

Sumean logiikan ja neuroverkkojen työkaluselvitys

Kari Haataja
VTT Elektronikka



ISBN 951-38-5283-0 (nid.)
ISSN 1235-0605 (nid)

ISBN 951-38-5284-9 (URL: <http://www.inf.vtt.fi/pdf/>)
ISSN 1455-0865 (URL: <http://www.inf.vtt.fi/pdf/>)

Copyright © Valtion teknillinen tutkimuskeskus (VTT) 1998

JULKAISIJA – UTGIVARE – PUBLISHER

Valtion teknillinen tutkimuskeskus (VTT), Vuorimiehentie 5, PL 2000, 02044 VTT
puh. vaihde (09) 4561, faksi 456 4374

Statens tekniska forskningscentral (VTT), Bergsmansvägen 5, PB 2000, 02044 VTT
tel. växel (09) 4561, fax 456 4374

Technical Research Centre of Finland (VTT), Vuorimiehentie 5, P.O.Box 2000, FIN-02044 VTT, Finland
phone internat. + 358 9 4561, fax + 358 9 456 4374

VTT Elektroniikka, Sulautetut ohjelmistot, Kaitoväylä 1, PL 1100, 90571 OULU
puh. vaihde (08) 551 2111, faksi (08) 551 2320

VTT Elektronik, Inbyggd programvara, Kaitoväylä 1, PB 1100, 90571 ULEÅBORG
tel. växel (08) 551 2111, fax (08) 551 2320

VTT Electronics, Embedded Software, Kaitoväylä 1, P.O.Box 1100, FIN-90571 OULU, Finland
phone internat. + 358 8 551 2111, fax + 358 8 551 2320

Tekninen toimitus Kerttu Tirronen

LIBELLA PAINOPALVELU OY, ESPOO 1998

Haataja, Kari. Sumean logiikan ja neuroverkkojen työkaluselvitys [Survey of fuzzy logic and neural network tools]. Espoo 1998, Valtion teknillinen tutkimuskeskus, VTT Tiedotteita – Meddelanden – Research Notes 1895. 58 s. + liitt. 7 s.

UDK 681.3.06:519.68:510.22

Avainsanat fuzzy logic, fuzzy systems, neural nets, software, commercialization

Tiivistelmä

Tiedote sisältää kartoituksen sumean logiikan ja neuroverkkotyökalujen tarjoamista mahdollisuuksista muutaman esimerkkityökalun avulla. Raportti ei ole täydellinen luettelo ohjelmistoista, vaan se pyrkii antamaan lukijalle käsityksen siitä, mitä käytettäviltä työkaluilta voi tällä hetkellä vaatia ja mitä ne tarjoavat. Kartoitukseen on valittu neljä sumean logiikan ja neljä neuroverkkojen työkalua.

Tiedotteessa selostetaan lyhyesti Suomessa kehitettyjä kaupallisia ja ei-kaupallisia sumean logiikan ja neuroverkkojen työkaluja. Samalla pohditaan näiden työkalujen mahdollisuuksia menestyä kaupallisesti kotimaassa ja ulkomailla.

Työkalujen kartoitus on liitetty kehitysprosessin malliin, joka esitetään varsin laajasti. Malli tarjoaa systemaattisen lähestymistavan kehitettäessä sumean logiikan ja neuroverkkojen sovelluksia. Kehitysprosessi jakaa projektin vaiheisiin ja tehtäviin, joita kehitystyön aikana tehdään.

Tiedotteessa annetaan esimerkkejä Suomessa tehdyistä projekteista ja tavoista, joilla projekteja on suoritettu onnistuneesti ja saatu aikaan toimivia sovelluksia.

Liitteenä on yhteystietoja kaupallisista sumean logiikan ja neuroverkko-ohjelmistojen valmistajista. Luettelo on hyödynnettävissä lisätietojen hankinnassa myös VTT:n web-palvelimella osoitteessa <http://www.ele.vtt.fi/projects/nessu>.

Haataja, Kari. Sumean logiikan ja neuroverkkojen työkaluselvitys [Survey of fuzzy logic and neural network tools]. Espoo 1998, Technical Research Centre of Finland, VTT Tiedotteita – Meddelanden – Research Notes 1895. 58 p. + app. 7 p.

UDC 681.3.06:519.68:510.22

Keywords fuzzy logic, fuzzy systems, neural nets, software, commercialization

Abstract

This survey focuses on the use of fuzzy logic and neural network software tools in engineering applications. A few software tools are selected as examples of current capabilities. This report does not list all of the available tools, but gives an idea what to expect and what to ask for while selecting a tool. Four fuzzy logic and four neural network tools are discussed.

Some commercial and non-commercial tools that have been developed in Finland are presented. In this same context a short discussion of commercialization of these tools in domestic and international market is provided.

Selected tools are related into a development process of fuzzy and neural applications. An extensive description of this process can be found in this report. The process gives a systematic approach to application development. Process divides a project into several stages and tasks to be carried out.

At the end of this report a few Finnish examples of projects and working methods are presented.

In the appendix, a list of commercial fuzzy logic and neural network software vendors and their contact information is given. This list can also be found in VTT's web-site <http://www.ele.vtt.fi/projects/nessu> as a link to further information on other software tools.

Alkusanat

NESSU-projekti (Neuroverkko ja sumean logiikan työkalut) käynnistyi syksyllä 1996 TEKESin tilaustutkimuksena ja päättyi kesäkuussa 1997. Sen toteuttamisesta on vastannut VTT Elektronikka.

NESSU-projektin tavoitteena oli kartoittaa nykyinen työkalutilanne sekä sumean logiikan että neuroverkkojen osalta. Ensisijaisesti oltiin kiinnostuneita työkaluista, joilla on vahva merkitys ja kannatus käyttäjäkunnassa ja jotka soveltuvat yritysten omaan sovelluskehitykseen. Kartoitukseen valittiin muutama työkalu, joiden ominaisuuksiin tutustuttiin lähemmin. Näiden keskeisimpien työkalujen ominaisuuksia sovitettiin neuroverkko- tai sumean logiikan projektin kehitysmalliin.

Tässä julkaisussa esitetään projektin pääasialliset tulokset. Julkaisun pääosan kirjoitti ja asiasisällön toimitti Kari Haataja. Tapio Rauma VTT Elektronikasta laati katsauksen sovelluskohteisiin (luku 4.2). Kotimaisten työkalujen kaupallistamisesta (luku 4.9) kirjoittivat Ossi Taipale Taipale Engineering Oy:stä ja Matti Sihto TEKESistä. Osuuden sumean logiikan soveltamisesta selluteollisuuteen (luku 5.2) laati Kari Lampela Valmet Automation Oy:stä.

Projektin vastuullisena johtajana kiitän kaikkia projektiin osallistuneita. Samoin kiitän tutkimuslaitosten ja yritysten edustajia rakentavista neuvoista ja kommentteista projektin aikana. TEKESiä kiitän hankkeen rahoittamisesta.

Oulussa 23. päivänä maaliskuuta 1998.

Kari Haataja
Projektipäällikkö

Sisällysluettelo

Tiivistelmä	3
Abstract	4
Alkusanat	5
1. Johdanto	8
2. Lyhyt katsaus sovelluskohteisiin	9
3. Kehitysprosessin malli	11
3.1 Yleistä	11
3.2 Ongelman määrittely	12
3.3 Asteittainen kehitys	13
3.4 Sovelluksen integrointi	14
3.5 Ylläpito	15
4. Työkalut	17
4.1 Sumean logiikan perustyökalut: fuzzyTECH ja TILShell	19
4.2 Neuroverkkojen perustyökalut: NeuralWorks ja BrainMaker	23
4.3 Matematiikkatyökalu: Matlab + Fuzzy Logic Toolbox & Neural Network Toolbox	26
4.4 Data Mining-työkalu: DataEngine	30
4.5 Tukityökalut: sumean mallin generointimenetelmät	33
4.6 Client/Server-työkalut: G2 + NeurOn-Line	35
4.7 Yhteenveto kaupallisista työkaluista	36
4.8 KOTIMAISET TYÖKALUT	42
4.8.1 Q-Opt	42
4.8.2 AboaFuzz	43
4.8.3 Neural-MILL	44
4.8.4 Tutkimuslaitosten työkalut	45
4.9 KOTIMAISTEN TYÖKALUJEN KAUPALLISTAMINEN	47
4.9.1 Ohjelmistoliiketoiminnan yleispiirteitä	47
4.9.2 Kaupallistamisen mahdollisuuksia ja vaikeuksia	49
4.9.3 Ratkaisumalleja	50
5. Case-tapaukset	51
5.1 Meesauunin sumea säätö	51
5.2 Sumean logiikan soveltaminen selluteollisuuteen	52
5.3 Neuroverkkojen soveltaminen paperiteollisuuteen	54

6. Yhteenveto	56
Lähdeluettelo	58

LIITTEET

Liite A: Yhteystietoja Suomessa

Liite B: Hintatietoja

Liite C: Sumean logiikan työkaluvalmistajien yhteystietoja

Liite D: Neuroverkkotyökaluvalmistajien yhteystietoja

1. Johdanto

Sumean logiikan ja neuroverkkojen sovellusten määrä on kasvanut viime vuosina eikä laantumista ole vielä näkyvässä. Aluksi sovellukset olivat lähtöisin pääosin tutkimuslaitoksista tai yliopistoista, mutta viime aikoina yritykset ovat lisänneet omaa panostustaan ja ryhtyneet tekemään sovelluksia myös omin voimin. Merkittävimmät syyt kehitykseen ovat tehtyjen sovellusten hyvät tulokset sekä tietämyksen ja osaamisen lisääntyminen.

Sumealla logiikalla ja neuroverkoilla toteutetut sovellukset saattavat tuntua hyvin monimutkaisilta ja vaikeasti hallittavilta, mutta pienellä asiaan perehtymisellä epäilyt hälvenevät. Tutkimusten ja sovellusten myötä julkaisujen määrä on kasvanut Suomessa ja osa julkaisuista onkin kirjoitettu suomeksi. Tekes on julkaissut selvitykset neuroverkoista [Koikkalainen toim. 1994] ja sumeasta logiikasta [Isomursu et al. 1993], joiden avulla on mahdollista tutustua tekniikoiden perusteisiin ja terminologiaan. Tehdyistä projekteista on julkaistu toteutuskokemuksia VTT Tiedotteita - Meddelanden - Research Notes -sarjassa Oppivien ja älykkäiden järjestelmien sovellukset -ohjelman puitteissa [Berg 1996].

Sovellettaessa oppivia ja älykkäitä menetelmiä teollisiin kohteisiin on syytä muistaa, että tarkoituksena on nimenomaan ratkaista ongelma siihen parhaiten soveltuvalla tekniikalla. Itse menetelmästä ei saisi tulla pääasia ongelmaa ratkaistaessa. Sovelluskehityksen alkuvaiheessa tulee huomioida kohdistaa lähtötietojen saatavuuteen ja laatuun sekä lopullisen integrointiympäristön vaatimuksiin ja tulevaan ylläpitoon. On hyvin erilaista toteuttaa järjestelmä vaatimattomaan ohjelmoitavaan logiikkaan kuin tehdä demonstraatio Windows-ympäristöön, sillä käytännön rajoitukset ovat aivan erilaiset. Sovelluksen kehitystä voidaan nopeuttaa ja helpottaa valitsemalla työkalut ja menetelmät oikein.

Tässä raportissa on tehty katsaus tyypillisiin kaupallisesti tarjolla oleviin sumean logiikan ja neuroverkkotyökaluihin. Vertailun vuoksi raportissa on mukana kotimaassa tehtyjä työkaluja, joista tosin kaikki eivät ole kaupallisesti saatavissa. Tähän liittyen raportissa on myös arvio kotimaisten ohjelmistojen mahdollisuuksista menestyä kaupallisesti. Raportissa käydään muutaman keskeisimmän ja laajimmalle levinneen työkalun avulla läpi niitä piirteitä, jotka ovat hyviä ja joita mahdollisesti tarvitaan sovelluksen kehityksessä. Työkalujen vertailemiseksi niiden toiminnot ja käyttö on selvityksessä liitetty kehitysprosessiin, joka on projektin tekemisen runkona. Siinä on määritelty tehtäviä ja välivaiheita, joiden avulla halutun ratkaisun saavuttamista voidaan helpottaa ja nopeuttaa. Raportissa on myös yhteenveto, johon on kerätty muutamassa Suomalaisessa sumean logiikan ja neuroverkkotekniikan projektissa käytetty projektirunko.

2. Lyhyt katsaus sovelluskohteisiin

Kirjallisuudessa on esitetty runsaasti erilaisia sovelluskohteita erilaisille älykkäiden järjestelmien tekniikoille. Monissa tapauksissa on kuitenkin annettu varsin optimistinen kuva tekniikoiden soveltuvuudesta erilaisiin tarpeisiin. On selvää, että esimerkiksi neuroverkkoja ja sumeaa logiikkaa voidaan usein soveltaa saman ongelman ratkaisuun, ja monissa tapauksissa kummallakin tekniikalla saavutetaan tuloksia. Tekniikoilla on kuitenkin omat vahvuutensa ja heikkoutensa. Onkin syytä pyrkiä etsimään sellaiset keskeiset tuntomerkit, joiden perusteella tekniikan valinta kannattaa tehdä. Useiden tuntomerkkien toteutuessa on syytä muistaa, että menetelmiä voidaan käyttää yhdessä hybridijärjestelmiä muodostettaessa.

Käytettävän tekniikan valinnassa tärkeimpiä huomioitavia seikkoja ovat seuraavat:

- Ongelmakohteesta saatavan tietämyksen muoto (formaatti) ja laatu tai onko tietämystä ylipäättään olemassa.
- Onko kyseessä ohjaus- vai säätöjärjestelmä, käyttäjän tukijärjestelmä, diagnostiikkajärjestelmä, luokittelujärjestelmä vai joku muu sovellus.
- Rakennettavan sovelluksen ympäristön rajoitukset.
- Onko kyseessä vahvasti räätälöitävä vai yleiskäyttöinen sovellus.

Tietämyksen formaatti voidaan pukea kahteen päämuotoon: asiantuntijatietämykseen ja dataan perustuvaan tietämykseen. Asiantuntijatietämyksellä tarkoitetaan henkilön kykyä hahmottaa ja mallintaa ongelma-aluetta ja siihen soveltuvaa ratkaisua. Asiantuntijatietämys tukee selvimmin sumean logiikan käyttöä ja data neuroverkkojen käyttöä. Käytännön tilanteessa tietämyslähteitä ei voi yleensä esittää näin yksinkertaisesti. Varsinkin jos sekä dataa että asiantuntijatietämystä on runsaasti saatavilla, ei tekniikkaa voi välttämättä yksiselitteisesti määrätä. Viime aikoina neuroverkoissa käytettyjä menetelmiä on siirretty helpottamaan ja nopeuttamaan sumean logiikan kehitystä. Tämä on johtanut menetelmien lähentymiseen. Samalla käytettävän datan merkitys sumean logiikan mallin kehityksessä on kasvanut. Neuroverkkojen, sumean logiikan ja geneettisten algoritmien yhteiskäyttö sovellusten kehityksessä on maailmalla laajan kiinnostuksen ja kehityksen kohteena. Tällä hetkellä yksikään menetelmistä ei ole saavuttanut kiistatonta suosiota.

Toteutettavan järjestelmän tyypin valinnassa on tärkeänä seikkana huomioitava se, onko päättelyn oltava jäljitettävissä. Tyypillisenä ulospäinnäkyvänä sumean päättelyn ja neuroverkon erona on mainittu neuroverkon olevan ns. musta laatikko kun taas sumean päättely on vaiheittainen. Jäljitettävyys on usein tärkeää diagnostiikka- ja tukijärjestelmissä, joissa muutoinkin käytetään asiantuntijatietämystä runsaasti. Esimerkiksi luokittelussa välituloksia ei sen sijaan juuri tarvita, minkä vuoksi

neuroverkot ovat erittäin käyttökelpoisia. Neuroverkon jäljitettävyys on toistaiseksi tutkimuksen alkuvaiheessa ja nykyiset työkalut eivät tarjoa siihen ratkaisuja.

