përhapje të pasme statik, por mënyra paralele kërkon algoritëm dinamik (forma e modifikuar e metodës me propagim të pasëm statik) për shkak të lakut të lidhjes së kundërt. Në të dy rastet rrjeti trajnohet për të përputhur sekuencat e përkohshme të kampioneve hyrje/dalje të objektit.

Pas trajnimit të kënaqshëm, rrjeti neural paraqet sistemin dinamik të objektit. Në metodën paralele, dalja e dëshiruar nga objekti krahasohet me daljen e llogaritur të rrjetit neural në çastin k , dhe gabimi $\xi(k)$ rifreskon peshat e rrjetit me anë të algoritmit dinamik me përhapje të pasme. Gabimi mund të mos konvergjojë gjithmonë në mënyrën paralele dhe trajnimi i rrjetit mund të dështojë. Në modelin seri-paralele, dalja e objektit (pothuaj si dalja e modelit të identifikimit) lidhet në të kundërt në modelin e identifikuar, siç tregohet në Figurën 2.10. Trajnimi seri-paralele përgjithësisht konvergjon, duke dhënë një emulim modeli të kënaqshëm. Megjithatë, nëse modeli nuk trajnohet me saktësi të mjaftueshme, modeli neural mund të mos jetë i qëndrueshëm. [31]

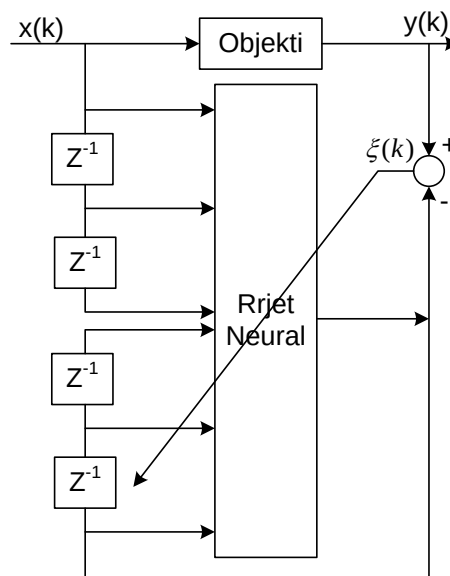


Figura 2.9 Identifikimi i modelit dinamik me modelin paralel

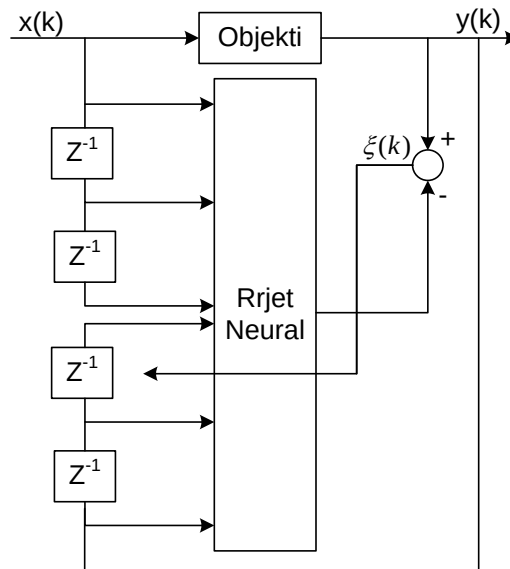


Figura 2.10 Identifikimi i modelit dinamik me modelin seri-paralel

Identifikimi i modelit i diskutuar më sipër është konsideruar jashtë-linje. Megjithatë, nëse parametrat e objektit ndryshojnë, rrjeti mund të trajnohet në-linjë (model adaptiv). Në përmbledhje, hapat për identifikimin neural të objektit me ndihmën e rrjeteve me vonesë në kohë mund të jepet si vijon:

- Përcaktohet struktura dhe rendi i modelit dinamik të objektit.
- Përcaktohet arkitektura e rrjetit neural artificial
- Merren të dhëna trajnimi të përkohshme nga eksperimenti ose simulimi (nëse është i disponueshëm modeli matematik).
- Duke përdorur algoritmin me përhapje të pasme (dinamik ose statik), trajnohet rrjeti me metodën paralele ose seri-paralele.
- Përdoren parametrat dhe peshat e rrjetit për të implementuar sistemin aktual në software ose hardware

2.3.4 Modeli dinamik invers

Gjer më tani është diskutuar identifikimi i drejtpërdrejtë i modelit të objektit. Gjithashtu është e mundur të indentifikohet një model invers i objektit, siç tregohet në Figurën 2.11. Në këtë rast, të dhënat e daljes të objektit fiksohen si të hyrja e rrjetit neural, dhe llogaritja e daljes të tij krahasohet me të dhënat e hyrjes (objektivin). Gabimi që rezulton trajnon rrjetin. Pas trajnimit të kënaqshëm, modeli $F^{-1}(\cdot)$ paraqet inversin e modelit të objektit $F^{-1}(\cdot)$.

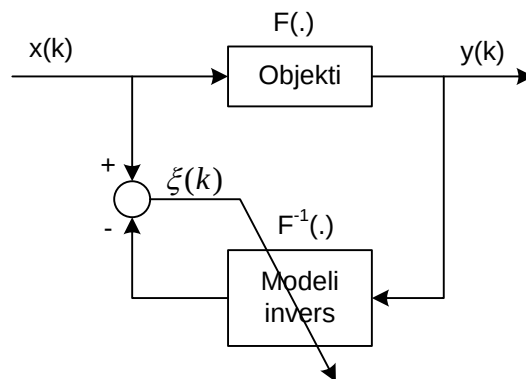


Figura 2.11 Modeli neural dinamik invers i objektit

2.3.5 Kontrolli me bazë rrjetet neurale

Rrjetet neurale mund të përdoren për kontrollin me lak të mbyllur të një sistemi. Figura 2.12 tregon një rregullator neural, i cili mund të trajnohet për të zëvendësuar rregullatorin. Nëse, për shembull, rregullatori është i tipit P-I Fuzzy dhe objekti është një motor me induksion, rregullatori neural (i tipit statik) mund të zëvendësojë rregullatorin fuzzy, duke thjeshtuar kështu implementimin.

Figura 2.13 tregon kontrollin adaptiv të një objekti bazuar në modelin dinamik invers të objektit. Nëse një rrjet neural artificial me model dinamik të saktë $F^{-1}(\cdot)$ është i disponueshëm, ai mund të lidhet në skajin fundor për të anuluar dinamikën e objektit [$F(\cdot) \cdot F^{-1}(\cdot) = 1$]. Atëherë, idealisht, sinjalet e daljes do të ndjekin sinjalin e hyrjes dhe nuk do jetë i nevojshëm kontrolli me lidhje të kundërt. Megjithatë, dalja do të devijojë nga hyrja nëse ka variacion parametrash të objektit ose ka gabim të fundmë në modelin invers. Figura 2.13 tregon një kontroll bazuar në modelin dinamik invers të një objekti ku gabimi e do të zëvendësojë gabimin e modelit dhe kjo për shkak të variacionit parametrik të objektit. Sinjali e i shtohet sinjalit $u(k)$ për të zëvendësuar sinjalin $u(k)$. Gabimi e mund të përdoret gjithashtu për të taruar rrjetin neural në-linjë.

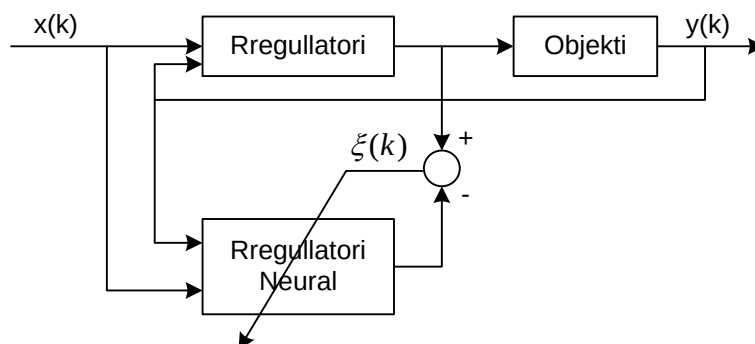


Figura 2.12 Trajnimi i rregullatorit neural për të emuluar rregullatorin aktual

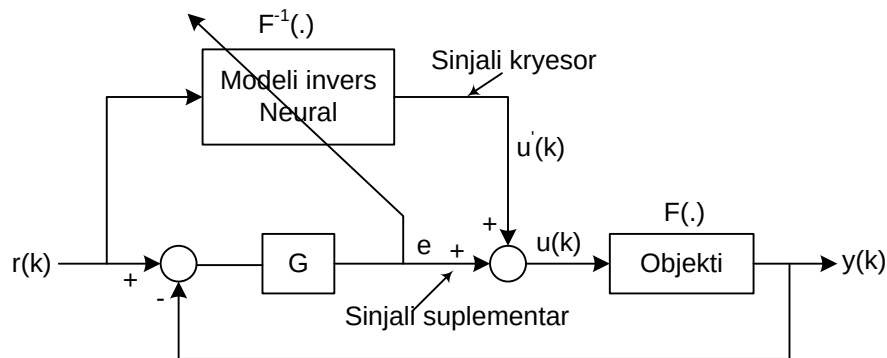


Figura 2.13 Kontrolli adaptiv i objektit me bazë modelin dinamik invers

Figura 2.14 tregon një kontroll adaptiv me model referencë (MRAC) që përdor një rrjet neural. Këtu, dalja e modelit referencë (objektivi) krahasohet me daljen e objektit. Rregullatori neural është trajnuar në mënyrë të tillë që, së bashku me modelin e objektit, gjurmon gjithmonë modelin referencë. Një problem në këtë skemë kontrolli është që objekti qëndron midis rregullatorit dhe gabimit dhe nuk ka ndonjë mënyrë për të propaganduar mbrapsht gabimin nëpër rregullator. Për të kapërcyer këtë problem, përdoret metoda me MRAC indirekt, e treguar në Figurën 2.15. Në fillim, modeli neural i identifikimit $F(\cdot)$ trajnohet për të emuluar modelin e drejtpërdrejtë të objektit. Më pas ky model vendoset në seri me rregullatorin neural për të ndjekur modelin referencë siç tregohet. Tarimi i rregullatorit neural tani është i përshtatshëm nga modeli neural $F(\cdot)$ në vend të objektit. Modeli neural mund të rifreskohet periodikisht nëse ka ndonjë variacion të parametrave të objektit. [30], [31], [36], [127]

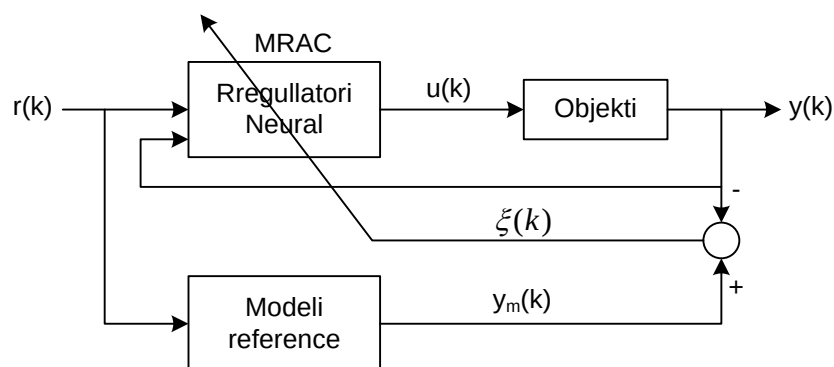


Figura 2.14 Kontrolli adaptiv me model reference me rrjet neural (metoda direkte)

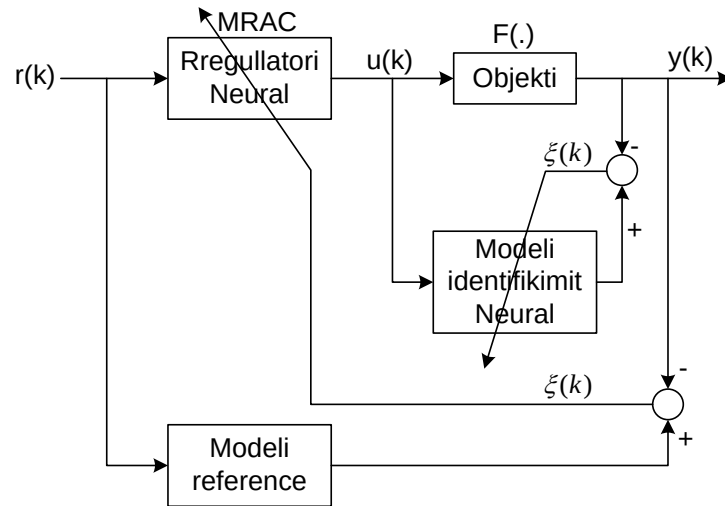


Figura 2.15 Kontrolli adaptiv me model reference me rrjet neural (metoda indirekte)

III. Simulime të rrjeteve neurale në identifikim dhe kontroll

3.1 Aplikimi i rrjeteve neurale në identifikimin/modelimin e objektit

Le të shohim zbatimin e rrjeteve neurale në identifikimin dhe modelimin e një objekti. Konsiderojmë si objekt një krah robotik planar me dy hallka si në figurën 3.1 [128]

Saktësia e pozicionit të manipulatorit robotik varet nga njohja e saktë e parametrave të tij kinematikë. Një nga këta parametra, që varet nga aktuatori i hallkës, është pozicioni këndor. Skemat e kontrollit të një manipulatori të tillë bazohen në lidhjen e kundërt sipas pozicionit dhe shpejtësisë [128], [129]. Të dy sinjalet, pozicioni dhe shpejtësia këndore janë të rëndësishëm për t'u monitoruar.

Duke patur parasysh dy faktorë:

- Këta sinjale ndikohen edhe nga faktorë të pamatshëm, pra nga ngacmime të jashtme
- Vetia e përmendur në paragrafët mësipërm e rrjeteve neurale për të hartëzuar sinjalet hyrje-dalje, pra për të ndërtuar modelin, pa patur nevojën e njohjes së një modeli matematik

lind interesi për aplikimin e rrjetave neurale në identifikimin e modelit që jep sinjalin që dëshirojmë të kemi nën vëzhgim.

Identifikimi i sistemit në vetvete përdor metoda statistikore për të ndërtuar modelet matematikore të sistemeve dinamike nga të dhënat e matura.

Në analogji, rrjeti neural përdor kampione të matura (të njohura) hyrje-dalje për të krijuar modelin që bën të mundur hartëzimin e sinjaleve hyrje-dalje.

Për të vëzhguar efikasitetin e rrjetit neural në modele të ndryshme, kryejmë identifikimin e modelit të shpejtësisë dhe të pozicionit duke përdorur të njëjtin rrjet neural.

Modelet matematikore të dhëna më poshtë i shërbejnë vetëm një qëllimi, simulimit të sinjaleve kampione hyrje-dalje, që do përdoren për trajnimin e rrjetit neural.

Hapat kryesorë të ndjekur do jenë:

- Marrja e Kampioneve Hyrje-Dalje nëpërmjet modeleve matematikore të aktuatorit
- Trajnimi i rrjetit neural me kampionet Hyrje-Dalje

- Krahasimi i përgjigjeve të marra nga rrjeti neural me ato të marra nga modeli real

3.1.1 Modeli i aktuatorit

Krahu robotik planar me dy hallka, paraqitur skematikisht në Figurën 3.1a [128] ka të dhënat e mëposhtme:

$$a_1 = a_2 = 1\text{m};$$

$$l_1 = l_2 = 0.5\text{m};$$

$$m_{l1} = m_{l2} = 50\text{kg};$$

$$I_{l1} = I_{l2} = 10\text{kg m}^2;$$

$$K_{r1} = K_{r2} = 100;$$

$$m_{m1} = m_{m2} = 5\text{kg};$$

$$I_{m1} = I_{m2} = 0.01\text{kg m}^2$$

Të dhënat e aktuatorit janë:

$$F_{m1} = F_{m2} = 0.01 \text{ N.ms/rad};$$

$$R_{a1} = R_{a2} = 10 \text{ ohm};$$

$$k_{t1} = k_{t2} = 2 \text{ N.m/A};$$

$$k_{v1} = k_{v2} = 2 \text{ v.s/rad}$$

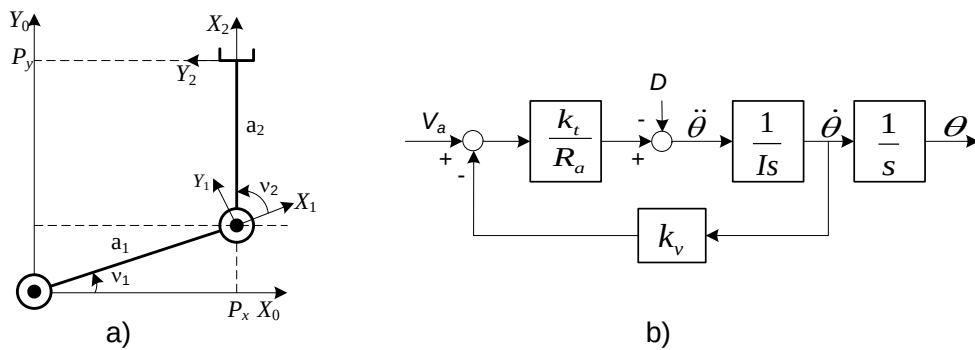


Figura. 3.1. a) Manipulatori me dy hallka, b) Bllokskema e sistemit të çiftuesit kinematik.

Lëvizja e imponuar në çiftuesit e manipulatorit realizohet nga sistemi aktuimit që konsiston në servomotor të rrymës së vazhduar [128], [129]. Bllok skema e çiftuesit i mund të paraqitet si në Figurën 3.1b. Variablet θ dhe $\dot{\theta}$ në figurë paraqesin respektivisht pozicionin këndor dhe shpejtësinë këndore të aktuatorit. I është inertësia mesatare e reduktuar në aksin e motorit, që varen nga parametrat fizikë të hallkave dhe çiftuesve fizikë si $I_i = \bar{b}_{ii} / K_{ri}^2$, për $i = 1, 2$, ku b_{ii} llogariten nga kinematika e manipulatorit si elemente të matricës së inercisë B në ekuacionet 3.1 &

3.2 . R_a është rezistenca e induktit (L_a , induktiviteti është neglizhuar), dhe k_t dhe k_v janë respektivisht konstantet e momentit dhe e motorit.[128], [129]

$$\mathbf{B}(q) = \begin{bmatrix} b_{11}(q_2) & b_{12}(q_2) \\ b_{21}(q_2) & b_{22}(q_2) \end{bmatrix} \quad (3.1)$$

$$\begin{aligned} b_{11} &= I_{l1} + m_{l1}l_1^2 + K_{r1}^2 I_{m1} + I_{l2} + m_{l2}(a_1^2 + l_2^2 + 2a_1l_2c_2) + I_{m2} + m_{m2}a_1^2 \\ b_{12} &= b_{21} = I_{l2} + m_{l2}(l_2^2 + a_1l_2c_2) + K_{r2}I_{m2} \\ b_{22} &= I_{l2} + m_{l2}l_2^2 + K_{r2}^2 I_{m2} \end{aligned} \quad (3.2)$$

Koeficienti c_2 në ekuacionin 3.2 llogaritet nga këndi i hallkës. Funkcioni i transmetimit të aktuatorit për pozicionin jepet në ekuacionin 3.3 dhe funksioni transmetimit për shpejtësinë në ekuacionin 3.4. Kontrolli i secilit çiftues konsiderohet i pavarur, megjithatë lëvizja e të dytit ndikon si shqetësim në çiftuesin e parë ([128], [129]).

$$M(s) = \frac{\theta}{V_a} = \frac{k_m}{s(1 + sT_m)} \quad \text{ku} \quad k_m = \frac{1}{k_v}; \quad T_m = \frac{R_a I}{k_v k_t} \quad (3.3)$$

$$M'(s) = \frac{\dot{\theta}}{V_a} = \frac{k_m}{1 + sT_m} \quad (3.4)$$

Modelet diskrete korresponduese të modeleve të dhëna nga ekuacionet 3.3 dhe 3.4 jepen me anë të ekuacioneve 3.7, 3.8 dhe 3.9.

$$M(z) = \frac{2.361 \cdot 10^{-4} z + 2.286 \cdot 10^{-4}}{z^2 - 1.907z + 0.9071} \quad (3.7)$$

Ek.7 jep:

$$y(k) = 1.907 y(k-1) - 0.9071y(k-2) + 2.361 \cdot 10^{-4} u(k-1) + 2.286 \cdot 10^{-4} u(k-2) \quad (3.8)$$

$$M'(z) = \frac{0.0467}{z - 0.9071} \Rightarrow y(k) = 0.9071y(k-1) + 0.0467 u(k-1) \quad (3.9)$$

3.1.2 Arkitektura e rrjetit neural

Për ndërtimin e modelit neural duhet të kemi në konsideratë:

- Çfarë arkitekture rrjeti neural do zgjedhim
- Çfarë metode të mësuari dhe algoritmi do përdorim për trajnimin e rrjetit
- Sa shtresa neuronesh do zgjedhim

- Sa neurone do ketë në secilën shtresë

Referuar teorisë të paraqitur në kapitullin I, zgjedhim një rrjet neural me lidhje të drejtpërdrejta me përhapje të pasme, tre shtresor; një shtresë hyrje, një të fshehur dhe një shtresë dalje.

Numrin e neuroneve në shtresën e hyrjes e marrim 6, sipas dinamikës të objektit të dhënë nga ekuacioni 3.8.

Numrin e neuroneve në shtresën e daljes e marrim 1, dhe korrespondon me daljen e modelit neural, pra sinjalin që duam të monitorojmë.

Numrin e neuroneve të shtresës së fshehur e marrim fillimisht të barabartë me 4, afërsisht sa gjysma e shumës së neuroneve të shtresës së hyrjes dhe daljes. Kjo sipas tabelës 1.1 në paragrafin 1.7.2. Këtë numër e rrisim ose zvogëlojmë në varësi të gabimit të përfutur.

Kërkojmë që gabimi kuadratik për daljen e modelit neural të jetë në rendin 10^{-3} , në mënyrë që të kemi një hartëzim të mirë hyrje-dalje nga rrjeti neural.

Arkitektura e zgjedhur jepet në mënyrë skematike në figurën 3.2. Kjo figurë paraqet identifikimin e modelit me anë të rrjetit neural.

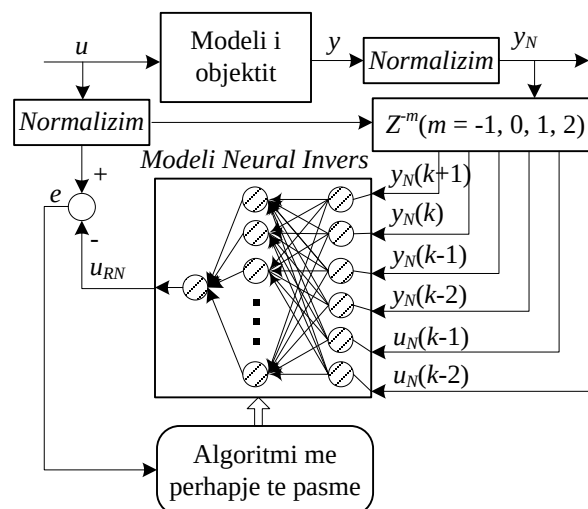


Figura 3.2 Blllokskema e identifikimi të modelit me rrjet neural

3.1.3 Trajnimi i rrjetit neural dhe rezultatet e simulimeve

Për trajnimin e rrjetit neural ndjekim metodën e dhënë në blllokskemën e figurës 1.8. Dy rastet e vëzhguara jepen si vijon në 3.1.3.1 dhe 3.1.3.2.

3.1.3.1 Identifikimi i modelit të pozicionit

Shohim fillimisht identifikimin e modelit të pozicionit. Modelet neurale të identifikuar do jenë modeli invers dhe modeli i pozicionit.

Kampionet hyrje-dalje që duhen për trajnimin e rrjetit neural i marrim me ndihmën e Matlab, duke simuluar sjelljen e modelit matematik 3.8 kundrejt një sinjali pseudorastësor. Sinjali i hyrjes zgjidhet pseudorastësor për të patur sa më shumë informacion për sjelljen e modelit kundrejt sinjalit të hyrjes të ndryshueshëm. Në këtë mënyrë rrjeti neural do arrijë të mësojë më mirë mbi dinamikën që duhet të imitojë. Këto kampione jepen në figurën 3.3

Për modelin neural invers, hyrja e rrjetit është dalja e modelit të ekuacionit 3.8. Për modelin neural, hyrja e rrjetit është hyrja e modelit të ekuacionit 3.8.

Arkitektura fillestare e rrjetit të marrë është $6 \times 4 \times 1$, siç është përshkruar në paragrafin 3.1.2. Numri i neuroneve të shtresës së fshehur ndryshohet sipas rezultateve të marra. Rrjeti trajnohet deri sa të merret një rezultat i kënaqshëm. Figurat 3.4 deri në 3.7 japin rezultatet e marra për rrjet neural me 7 neurone në shtresën e fshehur, pra rrjet $6 \times 7 \times 1$.

Figurat 3.4 dhe 3.5 paraqesin ecurinë e ndryshimit të peshave dhe spostimeve të shtresave të rrjetit.

Gabimi i rrjetit neural për modelin invers është: gabimi maksimal: 0.0048388; varianca e gabimit: 0.00010234

Figura 3.6 jep ecurinë e gabimit dhe daljen e modelit neural invers krahasuar me daljen e dëshiruar.

Rezultatet për modelin neural të pozicionit janë: gabimi maksimal : 0.039993; varianca e gabimit : 0.0014722

Ecuria e gabimit dhe dalja e modelit neural krahasuar me daljen e dëshiruar jepet në figurën 3.7.

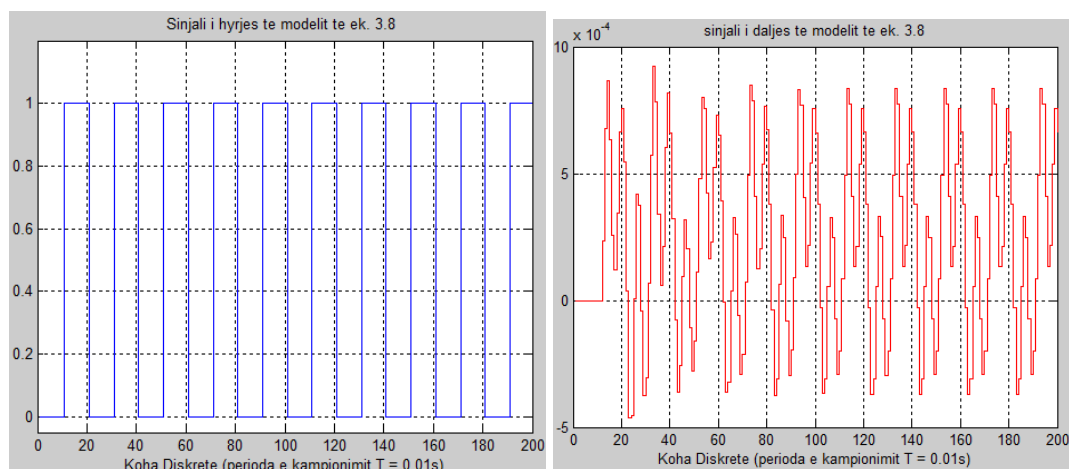


Figura 3.3 Kampionet H-D të përdorura për trajnimin e modelit neural të pozicionit

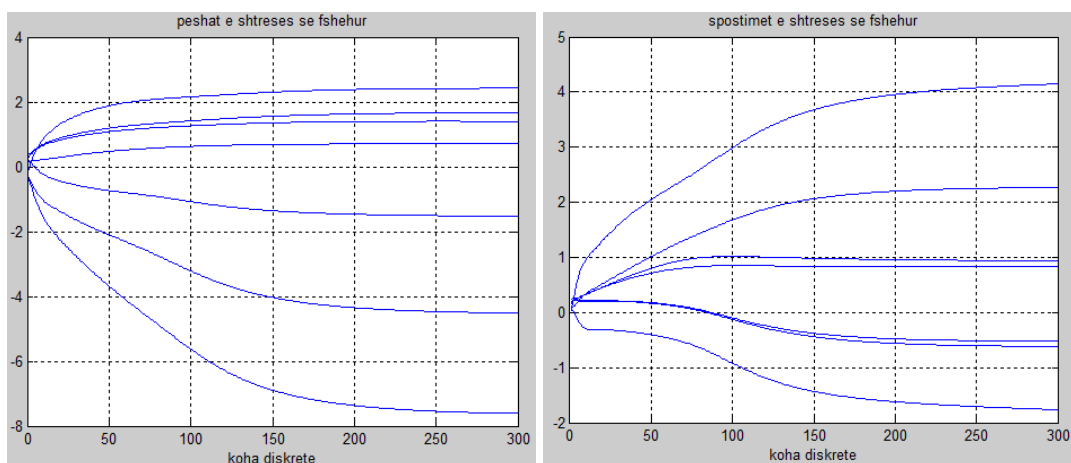


Figura 3.4 Ecuria e peshave dhe spostimeve të shtresës së fshehur

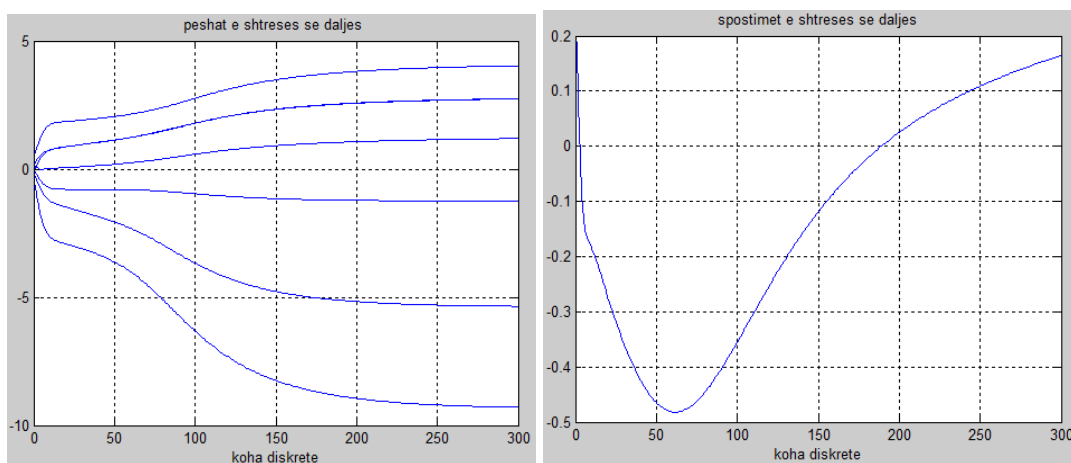


Figura 3.5 Ecuria e peshave dhe spostimit të shtresës së daljes

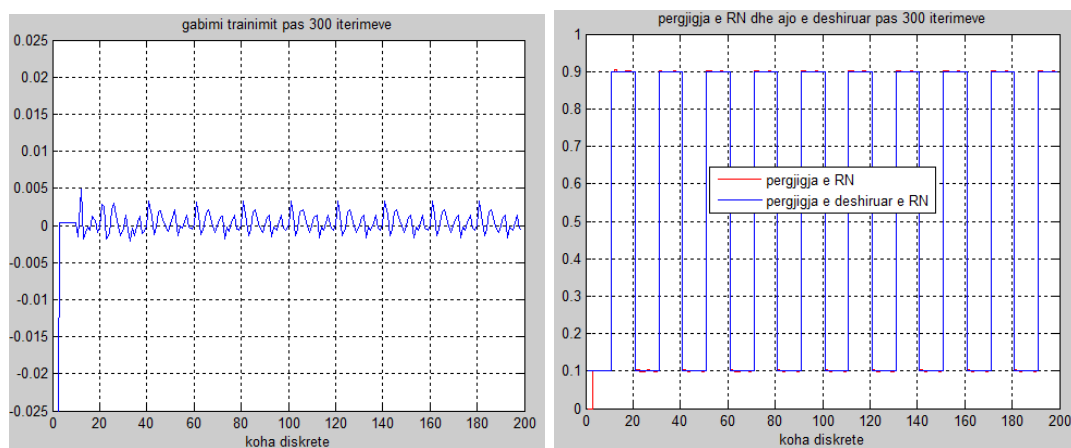


Figura 3.6 Gabimi i trajnimit dhe dalja e modelit neural invers të pozicionit

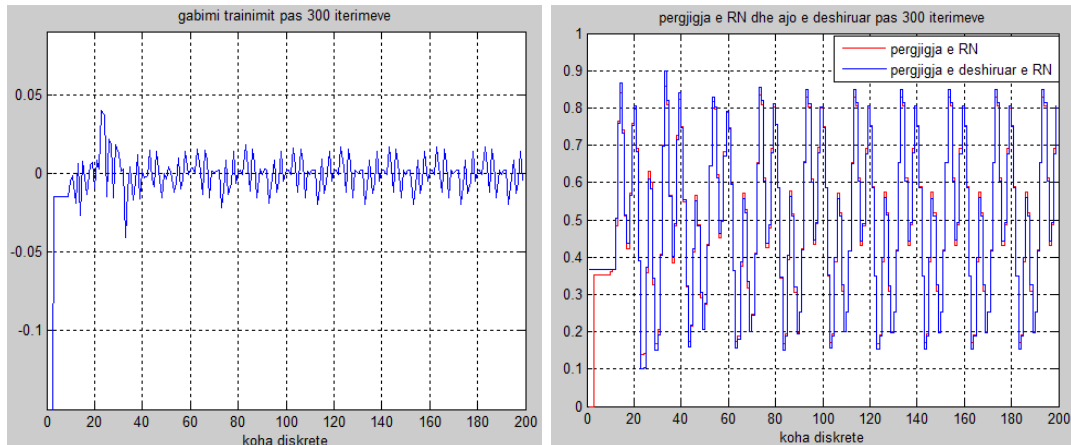


Figura 3.7 Gabimi i trajnimit dhe dalja e modelit neural të pozicionit

3.1.3.2 Identifikimi i modelit të shpejtësisë

Siç e përmendëm më sipër, për të vëzhguar vetinë e përgjithësimin të rrjetit neural, në kuptimin që e njëjta arkitekturë rrjeti mund të përdoret për identifikimin e modeleve të ndryshme, për identifikimin e modelit të ekuacionit 3.9 do aplikojmë të njëjtën strukturë rrjeti të gjetur për modelin e ekuacionit 3.8.

Kampionet hyrje-dalje për trajnimin e rrjetit me arkitekturë $6 \times 7 \times 1$, si në 3.1.3.1, jepen në figurën 3.8. Sinjali i hyrjes zgjidhet pseudorastësor, për të njëjta arsye të përmendura më sipër.

Ecuria e ndryshimit të peshave dhe spostimeve të shtresave të rrjetit jepet në figurat 3.9 dhe 3.10.

Gabimi i marrë pas trajnimit të modelit neural invers është:

- gabimi maksimal : 0.02314
- varianca e gabimit : 0.00011929

Figura 3.11 jep ecurinë e gabimit dhe daljen e modelit neural invers krahasuar me daljen e dëshiruar.

Rezultatet për modelin neural të shpejtësisë janë:

- gabimi maksimal : 0.026038
- varianca e gabimit : 0.00021176

Ecuria e gabimit dhe dalja e modelit neural krahasuar me daljen e dëshiruar jepen në figurën 3.12.

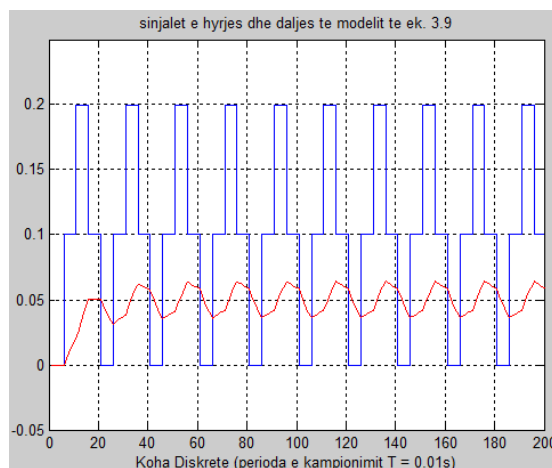


Figura 3.8 Kampionet H-D të përdorura për trajnimin e modelit neural të shpejtësisë

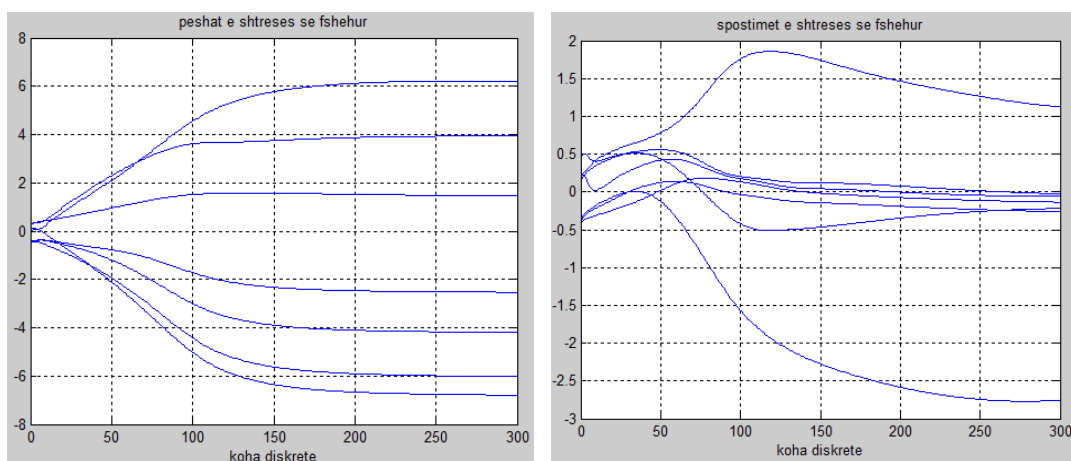


Figura 3.9 Ecuria e peshave dhe spostimeve të shtresës së fshehur

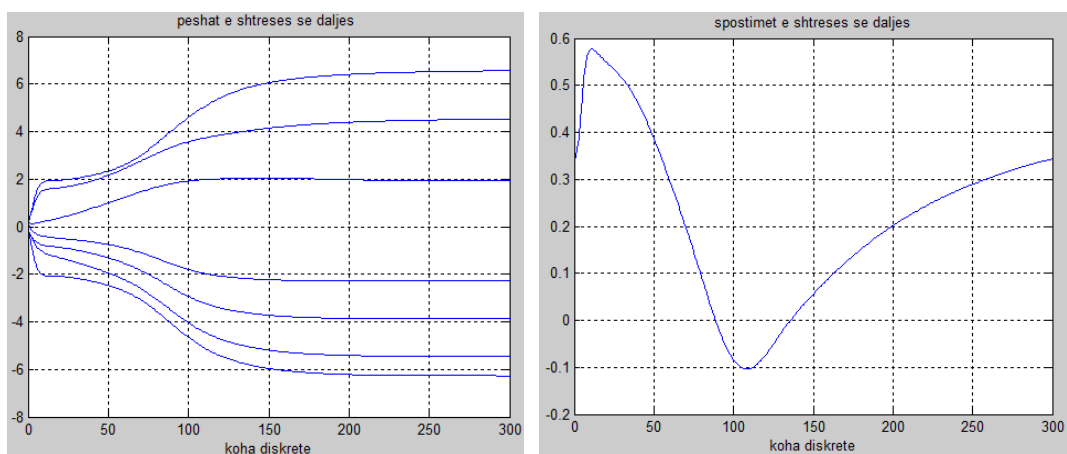


Figura 3.10 Ecuria e peshave dhe spostimit të shtresës së daljes

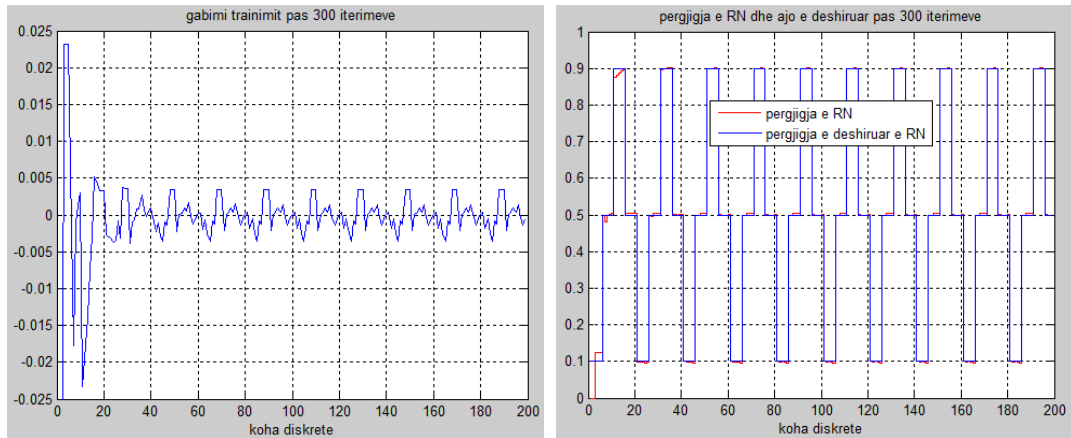


Figura 3.11 Gabimi i trajnimit dhe dalja e modelit neural invers të shpejtësisë

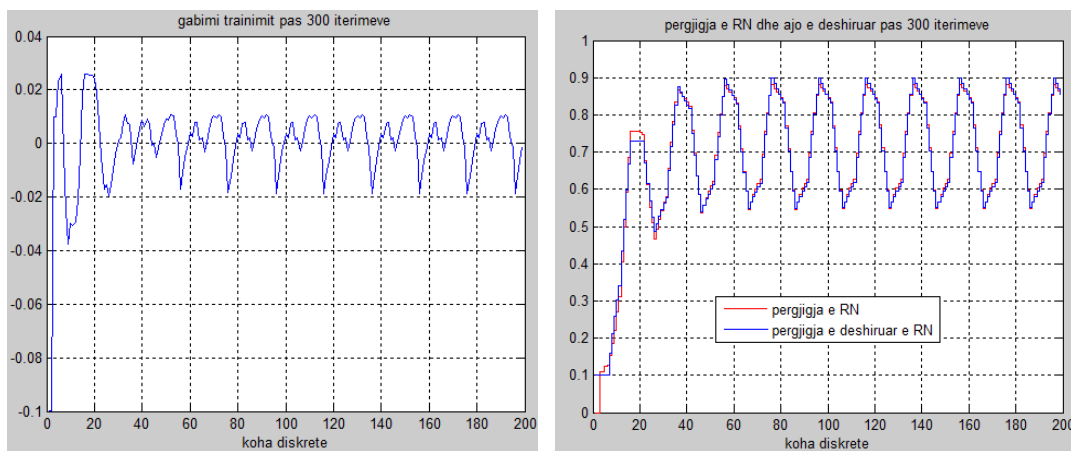


Figura 3.12 Gabimi i trajnimit dhe dalja e modelit neural të shpejtësisë

3.1.4 Përfundime

Modelet neurale u trajnuan vetëm duke patur kampionet hyrje-dalje të modeleve matematikore, pra vetëm kampione hyrje-dalje të objektit.

Nga rezultatet shihet që modelet neurale të trajnuara kanë sjellje të kënaqshme; ato përafrojnë sjelljen e modelit real me gabime të vogla, brenda vlerave të kërkuara. Kjo pa patur njohje të parametrave të objektit/sistemit. Si të tillë, rrjetet neurale janë të vlefshëm në përafrimin e modeleve të sistemit, qofshin këto lineare ose jolineare, kur parametrat e sistemit ndryshojnë me kohën ose kur ka shqetësime nga ambienti rrethues.

E njëjta arkitekturë rrjeti jep rezultate të kënaqshme për dy modele të ndryshme: modelin e pozicionit dhe modelin e shpejtësisë, edhe pse këto modele janë të rendit të dytë dhe të rendit të parë. Mund të themi që rrjetet neurale kanë vetinë e përgjithësimi [1], [29], [32]–[34].

3.2 Aplikimi i rrjetit neural në sistemin e mbyllur të kontrollit

Një nga skemat e njohura të kontrollit është ajo me model invers [30]–[32], [127], [130]. Le të shohim kontrollin e shpejtësisë të aktuatorit të dhënë në 3.1 me anë të kësaj skeme.

Modelin neural invers të trajnuar në 3.1.3.1 e vendosim në sistemin e mbyllur, si paraqitet në bllokskemën e figurës 3.13. Modeli i objektit në këtë skemë është ai i dhënë nga ekuacioni 3.9. Sinjali $r(k)$ është sinjali i referencës, sinjali $u(k)$ është dalja e modelit neural që shërben si sinjal komande, sinjali $y(k)$ është dalja e sistemit të mbyllur.

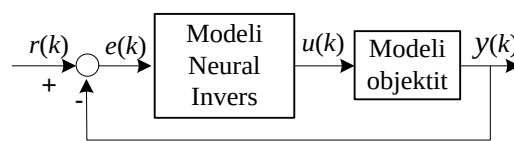


Figura 3.13 Bllokskema e sistemit të mbyllur të kontrollit me model neural invers

Treguesit e cilësisë që kërkojmë janë: mbirregullim 0%; gabim në regjim të vendosur $e = 0$; dhe kohë vendosje $t_v \leq 0.25$ sek.

Kryejmë simulimin e sistemit të mbyllur për dy sinjale të ndryshme reference $r(k)$:

1. $r(k) = 1$
2.
$$r(k) = \begin{cases} 0 & \text{per } r_1(k) < 0 \\ r_1(k) & \text{per } r_1(k) \geq 0 \end{cases} \quad \text{ku} \quad r_1(k) = \sin(\omega k)$$

Kjo për të parë nëse sjellja e sistemit ndryshon në varësi të sinjalit të hyrjes.

Rezultatet e marra jepen në figurat 3.14 dhe 3.15. Siç shihet nga këto figura, treguesit e cilësisë të sistemit të mbyllur janë më të mirë se treguesit e kërkuar.

Mbirregullimi është zero për të dy rastet, koha e vendosjes është më vogël se 0.3 sekonda dhe gabimi në regjim të vendosur është zero.

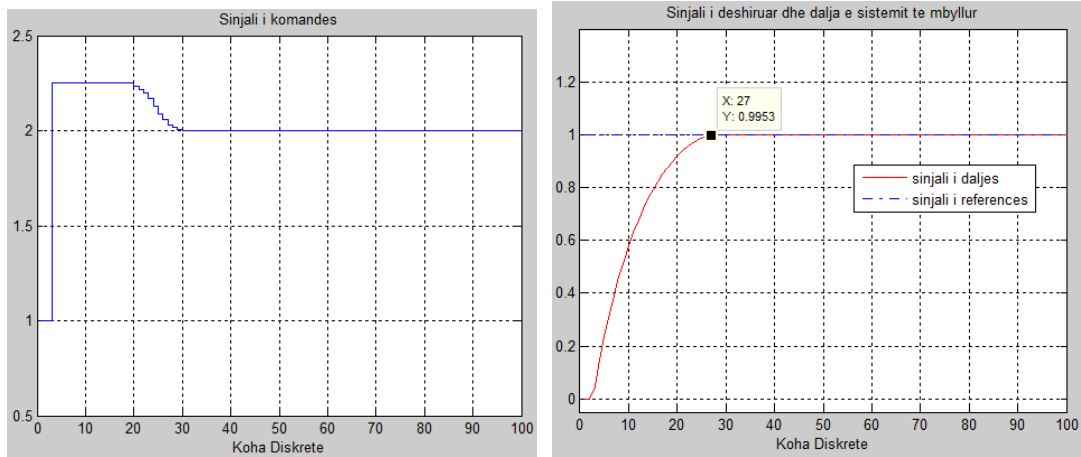


Figura 3.14 Sinjali i komandës dhe dalja e sistemit të mbyllur për sinjal hyrje shkallë njësi

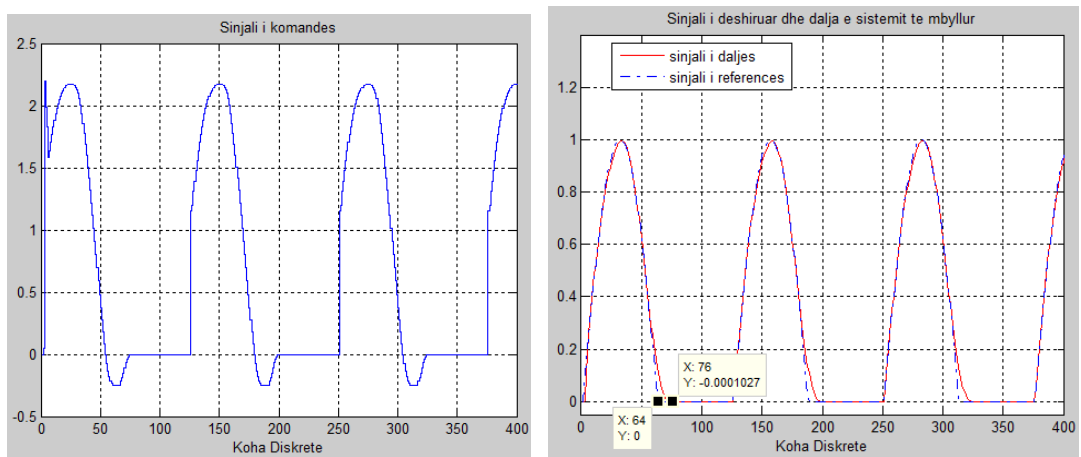


Figura 3.15 Sinjali i komandës dhe dalja e sistemit të mbyllur për sinjal hyrje sinusoidal

Sipas rezultateve të marra dhe literaturës [30]–[32], [130], mund të themi që skema e kontrollit me model neural invers është një nga aplikimet e mundshme të rrjeteve neurale në kontroll.

3.3 Vëzhgim mbi ndikimin e elementeve të ndërtimit të rrjetit neural në cilësinë e sistemit të mbyllur të kontrollit

Në literaturën e shfletuar nuk ka rregull të mirëfilltë për përcaktimin e arkitekturës të rrjetit neural në varësi të kërkesave për kontrollin e një sistemi të mbyllur. Lind pyetja se si ndikojnë elementet e ndërtimit të një rrjeti neural në cilësinë e sistemit të mbyllur.

Le të vëzhgojmë si ndikojnë këta elementë në skemën e kontrollit me model neural invers si në figurën 3.13. Si shembull marrim kontrollin e shpejtësisë të një motori të rrymës së vazhduar. Metoda e ndjekur e vëzhgimit jepet në figurën 3.16.

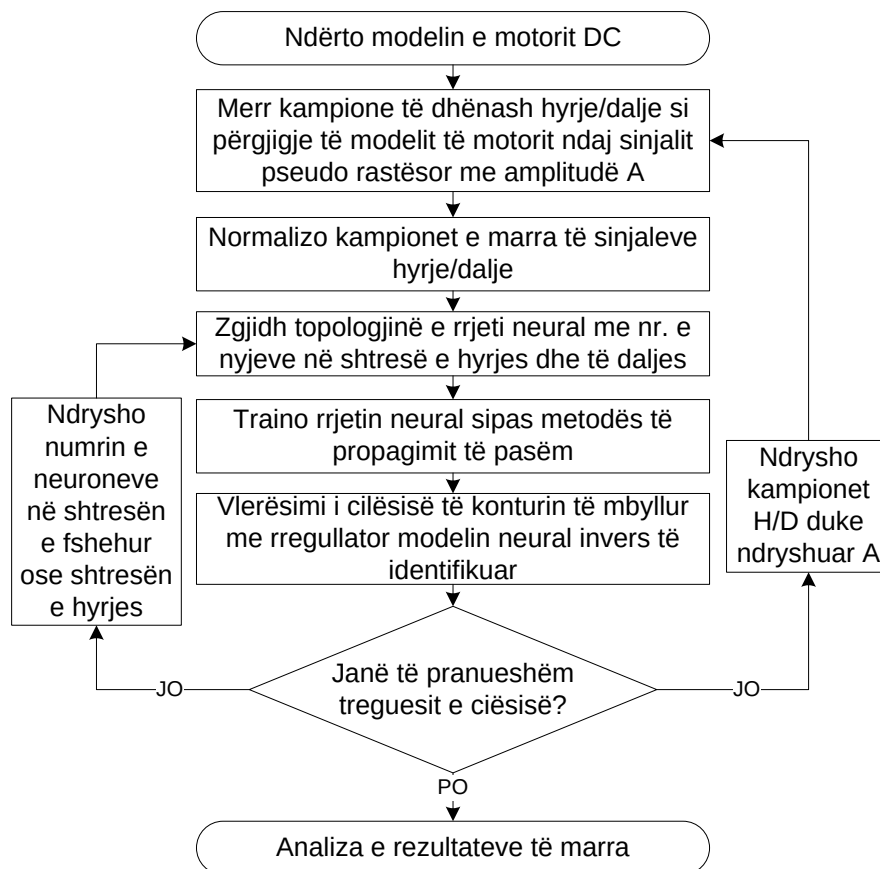


Figura 3.16 Metoda për trajnimin e rrjetit neural dhe vëzhgimin e treguesve të cilësisë të sistemit të mbyllur

Elementet e nevojshëm për ndërtimin e rrjetit apo modelit neural invers janë:

- Kampionet hyrje-dalje të përdorura për trajnimin e rrjetit
- Tipi i variableve në shtresën e hyrjes: cilat nga kampionet e marra do përdoren për trajnimin e rrjetit
- Numri i neuroneve të shtresës së hyrjes
- Numri i neuroneve të shtresës së fshehur
- Numri i neuroneve të shtresës së daljes

Simulimet do përqendrohen në ndryshueshmërinë e kampioneve hyrje-dalje, përzgjedhjen e variableve të hyrjes, dhe neuroneve të shtresës të hyrjes e të fshehur. Dalja e modelit neural është sinjali i komandës.

3.3.1 Modeli i motorit të rrymës së vazhduar DC

Motori i rrymës të vazhduar DC i marrë si shembull është ai i modulit G14 EV [131], që ndodhet pranë laboratorit të Automatizimit të Industrisë të Fakultetit të Inxhinierisë Elektrike. Parametrat teknikë të këtij motori janë:

- inercia e rotorit (J) = $350 \cdot 10^{-7} \text{ kg m}^2$;
- konstantja mekanike e kohës $T_m = 0.17\text{s}$;
- konstantja e forcës elektromotore ($K=K_e=K_t$) = 0.06 Nm/A ;
- rezistenca e rotorit (R_i) = 8 ohm ;
- induktiviteti i rotorit (L_i) = 7.6 mH

Skema strukturore e modelit të motorit jepet në Figurën 3.17a dhe funksioni transmetues i kontrollit të shpejtësisë jepet sipas ekuacionit (3.10) [132]. Hyrja e modelit është tensioni i armaturës V_i dhe dalja është shpejtësia ω .

$$G(s) = \frac{\omega(s)}{V_i(s)} = \frac{K_t}{(R_i + L_i s)(b + Js) + K_t K_e} \quad (3.10)$$

$$\text{Ku: } G(s) = \frac{2.256 \cdot 10^5}{s^2 + 1059s + 1.973 \cdot 10^4} = \frac{225563.91}{(s + 1040)(s + 18.98)} \quad (3.11)$$

Përgjigja e sistemit të hapur $G(s)$ ndaj hyrjes shkallë njësi jepet në figurën 3.17b. Sipas ekuacionit 3.11 përcaktojmë funksionin transmetues diskret (3.12) të motorit, për periudë kampionimi $T = 0.01\text{sek}$: [133]

$$H(z) = \frac{1.801z^{-1} + 0.1758z^{-2}}{1 - 0.8272z^{-1} + 2.529 \cdot 10^{-5}z^{-2}} \quad (3.12)$$

Nga ekuacioni 3.12 gjejmë modelin rekurent (3.13) të motorit [133]:

$$y(k) = 0.8272 y(k-1) - 2.529 \cdot 10^{-5} y(k-2) + 1.801 u(k-1) + 0.1758 \cdot u(k-2) \quad (3.13)$$

Ku $y(k)$ dhe $u(k)$ janë përkatësisht shpejtësia dhe sinjali i komandës në çastin kT , për $k = 1, 2, 3, \dots N$.

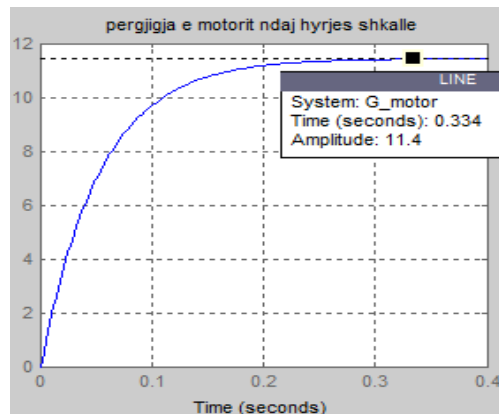
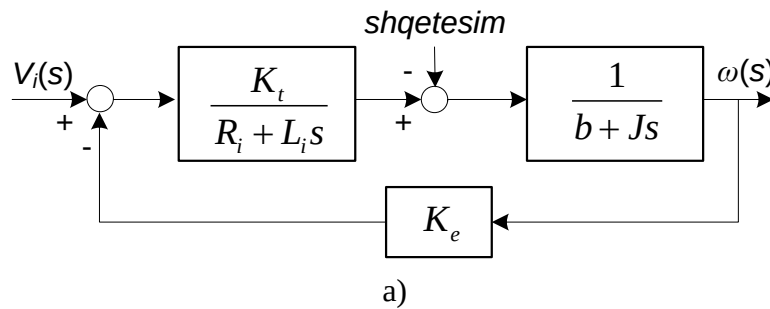


Figura 3.17. a) Skema strukturore e motorit DC b) Përgjigja shkallë njësi e sistemit me lak të hapur

3.3.2 Modeli neural invers

Rrejti neural i marrë për ndërtimin e modelit invers të motorit DC është tre shtresor, me lidhje të drejtpërdrejtë, i trajnuar me anë të algoritmit me përhapje të pasme. Arkitektura është e ngjashme me atë të trajtuar në paragrafin 3.1. Referuar literaturës [1], [31]–[34], [127], [130], një rrjet i tillë mund të përafrojë mirë çdo funksion linear ose jolinear.

Arkitektura e përgjithshme është e formës $n \times m \times 1$. Shtresa e daljes do ketë vetëm një neuron që korrespondon me sinjalin e komandës në skemën e sistemit të mbyllur të figurës 3.13.

Për të vëzhguar ndikimin e parametrave të ndërtimit të rrjetit në cilësinë e sistemit të mbyllur shqyrtojmë këto arkitektura:

A1: Arkitektura e parë $6 \times m \times 1$ ku si hyrje marrim $dy_N(i+1)$, $dy_N(i)$, $y_N(i)$, $y_N(i-1)$ u $N(i-1)$, $du_N(i-1)$. Këto janë sinjalet e normalizuar të komandës, të daljes të sistemit të hapur, dhe ndryshesat e tyre, në çastin korrent dhe një hapa para ose pas. Modelimi i rrjetit neural me këtë arkitekturë jepet në figurën 3.18a.

A2: Arkitektura e dytë $5 \times m \times 1$ ku si hyrje marrim $dy_N(i+1)$, $dy_N(i)$, $y_N(i)$, $y_N(i-1)$ u $N(i-1)$. Modelimi i rrjetit neural me këtë arkitekturë jepet në figurën 3.18b.

A3: Arkitektura e tretë $6 \times m \times 1$ ku si hyrje marrim $y_N(i+1)$, $y_N(i)$, $y_N(i-1)$, $y_N(i-2)$, $u_N(i-1)$, $u_N(i-2)$, që janë sinjalet e normalizuar të komandës dhe të daljes të sistemit të hapur në çastet $k+1$, k , $k-1$ dhe $k-2$. Modelimi i rrjetit neural me këtë arkitekturë jepet në figurën 3.19a.

A4: Arkitektura e katërt $5 \times m \times 1$ ku si hyrje marrim $y_N(i+1)$, $y_N(i)$, $y_N(i-1)$, $y_N(i-2)$, $u_N(i-1)$. Modelimi i rrjetit neural me këtë arkitekturë jepet në figurën 3.19b.

Zgjedhja e hyrjeve për secilën arkitekturë është marrë referuar ekuacionit 3.13 që jep dinamikën e motorit DC.

Numri i neuroneve shtresës së fshehur m do jetë i ndryshueshëm, nga 3 në 20.