Rakennettavan sovelluksen ympäristön rajoituksia ovat kohdejärjestelmä ja reaaliaikavaatimukset. Yleisiä vaatimuksia ovat mm. tuotettavan koodin sallittu koko ja yhteensopivuus laitteen eri versioihin. Mikäli reaaliaikavaatimuksia on, eli kyseessä ei ole kohdejärjestelmästä irrallinen järjestelmä tai algoritmi, täytyy varmistua, että toteutettava algoritmi pysyy vaatimusten rajoissa. Tämä yleensä estää sellaisten järjestelmien käyttämisen, joiden suoritusaikaa ei pystytä ennustamaan. Mm. tämän vuoksi geneettisten algoritmien käyttäminen reaaliaikajärjestelmissä on vaikeaa. Sumean logiikan ja neuroverkkojen välillä eroja ei juuri ilmene: kumpikin on nopea suorittaa ja suoritusnopeus ei vaihtele. Vasteajat voivat tosin vaihdella suuresti opetettaessa neuroverkkoja off-line-tyyppisesti tai adaptiivisesti on-line-järjestelmässä. Tällöin eräissä järjestelmissä on mahdollista käyttää erillisiä neurolaskentaan suunniteltuja prosessorikortteja vasteaikojen pienentämiseksi.

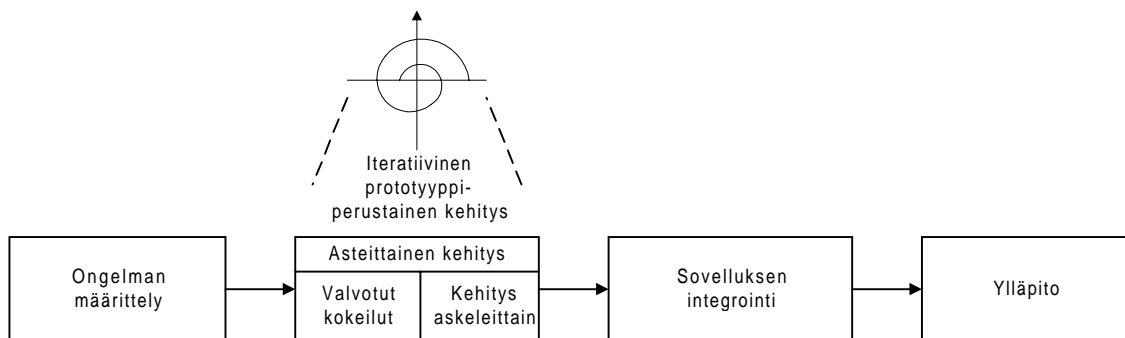
Räätälöinnillä voidaan saavuttaa yleiskäyttöisiä algoritmeja huomattavasti parempia tuloksia, mutta tällöin kehitys- ja ylläpitokustannukset kasvavat. Kokemusten mukaan sumealla logiikalla voidaan kehittää hyvin nopeasti yleiskäyttöisiä järjestelmiä, jotka toimivat ilman suurta räätälöintiä, mutta joiden tarkkuus ei ole välttämättä hyvä. Neuroverkko voidaan räätälöidä tuottamaan hyvä tulos, mutta tarvittava räätälöinnin määrä ja odotettavissa olevat tulokset ovat usein vaikeita arvioida.

3. Kehitysprosessin malli

3.1 Yleistä

Sumean logiikan ja neuroverkkosovellusten kehittäminen on mahdollista ilman kehitysmallia, mutta sen käyttäminen helpottaa projektin läpivientä. Mallin avulla voidaan nopeuttaa projektin toteuttamista ja saavuttaa usein parempi lopputulos. Kehitysprosessi toimii kokeneellekin suunnittelijalle muistilistana ja helpottaa töiden jakamista eri henkilöille tai alihankittaviksi. Valmiin standardin puuttuessa voidaan soveltaa yleisiä ohjelmistotekniikan menetelmiä. Näin on tehty esim. Isomursun väitöskirjassa [Isomursu 1995], joka käsittelee sumean logiikan sovelluksia.

Saksassa on vuonna 1996 perustettu komitea (VDI/VDE-GMA UA451) standardoimaan sumean logiikan käyttöä ja menetelmiä. Komitean vetäjänä on toiminut Constantin von Altrock, joka edustaa fuzzyTechin valmistajaa Informia (katso kappale 4.1). Tehtävä havaittiin yllättävän suureksi ja se on siirretty IEEE-organisaatiolle.



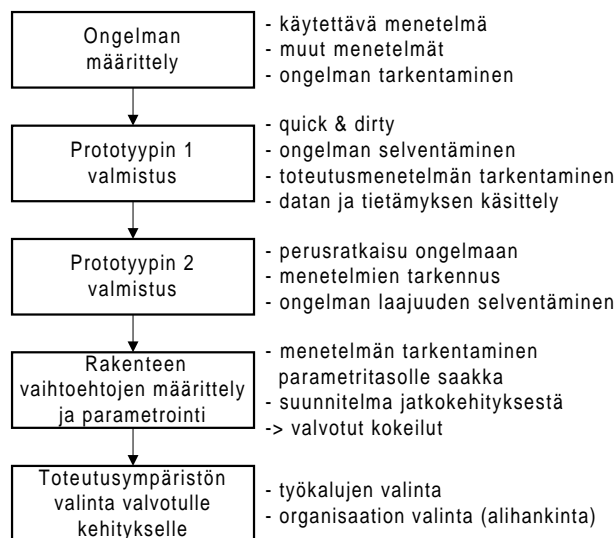
Kuva 1. Kehitysprosessin malli.

Kehitysprosessin tarkoituksena on jakaa projekti osiin ja tehtäviin, jotta suoritetuille tehtäville saadaan selvät välivaiheet ja projektin looginen eteneminen selkiytyy. Isomursun mukaan kehitysmalli voidaan ylimmällä tasolla jakaa kuvan 1 mukaisesti. Siinä ongelman määrittely, kehitys, integrointi ja ylläpito seuraavat toisiaan.

Kuvassa 1 esitettyjen lohkojen perusteella voitaisiin muodostaa kehitysprosessista perinteinen vesiputousmalli, jota käytetään yleisesti kuvaamaan vaiheittain etenevää työskentelyä. Siinä jokaisen lohkon lopuksi tehdään verifiointi ja validointi ja sen jälkeen siirrytään seuraavaan tehtävään [Hellendoorn 1993]. Myöhemmin havaitut ongelmat ongelman määrittelyssä ovat hankalia korjata käytettäessä tällaista lähestymistapaa. Isomursu ehdottaa kehitysprosessin malliksi lähestymistapaa, jossa iteratiivisesti valmistettavien prototyyppien kautta päästään haluttuun tulokseen. Edellisiin vaiheisiin voidaan palata prototyypin esille tuomien tarpeiden perusteella.

3.2 Ongelman määrittely

Ongelman määrittelyssä pyritään kartoittamaan varsinainen ongelma ja sen ratkaisemiseen mahdollisesti sopiva menetelmä. Samalla pohditaan, olisiko olemassa muita käyttökelpoisia menetelmiä. Asiaa käsitellään lyhyesti luvussa kaksi. Näiden päätösten jälkeen voidaan aloittaa varsinainen työskentely tarkentamalla ongelma- aluetta ja määrittelemällä tulevan toiminnan rajoja. Isomursu on jakanut suoritettavat tehtävät kuvan 2 mukaan. Siinä tarkennetaan määrittelyiden ja prototypoinnin avulla ratkaistavaa ongelmaa ja etsitään siihen mahdollisesti sopivaa ratkaisua.



Kuva 2. Ongelman määrittelyn vaiheet kehitysprosessissa.

Prototyyppien tarkoituksena on selventää ongelmaa ja hankkia tietämystä mahdollisesti käytettävistä menetelmistä. On mahdollista, että vasta tässä vaiheessa kaikille osapuolille selviää, millaista järjestelmää ollaan rakentamassa ja mitä sen rakentaminen vaatii. Ensimmäisen prototyypin peruste on sen nopeassa valmistamisessa ja siitä saatavan informaation suuressa määrässä. Ensimmäistä prototyyppiä voidaan käyttää muodostettaessa oikeanlaista käsitystä kehitettävän järjestelmän vaatimuksista ja toiminnasta. Sen valmistuksen aikana asiat tulevat kertaalleen mietityksi ja testatuksi käytännön tasolla.

Prototyypin kehittämisen aikana joudutaan ensimmäistä kertaa käyttämään sovellusalueesta saatavilla olevaa asiantuntijatietämystä sekä kerättyä mittausdataa. Ne on puettava muotoon, joka soveltuu parhaiten prototyypin valmistukseen. Kerättyä mittausdataa joudutaan käsittelemään, jotta se saataisiin työkalulle sopivaan muotoon. Neuroverkkoja kehitettäessä datasta muodostetaan opetus- ja testiaineistoa ja sumeasta logiikasta lähinnä testiaineistoa. Käytettäessä sumean logiikan generointimenetelmiä joudutaan muodostamaan myös opetusaineisto. Datat esikäsittelyssä pohditaan usein

mm. suhtautumista puuttuviin mittauksiin ja häiriöiden poistamiseen. Datan oikeellisuuden määrittämiseksi on syytä käyttää sovellusalueen asiantuntijaa, joka ohjaa datojen valinnassa ja poistossa. Tässä vaiheessa apuna voidaan käyttää mm. työkalujen luokittelumenetelmiä ristiriitaisuuksien poistamiseen.

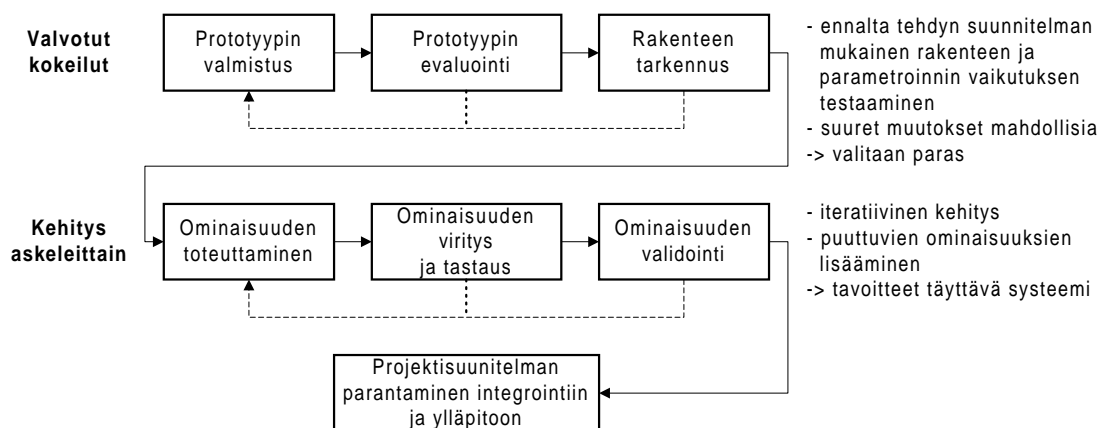
Toisen prototyypin tarkoitus on toimia eräänlaisena perusratkaisuna, jolle lopullista järjestelmää ryhdytään rakentamaan. Sitä kehitettäessä selviävät usein ratkaistavan ongelman laajuus ja siihen sopivat ratkaisumenetelmät.

Järjestelmän rakenteelle ja parametroinnille on lukuisia vaihtoehtoja. Niiden kirjaaminen myöhempiä tarkastelua varten nopeuttaa ja tarkentaa kehitystä. Sopivan työkalun ja kehitysympäristön valinnalla voidaan oleellisesti vaikuttaa projektin seuraaviin vaiheisiin.

Ongelman määrittelyn jälkeen on mahdollista nähdä, ovatko käytetyt menetelmät, esimerkiksi sumea logiikka tai neuroverkko, käyttökelpoisia ratkaisumenetelmiä.

3.3 Asteittainen kehitys

Asteittaisen kehityksen Isomursu jakaa valvottuihin kokeiluihin ja kehitykseen askeleittain, kuva 3.



Kuva 3. Asteittainen kehitys sumean logiikan ja neuroverkkojen kehitysprosessissa.

Valvotuissa kokeiluissa tehdään pikaisia prototyypikokeita, joissa systemaattisesti käydään läpi aiemmin annettuja vaihtoehtoisia määrittelyjä ja erilaisia parametrintivaihtoehtoja. Prototyypien avulla voidaan kokeilla erilaisten määrittelyjen ja parametrien vaikutusta lopputulokseen. Vaihtoehtojen kokeileminen on hyvin edullista ja nopeaa tässä vaiheessa, jos valittu työympäristö tukee konfigurointia ja

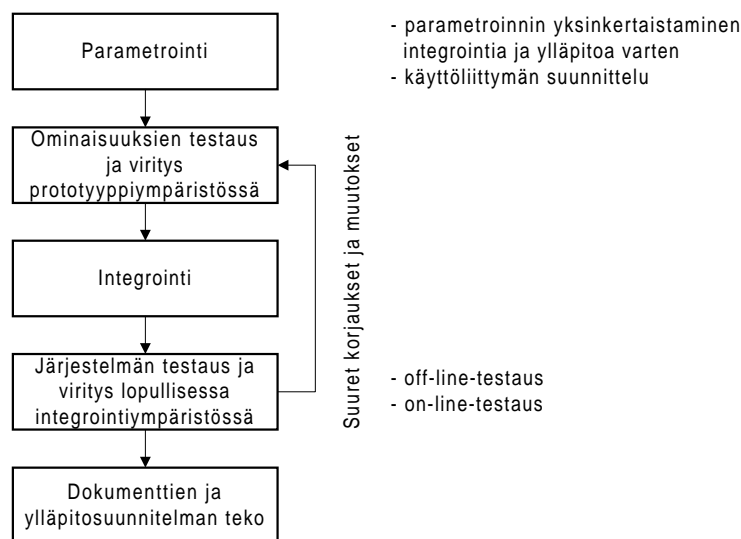
testausta. Yleensä rakenteen suuri muuttaminen integrointivaiheessa vaatii, että toteutuksessa palataan taaksepäin.

Prototyypikokeista hyväksytään paras, jota ryhdytään askeleittain edelleen parantamaan. Usein tähän mennessä on havaittu, että järjestelmästä puuttuu joitain ehdottomasti mukana oltava ominaisuuksia. Näiden ominaisuuksien kehittäminen tapahtuu asteittain lisäämällä, testaamalla, virittämällä ja validoimalla saatuja tuloksia.

Asteittaisen kehityksen jälkeen käytettävissä tulisi olla hyvin toimiva järjestelmä, joka täyttää sille asetetut tavoitteet. Vaiheen päätyttyä on syytä päivittää aiemmin tehtyjä suunnitelmia integrointia ja ylläpitoa varten. Lisäksi tulee dokumentoida kokemukset ja tulokset, joita vaiheen aikana on saatu.

3.4 Sovelluksen integrointi

Sovelluksen integroinnin tarkoitus on tuotteistaa prototyyppi lopulliseen toimintaympäristöön. Sovelluksen toiminta on tarkastettu ja hyväksytty ennen tätä vaihetta. Kuvassa 4 on esitetty vaiheet, joihin sovelluksen integrointi voidaan jakaa.



Kuva 4. Sovelluksen integroinnin työvaiheet kehitysprosessissa.

Integroinnin ensimmäisenä vaiheena on järjestelmän parametointi ja sen hallinta. Ylläpito- ja käyttöhenkilöstöä ei ole useinkaan koulutettu alan asiantuntijoiksi ja siksi heidän työtään on pyrittävä helpottamaan niin paljon kuin mahdollista. Järjestelmästä voidaan piilottaa ylimääräisiä parametreja tai tehdä muita vastaavia helpotuksia. Neuroverkkosovelluksissa parametrit sulautuvat yleensä täysin järjestelmään joitain pieniä poikkeuksia lukuun ottamatta. Samalla joudutaan miettimään mahdollista

käyttöliittymää. Parametroinnissa tehtyjen muutosten jälkeen on syytä uusia testaus ennen lopullista integrointia.