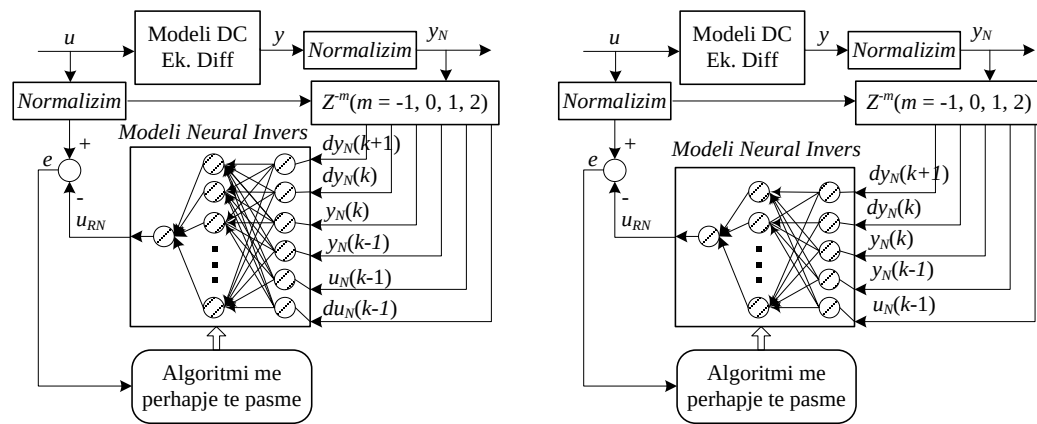


Figura 3.18 Skema e identifikimi të modelit neural invers të motorit DC me rrjet neural sipas a) arkitekturës së parë b) arkitekturës së dytë

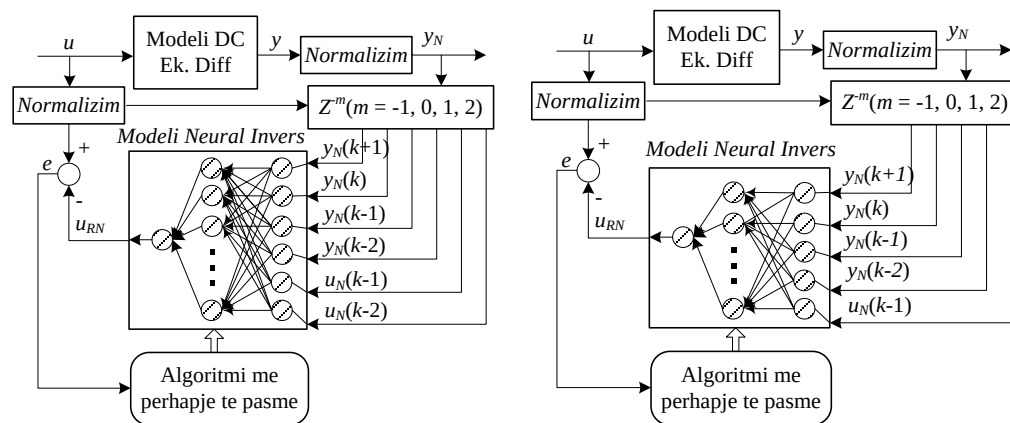


Figura 3.19 Skema e identifikimi të modelit neural invers të motorit DC me rrjet neural sipas a) arkitekturës së tretë b) arkitekturës së katërt

3.3.3 Simulime të sistemit të mbyllur me model neural invers

Hapat e simulimit janë dhënë përmbledhtasi në figurën 3.16. Marrim tre grupe të ndryshme kampionesh hyrje-dalje të modelit të motorit DC: tre sinjale të ndryshme pseudorastësore me amplituda $a=0.174$, $a=4$, $a=0.6$, që jepen në hyrje të modelit të ekuacionit 3.13 dhe merren daljet përkatëse si në figurat 3.20 dhe 3.21. Pasqyrohen simulimet për këto tre vlera tipike të amplitudës, një afër vlerës të sinjalit të daljes të modelit neural invers në sistemin e mbyllur (sinjalit të komandës) që jep rregullim të mirë, një vlerë e mesme dhe një vlerë e madhe. Kjo pasi është parë fillimisht reagimi i vetëm një arkitekture të rrjetit neural kundrejt vlerave të ndryshme të kësaj amplitude.

Kryejmë trajnimin e rrjetit neural për secilin grup kampionesh hyrje-dalje dhe secilën arkitekturë A1, A2, A3, A4 të dhënë në 3.3.2. Modelin neural invers të trajnuar e vendosim në sistemin e mbyllur dhe marrim daljen e tij për sinjal reference shkallë. Për secilin kombinim arkitekturë + kampione trajnimi hyrje-dalje, kryejmë 18 simulime për neurone të shtresës së fshehur $m = 3 \div 20$. Kjo për të parë ndikimin e numrit të neuroneve të shtresës së fshehur, neuroneve të shtresës së hyrjes dhe kampioneve hyrje-dalje që përdoren për trajnimin e modelit neural.

Gjithsej janë 12 çifte, 3 grupe kampionesh H-D $\times$ 4 arkitektura. Në total janë 216 (12×18) simulime të ndryshme. Për shkak të volumit të madh të këtyrë simulimeve, rezultatet janë paraqitur përmbledhtasi në tabelat S1 deri në S12 të shtojcës S1; vetëm vlerat e mbirregullimit dhe vlera në regjim të vendosur për përgjigjen e sistemit të mbyllur për secilin simulim janë paraqitur. Figurat 3.22 deri në 3.27 ilustrojnë përgjigjen e sistemit të mbyllur për secilin prej 12 kombinimeve kampion-arkitekture dhe për numër neuronesh të shtresës së fshehur $m = 6$. Paraqitja grafike e ecures të peshave dhe spostimeve të modeleve të rrjeteve neurale të trajnuara nuk po jepen për arsye hapësire, dhe sepse nuk sjellin vlerë të shtuar në analizën që kërkojmë të kryejmë.

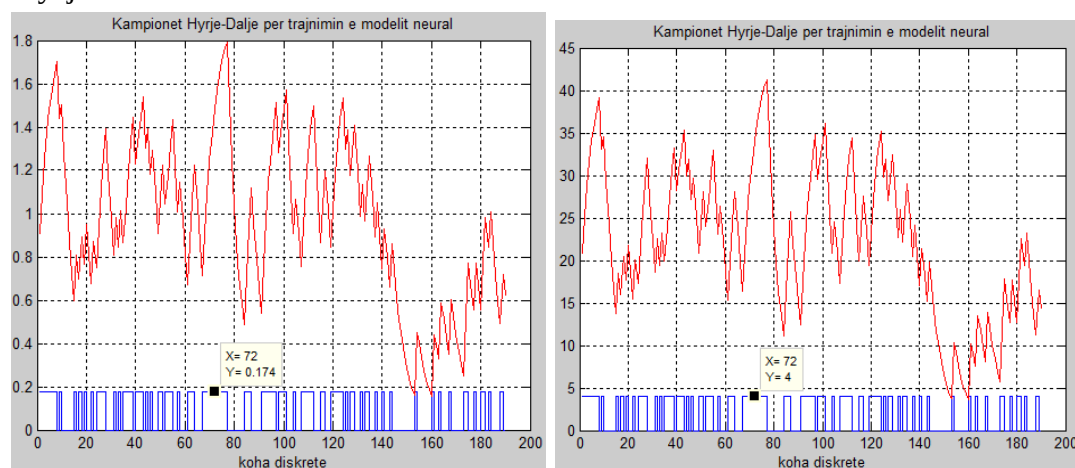


Figura 3.20 Grupi a) i parë dhe b) i dytë i kampioneve hyrje-dalje të përdorura për trajnimin e rrjetit neural

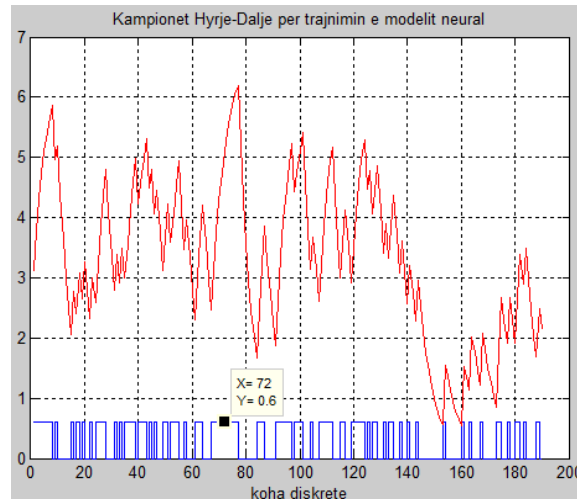


Figura 3.21 Grupi i tretë i kampioneve hyrje-dalje të përdorura për trajnimin e rrjetit neural

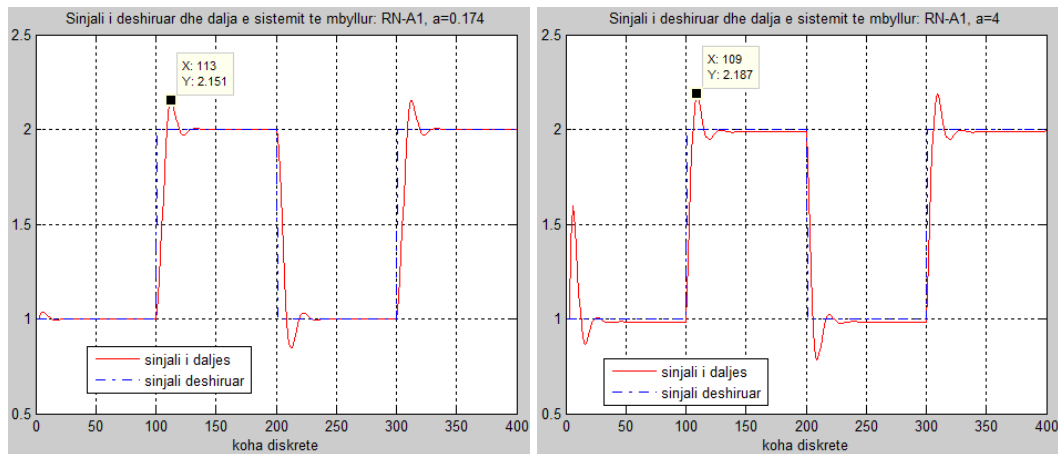


Figura 3.22 Dalja e sistemit të mbyllur për RN me A1 dhe trajnim të tij me kampione sipas a) fig. 3.20a dhe b) fig. 3.20b

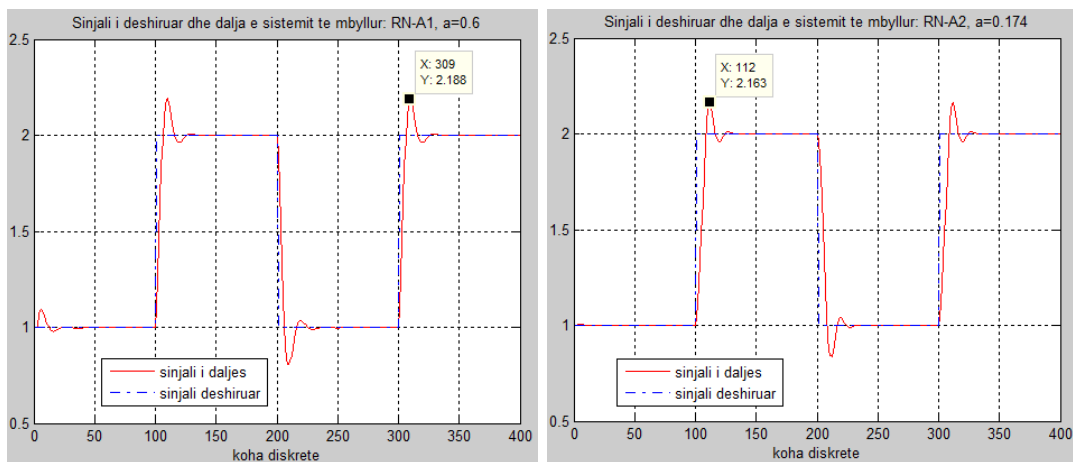


Figura 3.23 Dalja e sistemit të mbyllur për a) RN me A1 dhe trajnim të tij me kampione sipas fig. 3.21 dhe b) RN me A2 dhe trajnim të tij me kampione sipas fig. 3.20a

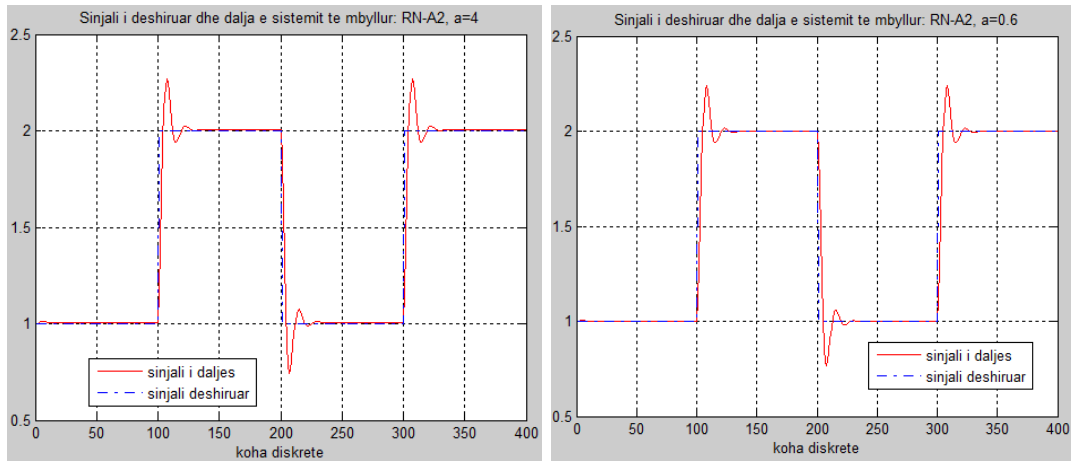


Figura 3.24 Dalja e sistemit të mbyllur për RN me A2 dhe trajnim të tij me kampione sipas a) fig. 3.20b dhe b) fig. 3.21

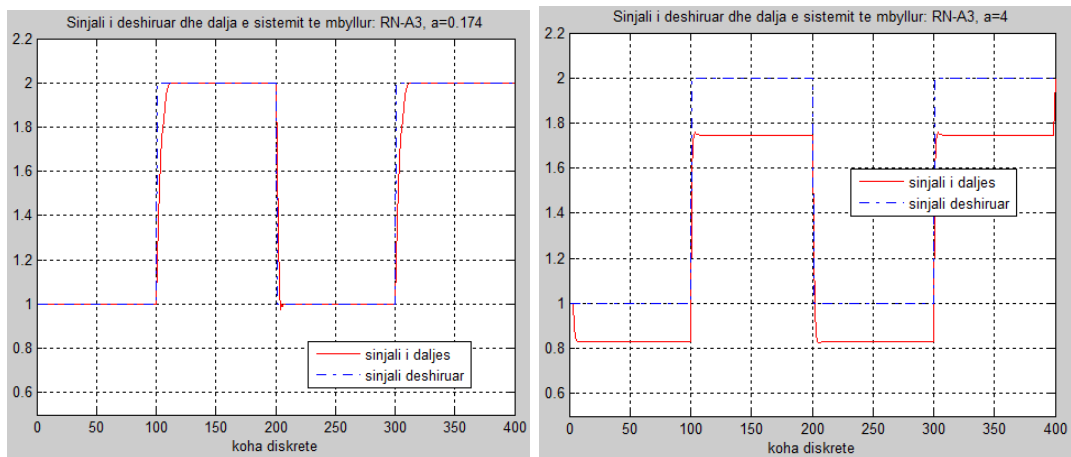


Figura 3.25 Dalja e sistemit të mbyllur për RN me A3 dhe trajnim të tij me kampione sipas a) fig. 3.20a dhe b) fig. 3.20b

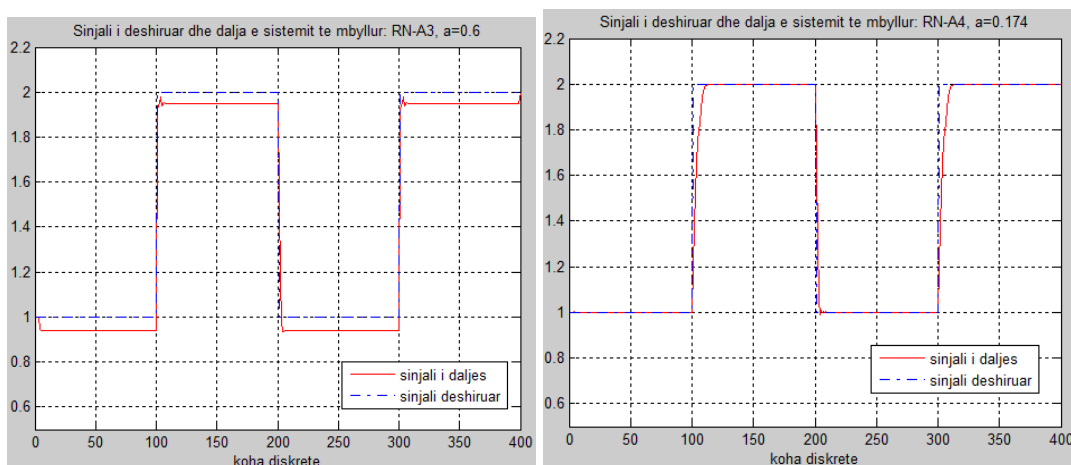


Figura 3.26 Dalja e sistemit të mbyllur për a) RN me A3 dhe trajnim të tij me kampione sipas fig. 3.21 dhe b) RN me A4 dhe trajnim të tij me kampione sipas fig. 3.20a

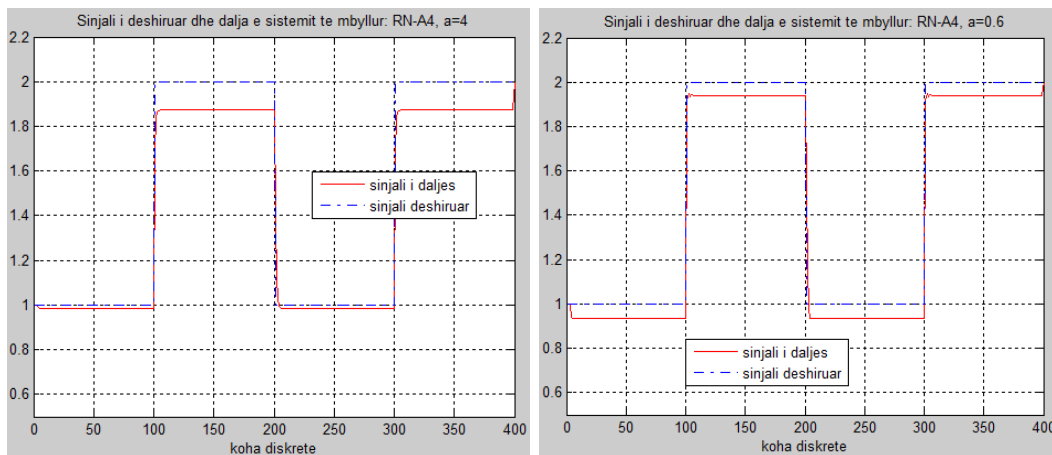


Figura 3.27 Dalja e sistemit të mbyllur për RN me A4 dhe trajnim të tij me kampione sipas a) fig. 3.20b dhe b) fig. 3.21

3.3.4 Krahاسimi i rezultateve të marra nga simulimet

Le të krahasojmë rezultatet duke vendosur në grafikë ecuritë përkatëse të dhëna në tabelat S1÷S12, për secilën arkitekturë dhe grup kampionesh hyrje-dalje. Për trajnim me kampione hyrje-dalje me amplitudë $a=0.174$, që sipas figurave 3.22÷3.27 jep rregullimin me të mirë, pasqyrojmë në të njëjtin graf rezultatet e gjithë arkitekturave.

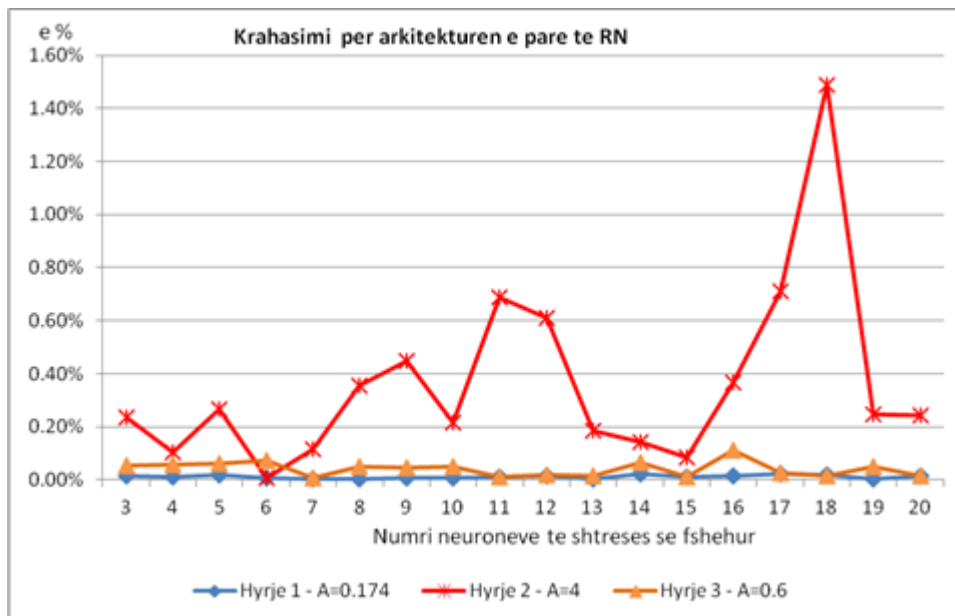


Figura 3.28 Gabimi i sistemit të mbyllur për RN-A1 dhe trajnim të modelit neural me kampione të ndryshme H-D

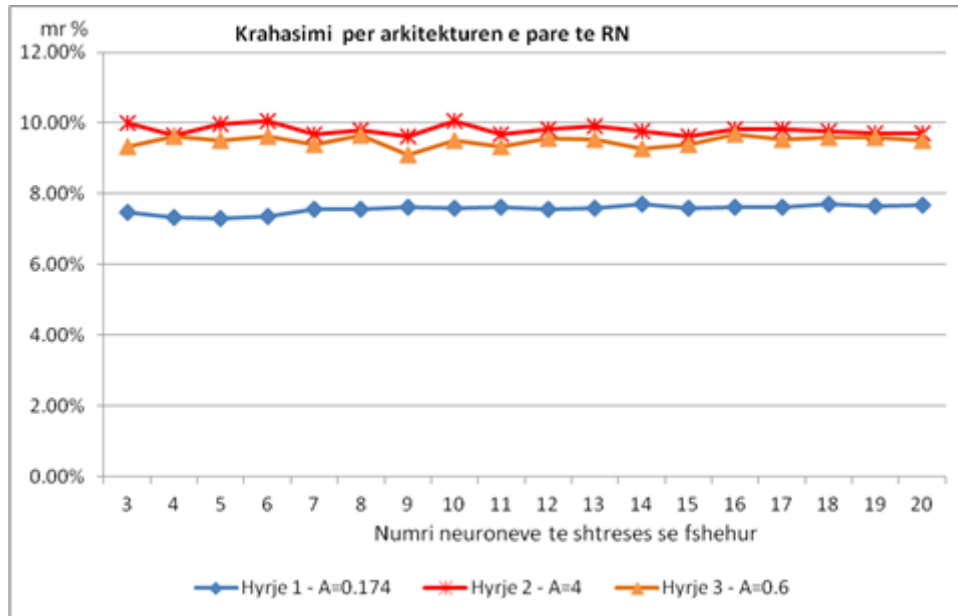


Figura 3.29 Mbirregullimi i sistemit të mbyllur për RN- A1 dhe trajnim të modelit neural me kampione të ndryshme H-D

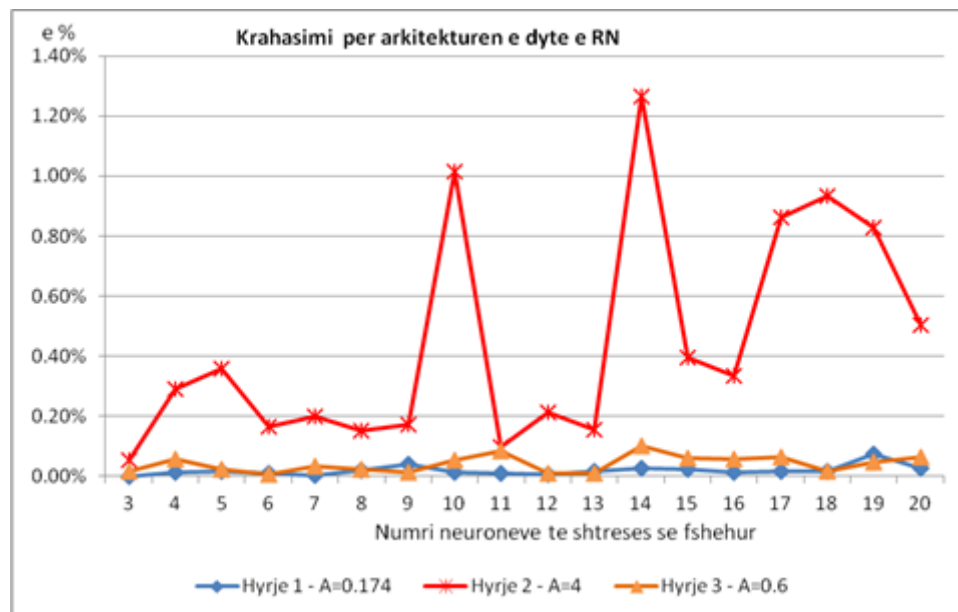


Figura 3.30 Gabimi i sistemit të mbyllur për RN- A2 dhe trajnim të modelit neural me kampione të ndryshme H-D

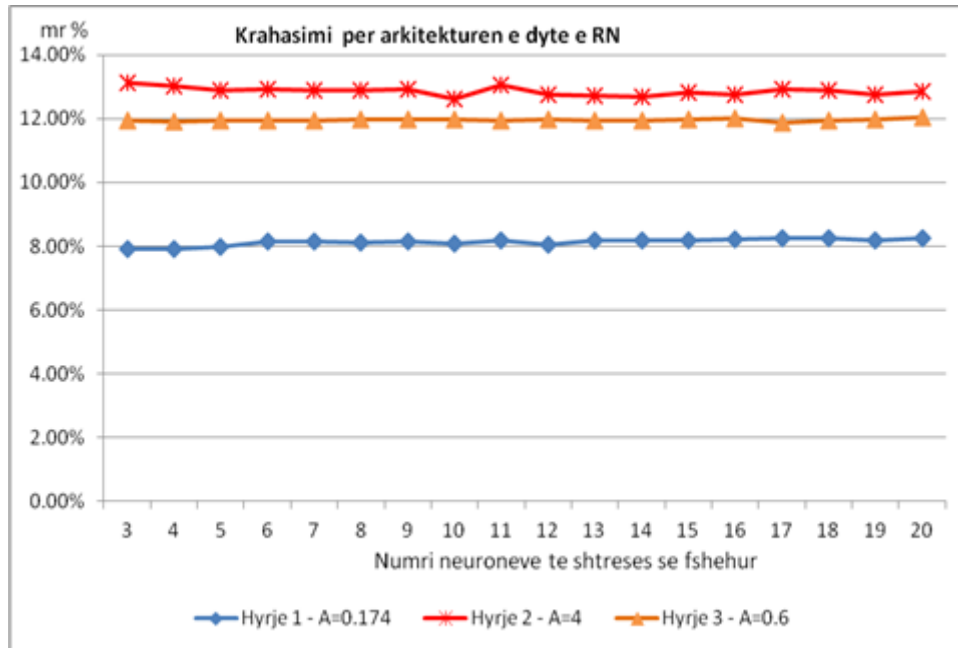


Figura 3.31 Mbirregullimi i sistemit të mbyllur për RN- A2 dhe trajnim të modelit neural me kampione të ndryshme H-D

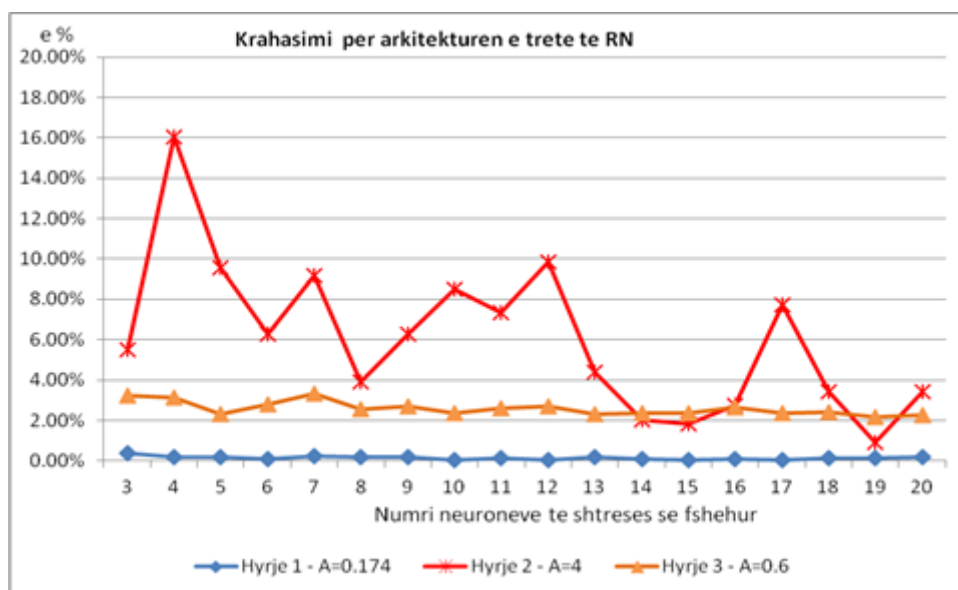


Figura 3.32 Gabimi i sistemit të mbyllur për RN- A3 dhe trajnim të modelit neural me kampione të ndryshme H-D

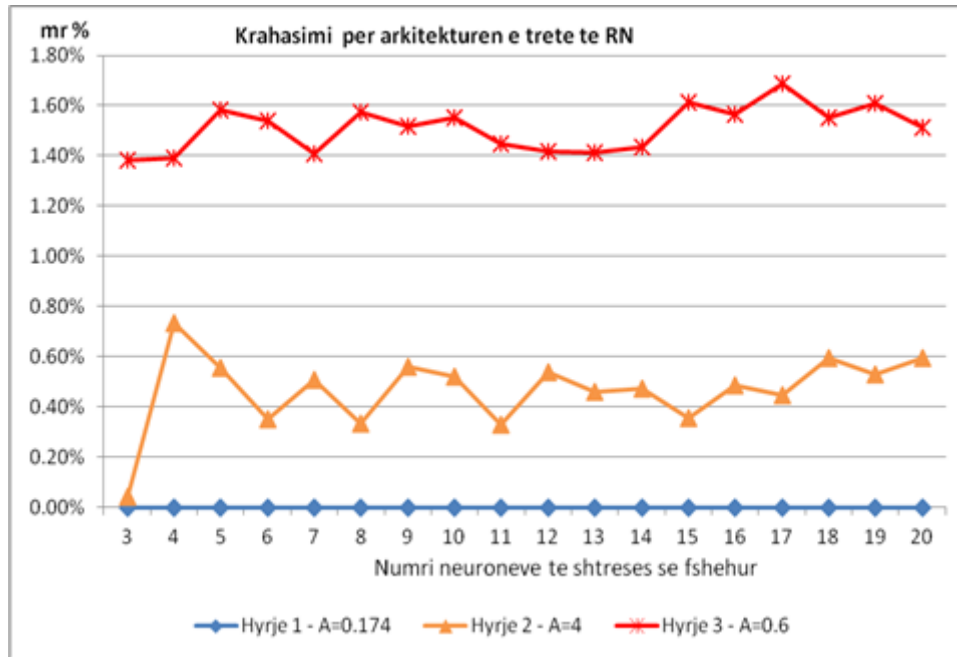


Figura 3.33 Mbirregullimi i sistemit të mbyllur për RN- A3 dhe trajnim të modelit neural me kampione të ndryshme H-D

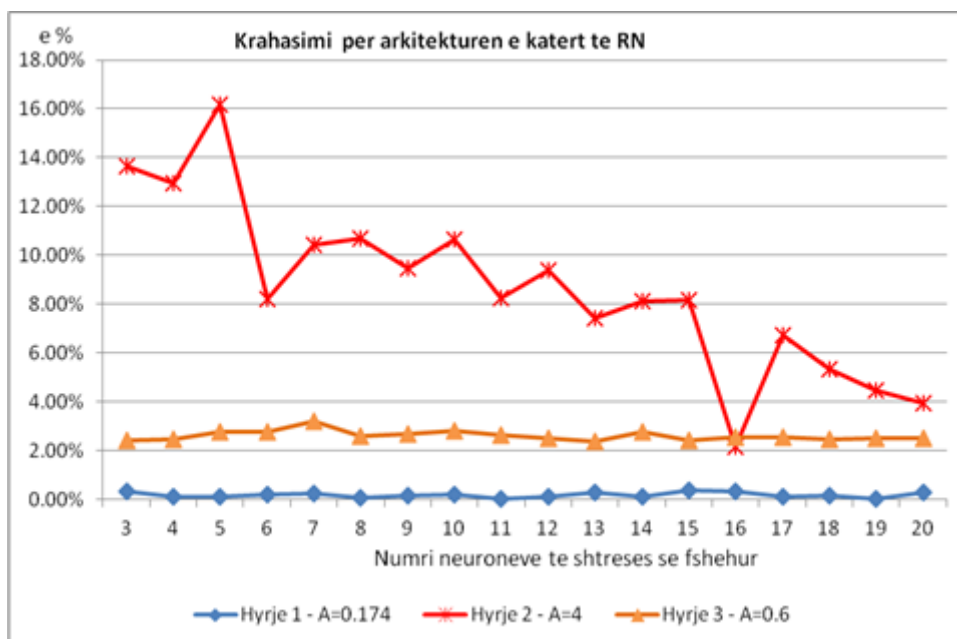


Figura 3.34 Gabimi i sistemit të mbyllur për RN- A4 dhe trajnim të modelit neural me kampione të ndryshme H-D

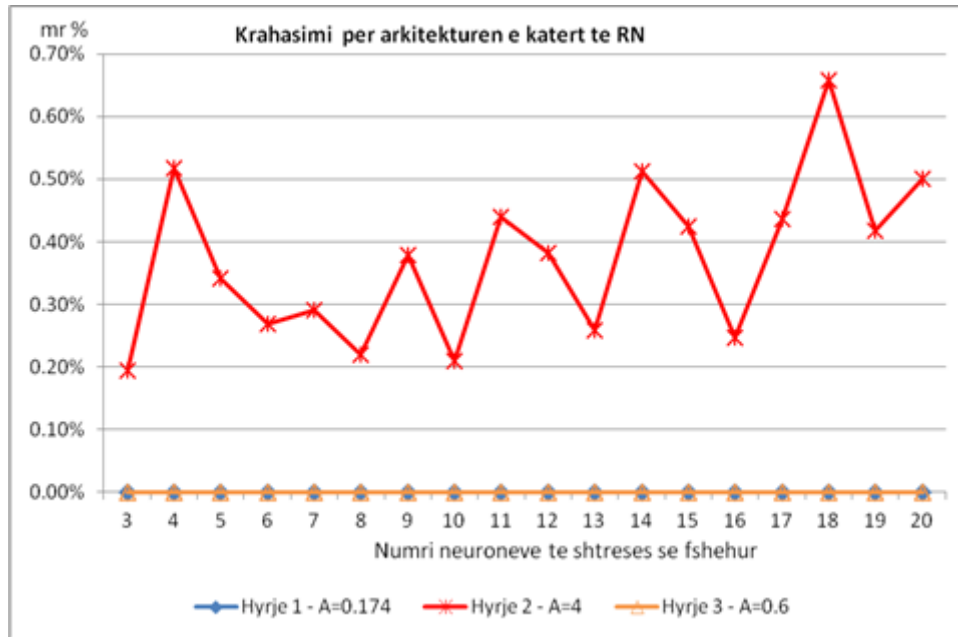


Figura 3.35. Mbirregullimi i sistemit të mbyllur për RN- A4 dhe trajnim të modelit neural me kampione të ndryshme H-D

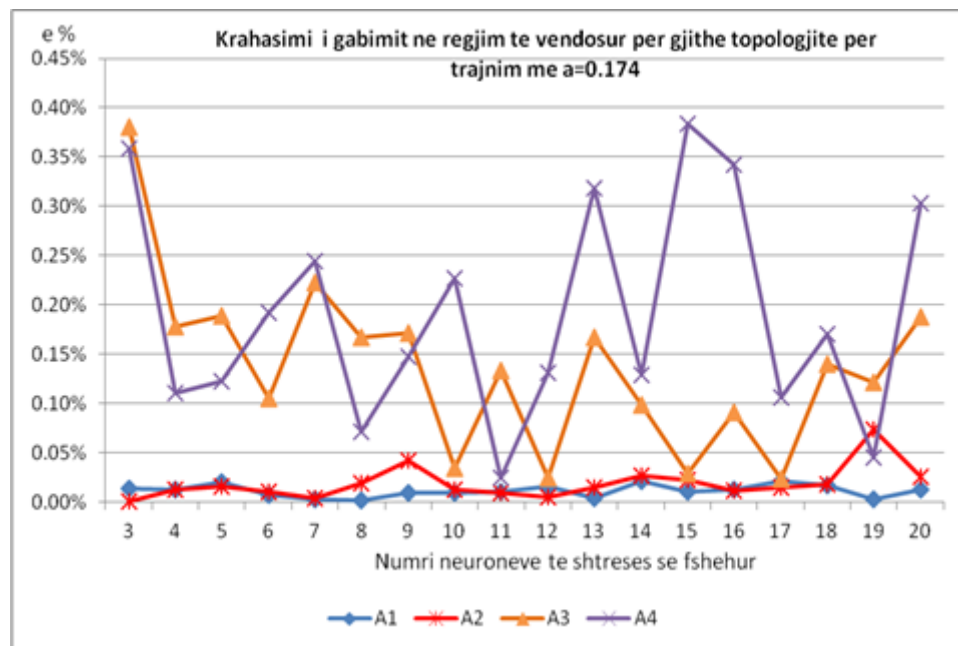


Figura 3.36 Gabimi i sistemit të mbyllur për gjithë arkitekturat dhe amplitudë të kampioneve H-D të trajnimit $a=0.174$

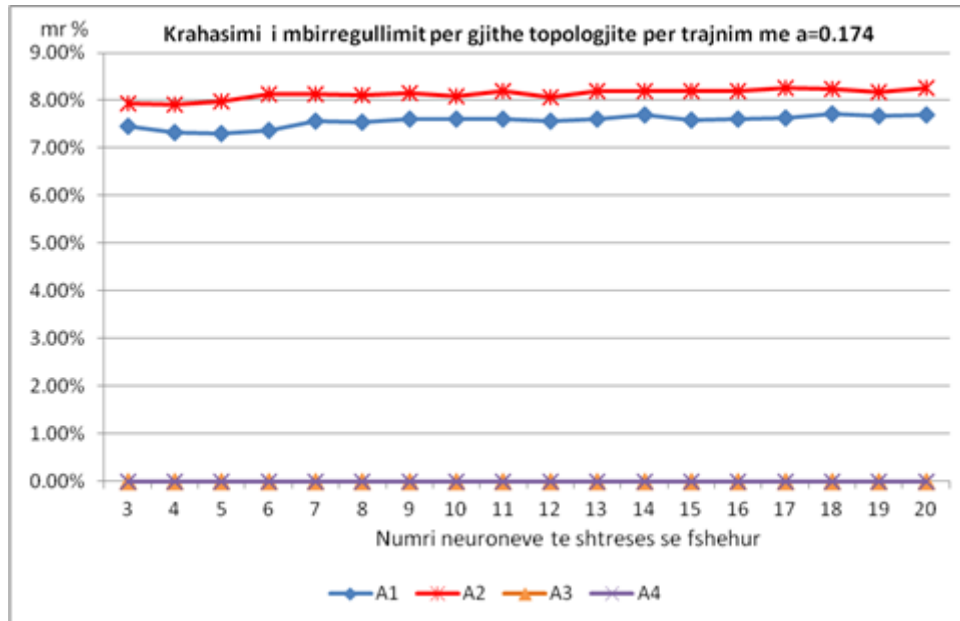


Figura 3.37 Mbirregullimi i sistemit të mbyllur për gjithë arkitekturat dhe amplitudë të kampioneve H-D të trajnimit $a=0.174$

Në secilin nga grafikët e figurave 3.28 deri në 3.35 jepen grafikisht ecuria e gabimeve në regjim të vendosur dhe mbirregullimeve për secilën arkitekturë kur modeli neural është trajnuar me grupet e ndryshme të kampioneve hyrje dalje të përshkruar në paragrafin 3.3.3, dhe neuronet e shtresës së fshehur ndryshojnë nga 3 në 20.

Vëzhgimet mund t'i ndajmë në grupe si më poshtë:

1. Ndikimi i kampioneve hyrje-dalje të përdorura për trajnimin e rrjetit

Nga figurat 3.28 deri në 3.35 vihet re që për gjithë arkitekturat, sa më afër amplitudës të sinjalit të komandës është amplituda e hyrjes e kampioneve hyrje-dalje të trajnimit aq më i mirë rregullimi; gabimi në regjim të vendosur dhe mbirregullimi janë në vlera më optimale. Gabimi në regjim të vendosur për arkitekturën e parë dhe të dytë është pothuaj i pandjeshëm nga amplituda e kampioneve të trajnimit.

2. Ndikimi i numrit të neuroneve të shtresës së fshehur

Nga grafikët e figurave 3.28÷3.37 shihet që rritja e numrit të neuroneve të shtresës së fshehur nuk sjell domosdoshmërisht përmirësim të cilësisë: mbi dimensionimi i rrjetit sjell edhe përkeqësim të cilësisë. Rritja e këtij numri sjell edhe kohë më të madhe trajnimi të modelit neural, gjë që u vu re gjatë simulimeve.

3. Ndikimi i tipit të variableve të shtresës së hyrjes të rrjetit neural

Nga krahasimi i figurave 3.28, 3.29 me 3.30, 3.31 (krahasimi i arkitekturës së parë me të dytën) vihet re që të dy arkitekturat japin rezultate krahasimisht të njëjta, me pak difference nga njëra tjetra. E njëjta gjë mund të thuhet nga krahasimi i figurave 3.32, 3.33 me 3.34, 3.35 (krahasimi i arkitekturës së tretë me të katërt). Pra ndryshimi nga 6 neurone në 5 për shtresën e hyrjes nuk duket të ketë ndikim të ndjeshëm në sjelljen e sistemit.

Nga një krahasim i gjithë rezultateve vëmë re ndryshime të dukshme midis dy grupeve të tipit të variableve të hyrjes të rrjetit neural. Këto u morën si:

1. A1 dhe A2 me ndryshesat dy, du plus sinjalet y dhe u në çastet e kohës $k+1$, k , $k-1$ dhe
2. A3 dhe A4 me sinjalet y dhe u në çastet e kohës $k+1$, k , $k-1$, $k-2$. A1 ka një variabël më tepër se A2 dhe po kështu A3 kundrejt A4.

Për treguesit e cilësisë në regjim të vendosur grupi i parë është i pothuaj i pandjeshëm kundrejt amplitudës të kampioneve hyrje-dalje të trajnimit, gabimi është gjithmonë zero ose shumë afër zeros (figurat 3.28 dhe 3.30). Grupi i dytë ka varësi nga këto kampione, për vlera të amplitudës afër daljes të dëshiruar të rrjetit neural, gabimi në regjim të vendosur është zero, kurse për vlera të mëdha të kësaj amplitude gabimi shkon deri në vlera 6% për një trajnim të mirë të rrjetit (figurat 3.32 dhe 3.34).

Për treguesit e cilësisë në regjim kalimtar grupi i parë ka gjithmonë mbirregullim që shkon nga 7.6% në 13% për ndryshim të amplitudës të kampioneve të trajnimit (figurat 3.29 dhe 3.30). Grupi i dytë jep mbirregullim zero për rastin e $a=0.174$ dhe 1.4% për $a=6$ (figurat 3.33 dhe 3.35).

Me zgjedhje të mirë të kampioneve hyrje-dalje të trajnimit arkitektura e tretë jep cilësi më të mirë të rregullimit, gabim dhe mbirregullim zero (figurat 3.36 dhe 3.37)

Nga sa më sipër mund të thuhet që tipi i zgjedhur i variableve të hyrjes të modelit neural dhe informacioni që japin kampionet e përdorura për trajnimin e rrjetit neural ndikojnë në cilësinë e rregullimit. [32], [36]

3.4 Krahasimi i kontrollit me model neural invers me kontrollin konvencional me rregullator PID

Pas vëzhgimit të simulimeve për identifikimin dhe kontrollin me anë të rrjeteve neurale, lind pyetja nëse kontrolli me model neural jep cilësi më të mirë kundrejt kontrollit me rregullatorë konvencionalë si për shembull PID.

Për të parë këtë, le të shohim kontrollin e shpejtësisë të motorit DC të modulit G14 EV me dy metodat, kontrollin me rregullator PID dhe kontrollin me model neural invers në skemën e sistemit të mbyllur.

Treguesit e cilësisë që kërkojmë të arrijmë janë:

Gabimi në regjim të vendosur: $e = 0$;

Mbirrregullimi në % $m_r < 15\%$

Koha e vendosjes: $t_v \leq 0.25$ sek

Modeli i motori DC është dhënë në paragrafin 3.3.1.

Dy metodat e kontrollit shtjellohen si vijon në paragrafin 3.4.1 dhe 3.4.2.

3.4.1 Kontrolli me rregullator PID

Blokskema e sistemit të mbyllur me rregullator PID jepet në Figurën 3.38, ku $G(s)$ është funksioni transmetues i motorit të rrymës së vazhduar DC dhe $G_R(s)$ është funksioni transmetues i rregullatorit PID dhënë në mënyrë të përgjithshme nga ekuacioni 3.14 [132]–[134]

$$G_R(s) = K_p \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \quad (3.14)$$

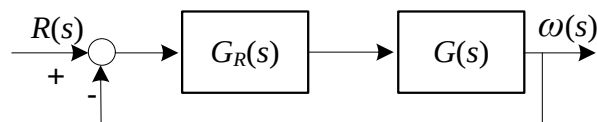


Figura. 3.38 Blokskema e sistemit të mbyllur me rregullator PID

Tarojmë parametrat e rregullatorit PID me anë të *pidtool* të Matlab, si në Figurën 3.39. Konstantet e rregullatorit, të gjetura me anë të kësaj metode janë:

$$K_p = 0.0358; T_i = 0.0154; T_d = 5.16 \cdot 10^{-6}$$

Funksioni i transmetimit të sistemit të mbyllur është:

$$H(s) = \frac{1.108 \cdot 10^{-8} s^2 + 0.002s + 0.139}{2.26 \cdot 10^{-7} s^3 + 0.0003s^2 + 0.007s + 0.139} \quad (3.15)$$

Funksioni transmetues diskret i sistemit të mbyllur, për periodë kampionimi $T_s = 0.01$ [133][xx] është:

$$H(z) = \frac{3.326 \cdot 10^{-5} z^3 + 0.106z^2 - 0.054z - 0.006}{z^3 - 1.721z^2 + 0.773z - 0.006} \quad (3.16)$$

Treguesit e cilësisë të sistemit të mbyllur të përfutur nga rregullatori i taruar PID janë:

Koha e rritjes: $t_r = 0.074$ sekonda

Koha e vendosjes: $t_v = 0.247$ sekonda

Mbirregullimi: 11%

Gabimi në regjim të vendosur: $e = 0$

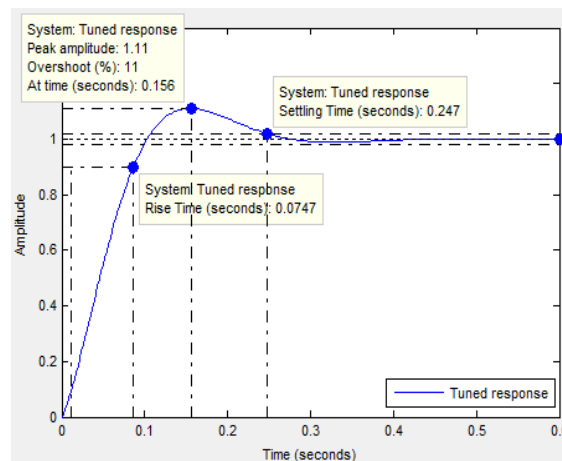


Figura 3.39 Tarimi i rregullatorit PID me anë të pidtool të Matlab

Përgjigja e sistemit të mbyllur ndaj sinjalit shkallë njësi jepet në figurën 3.40

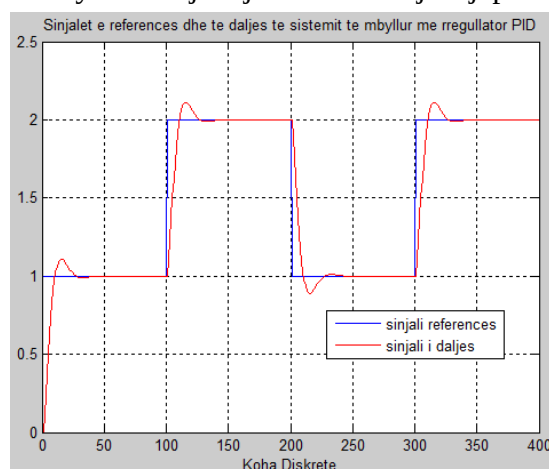


Figura 3.40 Përgjigja e sistemit të mbyllur me rregullator PID

Nga rezultatet e marra, shihet që sistemi i mbyllur me rregullatorin PID të taruar kënaq treguesit e kërkuar të cilësisë.

3.4.2 Kontrolli me model neural invers

Zëvendësojmë rregullatorin PID me modelin neural invers të motorit DC. Bllokskema e sistemit të mbyllur të kontrollit do jetë si në Figurën 3.13, dhënë në paragrafin 3.2.

Tentojmë të gjemë një model neural, që i vendosur në skemën e sistemit të mbyllur, jep tregues cilësie më të mirë se rregullatori PID.

Arkitekturën e rrjetit neural të modelit invers e zgjedhim si në figurën 3.2, paragrafi 3.1.2; rrjet tre-shtresor $6 \times m \times 1$ me përhapje të pasëm, ku m fillesatare është $m=4$, dhe m përfundimtare përcaktohet gjatë trajnimit.

Trajnimi i rrjetit kryhet jashtë linje (*offline*) sipas metodës të dhënë në paragrafin 3.3, figura 3.16.

Kampionet hyrje dalje të përdorura për trajnimin e rrjetit, jepen në figurën 3.41.

Pas disa tentativash marrim një model neural me arkitektur $6 \times 6 \times 1$. Gabimi i trajnimit i marrë për këtë rrjet është:

gabimi maksimal : 0.0075095

varianca e gabimit : 0.016036

Figurat 3.41 deri ne 3.43 japin në mënyrë grafike rezultatet e trajnimit të rrjetit.

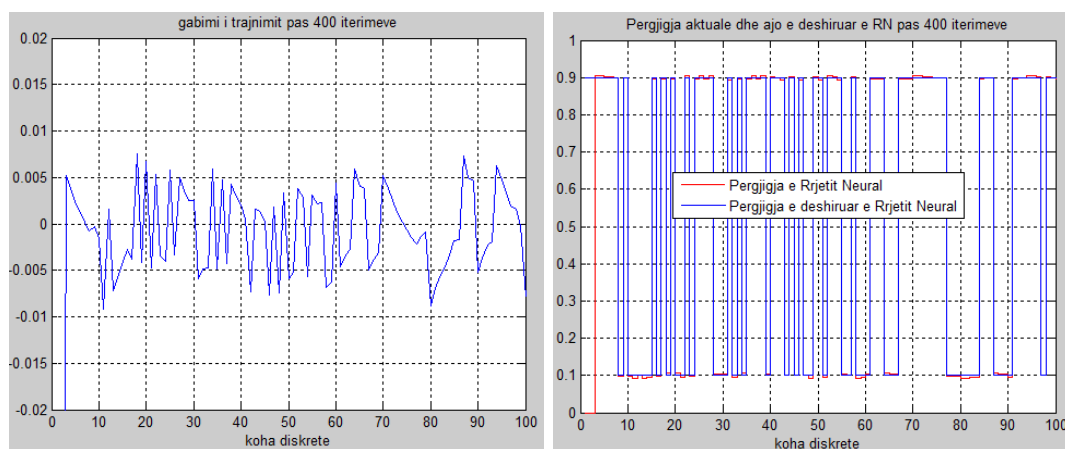


Figura 3.41 a) ecuria e gabimit të trajnimit b) dalja e modelit neural invers të trajnuar kundrejt daljes së dëshiruar

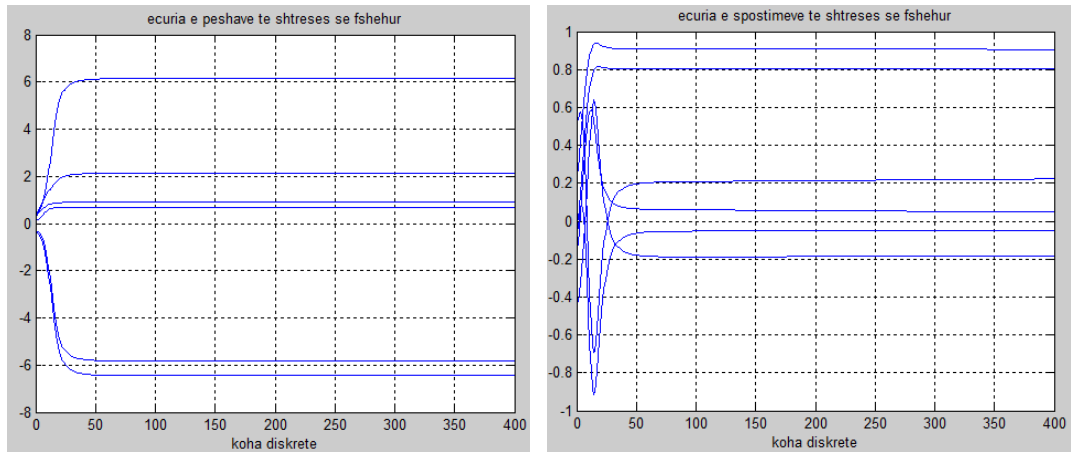


Figura 3.42 Ecuria e peshave dhe spostimeve të shtresës së fshehur gjatë trajnimit të modelit neural

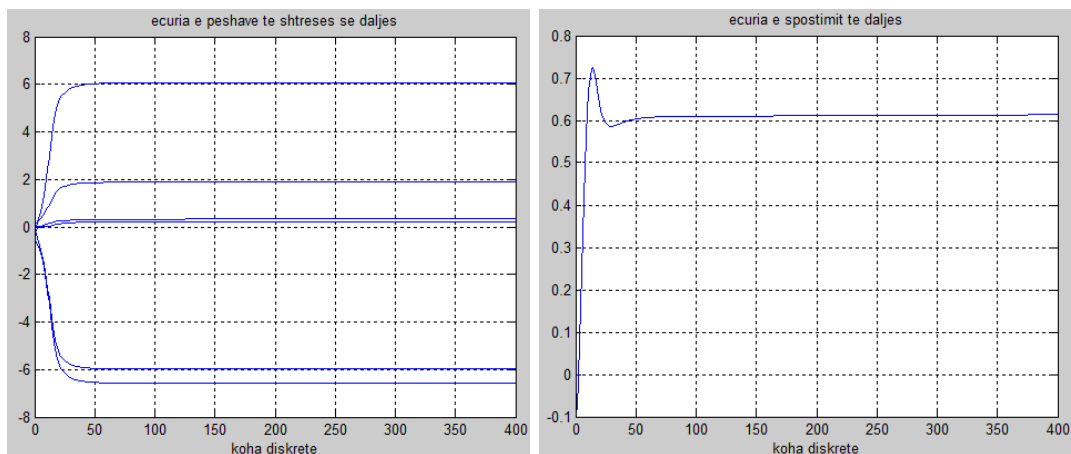


Figura 3.43 Ecuria e peshave dhe spostimit të shtresës së daljes gjatë trajnimit të modelit neural

Modelin neural invers të trajnuar e vendosim në skemën e sistemit të mbyllur dhe kryejmë simulimin për sinjal hyrje të sistemit shkallë. Rezultatet e marra jepen në figurën 3.44, ku paraqiten ecuria e sinjalit të daljes të sistemit dhe e sinjalit të komandës që është dalja e modelit neural invers.

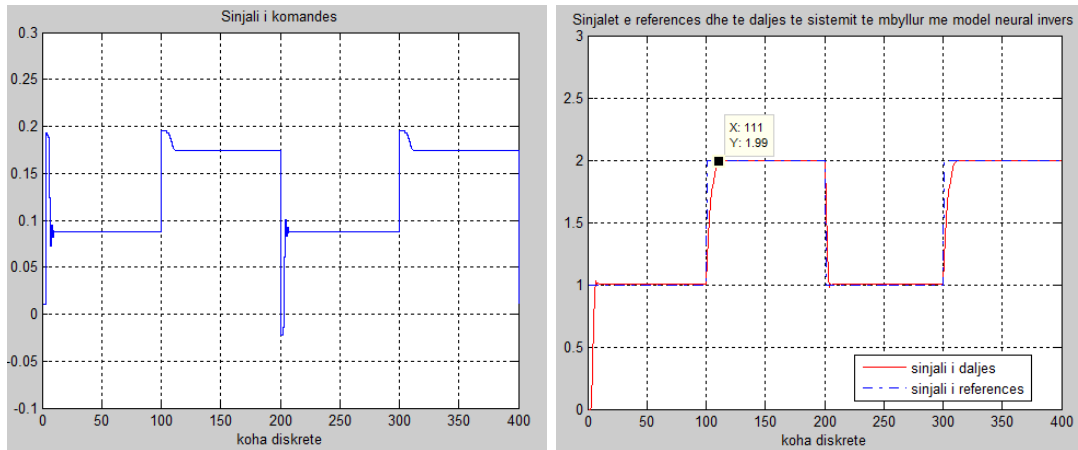


Figura 3.44 a) dalja e modelit neural invers në sistemin e mbyllur b) përgjigja e sistemit të mbyllur me model neural invers

Nga rezultatet shihet që sistemi i mbyllur ka gabim në regjim të vendosur $e = 0$, mbirregullim $m_r = 0\%$, dhe kohë vendosje $t_v = 0.11$ sekonda.

Këta tregues cilësie janë më të mirë se treguesit e cilësisë të kërkuar.

3.4.3 Krahassimi i dy metodave të kontrollit

Për krahasimin e rezultateve të marra për të dy metodat, vendosim:

- në tabelën 3.1 treguesit e cilësisë të dëshiruar dhe ata të marrë për secilën metodë,
- në figurën 3.45 përgjigjet e sistemit të mbyllur kundrejt daljes së dëshiruar.

Tabela 3.1 Treguesit e cilësisë të dëshiruar dhe të marrë nga simulimet

Treguesi i cilësisë	Vlera e dëshiruar e treguesit	Vlera e treguesit për rregullator PID	Vlera e treguesit për model neural invers
Gabimi në regjim të vendosur (e)	0	0	0
Mbirregullimi (m_r)	< 15%	11%	0%
Koha e vendosjes (t_v)	≤ 0.25 sek	0.247 sek	0.11 sek

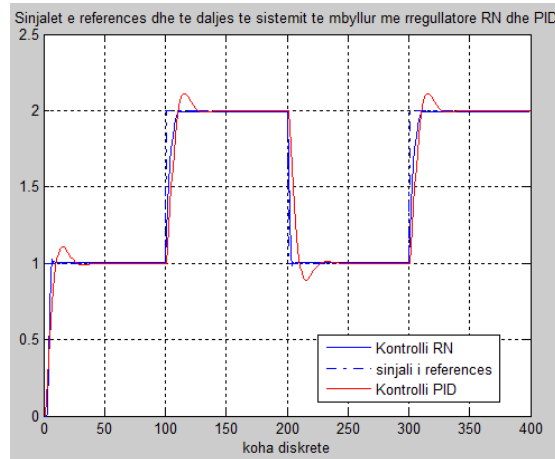


Figura 3.45 Krahاسimi i përgjigjeve të sistemit të mbyllur për kontrollin me PID dhe me model neural invers

Siç vihet re nga tabela 3.1 dhe figura 3.45, skema e kontrollit me model neural invers kënaq më mirë kërkesat për cilësinë e sistemit të mbyllur, kundrejt skemës me kontroll PID. Ndaj mund të themi që rrjetet neurale mund të shihen si një mënyrë alternative për të përmirësuar dinamikën e sistemit të mbyllur.

3.5 Vëzhgimi i cilësisë të sistemit të mbyllur me model neural invers pas shtimit të elementit integrues

Në kërkim të mënyrave për përmirësimin e dinamikës të sistemit të mbyllur, kur modeli neural nuk është trajnuar mjaftueshëm për të kënaqur treguesit e cilësisë, vëzhgojmë sjelljen e sistemit të mbyllur me model neural invers kur shtohet një element integrues.

Sistemi i mbyllur kontrollon shpejtësinë e motorit DC të modulit G14 EV, modeli i të cilit është dhënë në paragrafin 3.3.1.

Së pari trajnojmë një rrjet neural për skemën e kontrollit me model neural invers si në figurën 3.13.

Më pas shtojmë elementin integrues në skemën e sistemit të mbyllur si në figurën 3.46 [130] dhe vëzhgojmë rezultatet.

Treguesit e cilësisë që kërkojmë të arrijmë janë:

Gabimi në regjim të vendosur: $e = 0$;

Mbirregullimi në % $m_r < 15\%$

Koha e vendosjes: $t_v \leq 0.25$ sek

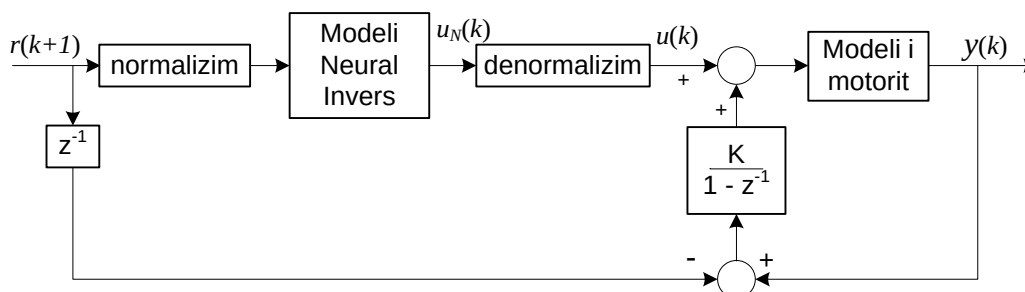


Figura 3.46 Skema strukturore e sistemit të mbyllur të kontrollit me model neural invers plus integrim

3.5.1 Kontrolli me model neural invers

Kjo skemë kontrolli është trajtuar tashmë në paragrafët 3.2, 3.3, dhe 3.4. Metodologjia e ndjekur në këtë paragraf është e njëjtë.

Arkitektura e rrjetit neural zgjidhet sipas figurës 3.19a. Kampionet hyrje-dalje të përdorur për trajnimin e modelit neural invers jepen në figurën 3.47.