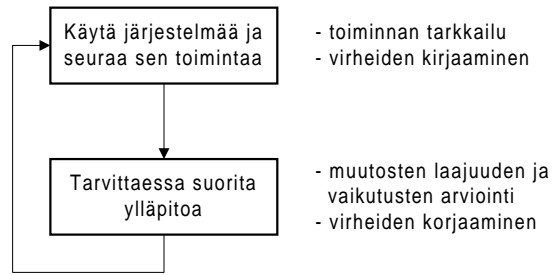
Integroinnin tarkoituksena on siirtää kehitetty järjestelmä lopulliseen kohdeympäristöön. Yleensä tämä tarkoittaa parametrien ja laskentakoodin siirtämistä pois testaus- ja kehitysympäristöstä toiseen ympäristöön, joka asettaa omat rajoituksensa ja vaatimuksensa järjestelmän toimivuudelle. Usein kehitys ja testaus suoritetaan PC:llä Windows-ympäristössä ja lopullinen toteutus integroidaan johonkin automaatiojärjestelmään. Lähes kaikki työkalut tukevat jossain muodossa C-kielen tai vastaavan generointia tai valmiiden C-kielisten kirjastojen käyttöä. Tämä mahdollistaa sovelluksen siirron suoraan kohdejärjestelmään pienin muutoksin. On kuitenkin huomattava, että on automaatiojärjestelmiä, joihin laskennan siirtäminen on käytännössä mahdotonta.

Helpoimmassa tapauksessa oikein valitulla työkalulla voidaan sovelluksen integrointi suorittaa automaattisesti käyttämällä esim. tuotettua C-kieltä. Joissain sumean logiikan työkaluissa on liitynyt yleisimpiin logiikoihin ja yksikkösäätimiin, jolloin niiden käyttämisestä voi olla apua etenkin lopullisessa on-line-testauksessa. Suuret automaatiotoimittajat ovat valmistaneet omia työkaluja järjestelmiinsä, joiden käyttämistä integrointivaiheessa kannattaa harkita. Mahdollisista ratkaisuksista kannattaa tiedustella automaatiotoimittajilta. Optimitapauksessa työkalu, jolla järjestelmän prototyyppi on suoritettu, tukee myös lopullista integrointivaihetta ja on-line-testausta.

Integroidun järjestelmän testaaminen on-line-tilassa normaalissa ja testiajossa kertoo sen lopullisen suorituskyvyn. Varsinkin säätösovellusten hyvyyden määrittely pelkän simuloinnin avulla voi olla vaikeaa. Saatujen tulosten perusteella voidaan arvoida kehitetyn projektin onnistuneisuus tavoitteisiin nähden.

3.5 Ylläpito

Viimeisenä vaiheena sovelluksen kehityksessä on ylläpito. Tämän tehtävän suoritus jää yleensä sovellusta käyttävän organisaation harteille, joten sille on luotava parhaat edellytykset toimia. Perusedellytyksenä ylläpitohenkilöiden toiminnalle on riittävän dokumentoinnin laatiminen ylläpidettävästä järjestelmästä. Lisäksi tarvitaan ylläpitäjien ja usein myös loppukäyttäjien koulutusta. Isomursu on jakanut suoritettavat tehtävät järjestelmän toiminnan seuraamiseen ja varsinaiseen ylläpitoon, kuva 5.



Kuva 5. Ylläpidon tehtävät kehitysprosessissa.

Ylläpitohenkilöstö tarvitsee kattavan dokumentoinnin käytettävästä sovelluksesta. Dokumentointiin tulee kuulua kuvaukset kaikista ohjelmassa käytetyistä parametreista ja menetelmistä. Tarvittavien muutosten ja lisäysten tekeminen helpottuu, jos niiden toteutusta on mietitty etukäteen.

Jotta kehitetyn järjestelmän toiminnasta voidaan tehdä varmoja päätelmiä, sen on oltava käytössä käyttökokemusten saamiseksi. Havaituista ongelmista on syytä pitää kirjaa ongelmien syiden jäljittämiseksi. Niiden perusteella päätetään mahdolliset jatkotoimet, jotka yleensä jakautuvat korjauksiin ja uusien puuttuvien piirteiden lisäämiseen. Suurten epäkohtien tapauksessa voidaan tarvittaessa valmistella laajemmat muutostyöt tai suunnitella joitain osia järjestelmästä uudelleen.

4. Työkalut

Kaupallisesti saatavien sumean logiikan ja neuroverkkotyökalujen määrä on kasvanut huomattavasti viimeisen viiden vuoden aikana. Osa ohjelmistoista on keskittynyt ominaisuuksiltaan tiettyihin teoreettisiin menetelmiin, ja ne ovat lähtöisin tutkijoiden kehittämistä ohjelmista. Suurin osa laajemmista ja yleiskäyttöisistä työkaluista on kuitenkin konsulttitalojen tai ohjelmistoyritysten tekemiä. Onkin hyvin yleistä, että työkalun valmistanut ohjelmistotalo tekee myös omia sovellusprojekteja ja käyttää itse kehittämiänsä työkaluja esim. konsultoinnin tukena. Tällainen tilanne on myös tähän raporttiin valittujen työkalujen kohdalla.

Tähän raporttiin valittiin erilaisia työkaluja neuroverkkojen ja sumean logiikan sovellusten kehittämistä varten. Niillä on mahdollista suorittaa luvussa 3 esitetyn kehitysprosessin kolme ensimmäistä vaihetta varsin kattavasti. Valitut työkalut ja niiden valmistajat ovat seuraavat:

- Sumean logiikan perustyökalut
 - Inform GmbH: fuzzyTECH
 - Ortech (ennen Togai Infra Logic): TILShell
- Neuroverkkojen perustyökalut
 - NeuralWare, Inc.: NeuralWorks
 - California Scientific Software: BrainMaker
- Muita työkaluja sumean logiikan ja neuroverkkosovellusten toteuttamiseksi
 - MathWorks, Inc.: Matlab + Fuzzy Logic Toolbox
 - MathWorks, Inc.: Matlab + Neural Network Toolbox
 - Management Intelligenter Technologien GmbH: DataEngine
 - Gensym Corp.: G2 + NeurOn-Line

Liitteissä A ja B on raportissa esiintyvien työkalujen yhteystietoja Suomessa ja hintatiedot raportin ilmestymishetkellä.

Selvityksessä mukana olevat työkalut ovat verrattain erityyppisiä, osa niistä toimii itsenäisesti ja on lähes suljettu käyttäjän muutoksilta ja lisäyksiltä. Tällaisia työkaluja ovat mm. fuzzyTech ja BrainMaker. Toinen ääripää antaa täydet mahdollisuudet ympäristön muuttamiselle ja määrittelemiselle ja omien funktioiden lisäämiselle, kuten Matlab ja G2.

Saatavilla on paljon muitakin ohjelmistoja, mutta valitut työkalut antavat varsin kattavan läpileikkauksen nykyisestä tarjonnasta. Ajan tasalla olevan luettelon laatiminen on mahdotonta ja toisaalta tarpeetonta jatkuvasti tulevien uusien työkaluversioiden ja kokonaan uusien ohjelmistojen vuoksi. Perustellusti voidaan sanoa, että ohjelmistotiedot

vanhenevat joiltain osin hyvin pian. Liitteisiin C ja D on koottu useiden sumean logiikan (C) ja neuroverkkojen (D) työkaluvalmistajien yhteystietoja.

Lähes kaikista työkaluista on tarjolla PC:n Windows-ympäristöön soveltuva versio, joka toimii myös Windows 95- ja NT-ympäristöissä. Joistain ohjelmista, lähinnä neuroverkkotyökaluista, on tarjolla myös työasemille tehtyjä versioita. Työasemaversioiden keskittyminen neuroverkkotyökaluihin johtuu lähinnä niiden suuremmasta laskentakapasiteetin tarpeesta opetusvaiheessa verrattuna sumean logiikan laskentaan.

Sumean logiikan työkalut muistuttavat erehdyttävän paljon toisiaan, koska sovellusten kehittäjien tarpeet ovat perusominaisuuksien osalta kohtuullisen samanlaisia. Neuroverkkotyökalut eroavat enemmän toisistaan jo laajemman menetelmäkirjonta vuoksi. Haettaessa ohjelmien välisiä eroavaisuuksia niitä löytyy lähinnä ns. apu-toimintojen määrästä ja laadusta. Yleisesti ohjelmien heikkoudet näkyvät kehitysprojektin alku- ja loppuvaiheessa tarvituista ominaisuuksista. Näitä ominaisuuksia ovat etenkin datan esikäsittely ja lopullisen sovelluksen testaaminen, analysointi ja integrointi. Datan käsittelyssä puuttuvia ominaisuuksia ovat mm. suodatukset, puuttuvien mittausten hallinta, mittausten välisten korrelaatioiden esittäminen ja etenkin mittausten välisten viiveiden huomioiminen. Integrointivaiheessa eroja on sekä off-line- ja on-line-testaus- että simulointimahdollisuuksissa. Neuroverkkosovelluksia rakennettaessa opetusalgoritmien lisääminen kehitettyyn järjestelmään vaihtelee tutkituissa ohjelmissa varsin paljon. Tätä ominaisuutta tarvitaan adaptiivisten sovellusten integroimisessa järjestelmään.

Tutkitut ohjelmat vaihtelevat suuresti käyttöliittymiltään ja käytettävyydeltään. Käyttäjät ovat viime aikoina tottuneet Windows-ympäristön kaltaiseen graafiseen ympäristöön, jossa ohjelmien ohjaus tapahtuu havainnollisesti valikoiden kautta. Tähän kuuluu yleensä runsas kuvaajien käyttö. Toimintojen ryhmittely ohjaa käyttäjää tarvittavien tehtävien suorittamisessa. Väärin tehdyistä valinnoistakaan ei aiheudu usein suurta vahinkoa. Tutkitut perustyökalut ovat valinneet keskeiseksi ympäristöksi Windowsin ja niiden käyttö voidaan opetella jopa yritys ja erehdys -periaatteella. Toisaalta ohjelmien eduksi voidaan laskea se, että ne ovat keskittyneet yhteen menetelmään ja sen kunnolliseen hoitamiseen. Yleiskäyttöiset työkalut, kuten Matlab, DataEngine ja G2, ovat ominaisuuksiltaan ja menetelmiltään laajoja ja monipuolisia, mutta tästä voi aiheutua sekaannusta ja yksinkertaisten toimintojen suorittaminen voi vaikeutua turhaan. Koska nykyään tietokoneiden levykapasiteetin ei pitäisi rajoittaa ohjelmien kokoa, niin soisi jokaisessa työkalussa olevan on-line help -tiedosto. Akuutin ongelman kohdatessa sen käyttö on huomattavasti nopeampaa kuin perinteisen paperiversion selaaminen. Paperiohjeen tulisikin olla enemmän ylemmän tason opas kuin yksityiskohtainen selvitys jonkin napin tehtävästä. Tässä NeuralWorks on onnistunut mainiosti.

4.1 Sumean logiikan perustyökalut: fuzzyTECH ja TILShell

Katsaukseen valittiin kaksi keskeisintä sumean logiikan työkalua, jotka ovat levinneet laajasti sumean logiikan soveltajille. Nämä ovat Inform GmbH:n fuzzyTech ja Ortechin TILShell. Molemmat ovat Windows-ohjelmia. Ne on tehty ainoastaan sumean logiikan kehitystä varten. Ohjelmistot ovat kehittyneet monipuolisiksi ja erittäin käyttäjäystävällisiksi työkaluiksi. Onkin havaittavissa, että ohjelmistojen valmistajat seuraavat tarkasti toistensa tekemiä muutoksia ja parannuksia. Yhteen työkaluun ilmestyneet hyvät ominaisuudet ilmestyvät pienellä viiveellä useiden muidenkin valmistajien uusiin versioihin. Osa muutoksista on tietenkin asiakkaiden ehdottamia ja kuuluu normaaliin kehitykseen.

Kumpikin työkalu vaatii perustietämystä sumeasta logiikasta, mutta kynnys ohjelmien käyttöönottoon on pieni. Työkalujen ohjekirjat antavat käyttäjälle kuvan sumean logiikan toiminnasta, vaikka eivät olekaan varsinaisia oppikirjoja. Oppikirjaksi on tarjolla parempiakin kirjoja alan asiantuntijoilta, kuten Koskolta [1992], Zimmermannilta [1991] ja Driankov et al:lta [1993].

FuzzyTech-työkalun suosio on kasvanut suuresti parin viime vuoden aikana. Tämä johtuu hyvin toimivasta kokonaisuudesta sekä kattavasta tuesta useiden laitevalmistajien järjestelmiin. Ohjelmasta on tarjolla versioita eri laitetoimittajien järjestelmiin kuten ohjelmitaviin logiikoihin, yksikkösäätimiin ja prosessoreihin. Pääosin nämä versiot eroavat tuotetun koodin (C tai assembler) ja on-line-testausominaisuuksien suhteen. Vastaavat perusominaisuudet löytyvät myös Togai InfraLogicin kehittämästä TILShell-työkalusta. Se oli yksi ensimmäisistä graafiseen käyttöliittymään perustuvista sumean logiikan kehitysohjelmista.

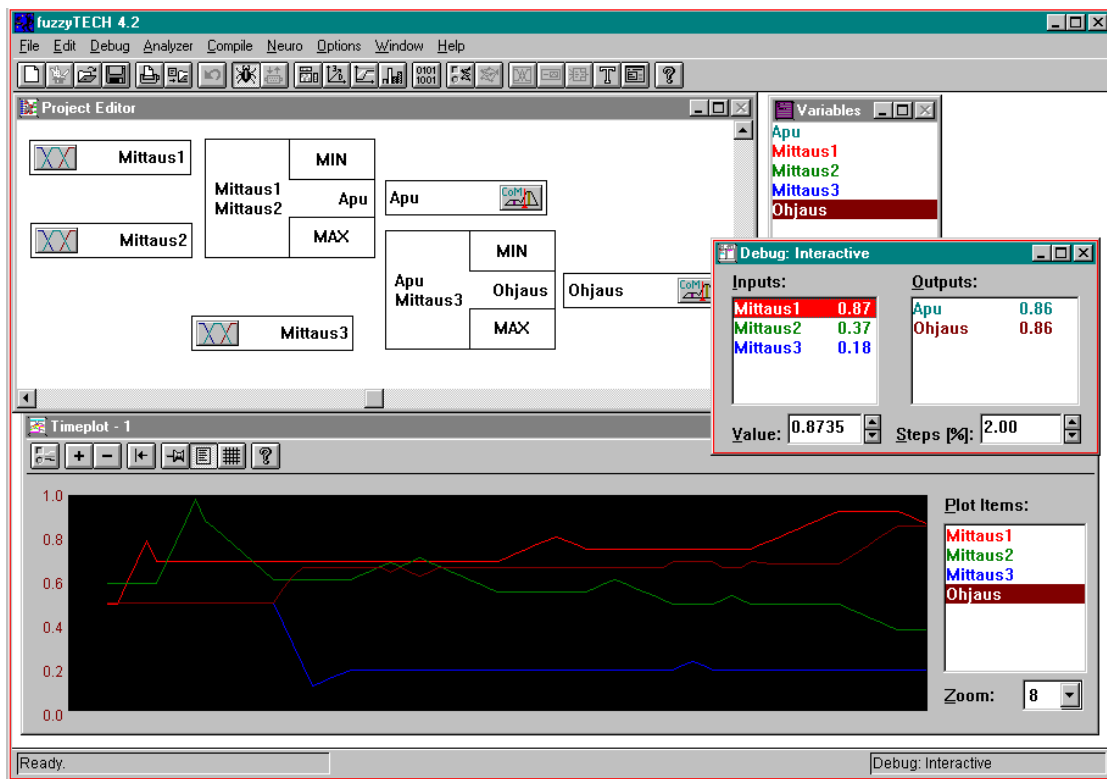
Datan esikäsittely

Vaikka sumean logiikan sovelluksen kehittäminen ei perinteisesti kuulu datapohjaisiin menetelmiin, joudutaan dataa käyttämään joissakin kehityksen vaiheissa. Valittaessa ja analysoitaessa muuttujia tai testattaessa sovelluksen toimintaa datan käyttö kuuluu oleellisena osana kehitykseen.