Kryejmë vetëm një grup të vogël iterimesh dhe tentativash për trajnimin e rrjetit neural. Arkitektura e marrë ka 6 neurone në shtresën e fshehur, pra ka arkitekturën $6 \times 6 \times 1$. Gabimi i trajnimit me anë të algoritmit me përhapje të pasme është:

Gabimi maksimal: 0.0078266

Varianca e gabimit : 0.016037

Figurat 3.48 deri në 3.50 japin përkatësisht:

- ecurinë e gabimit të trajnimit dhe daljen e rrjetit neural kundrejt daljes të dëshiruar për këtë rrjet.
- ecurinë e ndryshimit të peshave të shtresës së hyrjes dhe të fshehur gjatë trajnimit të rrjetit neural
- ecurinë e ndryshimit të spostimeve të shtresës të fshehur dhe të daljes gjatë trajnimit të rrjetit neural

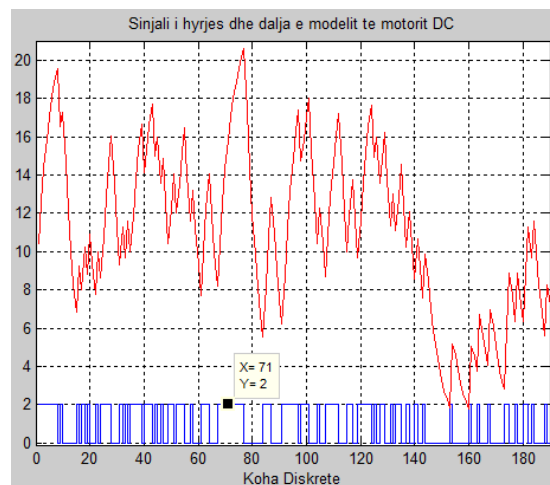


Figura 3.47 kampionet H-D për trajnimin e modelit neural invers

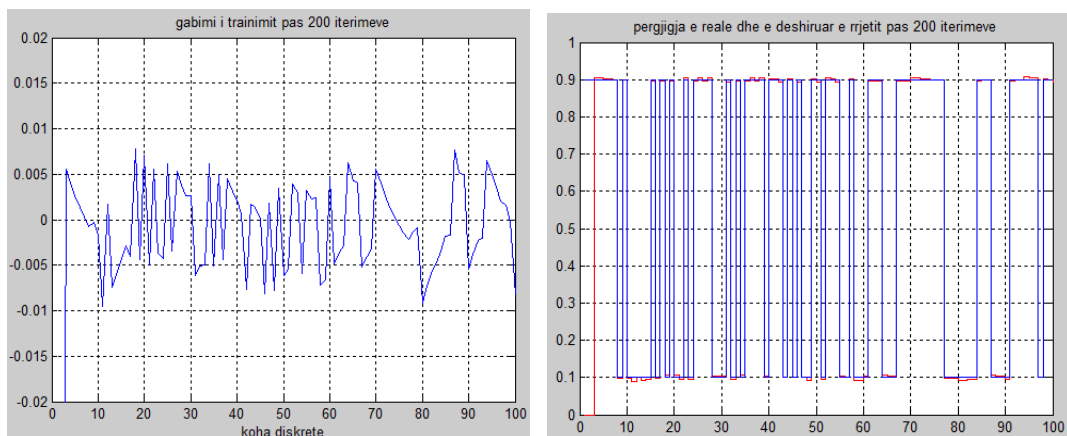


Figura 3.48 a) ecuria e gabimit të trajnimit b) dalja e modelit neural invers të trajnuar kundrejt daljes së dëshiruar

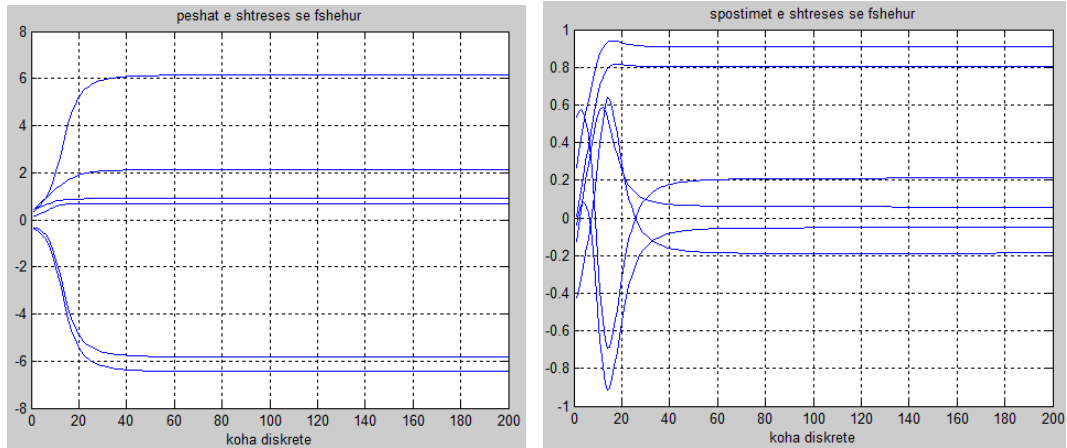


Figura 3.49 ecuria e ndryshimit të peshave dhe spostimeve të shtresës së fshehur

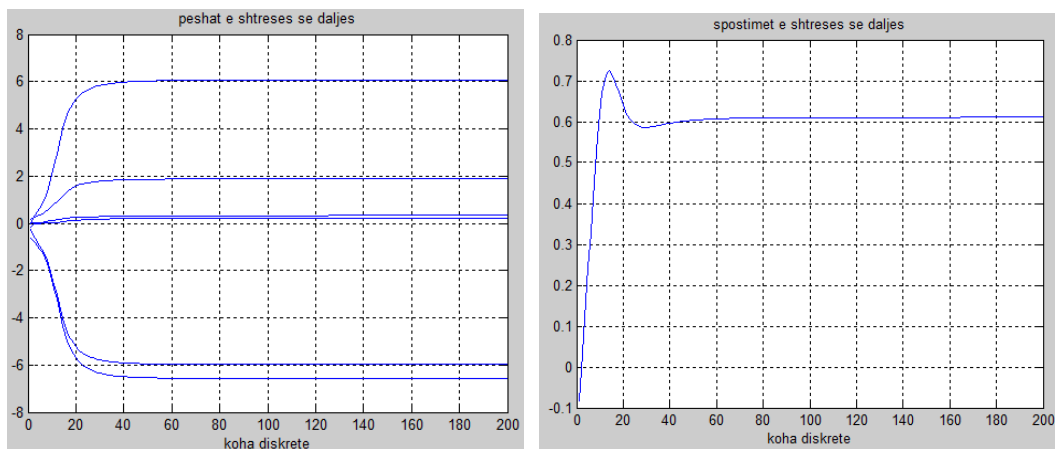


Figura 3.50 ecuria e ndryshimit të peshave dhe spostimit të shtresës së daljes

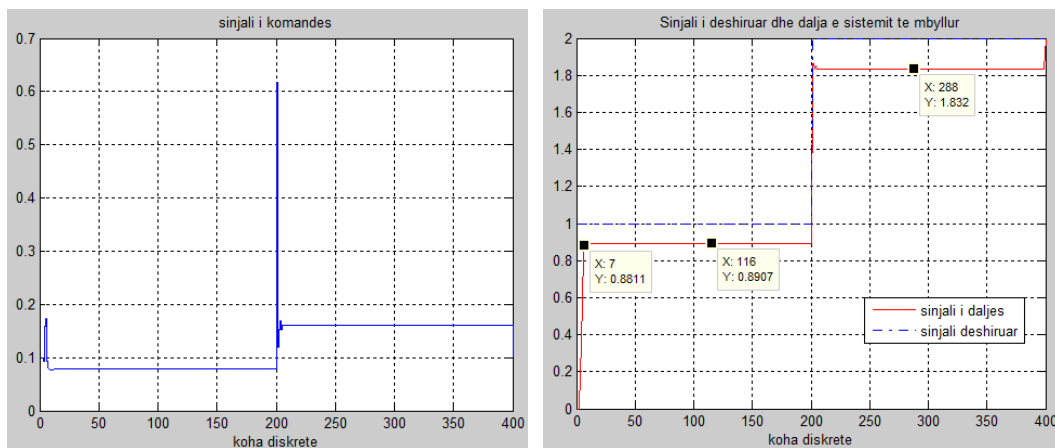


Figura 3.51 a) ecuria e daljes së modelit neural invers në skemën e mbyllur b) dalja e sistemit të mbyllur të kontrollit

Modeli neural invers i trajnuar, i vendosur në skemë e mbyllur jep rezultatet si në figurën 3.51. Figura 3.51a paraqet daljen e modelit neural invers që është sinjali i komandës.

Siç shihet nga figura 3.51b, gabimi në regjim të vendosur është $e=12\%$ për amplitudë njësi të sinjalit të hyrjes. Kjo vlerë nuk kënaq treguesit e cilësisë të kërkuar.

Mbirregullimi është pothuaj i papërfillshëm dhe koha e vendosjes është vetëm 0.07 sekonda.

3.5.2 Kontrolli me model neural invers dhe element integrues në sistemin e mbyllur

Modelin neural invers të trajnuar në 3.5.1, e vendosim në skemën e sistemit të mbyllur sipas Figurës 3.46.

Me ndihmën e Simulink, ndërtojmë sistemin e mbyllur, si në Figurën 3.52. Blloqet e shtresës së fshehur dhe të daljes në Simulink janë ndërtuar si në Figurën 3.53.

Dalja e sistemit të mbyllur dhe ecuria e sinjalit të komandës që është dalja e modelit neural invers, jepen në figurën 3.54.

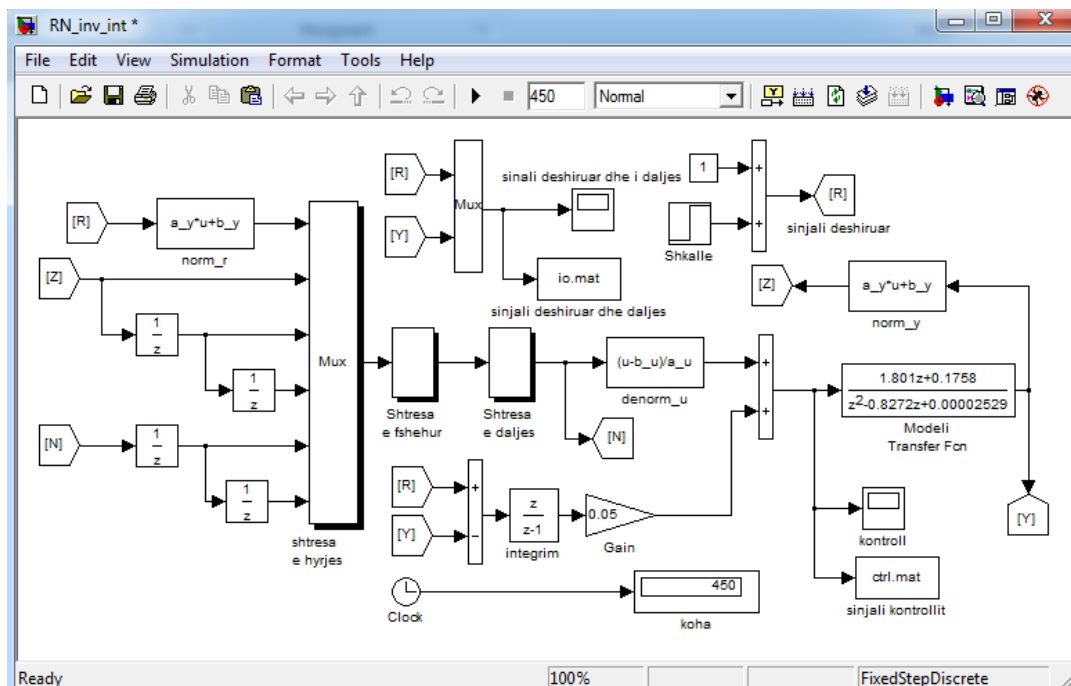


Figura 3.52 modeli në Simulink i sistemit të mbyllir të kontrollit me model neural invers dhe element integrues

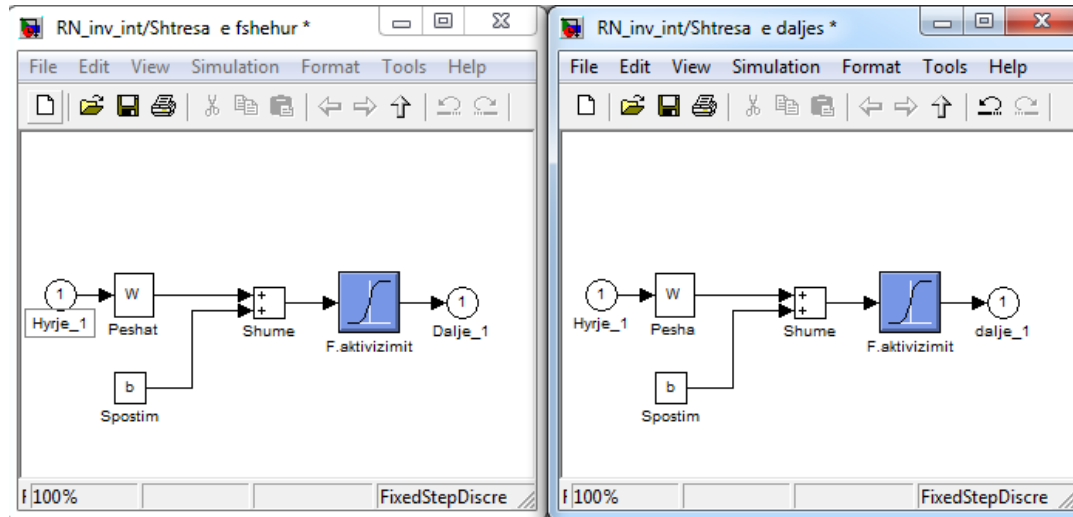


Figura 3.53 blloqet e shtresës së fshehur dhe të daljes të modelit neural në Simulink

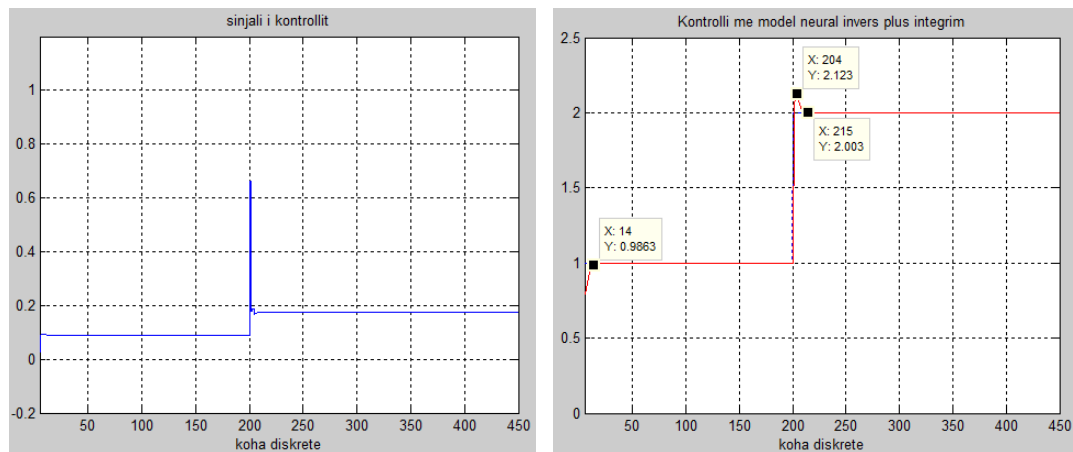


Figura 3.54 a) ecuria e daljes së modelit neural invers në skemën e mbyllur b) dalja e sistemit të mbyllur të kontrollit me element integrues

Siç shihet nga figura 3.54, gabimi në regjim të vendosur është $e=0$, mbirregullimi $m_r=6\%$, dhe koha e vendosje është 0.15 sekonda.

3.5.3 Krahasimi i dy skemave të kontrollit

Për krahasimin e cilësisë të sistemeve të trajtuara në 3.5.1 dhe 3.5.2, vendosim rezultatet e marra në tabelën 3.2 dhe paraqesim në të njëjtin graf si në figurën 3.55 daljet e të dy sistemeve.

Tabela 3.2 Krahاسimi i treguesve të cilësisë për dy skemat e kontrollit në 3.5.1 dhe 3.5.2

Treguesi i cilësisë	Vlera e dëshiruar e treguesit	Vlera e treguesit për model neural invers	Vlera e treguesit pas shtimit të elementit integrues
Gabimi në regjim të vendosur (e)	0	12%	0
Mbirregullimi (m_r)	< 15%	~0	6%
Koha e vendosjes (t_v)	≤ 0.25 sek	0.07 sek	0.15 sek

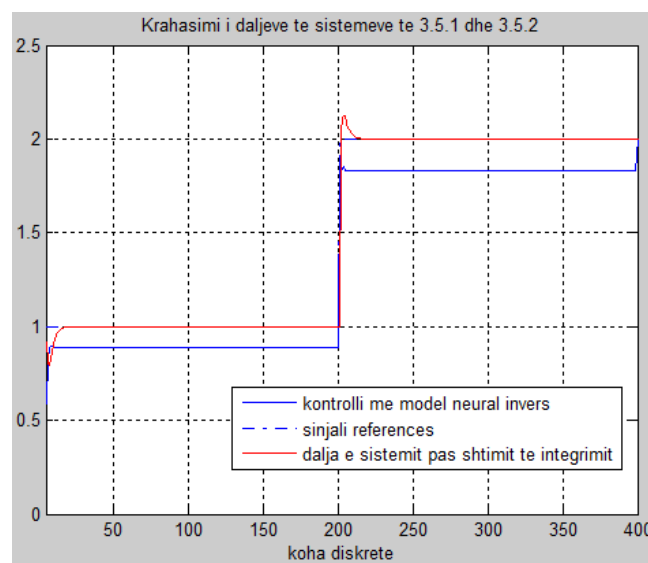


Figura 3.55 daljet e sistemeve të mbyllur me model neural invers dhe model neural invers plus integrim

Vihet re që shtimi i elementit integrues zbrit gabimin në regjim të vendosur nga 11% në 0%, dhe rrit lehtësisht mbirregullimin dhe kohën e vendosur. Këta tregues janë brenda vlerave të kërkuara.

Ashtu si edhe në metodat konvencionale të kontrollit, mund të thuhet që elementi integrues është një mjet për të marrë vlerë zero të gabimit në regjim të vendosur.

3.6 Kontrolli i shpejtësisë të motorit DC me model neural invers dhe ndryshim në linjë të parametrave të rrjetit neural

Një mënyrë tjetër për përmirësimin e dinamikës të sistemit të mbyllur është ndryshimi në linjë i parametrave të rrjetit neural që përbën modelin neural invers të motorit DC [30], [31], [130].

Përsëri, sistemi i mbyllur do kontrollojë shpejtësinë e motorit DC të modulit G14 EV, me model si në paragrafin 3.3.1.

Për të vëzhguar ndikimin pozitiv ose jo të ndryshimit në linjë të parametrevë të rrjetit neural, së pari marrim skemën e kontrollit me model neural invers si në figurën 3.13. Rrjeti neural nuk trajnohet mjaftueshëm jashtë linjë. Qëllimi është të kemi një cilësi jo të mirë me anë të kontrollit me model neural invers, që mund të përmirësohet me trajnimin në linjë të rrjetit neural.

Bllokskema e sistemit të mbyllur të kontrollit me ndryshim në linjë të modelit neural invers jepet në figurën 3.56.

Si referencë marrim të njëjtët tregues cilësie të dëshiruar:

Gabimi në regjim të vendosur: $e = 0$;

Mbirregullimi në % mr: $< 15\%$

Koha e vendosjes: $t_v \leq 0.25$ sek

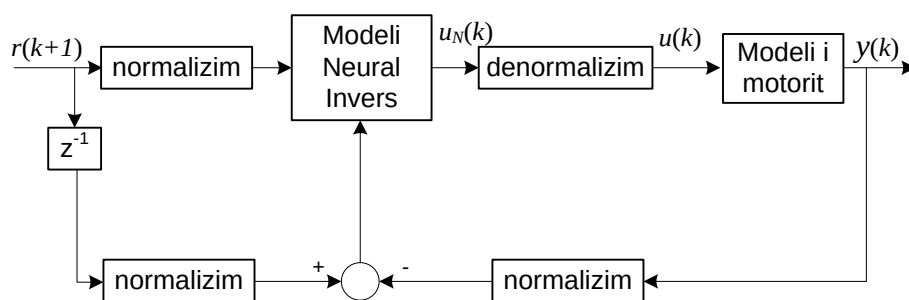


Figura 3.56 Bllokskema e kontrollit me model neural invers dhe ndryshim në linjë të parametrave të modelit neural

3.6.1 Kontrolli me model neural invers të trajnuar jashtë linje

Skema e kontrollit është trajtuar në paragrafët 3.2 deri në 3.5. Metodologjia e ndjekur këtu është e njëjtë.

Arkitektura e rrjetit neural merret si në figurën 3.19a. Numri i neuroneve të shtresës së fshehur zgjidhet i vogël, $m=4$, për të patur sa më pak elemente procesues, pra

përpunim sa më të shpejtë të informacionit për skemën e ndryshimit në linjë të parametrave të rrjetit.

Kampionet hyrje-dalje të modelit të motorit DC, të përdorura për trajnimin e modelit neural invers jepen në figurën 3.57. Amplituda e sinjalit të hyrjes merret e madhe $A=6$, dhe larg vlerave të vëzhguara të komandës në sistemin e mbyllur.

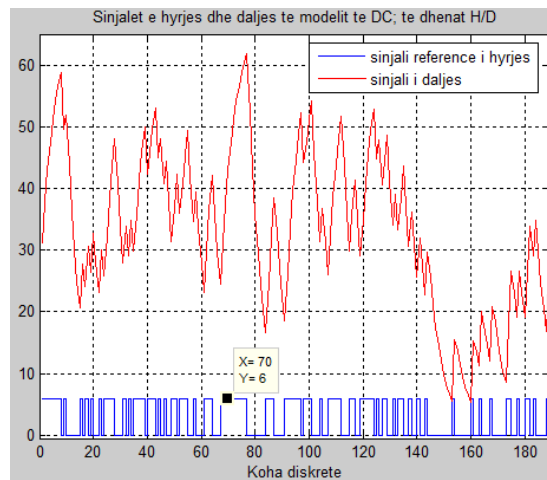


Figura 3.57 Kampionet e përdorura për trajnimin jashtë linjë të rrjetit neural

Kryejmë vetëm një cikël trajnimi të rrjetit neural pa tentuar të marrim cilësi të mirë të përgjigjes të sistemit të mbyllur. Gabimi i trajnimit është:

gabimi maksimal : 0.0078452

varianca e gabimit : 0.016034

Rezultatet e trajnimit të rrjetit jepen në figurat 3.58 deri në 3.60

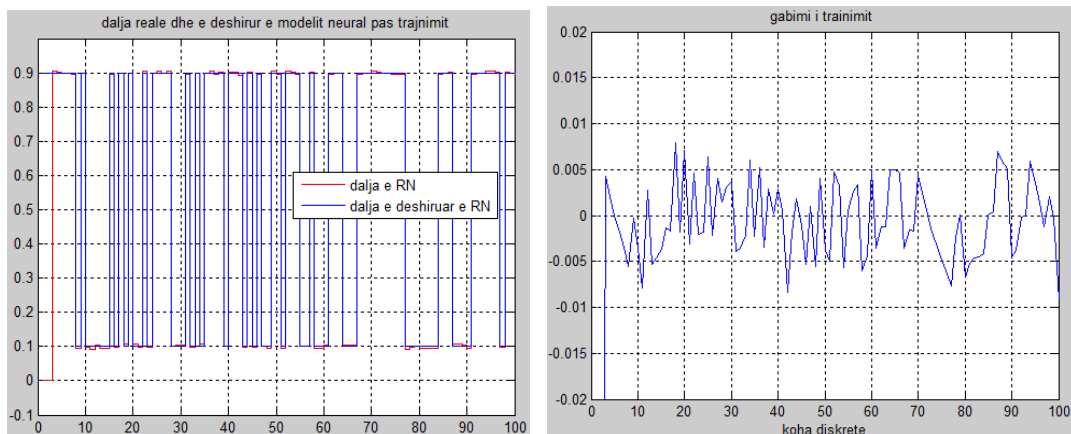


Figura 3.58 a) dalja e dëshiruar dhe reale e rrjetit neural b) ecuria e gabimit të trajnimit

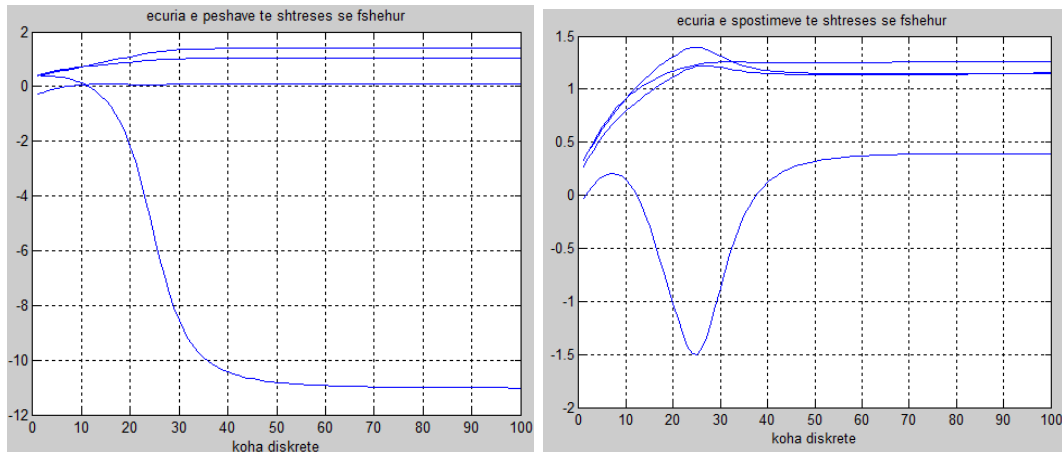


Figura 3.59 ecuria e peshave dhe spostimeve të shtresës së fshehur të rrjetit neural

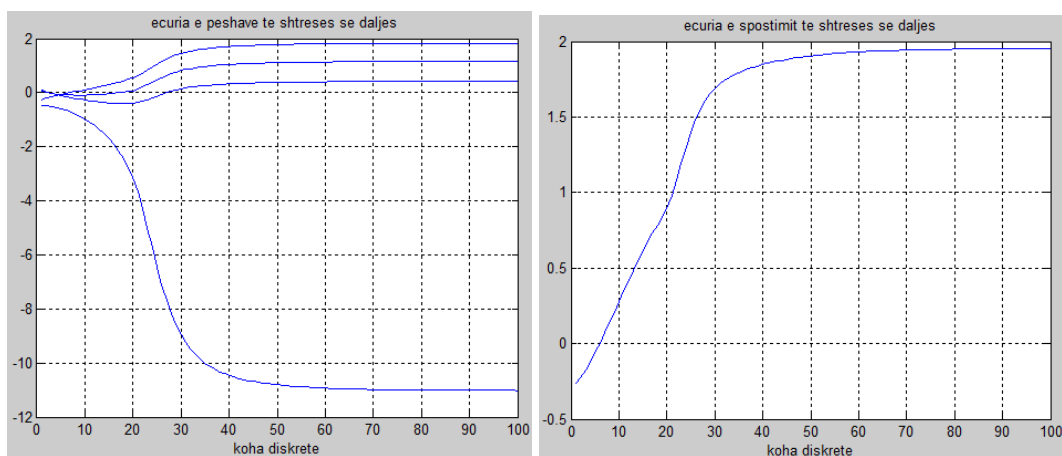


Figura 3.60 ecuria e peshave dhe spostimit të shtresës së daljes të rrjetit neural

Rrjetin neural të trajnuar e vendosim në skemën e mbyllur të kontrollit sipas figurës 3.13 dhe kryejmë simulimin për hyrje shkallë njësi. Dalja e modelit neural invers, që është sinjali i komandës, dhe dalja e sistemit të mbyllur jepen në figurën 3.61

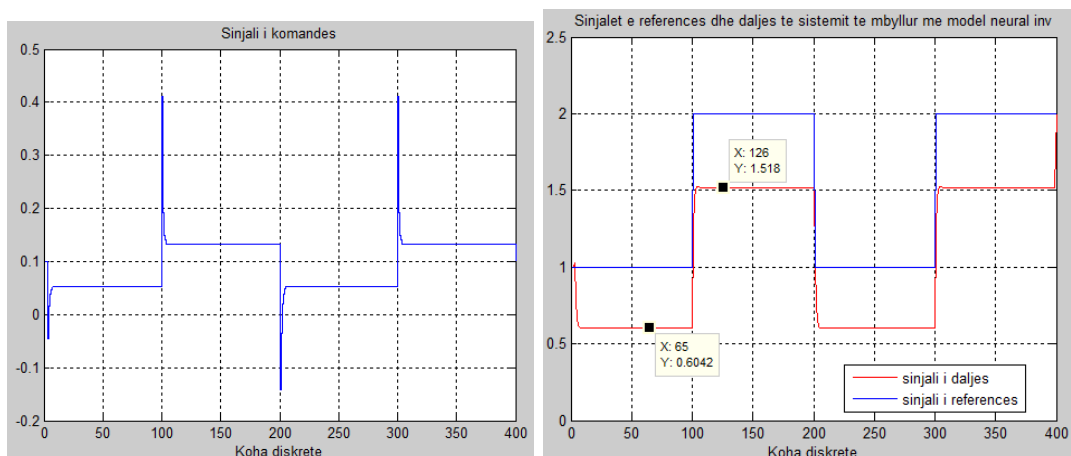


Figura 3.61 Dalja e modelit neural invers dhe dalja e sistemit të mbyllur

Siç shihet nga rezultati, cilësia e sistemit të mbyllur nuk është e kënaqshme, gabimi në regjim të vendosur është ~40% për hyrje njësi dhe ~25% për amplitudë të hyrjes 2 njësi.

3.6.2 Kontrolli me model neural invers të trajnuar në linjë

Modelin neural invers të trajnuar në 3.6.1 e vendosim në skemën e sistemit të mbyllur si në figurën 3.56.

Modeli neural invers trajnohet në linjë me anë të algoritmit me përhapje të pasme; gabimi përhapet në rrjetin neural në çdo çast kohe dhe peshat dhe spostimet e modelit neural ndryshohen në linjë sipas algoritmit në fjalë.

Figurat 3.62 deri në 3.65 paraqesin rezultatet e marra. Vihet re që cilësia e sistemit të mbyllur është e mirë, gabimi në regjim të vendosur anulohet, mbirregullimi është zero, dhe koha e vendosjes është $t_v=0.18$ sekonda. Kjo për të dy format e sinjalit në hyrje, shkallë njësi dhe sinusoidale. Të gjithë këta tregues janë më të mirë ose të njëjtë me treguesit e kërkuar.

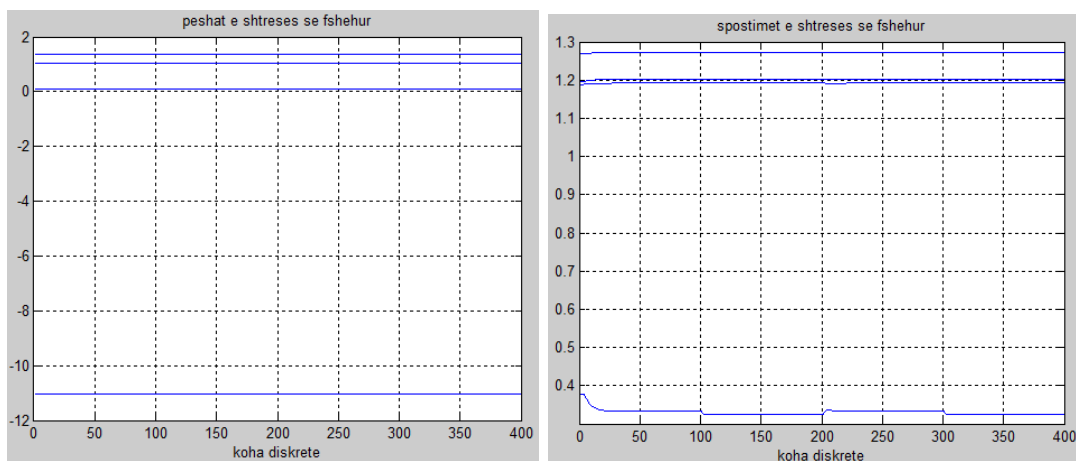


Figura 3.62 ecuria e peshave dhe spostimeve të shtresës së fshehur të ndryshuara në linjë

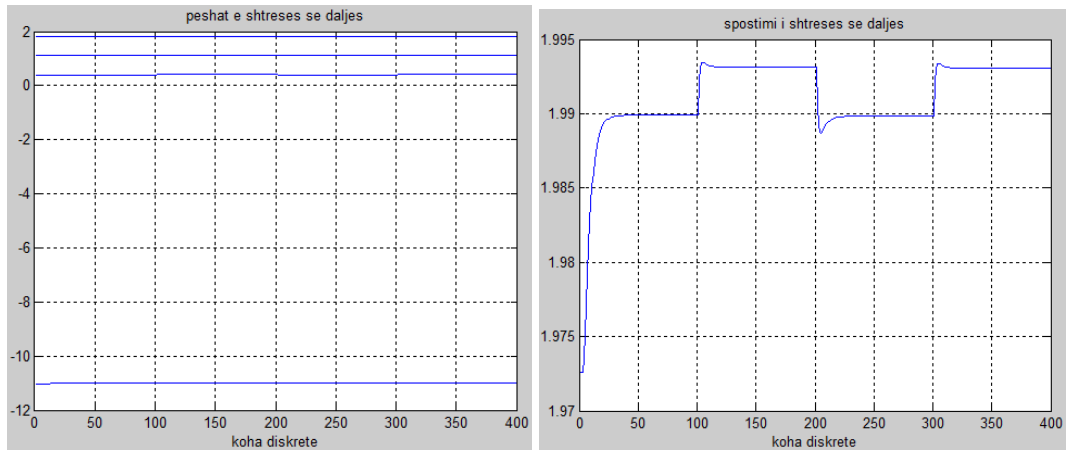


Figura 3.63 ecuria e peshave dhe spostimit të shtresës së daljes të ndryshuara në linjë

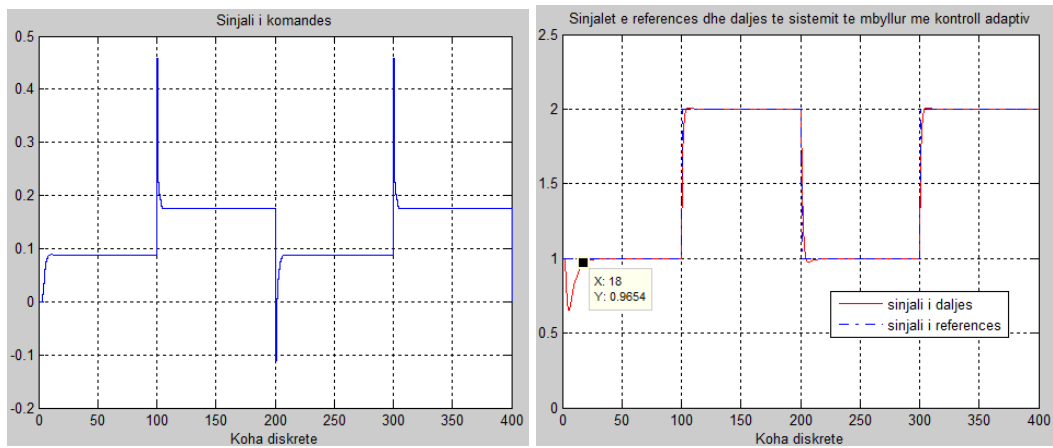


Figura 3.64 Dalja e modelit neural invers dhe dalja e sistemit të mbyllur për përshtatjen në linjë të peshave dhe spostimeve të rrjetit neural

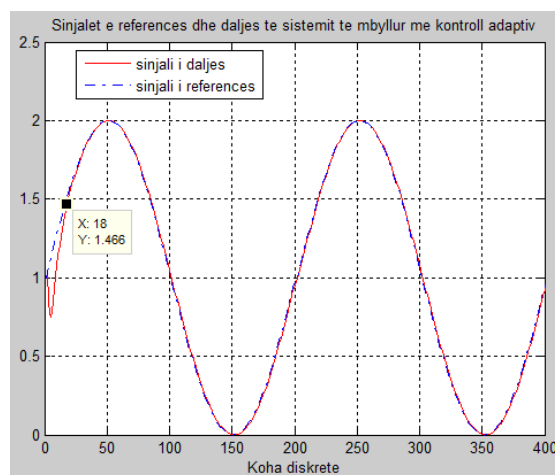


Figura 3.65 Dalja e sistemit të mbyllur për sinjal reference sinusoidal

3.6.3 Krahasimi i rezultateve

Rezultatet e simulimeve në dy paragrafët më sipër, pasqyrohen përmbledhtazi në tabelën 3.3 dhe figurën 3.66

Tabela 3.3 Krahasimi i treguesve të cilësisë për dy skemat e kontrollit në 3.6.1 dhe 3.6.2

Treguesi i cilësisë	Vlera e dëshiruar e treguesit	Vlera e treguesit për model neural invers	Vlera e treguesit për ndryshim në linjë të parametrave të RN
Gabimi në regjim të vendosur (e)	0	40% dhe 25%	0
Mbirregullimi (m_r)	< 15%	~0	~0%
Koha e vendosjes (t_v)	≤ 0.25 sek	0.07 sek	0.18 sek

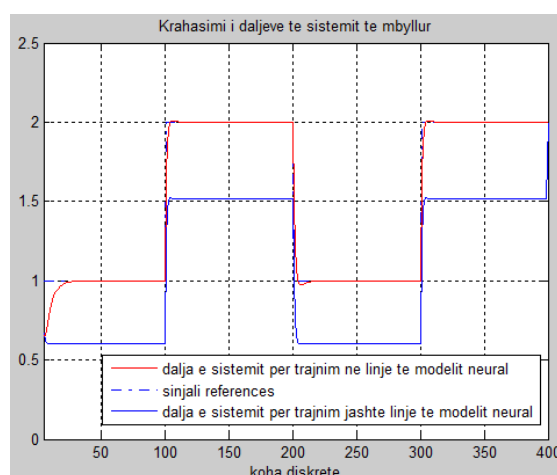


Figura 3.66 Daljet e sistemeve të mbyllura për modelin neural të trajnuar jashtë linje dhe në linjë

Numri i neuroneve që përbëjnë modelin neural në paragrafin 3.6.1, pra i elementeve procesues, u mor i vogël dhe u krye vetëm një cikël trajnimi. Me fjalë të tjera u krye vetëm një përcaktim fillestar i parametrave të modelit neural invers. Kjo rezultoi në një cilësi jo të mirë të kontrollit.

Pavarësisht kësaj, nga rezultatet e marra, vëzhgohet që ndryshimi në linjë i parametrave të modelit neural përmirëson ndjeshëm dinamikën e sistemit të mbyllur; gabimi në regjim të vendosur anulohet, koha e vendosjes rritet me 0.11 sekonda, por mbetet përsëri brenda treguesve të kërkuar të cilësisë.

Nga krahasimi i figurave 3.62 dhe 3.63 vihet re që gjatë trajnimit në linjë peshat e shtresave të modelit mbesin të pandryshueshme, ndërsa spostimet e shtresave ndryshojnë.[130]

Mund të thuhet që ndryshimi në linjë i parametrave të modelit neural është një mënyrë e mirë për optimizim të elementeve përpunues të rrjetit dhe përmirësim të dinamikës të sistemit të mbyllur. Optimizimi i numrit të neuroneve sjell lehtësi në implementimin hardware ose software të rrjetit. Nga ana tjetër, për implementimin e ndryshimit në linjë të parametrave të modelit neural të ndërfutur në një sistem të mbyllur kontrolli, duhen parë aspekte të ndryshme të implementimit disa prej të cilave janë dinamika e sistemit, kosto e implementimit, nevoja për këtë lloj ndryshimi.

3.7 Reagimi ndaj shqetësimeve i sistemit të mbyllur me model neural invers

Si pjesë të fundit të simulimeve të kryera, le të shohim reagimin ndaj shqetësimeve të sistemit të mbyllur me model neural invers.

Le të injektojmë shqetësime në dy raste të shqyrtuara më sipër, sistemin e mbyllur me model neural invers dhe atë me model neural invers me ndryshim parametrash në linjë.

Reagimin e sistemit e shohim për sinjal hyrje shkallë njësi dhe sinjal sinusoidal, për të parë nëse reagimi ndaj shqetësimit varet edhe nga forma e sinjalit.

Për rastin e parë marrim sistemin e mbyllur të përfutur në paragrafin 3.4.2, dhe për rastin e dytë sistemet e përfutur në paragrafët 3.6.1 dhe 3.6.2.

Rezultatet e simulimeve jepen përkatësisht në figurën 3.67 për rastin e parë dhe në figurën 3.68 për rastin e dytë.

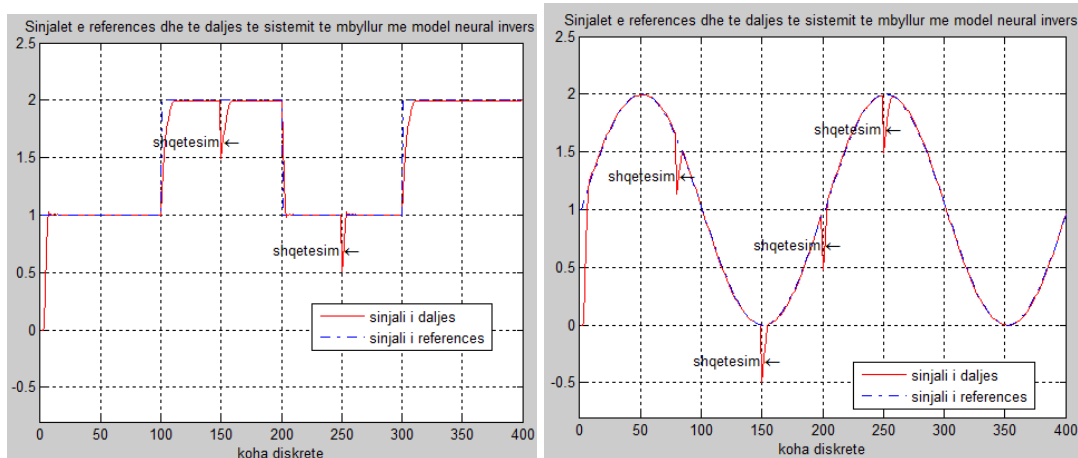


Figura 3.67 Reagimi ndaj shqetësimeve i sistemit të mbyllur me model neural invers për forma të ndryshme sinjali reference

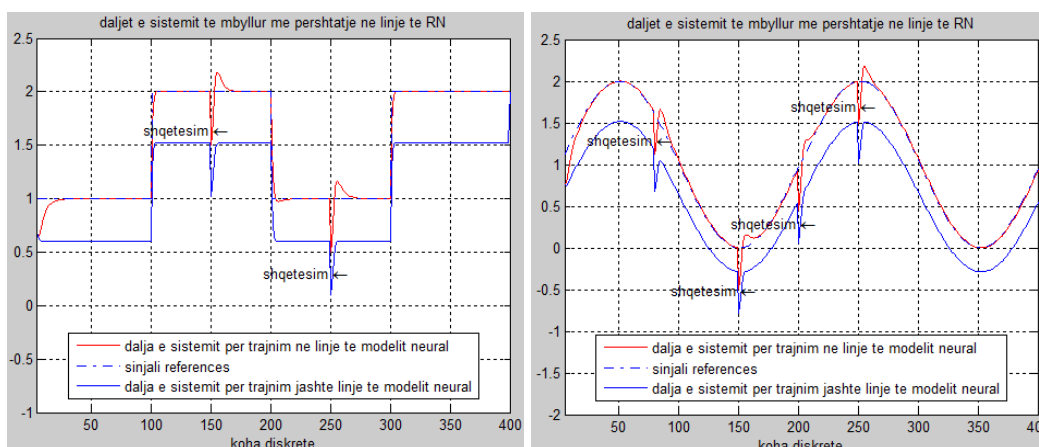


Figura 3.68 Daljet e sistemeve të mbyllura për modelin neural të trajnuar jashtë linje dhe në linjë, për forma të ndryshme sinjali reference dhe shqetësim në çaste k

Shqetësimet për sinjal reference shkallë njësi janë aplikuar në çastet $k = 150$ dhe $k = 250$, që korrespondojnë me dy pjesët e sinjalit shkallë; një pjesë ku amplituda është 2 njësi dhe një pjesë ku është 1 njësi.

Shqetësimet për sinjal reference sinusoidal janë aplikuar në çastet $k = 80$, $k = 150$, $k = 200$, dhe $k = 250$, që janë pjesët zbritëse, minimumi, pjesë rritëse dhe maksimumi i sinjalit.

Vëzhgohet që për sistemin me model neural invers përgjigja kthehet në gjendjen e qëndrueshme brenda 0.05 sekondash (5 kampione).

Sistemi ku parametrat e rrjetit neural ndryshohen në linjë shfaq një mbirregullim $\approx 9\%$ para se të arrijë gjendjen e qëndrueshme dhe arrin këtë gjendje për ≈ 0.13 sekonda (13 kampione).

Forma e sinjalit të referencës nuk ka ndikim të dukshëm në reagimin ndaj shqetësimeve.

Për të dy rastet sistemi i mbyllur reagon mirë ndaj shqetësimeve, gjendja e sistemit stabilizohet brenda treguesve të cilësisë të referuar në paragrafët 3.4÷3.6.

Përfundime

Rrjetet Neurale Artificiale janë një teknologji relativisht e re. Fillimet e studimit të mirëfilltë të tyre daton në vitet '40. Zhvillimi i kësaj fushe ka qenë i ngadaltë gjer në vitet '80. Kjo më së shumti si pasojë e mungesës të teknologjive mbështetëse për implementim të suksesshëm të tyre. Kompjuterizimi dhe zhvillimi në mënyrë eksponenciale i teknologjisë të informacionit pas viteve '80 e në vazhdim, ka sjellë gjithashtu zhvillim shumë të shpejtë në studimin dhe implementimin e rrjeteve neurale.

Rritja e kërkesave të konsumatorit, rritja e kompleksitetit, rritja drastike e sasisë të informacionit, nevoja për respektim të rregullave dhe ligjeve të vendosura nga entet rregullatore për sigurinë në punë e ndotjen e ambientit, konkurrenca në treg, nevoja për mbajtje optimale të kostove, kanë sjellë nevojën për implementim të metodave dhe teknologjive jokonvencionale në industri dhe shumë fusha të tjera. Inteligjenca artificiale, dhe rrjetet neurale artificiale si pjesë e saj, janë një nga fushat e të ardhmes, që ndihmon për t'u përballur me gjithë këto kërkesa në rritje. Kjo falë vetive të tilla të rrjeteve neurale si:

- Aftësia për të përafruar funksione jolineare
- Ata mund të trajnohen lehtësisht duke përdorur të dhëna nga regjistrime të mëparshme të objektit nën studim
- Gjenerojnë hartëzim hyrje-dalje pa kërkuar specifikimet e marrëdhënieve strukturore midis të dhënave hyrje – dalje
- Janë të aplikueshëm në sisteme me shumë variable
- Struktura paralele për përpunimin e informacionit, që tenton të imitojë rrjetin neural biologjik

Implementimi i rrjeteve neurale artificiale, si në paketa software ashtu edhe në hardware, është shumë i gjerë. Mund të jepen shembuj implementimesh të tilla në shumë fusha si kontroll industrial, robotikë, monitorim procesesh dhe identifikim në kohë të dëmtimeve, mjekësi, njohje dhe klasifikim të objektit, kërkim të dhënash, financë, ekonomi, etj.

Implementimi hardware i rrjeteve neurale artificiale është arritur me sukses si në qarqe analoge ashtu edhe në ato dixhitale. Teknologjia hardware e të ardhmes së afërt është shumë premtuese dhe do bëjë projektimin dhe implementimin e rrjeteve neurale më eficient në dy aspekte:

- rritja e fuqisë përpunuese të hardware do zhvillojë më tej implementimin software dhe algoritmeve më të fuqishëm të këtyre rrjeteve.
- Teknologjitë, si p.sh., qarqet nano dhe memristori, që do jenë së shpejti në prodhim komercial, do sjellin mundësi të reja për prodhimin e neuro-

hardware me numër shumë të madh elementesh procesues (neuronesh) dhe lidhjesh sinaptike. Pra mundësi të reja për implementim hardware të rrjeteve neurale artificiale shumë më të fuqishëm nga implementimet aktuale.

Rrjetet neurale janë përdorur me sukses në sistemet e kontrollit për identifikim dhe kontroll duke shfrytëzuar vetitë e përmendura më sipër të tyre. Meqë proceset janë zakonisht shumë komplekse, rrjetet neurale përdoren së bashku me teknikat konvencionale të kontrollit siç janë PI/PID, ose me teknika të bazuara në sistemet eksperte, logjikën fuzzy etj. Implementimi hibrid përmirëson performancën e përgjithshme të sistemit të kontrollit. Rrjetet neurale shumështresore me lidhje të drejtpërdrejtë dhe trajnim të mbikëqyrur janë më të përdorurit në aplikimet e sistemeve të kontrollit dhe identifikim.

Projektimi i rrjetit neural përfshin përcaktimin e arkitekturës, shtresave dhe neuroneve në çdo shtresë, dhe zgjedhjen e algoritmit të trajnimit.

Zgjedhja e parametrave të hyrjes të rrjetit neural është nga proceset më të vështira, sepse mund të ketë ndërvarësi dhe dublikim midis parametrave, disa parametra mund të mos merren parasysh për të ulur kompleksitetin e rrjetit, shpesh nuk ka njohje të mirë të marrëdhënieve midis hyrjes dhe daljes, ndaj nuk është e qartë se cili prej parametrave të hyrjes mund të jetë më i vlefshmi.

Numri i neuroneve në shtresat e fshehura gjendet duke përsëritur disa cikle trajnimi të rrjetit, bazuar në metodën me prova dhe dështime. Ky numër tentohet të mbahet sa më i vogël për efekt të kompleksitetit të rrjetit: sa më shumë neurone, pra sa më shumë elemente procesues, aq më shumë lidhje sinaptike dhe kompleksitet më i madh për implementimin software dhe hardware të rrjetit.

Numri i neuroneve në shtresën daljes zgjidhet i njëjtë me numrin e variableve që dëshirojmë të marrim si dalje të modelit.

Nga simulimet e kryera në këtë punim u vëzhguan rezultatet më poshtë, që janë në linjë me literaturën e shfletuar:

1. E njëjta arkitekturë rrjeti neural jep rezultate të kënaqshme në identifikimin e dy modele të ndryshme: modelin e pozicionit dhe modelin e shpejtësisë të motorit të rrymës të vazhduar, edhe pse këto modele janë të rendit të dytë dhe të rendit të parë. Pra, rrjetet neurale mund të përdoren në përafrimin e modeleve të sistemit, qofshin këto lineare ose jolineare, kur parametrat e sistemit ndryshojnë me kohën ose kur ka shqetësime nga ambienti rrethues.
2. Kontrolli me model neural invers jep rezultat të kënaqshëm, me tregues cilësie brenda vlerave të kërkuara. Pra, mund të themi që skema e kontrollit me model neural invers është një nga aplikimet e mundshme të rrjeteve neurale në kontroll.

3. Variablet e hyrjes të zgjedhur për ndërtimin e rrjetit neural dhe informacioni mbi dinamikën e sistemit që japin kampionet e përdorura për trajnimin e rrjetit neural ndikojnë në cilësinë e kontrollit të sistemit të mbyllur.
4. Nga krahasimi i skemës të kontrollit të sistemit të mbyllur me metodë konvencionale si ajo me rregullator PID, dhe skemës së kontrollit me model neural invers, u vu re që kjo e fundit kënaq më mirë kërkesat për cilësinë e sistemit të mbyllur. Ndaj mund të themi që rrjetet neurale mund të shihen si një mënyrë alternative për të përmirësuar dinamikën e sistemit të mbyllur.
5. Shtimi i elementit integruar në skemën e kontrollit me model neural invers është një mënyrë për të marrë gabim zero në regjim të vendosur. Shtimi i këtij elementi rrit kompleksitetin e skemës të kontrollit ndaj përdorimi i tij duhet konsideruar duke analizuar nevojën kundrejt kompleksitetit që sjell.
6. Ndryshimi në linjë i parametrave të modelit neural është një mënyrë e mirë për optimizim të elementeve përpunues të rrjetit dhe përmirësim të dinamikës të sistemit të mbyllur. Optimizimi i numrit të neuroneve sjell lehtësi në implementimin hardware ose software të rrjetit. Nga ana tjetër, për implementimin e ndryshimit në linjë të parametrave të modelit neural të ndërfutur në një sistem të mbyllur kontrolli, duhen parë aspekte të ndryshme të implementimit disa prej të cilave janë dinamika e sistemit, kosto e implementimit, nevoja për këtë lloj ndryshimi.
7. Skema e kontrollit me model neural invers reagon mirë ndaj shqetësimeve të jashtme. Përgjigja kthehet shpejt në regjim të vendosur. Forma e sinjalit të referencës nuk ka ndikim të dukshëm në reagimin ndaj shqetësimeve.

Në përfundim, mund të përmend si arritje modeste të këtij punimi:

- Paraqitje të zhvillimi të përgjithshëm të rrjeteve neurale
- Vlerësim teorik të zbatimit të rrjeteve neurale në identifikimin dhe kontrollin industrial
- Vlerësimin eksperimental të zbatimit të rrjeteve neurale në identifikimin dhe kontrollin në objekte të rendit të parë dhe të rendit të dytë
- Vlerësim të ndikimit të parametrave të modelit neural në cilësinë e sistemit të mbyllur

Shtojcë

S.1 Rezultatet e paragrafit 3.3

Tabelat më poshtë paraqesin përmbledhtazi rezultatet e gjithë simulimeve të kryera në paragrafin 3.3. Sinjali i referencës është shkallë që fillon me amplitudë 1 dhe vijon me amplitudë 2. Ndaj tabelat japin kalimin në gjendjet kalimtare dhe të vendosur për të dy pjesët e shkallës. Vlerat e variableve në tabela janë:

- y_{max} – vlera maksimale e sinjalit të daljes të sistemit të mbyllur
- y_s – vlera e vendosur e sinjalit të daljes të sistemit të mbyllur
- r – vlera e amplitudës të sinjalit të hyrjes (shkallë) të sistemit të mbyllur
- $m_r\%$ – mbirregullimi në përqindje
- $e\%$ – gabimi në regjim të vendosur në përqindje
- m – numri i neuroneve të shtresës së fshehur

Tabela S.1 Rezultatet për topologjinë e Figurës 3.18b (A2), $5 \times m \times 1$, kampion hyrje trajnimi me amplitudë $a = 0.174$

m	y_{max} për r=1	y_s për r=1	$m_r\%$ r=1	$e\%$ r=1	y_{max} për r=2	y_s për r=2	$m_r\%$ r=2	$e\%$ r=2
3	1.0083	1.0000	0.83%	0.00%	2.1585	2.0000	7.93%	0.00%
4	1.0084	0.9999	0.85%	0.01%	2.1587	2.0003	7.92%	0.01%
5	1.0084	1.0000	0.84%	0.00%	2.1599	2.0003	7.98%	0.02%
6	1.0087	1.0006	0.81%	0.06%	2.1624	1.9998	8.13%	0.01%
7	1.0087	1.0000	0.87%	0.00%	2.1628	2.0001	8.14%	0.00%
8	1.0086	0.9998	0.87%	0.02%	2.1628	2.0004	8.12%	0.02%
9	1.0088	1.0005	0.83%	0.05%	2.1620	1.9992	8.14%	0.04%
10	1.0086	1.0000	0.86%	0.00%	2.1615	1.9997	8.09%	0.01%
11	1.0087	1.0001	0.87%	0.01%	2.1639	2.0002	8.19%	0.01%
12	1.0086	0.9999	0.87%	0.01%	2.1613	2.0001	8.06%	0.01%
13	1.0087	1.0000	0.87%	0.00%	2.1635	1.9997	8.19%	0.02%
14	1.0088	1.0004	0.85%	0.04%	2.1634	1.9994	8.20%	0.03%
15	1.0088	1.0003	0.85%	0.03%	2.1635	1.9996	8.20%	0.02%
16	1.0088	1.0003	0.85%	0.03%	2.1638	1.9998	8.20%	0.01%
17	1.0089	1.0004	0.85%	0.04%	2.1647	1.9997	8.25%	0.02%
18	1.0088	1.0003	0.86%	0.03%	2.1651	2.0004	8.24%	0.02%
19	1.0089	1.0006	0.83%	0.06%	2.1619	1.9985	8.17%	0.07%
20	1.0086	0.9997	0.89%	0.03%	2.1657	2.0005	8.26%	0.03%

Tabela S.2 Rezultatet për topologjinë e Figurës 3.18b (A2), $5 \times m \times 1$, kampion hyrje trajnimi me amplitudë $A = 4$

m	$y_{\max}$ për r=1	y_s për r=1	$m_r\%$ r=1	e% r=1	$y_{\max}$ për r=2	y_s për r=2	$m_r\%$ r=2	e% r=2
3	1.0080	0.9990	0.91%	0.10%	2.2613	1.9989	13.13%	0.05%
4	1.0116	1.0060	0.55%	0.60%	2.2670	2.0058	13.02%	0.29%
5	1.0125	1.0073	0.52%	0.73%	2.2660	2.0072	12.90%	0.36%
6	1.0096	1.0026	0.70%	0.26%	2.2624	2.0033	12.93%	0.17%
7	1.0073	0.9959	1.15%	0.41%	2.2535	1.9960	12.90%	0.20%
8	1.0075	0.9967	1.08%	0.33%	2.2544	1.9970	12.89%	0.15%
9	1.0104	1.0041	0.63%	0.41%	2.2623	2.0035	12.92%	0.17%
10	1.0264	1.0207	0.56%	2.07%	2.2749	2.0203	12.60%	1.01%
11	1.0093	1.0019	0.74%	0.19%	2.2632	2.0020	13.05%	0.10%
12	1.0104	1.0039	0.65%	0.39%	2.2601	2.0042	12.76%	0.21%
13	1.0101	1.0035	0.66%	0.35%	2.2580	2.0031	12.73%	0.16%
14	1.0354	1.0281	0.71%	2.81%	2.2824	2.0253	12.70%	1.26%
15	1.0135	1.0084	0.50%	0.84%	2.2651	2.0079	12.81%	0.40%
16	1.0126	1.0073	0.52%	0.73%	2.2623	2.0067	12.73%	0.34%
17	1.0052	0.9806	2.51%	1.94%	2.2386	1.9828	12.90%	0.86%
18	1.0051	0.9768	2.90%	2.32%	2.2366	1.9813	12.88%	0.93%
19	1.0211	1.0161	0.50%	1.61%	2.2737	2.0166	12.75%	0.83%
20	1.0143	1.0094	0.49%	0.94%	2.2687	2.0101	12.86%	0.50%

Tabela S.3 Rezultatet për topologjinë e Figurës 3.18b (A2), $5 \times m \times 1$, kampion hyrje trajnimi me amplitudë $A = 0.6$

m	$y_{\max}$ për r=1	y_s për r=1	$m_r\%$ r=1	e% r=1	$y_{\max}$ për r=2	y_s për r=2	$m_r\%$ r=2	e% r=2
3	1.0082	0.9996	0.87%	0.04%	2.2383	1.9997	11.93%	0.02%
4	1.0091	1.0013	0.78%	0.13%	2.2396	2.0011	11.92%	0.06%
5	1.0088	1.0007	0.81%	0.07%	2.2389	2.0005	11.92%	0.02%
6	1.0087	1.0006	0.81%	0.06%	2.2391	2.0001	11.95%	0.01%
7	1.0084	0.9993	0.91%	0.07%	2.2377	1.9993	11.92%	0.03%
8	1.0082	0.9990	0.93%	0.10%	2.2389	1.9995	11.97%	0.02%
9	1.0085	0.9997	0.88%	0.03%	2.2393	1.9997	11.98%	0.01%
10	1.0082	0.9991	0.92%	0.09%	2.2380	1.9990	11.96%	0.05%
11	1.0095	1.0020	0.74%	0.20%	2.2409	2.0017	11.95%	0.08%
12	1.0086	0.9998	0.88%	0.02%	2.2389	1.9998	11.96%	0.01%
13	1.0092	1.0015	0.77%	0.15%	2.2392	2.0002	11.95%	0.01%
14	1.0094	1.0019	0.75%	0.19%	2.2410	2.0020	11.93%	0.10%
15	1.0091	1.0010	0.81%	0.10%	2.2407	2.0012	11.97%	0.06%
16	1.0091	1.0008	0.82%	0.08%	2.2413	2.0011	12.00%	0.06%

17	1.0092	1.0013	0.79%	0.13%	2.2391	2.0013	11.89%	0.06%
18	1.0085	0.9997	0.88%	0.03%	2.2385	1.9997	11.94%	0.02%
19	1.0092	1.0012	0.80%	0.12%	2.2407	2.0009	11.99%	0.05%
20	1.0087	1.0000	0.87%	0.00%	2.2391	1.9987	12.03%	0.06%

Tabela S.4 Rezultatet për topologjinë e Figurës 3.18a (A1), $6 \times m \times 1$, kampion hyrje trajnimi me amplitudë $A = 0.174$

m	$y_{\max}$ për $r=1$	y_s për $r=1$	$m_r\%$ $r=1$	$e\%$ $r=1$	$y_{\max}$ për $r=1$	y_s për $r=2$	$m_r\%$ $r=2$	$e\%$ $r=2$
3	1.0377	0.9998	3.79%	0.02%	2.1496	2.0003	7.46%	0.01%
4	1.0376	0.9996	3.80%	0.04%	2.1463	1.9997	7.33%	0.01%
5	1.0353	0.9996	3.56%	0.04%	2.1467	2.0004	7.31%	0.02%
6	1.0400	0.9997	4.03%	0.03%	2.1475	2.0001	7.37%	0.01%
7	1.0416	0.9998	4.17%	0.02%	2.1515	2.0001	7.57%	0.00%
8	1.0423	0.9999	4.24%	0.01%	2.1510	2.0000	7.55%	0.00%
9	1.0354	0.9997	3.57%	0.03%	2.1525	2.0002	7.61%	0.01%
10	1.0328	0.9997	3.31%	0.03%	2.1523	2.0002	7.61%	0.01%
11	1.0388	0.9999	3.89%	0.01%	2.1521	1.9998	7.62%	0.01%
12	1.0376	0.9997	3.79%	0.03%	2.1519	2.0003	7.58%	0.02%
13	1.0359	0.9998	3.61%	0.02%	2.1521	2.0001	7.60%	0.00%
14	1.0338	0.9997	3.41%	0.03%	2.1545	2.0004	7.70%	0.02%
15	1.0393	0.9996	3.97%	0.04%	2.1520	2.0002	7.59%	0.01%
16	1.0389	0.9998	3.91%	0.02%	2.1520	1.9997	7.61%	0.01%
17	1.0356	1.0003	3.53%	0.03%	2.1520	1.9996	7.62%	0.02%
18	1.0385	0.9994	3.91%	0.06%	2.1539	1.9997	7.71%	0.02%
19	1.0371	1.0000	3.71%	0.00%	2.1533	2.0001	7.66%	0.00%
20	1.0341	1.0000	3.41%	0.00%	2.1535	1.9997	7.69%	0.01%