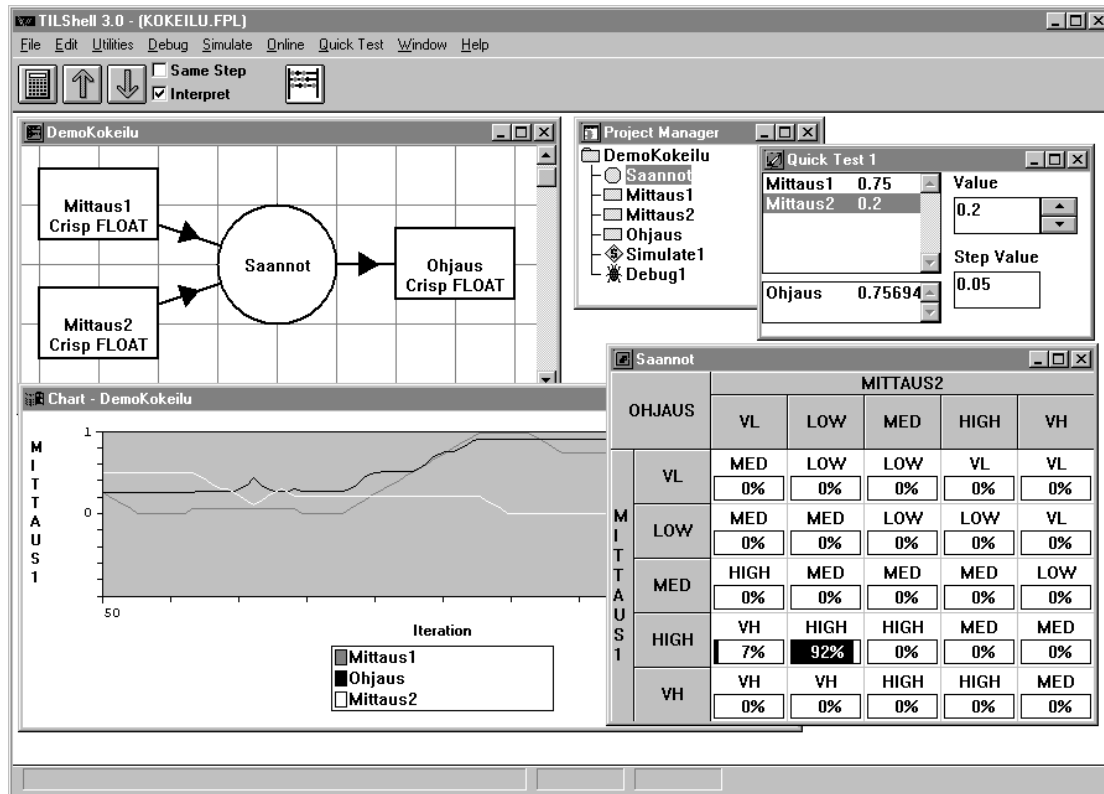
fuzzyTechin perusominaisuudet ovat varsin laajat, mutta niihin ei sisälly datan esikäsittelytoimintoja. Nämä ominaisuudet on saatavissa erillisenä DataAnalyzer-moduulina (DIAdem), joka on GfS Corporationin valmistama erillinen ohjelma. Se on mahdollista liittää fuzzyTechin rinnalle tai käyttää itsenäisesti omana ohjelmana. DIAdemin avulla mahdollistetaan useiden tiedostoformaattien käyttö ja voidaan suorittaa signaalin käsittelyä, data-analyysiä ja luokittelua. TILShell ei sisällä valmiita toimintoja mittausten esikäsittelyyn. Haluttuja toimintoja on tosin mahdollista tehdä FPL-kielellä muodostettuihin simulointilohkoihin.

Prototyypin kehitys

Kaikki fuzzyTechissä ja TILShellissä tapahtuva kehitys suoritetaan graafisen käyttöliittymän avulla. Järjestelmän rakentaminen aloitetaan määrittelemällä heräte- ja vastemuttajat ja niiden määrittelyt järjestelmän käyttöön. Sen jälkeen näiden välille luodaan lohkoilla rakenne halutun heräte-vaste-kuvauksen muodostamiseksi. Järjestelmä voidaan rakentaa hierarkkiseksi käyttämällä useita itsenäisiä sääntökantoja. Rakennettu sovellus tallennetaan fuzzyTechissä FTL-tiedostoon (Fuzzy Technology Language) ja TILShellissä FPL-tiedostoon (Fuzzy Programming Language). FPL-tiedostoon voidaan tehdä muutoksia ja lisäyksiä käyttämällä FPL-kieltä. Molemmat tiedostot ovat ASCII-pohjaisia ja pohjautuvat omiin rakenteisiinsa.



Kuva 6a. fuzzyTechin graafinen käyttöliittymä.



Kuva 6b. TILShellin graafinen käyttöliittymä.

Kuvissa 6a ja 6b ovat fuzzyTechin ja TILShellin käyttöliittymät ja niistä voidaan helposti huomata samankaltaisuuksia. Kuvassa 6a on esitetty fuzzyTechin esimerkkijärjestelmä, jossa on käytössä kolme herätettä, kaksi sääntökantaa ja yksi vaste. Vastaavasti kuvassa 6b on TILShellin esimerkkijärjestelmä, jossa on kaksi herätettä ja yksi vaste. Lisäksi siinä on esitetty sääntömatriisi-ikkuna, josta voidaan seurata testauksen aikana sääntöjen käyttäytymistä.

Työkaluissa on ikkunat muuttujien määrittelemiseksi ja sääntöjen kirjoittamiseksi. Sääntöjä voidaan lisätä, poistaa ja muuttaa useiden erityyppisten ikkunoiden, kuten taulukkoikkunan, tekstipohjaisen ikkunan ja matriisi-ikkunan, avulla. Ohjelmissa on käytettävissä useita erityyppisiä jäsenyysfunktioita, täsmällistämismenetelmiä ja päättelymenetelmiä.

fuzzyTech sisältää ns. wizard-ikkunan, jonka avulla käyttäjä voi luoda nopeasti ensimmäisen sumean järjestelmän. Wizardin antamat valitumahdollisuudet ovat rajalliset, mutta pikaisesti luotua versiota on mahdollista muuttaa ja optimoida wizardin käytön jälkeen. Jos mittausdataa on saatavissa, wizard kykenee analysoimaan siitä muutamia perusominaisuuksia, kuten muuttujien vaihteluvälit.

Prototyypin testaus ja viritys

Molemmissa työkaluissa voidaan kehityksen yhteydessä suorittaa tavanomaiset staattiset pikatestaukset kuten 3D-kuvaaja ja herätteiden syöttäminen käsin. 3D-kuvaajassa kahden herätteen ja yhden vasteen välille piirretään 3D-pintakuva. Silloin muut herätteet ovat vakioita. Prototyypin testatessa voidaan edellä mainittujen ikkunoiden lisäksi käyttää mm. heräte- ja vastemuuttujien kuvaajia, joiden määrittelyt on esitetty yhdessä reaaliaikaisen arvonn kanssa.

Käytettäessä mittausdataa tiedostosta tai simuloitaessa ulkopuolisen ohjelman avulla käyttäen DDE-linkkiä (dynamic data exchange - toiminta, jonka avulla kaksi Windows-ohjelmaa voivat siirtää tietoa keskenään), ikkunoista voidaan seurata reaaliaikaisesti muuttujien käyttäytymistä. TILShellin on lisäksi mahdollista tehdä omia simulointilohkoja kirjoittamalla simulointikoodi suoraan FPL-ohjelmointikielellä (FPL Script Language). Simulointilohko on mahdollista toteuttaa myös Windowsin DLL-kirjasto (dynamic link library - ohjelma kirjasto, jonka avulla voidaan lisätä perusohjelman toimintoja). TILShellin simulointiin voidaan yhdistää halutunlainen visualisointi DLL-kirjastoja ohjelmoimalla.

Molemmissa ohjelmissa voidaan testauksen aikana tehdä muutoksia ja lisäyksiä muuttujiin ja sääntöihin. Ne otetaan automaattisesti käyttöön seuraavalla laskentakierroksella. Kunkin säännön laukeamisaste ja painokerroin voidaan nähdä reaaliaikaisesti näytöltä. Lisäksi nähtävillä on sääntökannan jokaisen säännön tilastotietoja.

Simuloinnin keskeinen työkalu on muuttujien seuraaminen trendi-ikkunoiden avulla. Molemmassa työkalussa on mahdollista valita muuttujat trendi-ikkunaan piirtoa varten. TILShellissä voidaan lisäksi valita muuttujat, joiden arvot tallennetaan levyllä testauksen ajalta tai halutulta tapahtumajaksolta trigger-funktion avulla.

On-line-simulointi on mahdollista rajoitetusti molemmilla ohjelmilla. Tästä enemmän seuraavassa kohdassa.

Integrointi kohdejärjestelmään

fuzzyTechin ja TILShellin perusominaisuuksiin kuuluu tehdyn järjestelmän muuttaminen C-koodiksi sovelluksen integrointia varten. fuzzyTechistä on useita versioita, jotka soveltuvat joukolle yksikkösäätimiä, ohjelmoitavia logiikoita ja prosessoreita. Näihin laitteisiin tehtyä sovellusta voidaan seurata suoraan on-line-tilassa. Muutosten tekeminen on-line-tilassa muutamaan tuettuun laitteeseen onnistuu lisämoduulin avulla (RTRCD-moduuli). Muiden laitteiden käyttäytymistä voidaan vain

tarkkailla. TILShellissä nämä ominaisuudet ovat rajoittuneet ainoastaan muutamaankin erikoisympäristöön.

fuzzyTeehiin on mahdollista liittää PlusC-lisämoduuli, jonka avulla kehitetty systeemi voidaan suojata salaiseksi FTR-tiedostoksi. Se estää ulkopuolisia näkemästä ja muuttamasta toteutettua ratkaisua. Systeemiä voidaan käyttää levitettäessä sovellusta ajoaikaisen moduulin kanssa useisiin kohteisiin paljastamatta kehitettyä ratkaisua.

4.2 Neuroverkkojen perustyökalut: NeuralWorks ja BrainMaker

Vaikka neuroverkkomenetelmien ja -sovellusten historia on pitempi kuin sumean logiikan, eivät ohjelmistot ole viime vuosien laajasta kiinnostuksesta huolimatta vakiinnuttaneet kaikkia niiltä kaivattuja ominaisuuksia. Jatkuva kiinnostus ja kehitys onkin aiheuttanut kilpailua lähinnä menetelmien lisäämisessä ja useat tarpeelliset toiminnot, kuten datan esikäsittely ja lopullisen verkon analyysi, ovat jääneet työkaluissa pienemmälle huomiolle. Ohjelmien monipuolistuessa nämäkin ominaisuudet varmasti lisääntyvät ja työkaluista tehdään helpokäyttöisempiä.

Katsaukseen valituilla kahdella yleiskäyttöisellä neuroverkkotyökalulla on laaja levikki, mutta käyttäjäkunta on hivenen erilainen. NeuralWorks on eräs vanhimmista ja yleisimmistä lähinnä tutkijoiden parissa ja BrainMaker lähinnä taloussovelluksia kehittävien yritysten piirissä. Molemmista ohjelmista on tarjolla versiot Windows-ympäristöön. NeuralWorksista on lisäksi olemassa versiot yleisimpiin Unix-työasemiin.

Työkalujen ohjekirjat ovat NeuralWorksissa ja BrainMakerissa varsin laajat ja ne soveltuvat osittain jopa perusoppikirjoiksi. Varsinaisiksi oppikirjoiksi on kuitenkin tarjolla teoksia alan asiantuntijoilta, kuten Dayhoffilta [1990] ja Haykinilta [1994]. Laajempi lista kirjallisuudesta on Tekesin raportissa Neurolaskennan mahdollisuudet [Koikkalainen toim. 1994].

NeuralWorks on kehitysympäristö, joka tarjoaa laajat mahdollisuudet erilaisten menetelmien ja ratkaisujen kokeiluun ja mahdollistaa lisämoduulien avulla omien toimintojen ja menetelmien rajattoman lisäämisen. Sen käyttö vaatii jonkinasteista perehtymistä ohjelman rakenteeseen ja toimintaan. Ohjelman voi myös käynnistää ns. novice-moodiin, jolloin osa toiminnoista on piilotettu. NeuralWorksiä ei ole tehty alunperin Windows-ohjelmaksi vaan Unix-ympäristöön. BrainMaker puolestaan on tehty Windows-ympäristöön ja sen käyttö on helpompaa tottuneelle PC:n käyttäjälle. Helppous johtuu osittain pienemmästä toimintomäärästä. Sen menestys pohjautuu pääosin yksinkertaiseen käyttöliittymään, hienostelemattomiin ominaisuuksiin ja yleisimmän backpropagation-algoritmin käyttöön. BrainMakerista puuttuvat helposti saatavat on-line-manuaalit, mikä hidastaa ohjelmaan tutustumista.

BrainMakeriin on tarjolla useita neuroverkkolaskentaan kehitettyjä lisäkortteja, joiden avulla laskentaa voidaan nopeuttaa. Niiden merkitys korostuu suurta nopeutta vaativissa ja laajaan muuttujajoukkoon keskittyvissä sovelluksissa, esim. puheen tunnistuksessa.

Datan esikäsittely

Neuroverkkosovelluksia kehitettäessä datan esikäsittely on ehdoton edellytys hyvälle lopputulokselle. Se on syytä tehdä huolellisesti ja usein siihen kulutettu aika korvautuu nopeampana ja parempana opetuksena ja testauksena.

NeuralWorks ei tarjoa mahdollisuuksia datan esikäsittelyyn, vaan se on tehtävä jossain muussa ohjelmassa. Tähän tehtävään NeuralWare on kehittänyt erillisen DataSculpture-ohjelman. Sen avulla voidaan suorittaa datan visualisointia, suodatusta, korrelaatioita ja useita muita toimintoja. Työkalu kykenee lukemaan dataa useista tiedostoformaateista ja tallentamaan esikäsitellyn datan määrittäksineen NeuralWorks-ohjelmaa varten ASCII-tiedostoihin.

BrainMakerin mukana tulee oma versio NetMaker-ohjelmasta, jolla haluttu datan käsittely voidaan suorittaa. Sen avulla data esikäsitellään BrainMakerillä suoritettavaa opetusta varten. Kerätyistä mittauksista muodostetaan NetMakerin avulla neuroverkolle haluttu heräte- tai vaste-opetus- ja testijoukot. Ohjelma tukee datan tuontia useista eri tiedostoformaateista. Uusien muuttujien muodostaminen on mahdollista käyttämällä esikäsitteilytoimintoja tuotuihin mittauksiin. Käytössä on yksinkertaisia operaatioita kuten datan siirrot ylös- ja alaspäin (viiveen kompensointi), joitain laskennallisia operaatioita datojen välillä sekä suodatuksia. Ohjelmassa voidaan myös piirtää muuttujista kuvaajia ja laskea kahden muuttujan välisiä korrelaatioita. Ohjelma tallentaa määrittäykset opetus- ja testausdatan tiedostoihin BrainMaker-ohjelmalla suoritettavaa opetusta varten.