Tabela S.5 Rezultatet për topologjinë e Figurës 3.18a (A1), $6 \times m \times 1$, kampion hyrje trajnimi me amplitudë $A = 4$

m	$y_{\max}$ për $r=1$	y_s për $r=1$	$m_r\%$ $r=1$	$e\%$ $r=1$	$y_{\max}$ për $r=2$	y_s për $r=2$	$m_r\%$ $r=2$	$e\%$ $r=2$
3	1.6234	0.9939	63.33%	0.61%	2.1946	1.9953	9.99%	0.23%
4	1.6852	0.9949	69.38%	0.51%	2.1909	1.9979	9.66%	0.11%
5	1.6042	0.9918	61.75%	0.82%	2.1936	1.9947	9.97%	0.27%
6	1.5743	0.9987	57.64%	0.13%	2.2015	2.0002	10.07%	0.01%
7	1.6429	0.9949	65.13%	0.51%	2.1913	1.9977	9.69%	0.12%
8	1.6182	1.0045	61.09%	0.45%	2.2035	2.0071	9.79%	0.35%
9	1.6935	0.9874	71.52%	1.26%	2.1826	1.9910	9.62%	0.45%
10	1.5832	0.9945	59.20%	0.55%	2.1967	1.9957	10.07%	0.22%

11	1.6461	1.0126	62.56%	1.26%	2.2087	2.0138	9.68%	0.69%
12	1.6008	1.0109	58.35%	1.09%	2.2098	2.0122	9.82%	0.61%
13	1.5862	0.9940	59.58%	0.60%	2.1940	1.9963	9.90%	0.19%
14	1.6291	0.9939	63.91%	0.61%	2.1925	1.9972	9.78%	0.14%
15	1.6900	0.9947	69.91%	0.53%	2.1904	1.9983	9.61%	0.08%
16	1.6093	0.9901	62.54%	0.99%	2.1887	1.9926	9.84%	0.37%
17	1.6228	0.9809	65.45%	1.91%	2.1810	1.9858	9.83%	0.71%
18	1.6853	0.9634	74.93%	3.66%	2.1627	1.9702	9.77%	1.49%
19	1.6484	1.0027	64.39%	0.27%	2.1997	2.0049	9.72%	0.25%
20	1.6586	1.0009	65.70%	0.09%	2.1993	2.0049	9.70%	0.24%

Tabela S.6 Rezultatet për topologjinë e Figurës 3.18a (A1), $6 \times m \times 1$, kampion hyrje trajnimi me amplitudë $A = 0.6$

m	$y_{\max}$ për r=1	y_s për r=1	$m_r\%$ r=1	e% r=1	$y_{\max}$ për r=2	y_s për r=2	$m_r\%$ r=2	e% r=2
3	1.1200	0.9985	12.17%	0.15%	2.1855	1.9989	9.34%	0.05%
4	1.1005	0.9980	10.28%	0.20%	2.1914	1.9989	9.63%	0.06%
5	1.1092	0.9978	11.16%	0.22%	2.1885	1.9988	9.49%	0.06%
6	1.0969	1.0009	9.59%	0.09%	2.1939	2.0015	9.61%	0.07%
7	1.1129	0.9995	11.34%	0.05%	2.1880	2.0002	9.39%	0.01%
8	1.0998	0.9981	10.18%	0.19%	2.1919	1.9990	9.65%	0.05%
9	1.1261	0.9978	12.86%	0.22%	2.1811	1.9991	9.11%	0.05%
10	1.1079	0.9978	11.03%	0.22%	2.1891	1.9990	9.51%	0.05%
11	1.1161	0.9994	11.68%	0.06%	2.1862	1.9998	9.32%	0.01%
12	1.1020	0.9993	10.28%	0.07%	2.1920	2.0004	9.58%	0.02%
13	1.1078	1.0001	10.77%	0.01%	2.1911	2.0003	9.54%	0.02%
14	1.1167	1.0003	11.63%	0.03%	2.1867	2.0013	9.26%	0.07%
15	1.1127	0.9998	11.29%	0.02%	2.1878	2.0002	9.38%	0.01%
16	1.1004	1.0008	9.96%	0.08%	2.1958	2.0022	9.67%	0.11%
17	1.1082	0.9987	10.97%	0.13%	2.1901	1.9995	9.54%	0.03%
18	1.1042	1.0000	10.41%	0.00%	2.1920	2.0003	9.59%	0.02%
19	1.1039	1.0003	10.35%	0.03%	2.1929	2.0010	9.59%	0.05%
20	1.1075	0.9997	10.79%	0.03%	2.1898	1.9997	9.51%	0.01%

Tabela S.7 Rezultatet për topologjinë e Figurës 3.19a (A3), $6 \times m \times 1$, kampion hyrje trajnimi me amplitudë $A = 0.174$

m	$y_{\max}$ për r=1	y_s për r=1	$m_r\%$ r=1	e% r=1	$y_{\max}$ për r=2	y_s për r=2	$m_r\%$ r=2	e% r=2
3	1.0098	1.0065	0.32%	0.65%	1.9924	1.9924	0.00%	0.38%
4	1.0036	1.0022	0.14%	0.22%	1.9965	1.9964	0.00%	0.18%
5	1.0041	1.0026	0.15%	0.26%	1.9962	1.9962	0.00%	0.19%

6	1.0036	1.0022	0.14%	0.22%	1.9979	1.9979	0.00%	0.11%
7	1.0057	1.0037	0.19%	0.37%	1.9956	1.9956	0.00%	0.22%
8	1.0046	1.0029	0.17%	0.29%	1.9967	1.9966	0.00%	0.17%
9	1.0014	1.0006	0.08%	0.06%	1.9966	1.9966	0.00%	0.17%
10	1.0024	1.0013	0.10%	0.13%	1.9993	1.9993	0.00%	0.03%
11	1.0017	1.0008	0.09%	0.08%	2.0027	2.0027	0.00%	0.13%
12	0.9992	0.9967	0.25%	0.33%	1.9995	1.9995	0.00%	0.03%
13	0.9992	0.9990	0.03%	0.10%	2.0034	2.0033	0.00%	0.17%
14	1.0012	1.0004	0.08%	0.04%	2.0020	2.0020	0.00%	0.10%
15	1.0018	1.0010	0.09%	0.10%	2.0006	2.0006	0.00%	0.03%
16	1.0018	1.0009	0.09%	0.09%	2.0018	2.0018	0.00%	0.09%
17	1.0044	1.0028	0.15%	0.28%	1.9995	1.9995	0.00%	0.02%
18	1.0019	1.0010	0.09%	0.10%	1.9972	1.9972	0.00%	0.14%
19	1.0012	1.0005	0.07%	0.05%	2.0024	2.0024	0.00%	0.12%
20	1.0019	1.0009	0.10%	0.09%	2.0038	2.0038	0.00%	0.19%

Tabela S.8 Rezultatet për topologjinë e Figurës 3.19a (A3), $6 \times m \times 1$, kampion hyrje trajnimi me amplitudë $A = 4$

m	$y_{\max}$ për r=1	y_s për r=1	$m_r\%$ r=1	e% r=1	$y_{\max}$ për r=2	y_s për r=2	$m_r\%$ r=2	e% r=2
3	1.0473	1.0472	0.00%	4.72%	1.8907	1.8899	0.04%	5.51%
4	0.9992	0.9647	3.57%	3.53%	1.6919	1.6796	0.74%	16.02%
5	0.9992	0.9148	9.23%	8.52%	1.8187	1.8086	0.55%	9.57%
6	0.9992	0.9889	1.03%	1.11%	1.8814	1.8748	0.35%	6.26%
7	0.9992	0.9170	8.96%	8.30%	1.8256	1.8164	0.51%	9.18%
8	1.0476	1.0476	0.01%	4.76%	1.9287	1.9223	0.33%	3.89%
9	0.9992	0.9747	2.51%	2.53%	1.8853	1.8749	0.56%	6.26%
10	0.9992	0.9285	7.62%	7.15%	1.8392	1.8296	0.52%	8.52%

1								
1	0.9992	0.9748	2.49%	2.52%	1.8594	1.8533	0.33%	7.33%
1			11.57	10.44				
2	0.9992	0.8956	%	%	1.8127	1.8030	0.54%	9.85%
1								
3	1.0222	1.0221	0.01%	2.21%	1.9212	1.9123	0.46%	4.38%
1								
4	1.0735	1.0732	0.02%	7.32%	1.9692	1.9599	0.48%	2.00%
1								
5	1.0824	1.0823	0.02%	8.23%	1.9702	1.9632	0.36%	1.84%
1								
6	1.0523	1.0521	0.02%	5.21%	1.9544	1.9450	0.49%	2.75%
1								
7	0.9992	0.9433	5.92%	5.67%	1.8537	1.8455	0.45%	7.73%
1				19.59				
8	1.1972	1.1959	0.10%	%	2.0807	2.0684	0.59%	3.42%
1								
9	1.0910	1.0906	0.04%	9.06%	1.9927	1.9822	0.53%	0.89%
2								
0	1.0361	1.0359	0.02%	3.59%	1.9428	1.9313	0.60%	3.43%

Tabela S.9 Rezultatet për topologjinë e Figurës 3.19a (A3), $6 \times m \times 1$, kampion hyrje trajnimi me amplitudë $A = 0.6$

m	$y_{\max}$ për r=1	y_s për r=1	$m_r\%$ r=1	e% r=1	$y_{\max}$ për r=2	y_s për r=2	$m_r\%$ r=2	e% r=2
3	0.9992	0.9280	7.67%	7.20%	1.9626	1.9358	1.38%	3.21%
4	0.9992	0.9391	6.40%	6.09%	1.9643	1.9374	1.39%	3.13%
5	0.9992	0.9364	6.70%	6.36%	1.9849	1.9540	1.58%	2.30%
6	0.9992	0.9410	6.18%	5.90%	1.9739	1.9440	1.54%	2.80%
7	0.9992	0.9370	6.64%	6.30%	1.9607	1.9334	1.41%	3.33%
8	0.9992	0.9414	6.14%	5.86%	1.9798	1.9491	1.58%	2.54%
9	0.9992	0.9469	5.51%	5.31%	1.9754	1.9459	1.52%	2.70%
1								
0	0.9992	0.9488	5.31%	5.12%	1.9827	1.9524	1.55%	2.38%
1								
1	0.9992	0.9489	5.29%	5.11%	1.9760	1.9478	1.45%	2.61%
1								
2	0.9992	0.9496	5.22%	5.04%	1.9741	1.9464	1.42%	2.68%
1								
3	0.9992	0.9519	4.96%	4.81%	1.9808	1.9532	1.41%	2.34%

1 4	0.9992	0.9445	5.79%	5.55%	1.9806	1.9525	1.44%	2.37%
1 5	0.9992	0.9567	4.44%	4.33%	1.9839	1.9524	1.61%	2.38%
1 6	0.9992	0.9483	5.37%	5.17%	1.9775	1.9470	1.57%	2.65%
1 7	0.9992	0.9472	5.48%	5.28%	1.9853	1.9523	1.69%	2.38%
1 8	0.9992	0.9416	6.11%	5.84%	1.9818	1.9514	1.55%	2.43%
1 9	0.9992	0.9599	4.09%	4.01%	1.9879	1.9563	1.61%	2.18%
2 0	0.9992	0.9560	4.51%	4.40%	1.9838	1.9542	1.51%	2.29%

Tabela S.10 Rezultatet për topologjinë e Figurës 3.19b (A4), $5 \times m \times 1$, kampion hyrje trajnimi me amplitudë $A = 0.174$

m	$y_{\max}$ për r=1	y_s për r=1	$m_r\%$ r=1	e% r=1	$y_{\max}$ për r=2	y_s për r=2	$m_r\%$ r=2	e% r=2
3	1.0071	1.0049	0.22%	0.49%	1.9928	1.9928	0.00%	0.36%
4	1.0046	1.0032	0.14%	0.32%	1.9978	1.9978	0.00%	0.11%
5	1.0025	1.0016	0.08%	0.16%	1.9975	1.9975	0.00%	0.12%
6	1.0005	1.0002	0.03%	0.02%	2.0038	2.0038	0.00%	0.19%
7	1.0061	1.0043	0.17%	0.43%	1.9951	1.9951	0.00%	0.24%
8	0.9998	0.9996	0.01%	0.04%	2.0014	2.0014	0.00%	0.07%
9	1.0025	1.0017	0.08%	0.17%	2.0030	2.0030	0.00%	0.15%
1 0	1.0016	1.0010	0.06%	0.10%	2.0045	2.0045	0.00%	0.23%
1 1	1.0034	1.0024	0.10%	0.24%	2.0005	2.0005	0.00%	0.02%
1 2	1.0022	1.0015	0.07%	0.15%	2.0026	2.0026	0.00%	0.13%
1 3	0.9994	0.9993	0.00%	0.07%	2.0064	2.0064	0.00%	0.32%
1 4	1.0027	1.0019	0.09%	0.19%	2.0026	2.0026	0.00%	0.13%
1 5	1.0002	1.0000	0.02%	0.00%	2.0077	2.0077	0.00%	0.38%
1 6	1.0018	1.0011	0.06%	0.11%	2.0069	2.0069	0.00%	0.34%

1 7	1.0021	1.0014	0.07%	0.14%	2.0021	2.0021	0.00%	0.11%
1 8	1.0024	1.0016	0.08%	0.16%	2.0034	2.0034	0.00%	0.17%
1 9	1.0025	1.0017	0.08%	0.17%	2.0009	2.0009	0.00%	0.05%
2 0	1.0015	1.0009	0.05%	0.09%	2.0061	2.0061	0.00%	0.30%

Tabela S.11 Rezultatet për topologjinë e Figurës 3.19b (A4), $5 \times m \times 1$, kampion hyrje trajnimi me amplitudë $A = 4$

m	$y_{\max}$ për r=1	y_s për r=1	$m_r\%$ r=1	e% r=1	$y_{\max}$ për r=2	y_s për r=2	$m_r\%$ r=2	e% r=2
3	0.9992	0.8157	22.48 %	18.43 %	1.7272	1.7272	0.00%	13.64 %
4	0.9992	0.8381	19.22 %	16.19 %	1.7412	1.7412	0.00%	12.94 %
5	0.9992	0.7650	30.60 %	23.50 %	1.6764	1.6764	0.00%	16.18 %
6	0.9992	0.9524	4.91%	4.76%	1.8353	1.8353	0.00%	8.23%
7	0.9992	0.8902	12.24 %	10.98 %	1.7911	1.7911	0.00%	10.45 %
8	0.9992	0.8816	13.34 %	11.84 %	1.7857	1.7857	0.00%	10.71 %
9	0.9992	0.9149	9.21%	8.51%	1.8105	1.8105	0.00%	9.48%
1 0	0.9992	0.8806	13.47 %	11.94 %	1.7867	1.7867	0.00%	10.67 %
1 1	0.9992	0.9343	6.94%	6.57%	1.8352	1.8352	0.00%	8.24%
1 2	0.9992	0.9107	9.72%	8.93%	1.8119	1.8119	0.00%	9.40%
1 3	0.9992	0.9562	4.49%	4.38%	1.8511	1.8511	0.00%	7.44%
1 4	0.9992	0.9330	7.09%	6.70%	1.8377	1.8377	0.00%	8.11%
1 5	0.9992	0.9349	6.87%	6.51%	1.8362	1.8362	0.00%	8.19%
1 6	1.0683	1.0683	0.00%	6.83%	1.9568	1.9568	0.00%	2.16%
1 7	0.9992	0.9670	3.33%	3.30%	1.8651	1.8651	0.00%	6.75%

1 8	0.9992	0.9970	0.22%	0.30%	1.8934	1.8934	0.00%	5.33%
1 9	1.0172	1.0172	0.00%	1.72%	1.9109	1.9109	0.00%	4.45%
2 0	1.0281	1.0281	0.00%	2.81%	1.9212	1.9212	0.00%	3.94%

Tabela S.12 Rezultatet për topologjinë e Figurës 3.19b (A4), $5 \times m \times 1$, kampion hyrje trajnimi me amplitudë $A = 0.6$

m	$y_{\max}$ për r=1	y_s për r=1	$m_r\%$ r=1	e% r=1	$y_{\max}$ për r=2	y_s për r=2	$m_r\%$ r=2	e% r=2
3	0.9992	0.9369	6.65%	6.31%	1.9552	1.9515	0.19%	2.43%
4	0.9992	0.9346	6.91%	6.54%	1.9603	1.9502	0.52%	2.49%
5	0.9992	0.9397	6.33%	6.03%	1.9509	1.9443	0.34%	2.79%
6	0.9992	0.9363	6.72%	6.37%	1.9494	1.9441	0.27%	2.79%
7	0.9992	0.9305	7.38%	6.95%	1.9415	1.9359	0.29%	3.20%
8	0.9992	0.9361	6.74%	6.39%	1.9519	1.9476	0.22%	2.62%
9	0.9992	0.9393	6.37%	6.07%	1.9533	1.9459	0.38%	2.71%
1 0	0.9992	0.9369	6.65%	6.31%	1.9474	1.9433	0.21%	2.84%
1 1	0.9992	0.9452	5.71%	5.48%	1.9557	1.9471	0.44%	2.64%
1 2	0.9992	0.9465	5.56%	5.35%	1.9568	1.9494	0.38%	2.53%
1 3	0.9992	0.9402	6.27%	5.98%	1.9576	1.9525	0.26%	2.37%
1 4	0.9992	0.9482	5.37%	5.18%	1.9548	1.9448	0.51%	2.76%
1 5	0.9992	0.9463	5.59%	5.37%	1.9595	1.9512	0.42%	2.44%
1 6	0.9992	0.9451	5.72%	5.49%	1.9535	1.9487	0.25%	2.56%
1 7	0.9992	0.9481	5.38%	5.19%	1.9569	1.9485	0.44%	2.58%
1 8	0.9992	0.9550	4.62%	4.50%	1.9634	1.9505	0.66%	2.47%
1 9	0.9992	0.9483	5.36%	5.17%	1.9575	1.9494	0.42%	2.53%
2 0	0.9992	0.9525	4.90%	4.75%	1.9597	1.9500	0.50%	2.50%

S.2 Kodet në Matlab të përdorura për simulimet në kapitullin III

File *model.m*

```
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***%
% kodi per marrjen e kampioneve Hyrje-Dalje te modelit te motorit
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***%
clear all, close all, clc

% sinjali konstant + sekuenca pseudorastesore
time = 200;
a=0.087;
u=a*ones(1,time);

% gjenerimi i sekuences pseudorastesore
prbs = (-1).^(1:10);
for i = 11:length(u)
    prbs(i) = -prbs(i-10)*prbs(i-7);
end
u = u+a*prbs;

% modeli dhe dalja
num = [1.801 0.1758];
den = [1 -0.8272 2.529e-005];
y = (dlsim(num,den,u))';
u(1:10) = []; y(1:10) = []; % shkurtimi i vektorit te daljes

% shfaqja e sinjalit te komandes dhe daljes se modelit
stairs(u)
hold on
h = plot(y);
set(h, 'LineWidth',1.45, 'color', 'r')
title('Sinjali i kontrollit dhe dalja e modelit te motorit DC')
xlabel('Koha Diskrete'), grid, hold off
```

File *normalizim.m*

```
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***%
% kodi per funksionin e normalizimit te sinjaleve
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***%
function [xN,a,b] = normalizim(x,alpha,beta)
a = (beta-alpha)/(max(x)-min(x));
b = beta-a*max(x);
xN = a*x+b;
```

File *HD_normalizim.m*

```
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %  
% kodi per normalizimin i sinjaleve Hyrje-Dalje  
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %  
[u_normalizuar,aNorm_u,bNorm_u] = normalizim(u,0.1,0.9);  
[y_normalizuar,aNorm_y,bNorm_y] = normalizim(y,0.1,0.9);  
  
% ruajtja e sinjaleve ne matlab data file  
HD_normalizuar = [u_normalizuar;y_normalizuar];  
save sinjalet_HD HD_normalizuar  
  
% koeficientet a dhe b te normalizimit te sinjaleve H/D  
save ab_normalizim aNorm_u bNorm_u aNorm_y bNorm_y
```

File *trajnim.m*

```
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %  
% kodi per trajnimin i rrjetit neural  
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %  
clear all, close all, clc  
  
% leximi i te dhenave H/D te normalizuara  
load sinjalet_HD  
u_normalizuar = HD_normalizuar(1,:);  
y_normalizuar = HD_normalizuar(2,:);  
  
Nr_Hyrje = 6; % numri i neuroneve te shtreses se hyrjes  
Nr_Fshehur = 6; % numri i neuroneve te shtreses se fshehur  
Nr_Dalje = 1; % numri i neuroneve te shtreses se daljes  
  
% inicializimi i rastesishem i peshave W, Z dhe spostimeve w0,z0  
W = rand(Nr_Fshehur,Nr_Hyrje)-0.5; w0 = rand(Nr_Fshehur,1)-0.5;  
Z = rand(Nr_Dalje,Nr_Fshehur)-0.5; z0 = rand(Nr_Dalje,1)-0.5;  
  
Nr_iterimeve = 400; % Numri i iterimeve  
eta = 0.8; % koeficienti trajnimit  
k = 0;  
while k<=Nr_iterimeve  
    k = k+1;  
    for i = 3:length(HD_normalizuar)-1  
  
        x=[y_normalizuar(i+1) y_normalizuar(i) y_normalizuar(i-1)  
y_normalizuar(i-2) u_normalizuar(i-1) u_normalizuar(i-2)]';  
        f = logsig(W*x+w0);% pergjigja e shtreses se fshehur  
        d(i) = logsig(Z*f+z0); % pergjigja e shtreses se daljes  
        e(i) = u_normalizuar(i)-d(i);% gabimi ne daljen e rrjetit
```

```
% gabimi per t'u propaguar mbrapsht to be backpropagated
delta(i) = d(i).*(ones(Nr_Dalje,1)-d(i)).*e(i);

% rifreskimi i matricave te peshave Z dhe spostimit z0
Z = Z+eta*delta(i)*f'; z0 = z0+eta*delta(i);

% gabimi ne shtreshen e fshehur at hidden layer output
dh = f.*(ones(size(f))-f).*(Z'*delta(i));

% rifreskimi i matricave te peshave dhe spostimit te shtreses
se fshehur
W = W+eta*dh*x';
w0 = w0+eta*dh;
end % fundi i lakut "for"

peshat_W(:, :, k) = W; peshat_Z(:, :, k) = Z;
spostim_w0(:, :, k) = w0; spostim_z0(:, :, k) = z0;

% ruajtja e peshave dhe spostimeve ne matlab data file
save peshat peshat_W peshat_Z spostim_w0 spostim_z0

clc, disp(['k = ' num2str(k)])
end % fundi i lakut "while"
```

File verifikim_trajnimi.m

```
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %
% kodi per verifikimin e trajnimit te rrjetit neural
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %
clear all; close all; clc

% leximi i file-t te te dhenave te sinjaleve Hyrje/Dalje te
normalizuar
load sinjalet_HD
u_normalizuar = HD_normalizuar(1,:);
y_normalizuar = HD_normalizuar(2,:);

load peshat % leximi i file-t te te dhenave te spostimeve

N = 400; % numri i iterimeve
W = peshat_W(:, :, N); w0 = spostim_w0(:, :, N);
Z = peshat_Z(:, :, N); z0 = spostim_z0(:, :, N);

clear peshat_W peshat_Z spostim_w0 spostim_z0

for i = 3:length(HD_normalizuar)-1
```

```
x=[y_normalizuar(i+1) y_normalizuar(i) y_normalizuar(i-1)
y_normalizuar(i-2) u_normalizuar(i-1) u_normalizuar(i-2)];
y(i) = logsig(Z*(logsig(W*x+w0))+z0);% pergjigja e shtreses
se daljes
end % fundi i lakut "for"

y = y(1:100);
figure(1)
stairs(y, 'r')
hold
h = stairs(u_normalizuar(1:100));
set(h, 'LineWidth', 1.4)
title(['Pergjigja aktuale dhe ajo e deshiruar e RN pas ' num2str(N)
' iterimeve'])
axis([0 100 0 1])
xlabel('koha diskrete')
legend('Pergjigja e Rrjetit Neural', 'Pergjigja e deshiruar e
Rrjetit Neural', 'Location', 'Best')
grid
hold off

figure(2)
u_normalizuar = u_normalizuar(1:100);
error = y-u_normalizuar;
plot(error)
grid
axis([0 100 -0.02 0.02])
title(['gabimi i trajnimit ' 'pas ' num2str(N) ' iterimeve'])
xlabel('koha diskrete')

disp(['gabimi maksimal : ' num2str(max(error))])
disp(['varianca e gabimit : ' num2str(std(error)^2)])
```

File *lexim_pesha.m*

```
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** %
% kodi per paraqitjen grafike te ecurise te peshave dhe
% spostimeve te rrjetit neural
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** %
clear all, close all, clc
load peshat

N = length(peshat_W);
W = peshat_W(:, :, N);
w0 = spostim_w0(:, :, N);
Z = peshat_Z(:, :, N);
z0 = spostim_z0(:, :, N);
```



```
Nr_Hyrje = 6; % numri i neuroneve te shtreses se hyrjes
Nr_Fshehur = 6; % numri i neuroneve te shtreses se fshehur
Nr_Dalje = 1; % numri i neuroneve te shtreses se daljes

for k = 1:Nr_Fshehur
    x = peshat_W(k,1,1:400);
    plot(x(:))
    hold on
end
hold off
grid
title('ecuria e peshave te shtreses se fshehur')
xlabel('koha diskrete')

% paraqitja grafike e ecurise te spostimeve te shtreses se fshehur
figure(2)
for k = 1:Nr_Fshehur
    x = spostim_w0(k,1,1:400);
    plot(x(:))
    hold on
end
hold off
grid
title('ecuria e spostimeve te shtreses se fshehur')
xlabel('koha diskrete')

% paraqitja grafike e ecurise te peshave te shtreses se daljes
figure(3)
for k = 1:Nr_Fshehur
    x = peshat_Z(1,k,1:400);
    plot(x(:))
    hold on
end
hold off
grid
title('ecuria e peshave te shtreses se daljes')
xlabel('koha diskrete')

% paraqitja grafike e ecurise te spostimit te shtreses se daljes
figure(4)
x = spostim_z0(1:400);
plot(x(:))
grid
title('ecuria e spostimit te daljes')
xlabel('koha diskrete')
```

File *kontroll_RN.m*

```
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %
% kodi per kontrollin me model  neural invers
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %
clear all, close all, clc

load peshat % leximi i peshave
N = length(peshat_W);
W = peshat_W(:, :, N);
w0 = spostim_w0(:, :, N);
Z = peshat_Z(:, :, N);
z0 = spostim_z0(:, :, N);
clear peshat_W peshat_Z spostim_w0 spostim_z0

load ab_normalizim % leximi i koeficienteve te normalizimit

% gjenerimi i sinjalit te deshruar
stair1 = 1*ones(1,100); stair2 = 2*ones(1,100);
r1 = [stair1 stair2];
% t = 0:pi/100:2*pi;
% r1 = 2+1*sin(0.8*t);
r2=r1;
r = [r1 r2];
dalja_y = r; % inicializimi i daljes te modelit
u = 0.01*ones(size(r)); y_normalizuar = aNorm_y*dalja_y+bNorm_y;

y_normalizuar = aNorm_y*dalja_y+bNorm_y;
u_normalizuar = aNorm_u*u+bNorm_u;

for i = 3:length(r)-1
    dalja_y(1)=0;
    dalja_y(2)=0;
    % dalja e sistemit te mbyllur
    dalja_y(i) = 0.8272*dalja_y(i-1)-2.529e-005*dalja_y(i-2)+1.801*u(i-1)+0.1758*u(i-2);
    r_N = aNorm_y*r(i+1)+bNorm_y; % normalizimi i gabimit

    % normalizimi i daljes aktuale
    y_normalizuar(i) = aNorm_y*dalja_y(i)+bNorm_y;

    x=[r_N y_normalizuar(i) y_normalizuar(i-1) y_normalizuar(i-2)
u_normalizuar(i-1) u_normalizuar(i-2)]';

    % aktivizimi shtreses se fshehur
    f_in = W*x+w0;
    f_in = f_in.*(-5<=f_in<=5)-5.*(f_in<-5)+5.*(f_in>5);

    % dalja e shtreses se fshehur
```

```
f_out = logsig(f_in);

% aktivizimi i shtreses se dalje
d_in = Z*f_out+z0;
d_in = d_in.*(-5<=d_in<=5)-5.*(d_in<-5)+5.*(d_in>5);

% pergjigja e shtreses se daljes
u_normalizuar(i) = logsig(d_in);

% denormalizimi i sinjalit te kontrollit
u(i) = (u_normalizuar(i)-bNorm_u)/aNorm_u;
end % for loop

save dalja_y dalja_y
% RN=length(Z);
% dalja=[RN y_normalizuar];
% xlswrite('E:\PhD\Simulime\dalje_a2-h1.xlsx',dalja,'Sheet1','A1')

% paraqitja grafike e sinjaleve
figure(1), plot(dalja_y,'r'),
hold on
plot(r,'-.')
title('Sinjalet e references dhe te daljes te sistemit te mbyllur me model neural invers')
xlabel('koha diskrete')
axis([0 400 0 3])
legend('sinjali i daljes','sinjali i references','Location','Best')
grid
hold off

figure(2)
stairs(u)
title('Sinjali i komandes')
xlabel('koha diskrete')
grid
axis([0 400 -0.1 0.3])
```

File motor.m

```
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %
%kodi per ndertimin e modelit matematikor te motorit
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %
clear all, close all, clc

J = 350*10^(-7);
tao_m=0.17;
b = J/tao_m;
Ke = 0.06;
```

```
Kt=Ke;
Ra = 8;
La = 0.0076;
s = tf('s');
Ts=0.01;
K1=Kt/(J*La);
a1=(La*b+J*Ra)/(J*La);
a2=(Kt*Ke+b*Ra)/(J*La);

% modeli i vijuar i motorit DC
G_motor = Kt/((J*s+b)*(La*s+Ra)+Kt*Ke)
g2pol=zpk(G_motor);
step(G_motor)
title('pergjigja e motorit ndaj hyrjes shkalle')
grid

% modeli diskret i motorit DC
G_motor_z=c2d(G_motor,Ts,'zoh')

[NUM_M_z,DEN_M_z] = tfdata(G_motor_z);
save motor G_motor G_motor_z NUM_M_z DEN_M_z Ts %ruajta e modelit
ne skedarin motor.mat
```