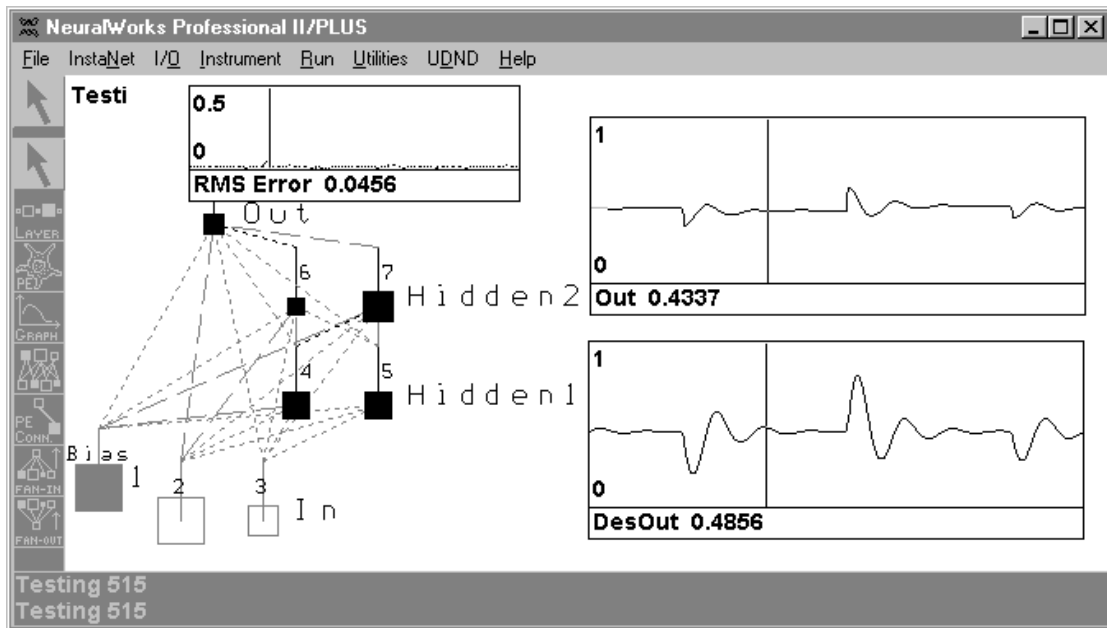
Prototyypin kehitys

NeuralWorks sisältää useita erilaisia menetelmiä neuroverkkojen opetukseen. Mukana on sekä valvotun opetuksen menetelmiä että itseoppivia verkkotyyppejä. BrainMakerin käyttämä opetusalgoritmi on tyypillinen backpropagation ja sen haittana on hitaus ja herkkyyys paikallisille minimeille. Lisäksi mukana on ns. hypersonic-opetus, joka on kehitetty lähinnä luokittelutehtäviin. Kummallakin ohjelmalla on opetuksessa mahdollista käyttää testijoukkoa verkon hyvyden määrittämiseen sekä opetuksen keskeyttämiseksi optimihetkellä.

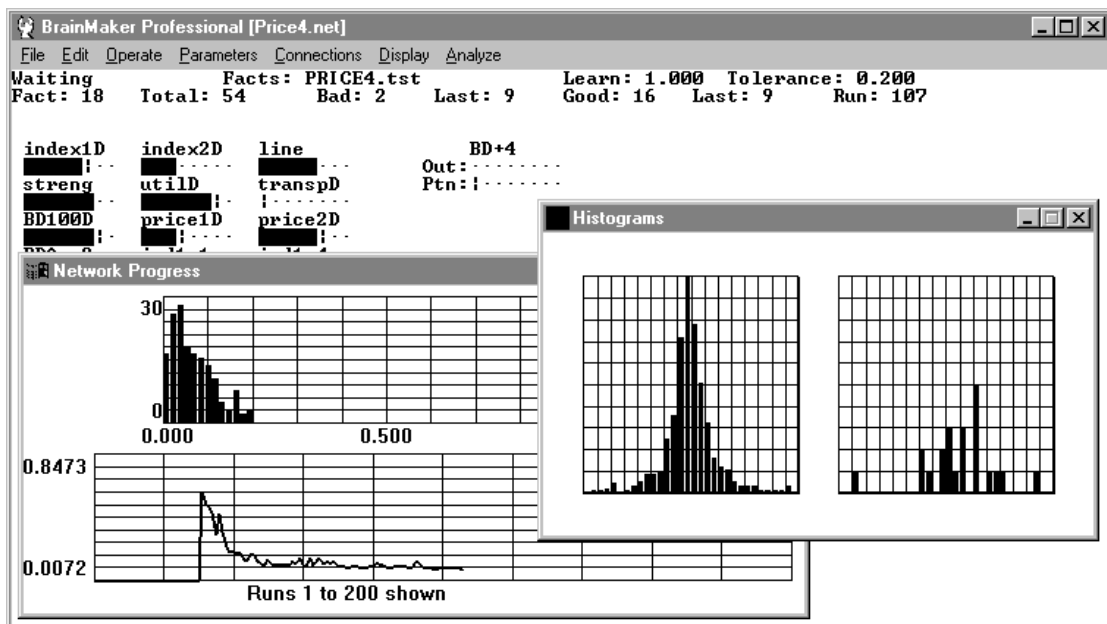
Menetelmät ja toiminnot valitaan valikkojen kautta molemmissa työkaluissa. Verkon määrittelyt suoritetaan graafisista asetusikkunoista, joista valitaan parametrit opetukseen. NeuralWorksissa voidaan hiiren avulla lisätä ja poistaa verkon kytchentöjä.

Prototyypin testaus ja viritys

Testauksen kannalta oleellinen asia on hyvien kuvaajien esittäminen käytetystä opetus- ja testiaineistosta sekä saadusta tuloksesta. Kuvassa 7a on esitetty NeuralWorksin työtila sekä joitain esille saatavia kuvaajia. Kuvassa 7b on esitetty BrainMakerin työtila sekä joitain kuvaajia opetuksen etenemisestä ja verkon parametreista.



Kuva 7a. NeuralWorksin työtila.



Kuva 7b. BrainMakerin työtila.

NeuralWorksiin on mahdollista muodostaa kuvaaja lähes jokaisesta verkossa esiintyvistä muuttujasta. BrainMakerissä voidaan valita valmiita kuvia, joiden avulla opetuksen etenemistä voidaan seurata. Molemmissa työkaluissa voidaan esittää mm. jokaisen piilokerroksen painokertoimista graafinen kuvaaja, jossa painokertoimien arvot on asetettu histogrammiin. BrainMakerista puuttuvat lasketun ja tavoitteena olleen vasteen esittävät ikkunat. Työkalun valmistaja on ilmeisesti ajatellut, että tarkoitukseen käytetään NetMaker-ohjelmaa, jolla mm. korrelaatiot saadaan laskettua.

Opetuksen tulosta voidaan pyrkiä parantamaan painokertoimien tai herätteiden tai molempien keinotekoisella häirinnällä. NeuralWorksissä ja BrainMakerissa on mahdollista lisätä häiriöitä verkon parametreihin. Lisäksi painokertoimet voidaan rajoittaa tietyille välille ja/tai poistaa raja-arvojen ulkopuolelle joutuneet arvot. BrainMaker mahdollistaa myös herätteiden häirinnän.

NeuralWorksissä on Explain Net -toiminto, jonka avulla voidaan suorittaa herkkyysanalyysiä muuttujien välillä. Se pyrkii määrittämään herätteen, jolla on suurin vaikutus verkon ulostuloon. BrainMaker sisältää mahdollisuudet suorittaa muodostetulle mallille erilaisia analyysejä, kuten verkon herkkyysanalyysi, sekä mahdollisuudet piirtää vastetta yhden tai kahden herätemuuttujan suhteen.

BrainMakerilla tehtyä sovellusta on mahdollista simuloida käyttämällä simulointiohjelmaa BrainMakerin rinnalla. Tällöin tarvittavat datat siirretään ohjelmien välillä käyttämällä DDE-linkkiä. Opetus ei ole mahdollista simuloinnin aikana ja tämä estää adaptiivisten sovellusten kehittämisen.

Integrointi kohdejärjestelmään

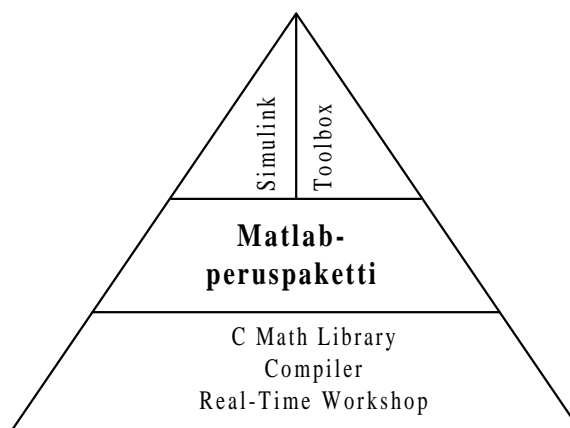
NeuralWorks kykenee generoimaan tehdystä sovelluksesta C-koodia, jonka avulla kehitetty sovellus voidaan integroida kohdejärjestelmään. BrainMakerissa on C-kielinen ohjelma, joka on tarkoitettu sen tuloksen integroimiseen kohdejärjestelmään. Tämä ohjelma kykenee lukemaan BrainMakerin muodostamat parametrit levyiltä ja laskemaan levyille tallennettujen herätteiden perusteella vasteet tiedostoon.

NeuralWarella on tarjota Designer Pack-moduuli, jonka avulla opetusalgoritmit voidaan liittää integroitavaan kohdejärjestelmään adaptiivisten toimintojen suorittamista varten.

4.3 Matematiikkatyökalu: Matlab + Fuzzy Logic Toolbox & Neural Network Toolbox

Matlab on MathWorksin perinteinen matriisilaskentaan perustuva ohjelmisto, jota useat yliopistot ja korkeakoulut käyttävät opetuksessa ja tutkimuksessa. Se tarjoaa melkein

rajattomat mahdollisuudet laajentamiseen ja omien sovellusten tekemiseen. Lähes kaikki siinä pyörivät rutiinit voidaan nähdä koodina, joka on eräänlainen versio Fortran-tai Basic-koodista. Tämä auttaa avointa järjestelmien kehitystä. Ohjelmasta on versiot PC:lle DOS- ja Windows-ympäristöön, yleisimpiin työasemiin (HP, Linux, SGI, VAX/VMS, DEC, Sun) ja Macintoshille. Siihen on tehty lukuisten yritysten toimesta runsaasti valmiita laskentakirjastoja eri tekniikoiden tutkimiseen. Se tarjoaa erinomaisen alustan erilaisten hybridiratkaisujen kehittämiseen ja testaamiseen sekä omien ideoiden toteuttamiseen. MathWorks on tehnyt Fuzzy Logic Toolbox -paketin sumean logiikan laskentaa varten ja Neural Network Toolbox -paketin neuroverkkolaskentaa varten. Matlabin perusfilosofia on esitetty kuvassa 8, jossa sen ominaisuuksia on kasvatettu lisäämällä siihen saatavilla olevia lisämoduuleja.



Kuva 8. MathWorksin Matlab-ohjelmiston modulaarinen rakenne.

Koska Matlabin peruskäyttöliittymä on täysin komentopohjainen, kynnys ohjelman käytölle on normaalia Windows-ohjelmaa suurempi. Pääsääntöisesti kaikki tarvittavat komennot on muistettava ulkoa.

Datan esikäsittely

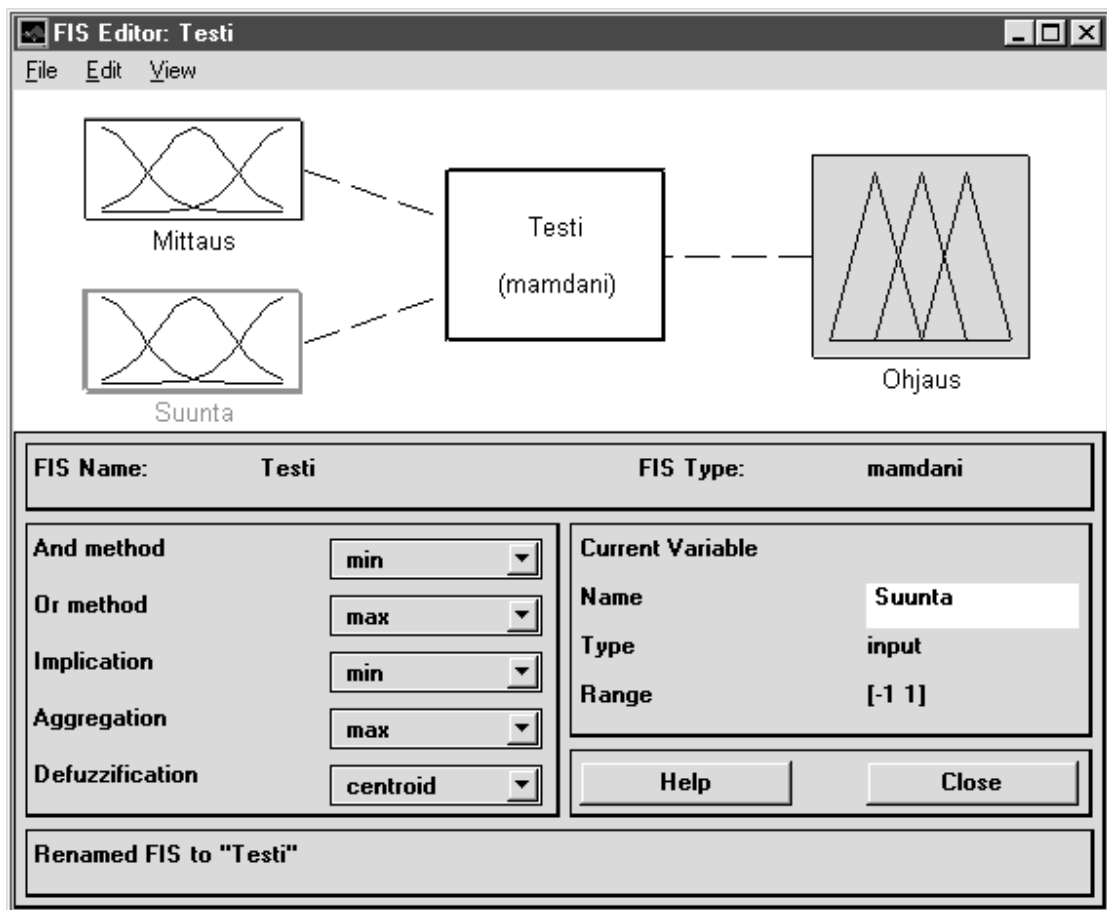
Data on helpointa lukea Matlabin työtilaan matriiseina ASCII-tiedostoista. Perinteiseen signaalinkäsittelyyn Matlabissa on olemassa runsaasti funktioita. Ohjelmasta kuitenkin puuttuvat valmiit funktiot ns. huonon ja puuttuvan datan käsittelylle sekä viiveellisen mittausdatan kohdistamiselle. Hyvin yleistä onkin, että mittaukset esikäsitellään osittain jollain muulla ohjelmalla. Vaihtoehtoisesti Matlabiin voidaan tehdä omia funktioita tai kerätä tarjolla olevista funktioista toiminnot haluttujen tehtävien suorittamiseksi.

Työkalusta puuttuvat valmiit funktiot mm. mittausten jakamiseen opetus- ja testausjoukkoihin, jota perinteisesti käytetään neuroverkko-opetuksessa. Neural Network Toolbox tarjoaa joukon funktioita datan normalisointiin.

Prototyypin kehitys (Fuzzy Logic Toolbox)

Työskentely tapahtuu osin graafisen ikkunan kautta, joka on poikkeus normaaliin Matlab-käyttöön. Tämä graafinen editori on esitetty kuvassa 9. Se tarjoaa mahdollisuuden määrittellä muuttujat herätteille ja vasteille sekä tarvittavan sääntökannan. Kuvassa 9 esitettyyn ikkunaan on FIS-editorilla kehitetty järjestelmä, joka koostuu kahdesta heräte- ja yhdestä vastemuuttujasta. Rakennettu järjestelmä voidaan tallentaa työtilaan tai levyllä ASCII-pohjaiseen FIS-tiedostoon.

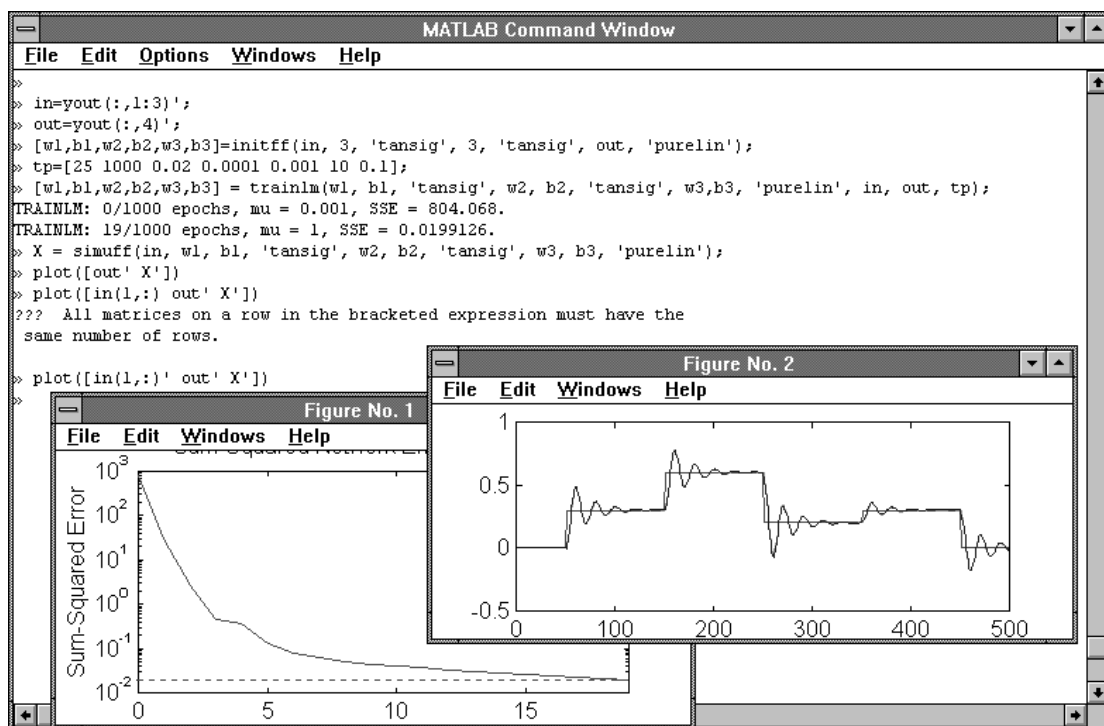
FIS-editori on pääikkuna joukolle editoreita, joiden avulla voidaan suorittaa sumean logiikan vaatimat määrittelyt heräte- ja vastemuuttujille, päättelymenetelmälle sekä sääntökannalle. Editorin käyttö ei vaadi suurta perehtymistä ja järjestelmän rakentaminen sujuu pääosin hiirtä käyttämällä. Editori on rakennettu käyttämällä Matlabin grafiikkaominaisuuksia, minkä vuoksi sen jouheva käyttäminen vaatii tietokoneelta kohtalaista laskentakapasiteettia (Pentium).



Kuva 9. Matlabin Fuzzy Logic Toolboxin FIS-editori.

Prototyypin kehitys (Neural Network Toolbox)

Neural Network Toolboxilla tehtävä kehitys suoritetaan täysin komentopohjaisen ikkunan avulla. Opetukseen Neural Network Toolbox tarjoaa yleisimmin käytetyt arkkitehtuurit ja algoritmit. Heikkoutena on, että Neural Network Toolbox ei kykene opetuksen aikana käyttämään testijoukkoa määrittelemään verkon hyvyttä. Tämä saattaa aiheuttaa joissain tilanteissa verkon ylioppimista ja tätä kautta ylioptimistisia odotuksia. Opetuksen eteneminen näkyy sekä numeerisesti että graafisesti. Kuvassa 10 on Neural Network Toolboxin funktioiden avulla suoritettu verkon opetus ja sen antamaa informaatiota opetuksen etenemisestä sekä kuvaaja opetuksen tuloksesta.



Kuva 10. Neural Network Toolboxin funktioiden avulla suoritettu verkon opetus.

Adaptiivisten järjestelmien kehitys on mahdollista, mutta niiden kehittämiseksi joudutaan tekemään omia funktioita kokonaisuuden hallitsemiseksi.

Prototyypin testaus ja viritys

Matlabin vahvuus on kehitetyn sovelluksen kehittäminen ympäristössä, jossa voidaan suorittaa laaja testaus käyttämällä sekä staattista että dynaamista simulointia. Sumean logiikan sovelluksen staattista käyttäytymistä testattaessa voidaan apuna käyttää sääntö- ja 3D-säätöpintaikkunaa kuten sumean logiikan perustyökaluissa. Erona perustyökaluihin on, että Matlabissa muutoksia ei voi tehdä simuloinnin aikana.

Opetetun neuroverkon analysointiin Neural Network Toolbox tarjoaa ainoastaan numeerisen ja graafisen raportin opetuksen etenemisestä sekä mahdollisuudet käyttää opetettua verkkoa simuloinnissa. Lisäksi on joukko menetelmäspesifisiä graafisia kuvaajia, joiden avulla tulosta voidaan analysoida. Keskeisimpänä puutteena Neural Network Toolboxin testausominaisuuksissa on verkon herkkyysanalyysin puuttuminen. Tämä on mahdollista lisätä itse, mutta keskeisenä analyysimenetelmänä sen toivoisi olevan mukana.

Matlab tarjoaa erinomaiset mahdollisuudet kehitetyn järjestelmän dynaamiseen simulointiin. Vastaavaa ominaisuutta ei perustyökaluista löydy. Käyttämällä Matlabin tarjoamaa simulointipakettia (Simulink) helpotetaan simuloinnin suoritusta. Dynaaminen simulointi edellyttää tietenkin, että sovelluskohteen malli on tiedossa. Simulink tarjoaa miellyttävät puitteet erilaisten hybridi- ja hierarkkisten ratkaisujen testaamiseen. Ohjelmasta puuttuvat kuitenkin ominaisuudet, jotka mahdollistavat sumean logiikan jäsenyysfunktioiden ja sääntöjen seurannan simuloinnin aikana. Järjestelmän hyvyden laskemiseen löytyy Matlabista tarvittavia apuvälineitä.

Integrointi kohdejärjestelmään

Fuzzy Logic Toolboxin mukana seuraa kaksi C-kielistä ohjelmarutiinia, joiden avulla kehitetty sovellus voidaan integroida muihin järjestelmiin. Omien lisäysten ja muutosten tekeminen tiedostoihin on mahdollista. Rutiinien oletuksena on, että parametrit ovat luettavissa tiedostosta, kun ohjelma käynnistyy.

Neural Network Toolbox ei tarjoa suoraan minkäänlaista tukea tehdyn sovelluksen integroimiseen kohdejärjestelmään. MathWorks on julkaissut C-koodigeneraattorin, joka kykenee muuttamaan käytetyt Matlab-tiedostot C-kielille. Sen avulla voidaan suorittaa tarvittavien funktioiden generointi automaattisesti. Tarvittava laskenta on kuitenkin yksinkertainen ja funktiot on mahdollista ohjelmoida nopeasti pienellä C-kielen tunteuksella.

Joillain automaatiotoimittajilla on olemassa valmiit moduulit, joiden avulla työasemassa toimivaan Matlabiin on mahdollista tuoda ja viedä mittauksia on-line-toiminnassa suoraan automaatiojärjestelmään. Ne ovat kuitenkin hyvin valmistaja- ja työasemakohtaisia ominaisuuksia.

4.4 Data Mining-työkalu: DataEngine

DataEngine on saksalaisen Management Intelligenter Technologien (MIT) valmistama tuote, joka on kehitetty lähinnä data-analyysiä varten. Ohjelma sisältää laajat ominaisuudet mittausten esikäsittelyyn ja niiden analysointiin. Sen menetelmäkirjastoon

kuuluvat sumea logiikka ja neuroverkot. Siitä on tehty versiot PC:lle Windows 3.1 ...NT -ympäristöihin. Ohjelmasta puuttuu on-line help -tiedosto. Ohjelman toiminnan logiikan oppiminen vaatii hieman totuttelua.

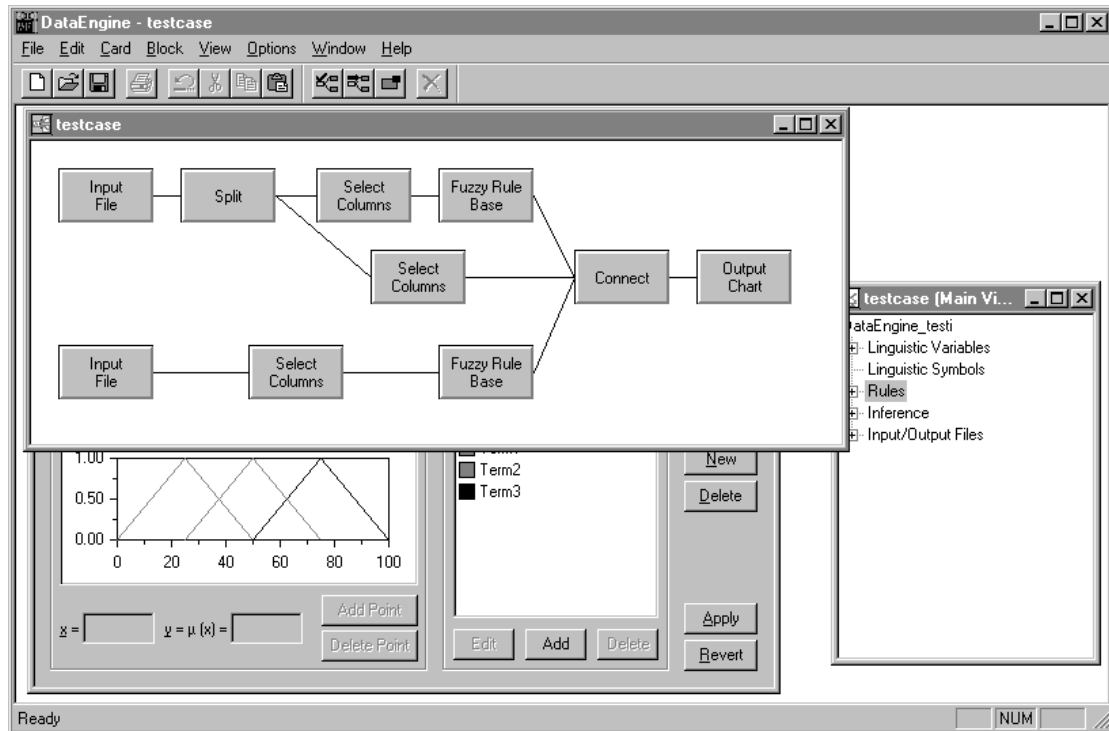
Datan esikäsittely

Data voidaan tuoda esikäsittelyyn DataEngineen ASCII- tai Excel-tiedostoista. Se voidaan suorittaa taulukkolaskentatyypin ikkunassa tai koota lohkoista Card-ikkunaan. Siinä voidaan tehdä tavallisia signaalinkäsittelyoperaatioita ja tutkia tuloksia 2D- ja 3D-kuvaajista. Esikäsitelty data tallennetaan levyille myöhempää käyttöä varten.

Prototyypin kehitys

DataEnginellä valitaan käytettävä menetelmä ja suoritetaan tarvittavat määrittelyt ja opetukset. Opetukseen ja testaukseen käytetään esikäsitellyssä tallennettua dataa. Valittavat menetelmät ovat sumea logiikka, perceptron, Kohosen kartta ja fuzzy-Kohonen verkot sekä fuzzy C-means -luokittelu. Tulokset tallentuvat myös tiedostoihin. Saadut tulokset joudutaan lukemaan aina dataeditoriin, jossa niitä voidaan tutkia graafisesti ja numeerisesti.

Laajemman kokonaisuuden kehittämiseen voidaan käyttää ns. Card-ikkunaa. Kuvassa 11 on esitetty DataEngin työtila, jossa Card-ikkunaan on muodostettu kahden sääntökannan järjestelmä. Se käyttää herätteitä kahdesta tiedostosta ja muodostaa sääntökantojen avulla yhden vasteen graafiseen ikkunaan.



Kuva 11. DataEnginen Card-ikkunaan rakennettu kahden sääntökannan järjestelmä.

Prototyypin testaus ja viritys

Itsenäisten menetelmien (ei Card-ikkunaa) testaamiseen voidaan käyttää testi- ja vertailuaineistoa. Herätteiden syöttäminen käsin ei ole mahdollista. Saadut tulokset tallentuvat tiedostoihin ja ne joudutaan lukemaan ohjelman dataeditoriin. Huomattavasti helpompaa olisi, jos tulokset olisivat käytettävissä automaattisesti.

Käytettäessä Card-ikkunaa voidaan usein toistuvia toimintoja rakentaa kokonaisuuksiksi. Card-ikkunaan voidaan rakentaa mm. kaikki DataEnginen tarjoamat datan esikäsittelyominaisuudet. Tällöin tiedostoista voidaan määritellä tarvittavat sarakkeet herätteiksi ja suorittaa esikäsittely ennen haluttua menetelmää. Ohjelma ei tarjoa mahdollisuuksia simulointiin ulkopuolisen ohjelman avulla.

Ohjelman menetelmät ovat rajalliset viritystoimintojen osalta. Ohjelmasta puuttuvat mm. mahdollisuudet muodostetun sumean päättelyn seuraamiseen, neuroverkkojen herkkyyksianalyysin suorittamiseen ja painokertoimien häirintään.

Integrointi kohdejärjestelmään

DataEnginen peruspaketti ei tuota koodia kehitetystä järjestelmästä. DataEngine ADL (Application Development Library) on C++ -kirjasto, joka mahdollistaa DataEnginellä

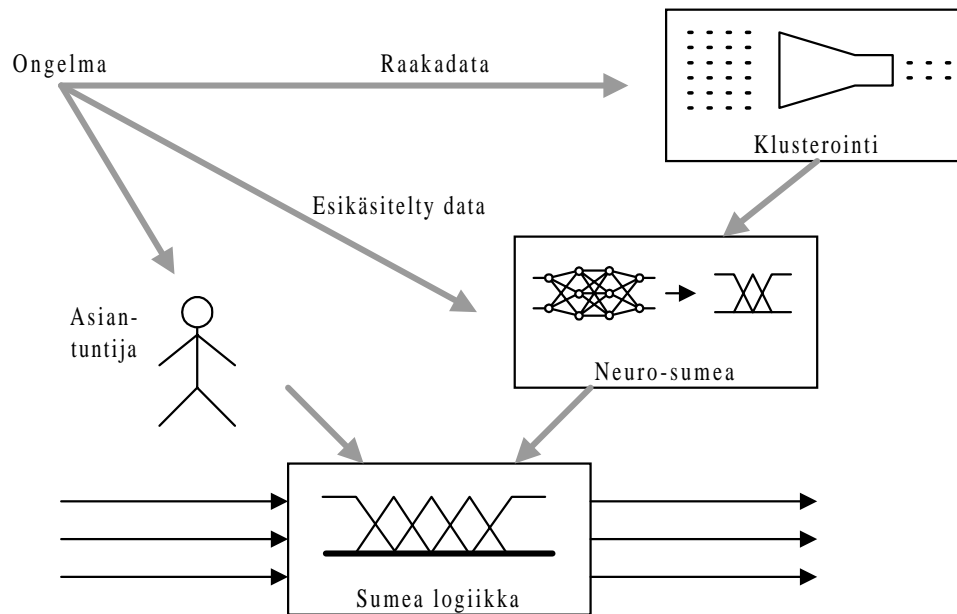
kehitettyjen sovellusten integroimisen haluttuun järjestelmään. Kirjasto toimitetaan C++-objektikirjastoina tai valmiina DLL-funktioina Windows-ympäristöön. DataEngine V.i. on LabView-järjestelmän lisämoduuli ja se mahdollistaa DataEngineä vastaavan sovelluskehittämisen suoraan LabView-ympäristössä.

4.5 Tukityökalut: sumean mallin generointimenetelmät

Useisiin sumean logiikan työkaluihin kuuluu nykyään menetelmä, jolla voidaan dataa analysoimalla generoida tarvittavat muuttujat ja sääntökanta. Niiden avulla voidaan prototyypin kehitykseen saada nopeutta ja uusia ajatuksia. Generointimenetelmien ajatuksena on hyödyntää saatavissa oleva mittausdata maksimaalisesti mallinnettaessa toimintaa tai käyttäjää. Samalla voidaan kehityksessä saavuttaa suuria aikasäästöjä. Jos kehitettävä sovellus on säätäjä, saavutettava hyöty voi kuitenkin olla minimaalinen haluttaessa oleellisesti parantaa aiempaa säätäjää. Seuraavassa listassa ovat selvityksessä mukana olevien työkalujen sisältämät tai erikseen saatavat opetusmenetelmät:

- fuzzyTech - NeuroFuzzy Module (fuzzyTech -lisämoduuli)
- TILShel - TILGen (Professional -versiossa)
- Fuzzy Logic Toolbox - Anfis (peruspaketissa)
- Muut valmistajat: WinRosa (myyjä MIT (DataEngine))

Kuvassa 12 on esitetty, millaista tietämystä erityyppiset menetelmät käyttävät ja miten niitä voidaan hyödyntää kehityksessä. Perinteisesti sumea järjestelmä kehitetään pelkästään asiantuntijatietämystä käyttämällä. Tällöin dataa käytetään lähinnä testauksessa. Dataan perustuvat generointi ja opetusmenetelmät pohjautuvat yleensä neuroverkoissa käytettyihin algoritmeihin tai tilastollisiin menetelmiin. Opetettu järjestelmä on yleensä yhtä hyvä kuin opetuksessa käytetty aineisto. Tästä syystä kannattaa datan esikäsittely tehdä huolellisesti.