File sinjali_deshiruar.m

```
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %
% kodi per gjenerimin e sinjalit te references
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %
close all, clear all, clc
% gjenerimi i sinjalit te deshiruar
stair1 = 1*ones(1,100);
stair2 = 2*ones(1,100);
r1 = [stair1 stair2];

stair3 = 1*ones(1,100); stair4 = 2*ones(1,100);
r2=[stair3 stair4];
r = [r1 r2];
save target_sig r
```

File kontroll_PID.m

```
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %
% kodi per kontrollin me rregullator PID
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** %
clear all, close all, clc

load motor
load target_sig

%parametrat e rregullatorit PID
```

```
Kp = 0.0358;
Ti = 0.0154; Td = 5.16e-006;
Ki = Kp/Ti;
Kd = Kp*Td;
Tf=0;

%funksioni transmetues i rregullatorit PID
rreg_pid = pid(Kp,Ki,Kd,Tf);
G=series(G_motor,rreg_pid);

%funksioni transmetues diskret i rregullatorit PID
rreg_pid_z = pid(Kp,Ki,Kd,Tf,Ts, 'IFormula', 'BackwardEuler');
Gz=series(G_motor_z,rreg_pid_z);

%funksioni transmetues i sistemit te mbyllur
H=feedback(G,1);
[NUM,DEN] = tfdata(H);

%funksioni transmetues diskret i sistemit te mbyllur
Hz=feedback(Gz,1);
[NUMz,DENZ] = tfdata(Hz);

y_pid = (dlsim(NUMz,DENZ,r))';

figure
plot(r), hold,plot(y_pid,'r')
grid
title('Sinjalet e references dhe te daljes te sistemit te mbyllur  
me rregullator PID')
xlabel('Koha Diskrete')
legend('sinjali references','sinjali i daljes','Location','Best')

save PID_ctrl H NUM DEN Hz NUMz DENz y_pid
```

File plot_rezultatet.m

```
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
*** %
% kodi per krahasimin grafik te dy metodave PID & Model Neural
Invers
% *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
*** %
clear all, close all, clc

load target_sig %ngarkimi i sinjalit te deshruar
load PID_ctrl %ngarkimi i pergjigjes per rregullim PID
load dalja_y %ngarkimi i pergjigjes per arkitekturen e dyte
```

```
% grafika e sinjaleve te ndryshme
figure(1)
plot(dalja_y)
hold on
plot(r, '-.')
plot(y_pid, 'r')
hold off
title('Sinjalet e references dhe te daljes te sistemit te mbyllur
me rregullatore RN dhe PID')
xlabel('koha diskrete')
% axis([0 400 0.7 2.8]),
legend('Kontrolli RN', 'sinjali i references', 'Kontrolli
PID', 'Location', 'Best')
grid
```

File kontroll2_adaptiv

```
% *** *** *** *** *** *** *** *** *** *** *** *** *** %
% kontrolli adaptiv me model neural invers
% arkitektura e dyte e rrjetit neural
% *** *** *** *** *** *** *** *** *** *** *** *** *** %
clear all, close all, clc

% leximi i peshave
load peshat2
N = length(peshat_W);
W = peshat_W(:, :, N);
w0 = spostim_w0(:, :, N);
Z = peshat_Z(:, :, N);
z0 = spostim_z0(:, :, N);
clear peshat_W peshat_Z spostim_w0 spostim_z0

% leximi i koeficienteve normalizues
load koef_norm2

eta = 0.06; % koeficienti i adaptimit te peshave

% sinjali i references
r1 = 1*ones(1,100); r2 = 2*ones(1,100);
r = [r1 r2 r1 r2];
% t = 0:pi/100:2*pi;
% r1 = 1+sin(t);
% r = [r1 r1];

y_ctr2_adapt = r; % inicializimi i daljes te sistemit te mbyllur
u = 0*ones(size(r)); % inicializimi i daljes te modelit neural

y_N = a_y*y_ctr2_adapt+b_y; %normalizimi i daljes te sistemit
```

```
u_N = a_u*u+b_u; %normalizimi i daljes te modelit neuro

% vlerat fillestare te peshave dhe spostimeve
peshat_W = cat(3,W,W,W);
spostim_w0 = cat(3,w0,w0,w0);
peshat_Z = cat(3,Z,Z,Z);
spostim_z0 = cat(3,z0,z0,z0);

for i = 3:length(r)-1
    % dalja e sistemit
    y_ctr2_adapt(i) = 0.8272*y_ctr2_adapt(i-1)-2.529e-
005*y_ctr2_adapt(i-2)+1.801*u(i-1)+0.1758*u(i-2);

    r_N = a_y*r(i+1)+b_y; % normalizimi i gabimit

    % normalizimi i daljes aktuale
    y_N(i) = a_y*y_ctr2_adapt(i)+b_y;

    % hyrja e NN
    x=[r_N y_N(i) y_N(i-1) y_N(i-2) u_N(i-1) u_N(i-2)]';

    h_in = W*x+w0;% qelizat e shtreses se fshehur
    h = logsig(h_in);% dalja e qelizave te shtreses se fshehur

    o_in = Z*h+z0;% qelizat e shtreses se daljes

    % dalja e qelizave te shtreses se daljes
    o = logsig(o_in);
    u_N(i) = o;

    % denormalizimi i daljes te rrjetit neural
    u(i) = (u_N(i)-b_u)/a_u;

% krijimi i shqetesimit
% y_ctr2_adapt(i) = y_ctr2_adapt(i)-(i==150)/2;
% y_ctr2_adapt(i) = y_ctr2_adapt(i)-(i==80)/2;
% y_ctr2_adapt(i) = y_ctr2_adapt(i)-(i==200)/2;
% y_ctr2_adapt(i) = y_ctr2_adapt(i)-(i==250)/2;

%%% ripershtatja e peshave dhe spostimeve %%%
e = r(i)-y_ctr2_adapt(i); %gabimi reference-dalje
delta = o.*(ones(1,1)-o).*e;% gabimi per pwrhapjein e pasem

% pershtatja e peshave Z dhe spostimit z0
Z = Z+eta*delta*h';
z0 = z0+eta*delta;

% gabimi ne daljen e shtreses se fshehur
```

```
dh = h.*(ones(size(h))-h).*(Z'*delta);

% pershtatja e peshave W dhe spostimit w0
W = W+eta*dh*x';
w0 = w0+eta*dh;

% ruajtja e peshave dhe spostimeve
peshat_W = cat(3,peshat_W,W);
spostim_w0 = cat(3,spostim_w0,w0);
peshat_Z = cat(3,peshat_Z,Z);
spostim_z0 = cat(3,spostim_z0,z0);
end % mbyllja e lakut for

% save y_ctr2_adapt y_ctr2_adapt
RN=length(Z);
dalja=[RN y_ctr2_adapt];

% paraqitja grafike e sinjaleve
figure(1), plot(y_ctr2_adapt,'r'), hold, plot(r,'-.')
title('Sinjalet e references dhe daljes te sistemit te mbyllur me kontroll adaptiv')
xlabel('Koha diskrete'), axis([0 400 0 2.5]),
legend('sinjali i daljes','sinjali i references','Location','Best')
grid
% text(106,1.7,'\leftarrow sinjali i daljes',
'HorizontalAlignment','left')
figure(2), stairs(u), title('Sinjali i komandes')
xlabel('Koha diskrete'), grid, axis([0 400 -0.2 0.4])

% *****
load y_ctr2

figure(3), plot(y_ctr2),
hold on
plot(r,'-.'),
hold on
plot(y_ctr2_adapt,'r')
% plot(t,r,':'), hold on, plot(t,y,'r'), plot(t,y_ctr2)
axis([5 400 0 2.5])
title('Krahasimi i daljeve te sistemit te mbyllur')
xlabel('koha diskrete'), hold off
legend('dalja e sistemit per trajnim ne linje te modelit neural','sinjali references','dalja e sistemit per trajnim jashte linje te modelit neural','Location','Best')
grid on
```


Bibliografia

- [1] S. S. Haykin, *Neural networks and learning machines*, 3rd ed. New York: Prentice Hall, 2009.
- [2] Bernard Widrow and Peter N. Stearns, *Adaptive Signal Processing*. New Jersey: Prentice-Hall, 1985.
- [3] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.
- [4] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*. New York: Wiley, 1949.
- [5] B. Widrow and M. E. Hoff, "Adaptive Switching Circuits," presented at the IRE WESCON Convention Record, 1960, pp. 96–104.
- [6] K. Steinbuch, *Die lernmatrix. Kybernetik (Biological Cybernetics)*. 1961.
- [7] M. Minsky and S. Papert, *Perceptrons*. MIT Press, Cambridge, Mass, 1969.
- [8] T. Kohonen, "Correlation matrix memories," *IEEEtC*, pp. 353–359, 1972.
- [9] J. A. Anderson, "A simple neural network generating an interactive memory," *Mathematical Biosciences*, vol. 14, pp. 197–200, 1972.
- [10] C. von der Malsburg, "Self-organizing of orientation sensitive cells in striate cortex," *Kybernetik*, pp. 85–100, 1973.
- [11] P. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," PhD Thesis, Harvard University, 1974.
- [12] S. Grossberg, "Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors," *Biological Cybernetics*, pp. 121–134, 1976.
- [13] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, p. 43:59–69, 1982.
- [14] T. Kohonen, "The self-organizing map," *Neurocomputing*, p. 21(1-3):1–6, 1998.
- [15] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proc. of the National Academy of Science, USA*, 1982, p. 79:2554–2558.
- [16] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A neural network model for a mechanism of visual pattern recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, p. 13(5):826–834, 1983.
- [17] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, p. 323:533–536, 1986.
- [18] Grossberg, S., *Neural Networks and Natural Intelligence*. Cambridge, MA: MIT Press, 1988.

- [19] P. Kerlirzin and F. Vallet, "Robustness in Multilayer Perceptrons," *Neural Computation*, vol. 5, no. 4, pp. 473–482, May 1993.
- [20] Arbib, M.A., *Brains, Machines, and Mathematics*, 2d ed. New York: Springer-Verlag, 1987.
- [21] Ramón y Cajál, S, *Histologie du Systéms Nerveux de l'homme et des vertébrés*. Paris: Maloine, 1911.
- [22] Shepherd, G.M and C.Koch, "Introduction to synaptic circuits," in *The Synaptic Organization of the Brain*, New York: Oxford University Press, 1990, pp. 3–31.
- [23] Eggermont, J.J., *The Correlative Brain: Theory and Experiment in Neural Interaction*. New York: Springer-Verlag, 1990.
- [24] Churchland, P.S and T.J. Sejnowski, *The Computational Brain*. Cambridge, MA: MIT Press, 1992.
- [25] Eric R. Kandel, James H. Schwartz, Thomas M. Jessell, Steven A. Siegelbaum, and A. J. Hudspeth, *Principles of neural science*. New York: McGraw-Hill Medical, 2012.
- [26] Freeman, W.J., *Mass Action in the Nervous System*. New York: Academic Press, 1975.
- [27] Anderson, J.A, *Introduction to Neural Networks*. Cambridge, MA: MIT Press, 1995.
- [28] Brodal, A, *Neurological Anatomy in Relation to Clinical Medicine*, 3d ed. New York: Oxford University Press, 1981.
- [29] Ben Krose and Patrick van der Smagt, *An Introduction to Neural Networks*. University of Amsterdam, 1996.
- [30] R. S. Burns, *Advanced control engineering*. Oxford ; Boston: Butterworth-Heinemann, 2001.
- [31] B. K. Bose, *Modern power electronics and AC drives*. Upper Saddle River, NJ: Prentice Hall, 2002.
- [32] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural network design*. 1996.
- [33] P. Peretto, *An introduction to the modeling of neural networks*. Cambridge ; New York: Cambridge University Press, 1992.
- [34] R. Rojas, *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.
- [35] B. Yegnanarayana, *Artificial Neural Networks*. Prentice-Hall of India, 2005.
- [36] L. Jain and A. M. Fanelli, *Recent advances in artificial neural networks*. CRC Press, 2000.
- [37] Chung, Y and Kusiak, A, "Grouping parts with a neural network," *Journal of Manufacturing Systems*, vol. 13, no. 4, pp. 262–275, 1994.
- [38] Kusiak, A and Lee, H, "Neural computing based design of components for cellular manufacturing," *International Journal of Production Research*, vol. 34, no. 7, pp. 1777–1790, 1996.

- [39] I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *J. Microbiol. Methods*, vol. 43, no. 1, pp. 3–31, 2000.
- [40] L. C. Jain and N. M. Martin, Eds., *Fusion of neural networks, fuzzy sets, and genetic algorithms: industrial applications*. Boca Raton: CRC press, 1999.
- [41] M. A. Shahin, M. B. Jaksa, and H. R. Maier, "State of the art of artificial neural networks in geotechnical engineering," *Electron. J. Geotech. Eng.*, vol. 8, pp. 1–26, 2008.
- [42] Cybenko, G, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [43] Bounds, D.G, Lloyd, PJ, Mathew, B, and Waddell, G, "A multilayer perceptron network for the diagnosis of low back pain," presented at the Proc. of 2nd IEEE Annual Int'l Conf. on Neural Networks, San Diego, NJ, USA, 1988, pp. 481–489.
- [44] Chester, D.L, "Why two hidden layers are better than one," presented at the Proc. of 4th IEEE Annual Int'l Conf. on Neural Networks, Washington, DC, NJ, USA, 1990, pp. 1265–1268.
- [45] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in neural information processing systems*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, 1990, pp. 524–532.
- [46] Lawrence, J, *Introduction to Neural Networks: Design, Theory, and Applications*, 6th ed. Nevada City, CA: California Scientific Software, 1994.
- [47] Swingler, K, *Applying Neural Networks: A Practical Guide*. San Francisco: Morgan Kaufmann Publishers Inc, 1996.
- [48] Hecht-Nelson, R., "Kolmogorov's mapping neural network existence theorem," in *Proc. of 1st IEEE Annual Int'l Conf. on Neural Networks*, San Diego, NJ, USA, 1987, p. 111.11-111.14.
- [49] Lawrence, J and Fredrickson, J, *BrainMaker User's Guide and Reference Manual*, 7th ed. Nevada City, CA: California Scientific Software, 1988.
- [50] Marchandani, G and Cao, W, "On hidden nodes for neural nets," *IEEE Trans. on Circuits and Systems*, vol. 36, no. 5, pp. 661–664, 1989.
- [51] Y. Kang, M.-H. Chu, and M.-C. Chen, *The Artificial Neural Network Applied to Servo Control System*. Springer, 2007.
- [52] J. R. Rabunal and J. Dorado, Eds., *Artificial neural networks in real-life applications*. Hershey PA: Idea Group Pub, 2006.
- [53] K. Suzuki, *Artificial neural networks - industrial and control engineering applications*. Rijeka: InTech : Intech, 2011.
- [54] Krzysztof Patan, *Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes*, vol. 377. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

- [55] Haider A. F. Almurib, Ahmad A. Mat Isa, and Hayder M.A.A. Al-Assadi, “Direct Neural Network Control via Inverse Modelling: Application on Induction Motors,” in *Artificial Neural Networks - Industrial and Control Engineering Applications*, InTech, 2011.
- [56] J. Schumann, P. Gupta, and Y. Liu, “Application of Neural Networks in High Assurance Systems: A Survey,” in *Applications of Neural Networks in High Assurance Systems*, Springer, 2010, pp. 1–19.
- [57] “SPPA-D3000 Plant Monitor Technical Description.” Siemens, Jun-2010.
- [58] Christine R  th, “Sustainable Power Generation: Neural Networks - Energy & Efficiency.” [Online]. Available: <http://www.siemens.com/innovation/en/home/pictures-of-the-future/energy-and-efficiency/sustainable-power-generation-neural-networks.html>.
- [59] “NASA - Dryden Flight Research Center - News Room: News Releases: NASA NEURAL NETWORK PROJECT PASSES MILESTONE | NASA.” [Online]. Available: <http://www.nasa.gov/centers/dryden/news/NewsReleases/2003/03-49.html#.VvIKXOYuvIU>. [Accessed: 28-Mar-2016].
- [60] “Aerodynamic Design Using Neural Networks | NASA,” 16-Nov-2015. [Online]. Available: <http://www.nasa.gov/ames-partnerships/information-technology-and-software/technology-opportunity-aerodynamic-design-using-neural-networks>.
- [61] J. J. Hageman, M. Smith, and S. Stachowiak, “Integration of online parameter identification and neural network for in-flight adaptive control,” *INTEGRATION*, vol. 11, p. 14, 2003.
- [62] P. S. Williams-Hayes, “Selected Flight Test Results for Online Learning Neural Network-Based Flight Control System,” NASA, NASA/TM-2004-212857, 2004.
- [63] Bernhard M  ller and Otthein Herzog, “Industry 4.0 and Urban Development: The Case of India,” Deutsche Akademie der Technikwissenschaften, Aug. 2015.
- [64] M. Brettel, N. Friederichsen, M. Keller, and M. Rosenberg, “How virtualization, decentralization and network building change the manufacturing landscape: An Industry 4.0 Perspective,” *Int. J. Mech. Ind. Sci. Eng.*, vol. 8, no. 1, pp. 37–44, 2014.
- [65] D. Huber and ABB, “Industrie 4.0 - Possible use cases in the process industry,” International Conference - Industry 4.0 in Practice, 2015.
- [66] I. Lauri and NORTAL, “How to use Big Data in Industry 4.0 implementations,” International Conference - Industry 4.0 in Practice, 2015.
- [67] “Industry 4.0 - Fraunhofer IPT.” [Online]. Available: <http://www.ipt.fraunhofer.de/en/industry40.html>. [Accessed: 12-Mar-2016].
- [68] “Industry 4.0 – Connected, adaptive production - brochure.” Fraunhofer Institute for Production Technology IPT, 2015.

- [69] “The Internet of Things: Industrie 4.0 vs. the Industrial Internet | MAPI,” 23-Jul-2015. [Online]. Available: <https://www.mapi.net/forecasts-data/internet-things-industrie-40-vs-industrial-internet>.
- [70] R. Anderl, “What is Industrie 4.0 and how will it create the new growth,” International Conference - Industry 4.0 in Practice, 2015.
- [71] R. Kallenbach and BOSCH, “The Internet of Things: Connected products – connected manufacturing,” International Conference - Industry 4.0 in Practice, 2015.
- [72] G. Hotson, D. P. McMullen, M. S. Fifer, M. S. Johannes, K. D. Katyal, M. P. Para, R. Armiger, W. S. Anderson, N. V. Thakor, B. A. Wester, and N. E. Crone, “Individual finger control of a modular prosthetic limb using high-density electrocorticography in a human subject,” *J. Neural Eng.*, vol. 13, no. 2, p. 26017, Apr. 2016.
- [73] M. Mano, G. Capi, N. Tanaka, and S. Kawahara, “An Artificial Neural Network Based Robot Controller that Uses Rat’s Brain Signals,” *Robotics*, vol. 2, no. 2, pp. 54–65, Apr. 2013.
- [74] Mike James, “A Robot Learns To Do Things Using A Deep Neural Network,” 27-May-2015. [Online]. Available: <http://www.i-programmer.info/news/105-artificial-intelligence/8619-a-robot-learns-to-do-things-using-a-deep-neural-network.html>. [Accessed: 05-May-2016].
- [75] “Brainpower under the bonnet | The Economist.” [Online]. Available: <http://www.economist.com/node/7001748>. [Accessed: 05-May-2016].
- [76] Deng, Jiamei, Stobart, Richard, and M. Basti, “The Applications of Artificial Neural Networks to Engines,” in *Artificial Neural Networks - Industrial and Control Engineering Applications*, K. Suzuki, Ed. InTech, 2011.
- [77] “Qualcomm Announces Breakthrough Automotive Processor with Integrated LTE Modem and Machine Intelligence to Further its Leadership in the Connected Car | Qualcomm,” 05-Jan-2016. [Online]. Available: <https://www.qualcomm.com/news/releases/2016/01/05/qualcomm-announces-breakthrough-automotive-processor-integrated-lte-modem>.
- [78] C. T. Barnett and P. M. Williams, “Using Geochemistry and Neural Networks to map Geology under Glacial Cover Geoscience BC Project 2008-003,” BW Mining, Final Report for Geoscience BC Project 2008-003 Geoscience BC Report 2009-003, 2009.
- [79] S. Diersen, E.-J. Lee, D. Spears, P. Chen, and L. Wang, “Classification of Seismic Windows Using Artificial Neural Networks,” *Procedia Comput. Sci.*, vol. 4, pp. 1572–1581, 2011.
- [80] Chi-Leung Hui, Ed., *Artificial Neural Networks - Application*. INTECH Open Access Publisher, 2011.
- [81] M. Sordo, “Introduction to neural networks in healthcare,” *Open Clin. Doc.*, 2002.

- [82] S. Cerutti and C. Marchesi, Eds., *Advanced methods of biomedical signal processing*. Hoboken, N.J. : Piscataway, NJ: Wiley ; IEEE Press, 2011.
- [83] A. Amirov, O. Gerget, D. Devjatyh, and A. Gazaliev, “Medical Data Processing System based on Neural Network and Genetic Algorithm,” *Procedia - Soc. Behav. Sci.*, vol. 131, pp. 149–155, May 2014.
- [84] D. L. Hudson and M. E. Cohen, *Neural networks and artificial intelligence for biomedical engineering*. New York: Institute of Electrical and Electronics Engineers, 2000.
- [85] D. Farina, W. Jensen, and M. Akay, Eds., *Introduction to neural engineering for motor rehabilitation*. Hoboken, N.J: John Wiley & Sons, 2013.
- [86] K. S. Nikita and Institute of Electrical and Electronics Engineers, Eds., *Handbook of biomedical telemetry*. Hoboken, New Jersey: Wiley : IEEE Press, 2014.
- [87] M. Akay, *Handbook of neural engineering*. John Wiley & Sons, 2007.
- [88] M. Akay and IEEE Engineering in Medicine and Biology Society, Eds., *Nonlinear biomedical signal processing*. New York: IEEE Press, 2000.
- [89] M. Zhang, Ed., *Artificial higher order neural networks for economics and business*. Hershey: Information Science Reference, 2009.
- [90] L. Falat, Z. Stanikova, M. Durisova, B. Holkova, and T. Potkanova, “Application of Neural Network Models in Modelling Economic Time Series with Non-constant Volatility,” *Procedia Econ. Finance*, vol. 34, pp. 600–607, 2015.
- [91] E. Kalyvas, “Using neural networks and genetic algorithms to predict stock market returns,” University of Manchester, 2001.
- [92] Simon Bisson, “How Facebook’s Open AI Research Uses GPU Neural Networks - The New Stack,” 2015. [Online]. Available: <http://thenewstack.io/facebooks-open-ai-research-uses-gpu-neural-networks/>.
- [93] “Google’s Smart Reply for Inbox Uses AI to Answer Emails | Digital Trends.” [Online]. Available: <http://www.digitaltrends.com/mobile/google-inbox-smart-reply/>. [Accessed: 29-Mar-2016].
- [94] “Google Applies Machine Learning to Gmail Spam Filters | Digital Trends,” 10-Jul-2015. [Online]. Available: <http://www.digitaltrends.com/android/google-starts-applying-artificial-intelligence-to-gmail-spam-filters/>.
- [95] S. Draghici, “Neural networks in analog hardware—Design and implementation issues,” *Int. J. Neural Syst.*, vol. 10, no. 1, pp. 19–42, 2000.
- [96] Y. Liao, “Neural networks in hardware: A survey,” *Dep. Comput. Sci. Univ. Calif.*, 2001.
- [97] M. Forssell, “Hardware Implementation of Artificial Neural Networks,” <http://users.ece.cmu.edu/~pgrover/teaching/files/NeuromorphicComputing.pdf>, 2014.

- [98] I. Aybay, S. Cetinkaya, and U. Halici, "Classification of neural network hardware," *Neural Netw. World*, vol. 6, no. 1, pp. 11–27, 1996.
- [99] Ö. AKIN and I. Alan, "The use of FPGA in field-oriented control of an induction machine," *Turk. J. Electr. Eng. Comput. Sci.*, vol. 18, no. 6, pp. 943–962, 2010.
- [100] Asha c.n and Mr. N.Jayakumar, "FPGA implementation for speed monitoring and control of DC motor using Soft Computing technique based controller," *International Journal of Engineering Science and Computing*, Apr. 2014.
- [101] J. A. Caballero Hernández, M. Díaz Salazar, M. Moradillos Paz-Lago, and S. Pavoni Oliver, "Implementación de la Función Sigmoidal Logarítmica en un FPGA," *Ing. Electrónica Automática Comun.*, vol. 35, no. 2, pp. 35–44, 2014.
- [102] Gabriele-Maria LOZITO , Antonino LAUDANI , Francesco RIGANTI-FULGINEI , Alessandro SALVINI, and Gabriele-Maria LOZITO , Antonino LAUDANI , Francesco RIGANTI-FULGINEI , "FPGA Implementations of Feed Forward Neural Network by using Floating Point Hardware Accelerators," *Adv. Electr. Electron. Eng.*, vol. 12, no. 1, 2014.
- [103] R. Gadea, J. Cerdá, F. Ballester, and A. Mocholí, "Artificial neural network implementation on a single FPGA of a pipelined on-line backpropagation," in *Proceedings of the 13th international symposium on System synthesis*, 2000, pp. 225–230.
- [104] A. Gomperts, A. Ukil, and F. Zurfluh, "Implementation of Neural Network on Parameterized FPGA.," in *AAAI Spring Symposium: Embedded Reasoning*, 2010, pp. 45–51.
- [105] D. E. Khodja, A. Kheldoun, and L. Refoufi, "Sigmoid function approximation for ANN implementation in FPGA devices," 2010.
- [106] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, Aug. 2014.
- [107] K. Ovtcharov, O. Ruwase, J.-Y. Kim, J. Fowers, K. Strauss, and E. S. Chung, "Accelerating deep convolutional neural networks using specialized hardware," *Microsoft Res. Whitepaper*, vol. 2, 2015.
- [108] "CogniMem Technologies Inc. • Products: Chips & Modules: CM1K Chip." [Online]. Available: <http://www.cognimem.com/products/chips-and-modules/CM1K-Chip/>.
- [109] E. Kussul, T. Baidyk, and D. C. Wunsch, *Neural Networks and Micromechanics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

- [110] “Introducing Qualcomm Zeroth Processors: Brain-Inspired Computing | Qualcomm,” 11-Oct-2013. [Online]. Available: <https://www.qualcomm.com/news/onq/2013/10/10/introducing-qualcomm-zeroth-processors-brain-inspired-computing>.
- [111] E. Z. Mohammed and H. K. Ali, “Hardware Implementation of Artificial Neural Network Using Field Programmable Gate Array,” *Int. J. Comput. Theory Eng.*, pp. 780–783, 2013.
- [112] “IBM Research: Brain-inspired Chip.” [Online]. Available: <http://www.research.ibm.com/articles/brain-chip.shtml>.
- [113] M. A. Arbib, Ed., *The handbook of brain theory and neural networks*, 2nd ed. Cambridge, Mass: MIT Press, 2003.
- [114] M. Valle, “Analog VLSI Implementation of Artificial Neural Networks with Supervised On-Chip Learning,” *Analog Integrated Circuits and Signal Processing*, vol. 33, no. 3, pp. 263–287, Dec. 2002.
- [115] K. P. Sridhar, B. Vignesh, S. Saravanan, M. Lavanya, and V. Vaithiyanathan, “Design and Implementation of Neural Network Based circuits for VLSI testing,” *World Applied Sciences Journal*, pp. 11–117, 2014.
- [116] R. A. Nawrocki, R. M. Voyles, and S. E. Shaheen, “Neurons in Polymer: Hardware Neural Units Based on Polymer Memristive Devices and Polymer Transistors,” *Electron Devices IEEE Trans. On*, vol. 61, no. 10, pp. 3513–3519, 2014.
- [117] J. A. Starzyk and others, “Comparison of two memristor based neural network learning schemes for crossbar architecture,” in *Advances in Computational Intelligence*, Springer, 2013, pp. 492–499.
- [118] D. Querlioz, O. Bichler, and C. Gamrat, “Simulation of a memristor-based spiking neural network immune to device variations,” in *Neural Networks (IJCNN), The 2011 International Joint Conference on*, 2011, pp. 1775–1781.
- [119] Tom Simonite, “A Better Way to Build Brain-Inspired Chips,” 06-May-2015. [Online]. Available: <https://www.technologyreview.com/s/537211/a-better-way-to-build-brain-inspired-chips/>.
- [120] G. E. Moore, “Progress in digital integrated electronics,” presented at the International Electron Devices Meeting, IEEE, 1975, vol. 178, pp. 179–181.
- [121] “Moore’s law - Wikipedia, the free encyclopedia,” 15-May-2016. [Online]. Available: https://en.wikipedia.org/wiki/Moore%27s_law.
- [122] Allison Mills, “Memristors: Making a New Generation for Digital Memory and Computation,” 02-Feb-2016. [Online]. Available: <http://www.mtu.edu/news/stories/2016/february/memristors-making-new-generation-for-digital-memory-computation.html>.
- [123] Russ Arensman, “Despite HP’s Delays, Memristors are Now Available | Electronics360,” 08-Mar-2016. [Online]. Available:

- <http://electronics360.globalspec.com/article/6389/despite-hp-s-delays-memristors-are-now-available>.
- [124] Sebastian Anthony, “HP and SanDisk join forces to finally bring memristor-like tech to market | Ars Technica,” 12-Oct-2015. [Online]. Available: <http://arstechnica.com/gadgets/2015/10/hp-and-sandisk-join-forces-to-finally-bring-memristor-like-tech-to-market/>.
- [125] Leon Chua, “Memristor-The missing circuit element,” *IEEE*, vol. 18, no. 5, pp. 507–519, Sep. 1971.
- [126] A. J. Maren, C. T. Harston, and R. M. Pap, *Handbook of neural computing applications*. San Diego: Academic Press, 1990.
- [127] M. T. Hagan, H. B. Demuth, and O. D. Jesús, “An introduction to the use of neural networks in control systems,” *Int. J. Robust Nonlinear Control*, vol. 12, no. 11, pp. 959–985, 2002.
- [128] K. Veisllari and E. Buzi, “Kontrolli në hapësirën e çiftuesit kinematik të një krah robotik dhe simulimi në ambientin MATLAB,” *Univ. Politek. Tiranes*, vol. 1, 2004.
- [129] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics Modelling, Planning and Control*. Springer-Verlag London Limited, 2009.
- [130] M. Mokhtari and M. Marie, *Engineering applications of MATLAB 5.3 and SIMULINK 3*. London ; New York: Springer, 2000.
- [131] ElettronicaVeneta, “Student-Trainer MPT/EV. Controllo di velocita per motore d.c. MODULO G14/EV.”.
- [132] P. Marango, *Kontrolli i Proceseve*. SHBLU, 2001.
- [133] K. Veisllari, *Kontrolli Numerik 1*. SHBLU, 1997.
- [134] K. J. Åström and T. Hägglund, *Advanced PID control*. Research Triangle Park, NC: ISA-The Instrumentation, Systems, and Automation Society, 2006.