Kuva 12. Erityyppisen datan käyttö sumeaa logiikkaa kehitettäessä.[Inform 1996]

Kolme edellä mainituista työkaluista käyttää neuroverkkotekniikkaa ja yksi tilastollisia menetelmiä muodostaessaan sumean logiikan päättelyn. Generointimenetelmien kehitys on laajan tutkimuksen kohteena maailmalla eikä mikään käytetyistä menetelmistä ole vielä vakiinnuttanut asemaansa. Taulukossa 1 on kuvattu lyhyesti näiden työkalujen käyttämiä menetelmiä ja niiden antamia tuloksia.

Taulukko 1. Opetusmenetelmissä käytetyt menetelmät ja niiden antamat tulokset.

Matlab Fuzzy Logic Toolbox & Anfis	<ul style="list-style-type: none"> – Neuroverkkojen rakenne on muutettu vastaamaan sumean logiikan järjestelmää. – Vastaa päättelyltään Sugenen menetelmää. – Generoi heräte- ja vastemuuttujille tarvittavat parametrit.
fuzzyTECH & NeuroFuzzy Module	<ul style="list-style-type: none"> – Menetelmänä neuroverkot (modifioitu error-backpropagation). Mahdollista tehdä omia lisäyksiä. – Generoi heräte- ja vastemuuttujien jäsenyysfunktioiden paikat ja sääntökannan. – Suorittaa sääntökannan optimoinnin.
TILShell & TILGen	<ul style="list-style-type: none"> – Menetelmänä neuroverkot (differential competitive learning - DCL). – Generoi säännöt kun heräte- ja vastemuuttujille on määritelty jäsenyysfunktiot.
WinRosa	<ul style="list-style-type: none"> – Menetelmä pohjautuu tilastollisiin menetelmiin ja hakualgoritmeihin. – Generoi rajat heräte- ja vastemuuttujien jäsenyysfunktioille sekä supistetun sääntökannan. Sisältää tuloksen analysoinnin. – Lukee/kirjoittaa Matlab, fuzzyTech ja DataEngine -tiedostoja.

Tutkituista työkaluista Matlab, fuzzyTech ja DataEngine sisältävät toimintoja mm. datan luokitteluun. Niissä on käytettävissä mm. Fuzzy C-Means -algoritmi. FuzzyTechissä tämä ominaisuus on saatavilla lisämoduulina.

4.6 Client/Server-työkalut: G2 + NeurOn-Line

G2 on Yhdysvaltalaisen Gensymin ohjelmistokehittäjä, joka tarjoaa client- tai server-ratkaisuja laajojen tietämispohjaisten sovellusten toteuttamiseksi. Siihen on saatavissa lisämoduuleja, joilla saadaan valmiita toimintapaketteja erityyppisten ratkaisujen kehittämiseen, samoin kuin lisämoduuli NeurOn-Line -neuroverkkosovellusten kehitykseen, joka lisää kehittäjien valmiita funktioita neuroverkkosovelluksen tekemiseksi. G2 + NeurOn-Line -paketista voi hyvin sanoa, että sitä ei ole tehty asiaan tutustumista ja nopeaa testailua varten. NeuroOn-Linen tehtävä on ennen kaikkea tarjota jo käytössä oleville installaatioille työkalu neuroverkkosovellusten kehittämiseen. Sovelluksen peruspiirre on, että sen tulee toimia G2:ssa eikä sitä ole tarkoitettu siirrettäväksi muualle. Yleisempää onkin, että jollain toisella työkalulla kehitetty ratkaisu toteutetaan NeurOn-Linellä lopulliseen integrointiympäristöön. G2:n perusjärjestelmä on olio-pohjainen ja perustuu pitkälti olioiden välisiin relaatioihin ja sääntöihin. Sumea logiikka kuuluu osana G2:n perusominaisuuksiin. G2:sta on saatavana versiot Windows NT-ympäristöihin (Intel, Alpha, Mips) sekä yleisimpiin työasemiin (Sun, Sgi, HP, IBM).

Prototyypin kehitys

Neuroverkkosovelluksen kehittäminen NeurOn-Linessä tehdään pääosin lohko-ohjelmoinnilla. Työkalun mukana tulee suuri määrä lohkoja, joiden avulla sovellus rakennetaan graafisesti G2:n päälle. Työkalu tarjoaa hyvät ominaisuudet datan käsittelylle ja visualisoinnille. Perusominaisuuksiin datan esikäsittelyssä kuuluvat mm. mittausten skaalaus, kopiointi, suodatukset, lajittelu ym. operaatiot. Työkalu onkin omiaan rakennettaessa järjestelmää, joka tehdään on-line-oppivaksi ja johon liitetään kattava datan esikäsittely ja lajittelu opetus- ja testiaineistoksi. Opetusmenetelminä NeurOn-Line sisältää perinteisen backpropagation-menetelmän sekä autoassosiativ-, radial basis- ja rho-verkot.

NeurOn-Line sisältää lohkon verkon herkkyysanalyysin suorittamiseksi. Tämän avulla voidaan valita verkkoon tuotujen mittausten merkitystä järjestelmän toimintaan. Lisäksi työkalussa on lohko eri määrittelyillä opettujen verkkojen paremmuuden vertaamiseksi.

Prototyypin testaus ja viritys

NeurOn-Line sisältää lohkot adaptiivisesti suoritettavaan opetukseen. Siihen kuuluvat lohkot, joiden avulla voidaan valita paras verkko useista vaihtoehdoista, uuden verkon käyttöönotosta ja edellisen verkon palauttamisesta. Työkalu sisältää mm. testilohkon, jonka avulla opetetun verkon virhe voidaan laskea halutulla datajoukolla.

G2:n perusominaisuuksiin kuuluu myös graafisten esitysten tekeminen, mutta NeurOn-Line sisältää omat lohkot haluttujen datojen esityksiin.

Integrointi kohdejärjestelmään

G2 + NeurOn-Line toimii lopullisena kohdejärjestelmänä eikä erillistä integrointia tarvitse tehdä.

4.7 Yhteenveto kaupallisista työkaluista

Tutkimuksessa mukana olleet työkalut ovat suosittuja sumean logiikan ja neuroverkkojen soveltajien käytössä. Samalla ne edustavat monenlaista näkemystä sovellusten kehittämiseen ja testaukseen. Useissa yliopistoissa ja tutkimuslaitoksissa on kuitenkin tehty omia työkaluja lähinnä opetus- ja tutkimustarkoituksiin. Näiden ohjelmien saatavuus ja tuki vaihtelevat suuresti. Siksi selvityksessä keskityttiin ainoastaan kaupallisiin työkaluihin.

Tutkituilla ohjelmilla sumean logiikan ja neuroverkkojen sovellukset pystytään kehittämään varsin hyvin. Niiden ominaisuudet ovat kuitenkin painottuneet eri kohtaan kehitysprosessia ja käytettävyys vaihtelee komentopohjaisesta valikkotyypiseen toimintaan ja lohko-ohjelmointiin asti. Työkalun valintaan vaikuttavat oleellisesti ratkaistavan ongelman luonne ja lopullisen kohdejärjestelmän ominaisuudet sekä käyttäjän aiemmat kokemukset työkaluista. Windows-ohjelmien valikkotyypinen toiminta on usein nopein ja helpoin oppia. Usein tuloksia saadaan jo pelkän kokeilun ja erehdyksen kautta tutustumatta ohjeisiin. Kun työkalun liittämistä ajatellaan kappaleessa 3 esitettyyn kehitysprosessin malliin, voidaan työkaluista löytää aina ohjelma tarvittavien tehtävien suorittamiseen ylläpitoa lukuunottamatta.

Datan esikäsittely

Datan käyttö on ehdoton edellytys sovelluksen kehityksessä ja testauksessa, jolloin joudutaan usein tekemään laajaa datan käsittelyä.

Datan esikäsittely on vertailtujen ohjelmien heikko kohta. Yhdenkään ohjelman esikäsittelyominaisuudet eivät ole helppokäyttöisiä, ja visuaalisuus jättää osin runsaasti toivomisen varaa. NeurOn-Line laajan lohkokirjastonsa kanssa on monipuolinen, mutta käytöltään raskas. BrainMaker, Matlab ja DataEngine ovat ominaisuuksiltaan välttäviä ja pienillä muutoksilla niistä saisi monipuolisia ja helppokäyttöisiä kokonaisuuksia. TILShelliin ja Matlabiin on mahdollista tehdä halutunlaisia rutiineja, mutta se saattaa olla työlästä.

Kaikki ohjelmat hallitsevat datojen lukemisen matriisimuotoisesta ASCII-tiedostosta. Lisäksi Excel-taulukkolaskentatiedostot ovat käytettävissä useille ohjelmille.

fuzzyTech ja NeuralWorks ovat liittäneet työkaluihinsa erilliset lisämoduulit, joiden avulla toimintojen määrää lisätään ja helpotetaan. NeuralWorksissä ei itsessään ole minkäänlaisia esikäsittelyominaisuuksia, mutta erillisen ohjelman avulla tarvittavia toimenpiteitä on mahdollista suorittaa.

Prototyypin kehitys (sumean logiikan työkalut)

Kaikissa tutkituissa ohjelmissa sovelluksen rakentaminen tapahtuu pääosin graafisen käyttöliittymän avulla. Sumean logiikan työkalut sisältävät editoreita tarvittavien määrittysten suorittamiseksi, esim. jäsenyysfunktio- ja sääntöeditorit. Jäsenyysfunktioeditorit ovat kaikissa työkaluissa lähes samanlaisia, mutta sääntöeditoreissa on eroja. Säännöt on mahdollista syöttää valikkojen avulla taulukkoa täyttämällä, kirjoittamalla tekstiä tai indeksejä tekstikenttään tai täyttämällä hiiren avulla matriisi-ikkunaa. FuzzyTech ja TILShell ovat kehitetty pisimmälle käyttäjäystävälliseen suuntaan. Niissä etenkin sääntöjen muodostaminen on huomattavasti helpompaa kuin muissa sumean logiikan työkaluissa.

Sääntökantojen ketjuttaminen hierarkkisiksi tai rinnakkaistaminen loogisiksi kokonaisuuksiksi saattaa helpottaa sovelluksen kokonaisuuden hallintaa. Jokaisella työkalulla voidaan rakentaa järjestelmä, jossa käytetään useaa sumeaa sääntökantaa. Matlabissa ja DataEnginessä laajan sovelluksen analysointi on työlästä puutteellisten monitorointiominaisuuksien vuoksi.

Prototyypin kehitys (neuroverkkojen työkalut)

Tottuneelle Windows-käyttäjälle BrainMakerin käyttö ei tuota vaikeuksia. Valinnat ja määrittelyt hoidetaan pääosin menu-valikoista ja kyselyikkunoista. Lähes vastaava on NeuralWaren käyttö, jota vaikeuttavat kuitenkin suuresti laajat ominaisuudet. Lähes kaikkea sen toimintaa voidaan jollain parametrillä konfiguroida. Sen käyttöä voidaan yksinkertaistaa ns. novice-moodin käytöllä. Matlabin käyttöä haittaa kynnyksen perusteiden opettelussa, minkä jälkeen käyttö on suoraviivaista, kunhan muistaa tarvittavat

komennot ja matriisien käsittelyt. G2 NeurOn-Linen käyttökyky on suurin. Jotta työkalusta saisi suurimman hyödyn, tarvitaan pitkää työskentelyä työkalun parissa. Monella soveltajalla ei ole varaa panostaa niin paljon ainoastaan yhden sovelluksen varaan, jonka lopputuloksesta ei olla täysin varmoja.

Menetelmiltään NeuralWorks on ylivoimainen muihin työkaluihin verrattuna. Matlab Neural Network Toolbox on hyvä kakkonen, mutta siinä näkyy jo version vanhuus. BrainMaker ja G2 NeurOn-Line perustuvat perinteiseen backpropagation-verkkoon ja sen variaatioihin.

Kehityksessä hyvin tärkeänä perusominaisuutena voidaan pitää testidatan käyttöä opetuksen aikana. NeuralWorks, BrainMaker ja NeurOn-Line mahdollistavat testidatan käytön verkon hyvyden määrittämiseen ja opetuksen keskeyttämiseen. Matlabista tämä ominaisuus puuttuu. Muita yleisiä ominaisuuksia ovat muuttujien ja tulosten esittäminen graafisesti opetuksen etenemisestä.

Graafiset kuvaajat ovat kaikissa selvityksen ohjelmissa verraten hyviä. NeuralWorks ja Matlab tarjoavat parhaat mahdollisuudet muuttujien ja tulosten esittämiseen ja niiden kuvaajien parametrintimahdollisuudet ovat laajimmat. Matlabin etuna on sen kyky esittää suuriakin datamääriä.

Prototyypin testaus ja viritys

Sumean logiikan sovellusta kehitettäessä työkalun tarjoamat ominaisuudet päättelyn seuraamiseen ovat tärkeitä. Ne vaihtelevat vertailuissa ohjelmissa suuresti. Heikoimmaksi nämä ominaisuudet havaittiin Matlabissa ja DataEnginessä, jotka eivät tarjoa juuri muuta kuin staattisten muutosten tarkastelun. Kaikki työkalut tarjoavat kuitenkin mahdollisuuden 3D-säätöpinnan piirtämiseksi kahden herätemuuttujan ja yhden vastemuuttujan välille. Usean muuttujan tapauksessa tätä on hankalaa käyttää, mutta se havainnollistaa usein selvät puutteet. Lisäksi on mahdollista antaa arvoja manuaalisesti herätemuuttujille.

Työkaluilla kehitettyjen sovellusten dynaamiseen simulointiin on kolme ratkaisutapaa. Ne ovat sovelluksen siirtäminen simulointiympäristöön, simuloinnin tekeminen kehitysympäristössä tai työkalun ja simulointiohjelman keskinäinen kommunikointi. fuzzyTech tukee sovelluksen siirtämistä VisSim- tai Matlab-ympäristöön. TILShell tarjoaa mahdollisuuden oman simulointilohkon ohjelmoimiseksi. Matlab, fuzzyTech ja TILShell tarjoavat DDE-linkin ohjelmien väliseen kommunikointiin ja muuttujien siirtämiseen. Simuloinnissa ja off- tai on-line-testauksessa on käyttöä mm. laukeavien sääntöjen esittämisellä ja kumulatiivisella esityksellä. Lisäksi normaalin trendipiirron avulla voidaan seurata testauksen etenemistä.

Neuroverkon analyysiä voidaan tehdä ns. herkkyysanalyysin avulla. Siinä tutkitaan esim. yhden herätemuuttujan vaikutusta vasteeseen muiden herätteiden ollessa vakioita. Siten voidaan analysoida mm. herätteiden tarpeellisuutta. NeuralWorks ja BrainMaker sisältävät toimintoja tämän analyysin suorittamiseksi.

Opetetun verkon hyvyyden määrittäminen simuloimalla verkkoa kehitysympäristössä onnistuu vain Matlabissa. Simulointi on mahdollista suorittaa ajamalla neuroverkko-ohjelmaa simulointiohjelman rinnalla ja siirtämällä dataa niiden välillä. BrainMakerin pystyy kommunikoimaan simulointiohjelman kanssa DDE-linkin avulla ja NeurOn-Line voidaan asettaa lukemaan data tiedostosta tietyin välein. NeuralWorks voidaan käynnistää eräajona simulointiohjelmasta, jolloin datojen siirto hoidetaan tiedostojen kautta.

Integrointi kohdejärjestelmään

Kehitetyn järjestelmän integroinnissa yleisin vaihtoehto on siirtää ohjelma C-kielisenä kohdejärjestelmään. Pienissä laboratoriotyypisissä kokeiluissa sovellus saattaa pyöriä PC:llä kehitysympäristössä mutta harvemmin teollisuusympäristössä. Lähes kaikki tutkitut työkalut tukevat C-kielen generointia tai tarjoavat kirjastot laskennan siirtämiseksi uuteen ympäristöön. G2 eroaa joukosta siinä, että se toimii myös lopullisena integrointiympäristönä.

Neuroverkkosovellukset voidaan jakaa karkeasti off- ja on-line-opetettaviin sovelluksiin. Näiden vaatimukset integroinnille kohdejärjestelmään ovat hivenen erilaiset. Off-line opetettavan verkon tuloksen siirtäminen ja laskennan muodostaminen on varsin yksinkertaista. Se tehdään yleensä C-kielillä tai automaatiojärjestelmien lohko-ohjelmoinnilla. On-line oppivat adaptiiviset järjestelmät ovat huomattavasti laajempitöisiä ja vaativat opetusalgoritmien integroinnin. Tehtävä edellyttää jonkin ohjelmointikielen käyttöä, yleensä C-kielen käyttöä. On-line opetusalgoritmien integroimiseen antavat ainoastaan NeuralWare ja DataEngine mahdollisuuden lisämoduulien avulla.

Ylläpito

Tutkituista ohjelmista fuzzyTech ja TILShell generoivat dokumentin tehdystä sovelluksesta. FuzzyTechin tuottama dokumentti on kuitenkin laajempi. FuzzyTechissa on lisäksi ominaisuus, jonka avulla voidaan suorittaa automaattista versionhallintaa kehityksen aikana.

Taulukossa 2 on yhteenveto sumean logiikan ja neuroverkkotyökalujen keskeisimmistä ominaisuuksista. On huomattava, että Matlabiin lähes kaikki ominaisuudet on mahdollista tehdä lisäämällä omia funktioita. Sitä ei ole kuitenkaan erikseen taulukkoon

merkitty. G2 NeuroOn-Line on jätetty pois taulukosta, koska G2:n hankkiminen ainoastaan neuroverkkosovelluksen rakentamista varten on perusteetonta, jos sen muita ominaisuuksia tietämuspohjaisen sovelluksen rakentamiseksi ei hyödynnetä.

Taulukko 2. Yhteenveto neuroverkkotyökalujen ominaisuuksista. K 'ominaisuus on olemassa', L 'ominaisuus on mahdollista liittää lisämoduulin avulla' ja # 'ominaisuus ei ole menetelmän kanssa tarpeellinen'.

	fuzzyTech	TILShell	Fuzzy Logic Toolbox	DataEngine	NeuralWare Pro II/+	BrainMaker Professional	Neural Network Toolbox
Datan esikäsittely							
Datatieostot							
ASCII	K	K	K	K	K	K	K
Excel	L		L	K	L	K	L
Puuttuvat datat	L			K	L		
Suodatus	L			K	L	K	
Luokittelu							
Fuzzy C-Means	L		K	K			
Kohosen verkko				K	K		K
Kehitys							
Graafinen	K	K	K	K	K	K	
Muuttujaeditori	K	K	K	K	#	#	#
Sääntöeditori	K	K	K	K	#	#	#
Testaus							
Staattinen testaus							
3D-säätöpinta	K	K	K		#	#	#
Mittaustiedostot	K	K	K	K	K	K	K
Manuaaliarvot	K	K	K	K	K	K	K
Menetelmät							
Backpropagation				K	K	K	K
Kohonen				K	K		K
Dynaaminen simulointi							
Simulointimalli työkalussa		K	K				K
Simulointimalli toisessa ohjelmassa	K	K	K			K	K
Simulointi toisessa ympäristössä	K						
Muuttujien seuraaminen	K	K			K		
Laukeavien sääntöjen seuraaminen	K	K			#	#	#
Muutosten tekeminen testauksen aikana	K	K					
Integrointi							
C-kieli valmius / generointi	K	K	K	L	K	K	L
Testaus integrointiympäristössä	K	K					
Opetuksen integrointi				L	L		
Dokumentin generointi	K	K					
On-line ohjeet	K	K	K		K		K

4.8 KOTIMAISET TYÖKALUT

4.8.1 Q-Opt

Yleistä

Q-Opt on Lappeenrantalaisen Taipale Engineeringin ja Lappenrannan teknillisen korkeakoulun yhteistoiminnan tuloksena syntynyt neuroverkko-ohjelmisto, joka on kehitetty lähinnä prosessiteollisuuden sovelluksiin. Ohjelman kehityksen lähtökohtana on ollut erityyppisen tietämyksen yhdistäminen prosessin mallinnukseen. Mallinnuksessa voidaan käyttää hyväksi prosessin tunnettua fysikaalista tietoa, opittua sumeaa tietämystä, prosessimittauksia ja prosessin rakennetta.

Q-Opt:n toiminta on rakennettu osin Matlabin ytimen päälle ja osin erillisillä ohjelmilla.

Prototyypin kehitys

Prototyypin valmistuksessa Q-Opt käyttää useita erilaisia tiedostoja, joista osaa muutetaan käsin ja osaa Matlabissa editorin avulla. Q-Optin muuttujien määrittelyt tehdään tiedostoon erityisessä ASCII-formaatissa ja mittaustiedot tallennetaan levyllä Matlab-formaatissa. Mittaustietojen esikäsittely on tehtävä ennen niiden tallennusta levyllä. Työkalu sisältää Matlab-rutiinin, jonka avulla on mahdollista muodostaa verkosta lohkomainen prosessin rakennetta vastaamaan. Työkalussa on erillinen ohjelma tallennettujen mittaustietojen jakamiseksi opetus-, testi- ja validointiaineistoksi. Matlabiin tehtyjen rdesign- ja rparser-toimintojen avulla luodaan säännöt verkon muodostamiseen. Tämän jälkeen esikäsitellään säännöt tähän tarkoitettu ohjelmalla. Lopuksi suoritetaan mallin opetus omalla ohjelmalla. Valmis malli voidaan ajaa Matlabissa.

Prototyypin testaus ja viritys

Q-Opt sisältää Matlabissa toimivan pääohjelman, jonka valikoista tehdyn sovelluksen toimintaa on mahdollista testata. Keskeisin tapa tutkia muuttujien käyttäytymistä on käyttää muuttujalista-ikkunaa. Siinä on mahdollista muuttaa halutun muuttujan arvoa ja samalla seurata sen vaikutusta muihin muuttujiin.

Sovelluksen testaus voidaan suorittaa tutkimalla muuttujien käyttäytymistä graafisista kuvaajista. Testauksessa käytettävää dataa on mahdollista vaihtaa.

Prototyypin integrointi

Ohjelma on mahdollista integroida Matlabia käyttävään tietokoneeseen.

4.8.2 AboaFuzz

Yleistä

AboaFuzz on Åbo Akademin kehittämä Windows-ympäristössä toimiva sumean logiikan työkalu, joka mahdollistaa sovelluskohteesta saadun tietämyksen ja kerätyn mittausdatan hyödyntämisen sovellusta kehitettäessä. Työkalun käyttökohteena ovat etenkin säätösovellukset. Ohjelman toiminta perustuu ennen kaikkea sumean luokittelun (fuzzy c-means) tuloksen hyödyntämiseen sääntökannan muodostamisessa. Ohjelma on suunniteltu avoimeksi ja siihen on mahdollista lisätä uusia menetelmiä tutkimuksen edetessä. Työkalun myyntiä hoitaa FF-Automation.

Prototyypin kehitys

AboaFuzz koostuu kahdesta moduulista joiden avulla tarvittavat toiminnot muodostetaan. AboaFuzz@Control-moduulin avulla määritellään manuaalisesti heräte- ja vaste-muuttujien sekä sääntökannan tarvitsemat parametrit. AboaFuzz@Cluster-moduulin avulla voidaan automaattisesti muodostaa sääntökanta ja virittää jäsenyysfunktioiden parametrit.

AboaFuzz@Control-moduuli on graafinen käyttöliittymä määriteltäessä sumean logiikan tarvitsemia parametreja. Jäsenyysfunktiot voidaan asettaa hiiren avulla, ja sääntöjen määrittämiseksi on käytettävissä taulukkomuotoinen ikkuna. Ikkunassa valinnat tehdään valitsemalla listasta halutut muuttujat.

Prototyypin testaus ja viritys

AboaFuzz@Control-moduulissa on asiakas/palvelin DDE-linkki, jonka avulla herätteet ja vasteet saadaan siirrettyä toiseen ohjelmaan. Tämä toinen ohjelma voi olla esim. valvomo-ohjelmisto, joka on liitetty ohjattavaan prosessiin. Ohjelmasta puuttuvat testauksen aikaiseen monitorointiin tarvittavat ikkunat.

AboaFuzz@Cluster-moduulia käytetään muodostamaan sumea sääntökanta kerätystä mittausdatasta. Moduuli käyttää annettua painokerrointa ja luokkamääräväliä luokituksen suorituksessa. Luokituksen tuloksia vertaillaan testausikkunassa, jossa tulos on mahdollista laskea tiedostoon käyttämällä luokittelussa saatuja sääntökantoja. Tulos

joudutaan analysoimaan jossain toisessa ohjelmassa. Valittu sääntökanta tallennetaan tiedostoon AboaFuzz@Control -moduulin käytettäväksi.

Prototyypin integrointi

AboaFuzz-työkalulla tehty sovellus voidaan integroida kahdella eri tavalla kohdejärjestelmän mukaan. Helpoin tapa on asettaa AboaFuzz@Control -moduuli toimimaan DDE-linkin avulla valvomo-ohjelmiston rinnalla. Tämä soveltuu lähinnä testaus- ja laboratoriokäyttöön. Toinen mahdollisuus on käyttää AboaFuzz@Control -moduulin generoimaa C- ja C++ koodia. Se antaa käyttöön joukon perusfunktioita, joiden avulla tarvittavat toiminnot voidaan sulauttaa kohdejärjestelmään.

4.8.3 Neural-MILL

Yleistä

Neural-MILL on Control Express Finland Oy:n valmistama neuroverkko-ohjelma, joka perustuu itsejärjestyvään karttaan (SOM, Self Organizing Map). Sitä on kehitetty lähinnä prosessinohjauksen sovelluksiin off-line-työkaluksi ajotapojen määrittelemiseen. Ohjelma toimii Windows-ympäristössä.

Prototyypin kehitys

Neural-MILLin perusoletus on, että mittausten esikäsittely tehdään jollain toisella työkalulla kuten taulukkolaskennalla. Ennen mittausten tallennusta joudutaan mittaustiedostoon sijoittamaan Neural-MILLin vaatimat perusmäärittelyt, koska niiden lisääminen ei ole mahdollista työkalusta käsin. Esikäsiteltyjen mittausten on oltava ASCII-tiedostona.

Neural-MILL tarjoaa mittausten esikäsittelyominaisuuden, vaikka edellä suositeltiin muun työkalun käyttöä. Neural-MILLiin on mahdollista rakentaa suodatin, joka tutkii jokaisen mittaustapahtumarivin ja poistaa tai muuttaa ehdon täyttäneen rivin. Pois siirrettävät rivit siirretään toiseen tiedostoon. Ehtoja voidaan asettaa useita.

Neural-MILL käyttää menetelmänä itseoppivaa neuroverkkoa (itsejärjestyvä kartta), jonka avulla pyritään identifioimaan prosessin tiloja. Ohjelma generoi muuttujakohtaiset kartat, joiden avulla voidaan hahmottaa prosessin riippuvuuksia. Opetuksen käyttö on suoraviivaista, koska käyttäjällä ei ole mahdollisuutta vaikuttaa opetuksen parametreihin.

Prototyypin testaus ja viritys

Neural-MILL on tehty mahdollisimman visuaaliseksi, koska käytetty menetelmä vaatii karttojen esittämisen tulosta analysoitaessa. Työkalu sisältääkin havainnollisen ja helppokäyttöisen kyselytoiminnon, jonka avulla mittausten välisiä riippuvuuksia voidaan tutkia.

Työkalussa on käytettävissä optimointitoiminto, jonka avulla voidaan etsiä optimia haluttuun vasteeseen pääsemiseksi. Optimointiprosessille on suunniteltava ensin strategia, joka sisältää joukon ehtoja, jotka ns. optimointisolun tulee täyttää sekä joukon tavoitteita. Strategioita voi olla useita ja kukin voi sisältää useita ehtoja ja tavoitteita.

U-matriisi-toiminnon avulla voidaan havainnollistaa kartassa esiintyviä muutoskohteita.

Prototyypin integrointi

Neural-MILLillä tehtyjä sovelluksia ei ole tarkoitettu integroitavaksi toiseen järjestelmään, koska kysely- ja optimointitoiminnot ovat hyvin keskeisiä ohjelman toiminnassa.

4.8.4 Tutkimuslaitosten työkalut

Yliopistot ja tutkimuslaitokset ovat olleet hyvin aktiivisia sumean logiikan ja neuroverkkojen tutkimuksessa viimeiset viisi vuotta. Tänä aikana on moneen eri tutkimusryhmään syntynyt työkaluja, joiden tarkoituksena on ollut omien taitojen, menetelmien ja ideoiden kartoittaminen. Osa edellä esitetyistä työkaluista onkin lähtöisin jossain määrin tutkimuksen piiristä. Muita tiedossa olevia työkaluja ovat sumean logiikan puolella Oulun yliopiston kehittämä FuzzyCon/FuzzyTune ja VTT Elektroniikan kehittämä FINTOOL. Ne eivät ole kuitenkaan yleisessä jakelussa. Neuroverkkojen puolella ryhmät ovat olleet aktiivisempia ja siellä on syntynyt useita työkaluja, joista osa on vapaasti levitettäviä ja jatkuvan kehityksen tukemia. Yleisimmin tunnettuja ovat Teknillisessä korkeakoulussa kehitetyt SOM_PAK ja LVQ_PAK. Lisäksi Lappeenrannan teknillisen korkeakoulun ja Jyväskylän yliopiston yhteistyönä ovat syntyneet CARELIA ja TS-SOM.

FuzzyCon/FuzzyTune

FuzzyCon on Oulun yliopistossa kehitetty Windows-ohjelma, jonka avulla voidaan kehittää ja testata sumean logiikan toimintaa. Tehty graafinen käyttöliittymä on yksinkertainen mutta tehokas. Ohjelma mahdollistaa staattisen testauksen käyttämällä herätetiedostoja ja manuaalisesti syötettyjä arvoja sekä dynaamisen testauksen

käyttämällä DDE-linkkiä simulointi- tai kommunikointiohjelman kanssa. Testauksen aikana on mahdollista seurata muuttujien käyttäytymistä sekä lauenneita sääntöjä.

FuzzyTune on FuzzyConin rinnalle kehitetty työkalu, jonka avulla voidaan kehittää säännöt ja jäsenyysfunktiot sumealle säätimelle. Ohjelma pohjautuu lingvististen yhtälöiden menetelmään, joka on kehitetty Oulun yliopistossa. Siinä mittausdataa hyödyntämällä saadaan generoitua tarvittavat määrittelyt järjestelmän muodostamiseksi. Menetelmään on myös mahdollista liittää asiantuntijatietämystä, jonka avulla voidaan määrittää muuttujien välisiä suhteita.

FINTOOL

FINTOOL (Fuzzy Inference Tool) on VTT Elektroniikan Oulussa kehittämä sumean logiikan kehitysympäristö, jonka avulla voidaan toteuttaa ja testata säätöjärjestelmiä. Kehitysympäristö toimii Windows-järjestelmässä. FINTOOL-työkalu muodostuu erilaisista editoreista, joiden avulla käyttäjä voi graafisesti määrittellä heräte- ja vastemuuttujat sekä sääntökannan. Ohjelmassa on mahdollista suorittaa staattinen testaus käyttämällä Run-editoria, jolloin mittaustiedostojen tai manuaalisesti syötettyjen arvojen käyttö on mahdollista. Dynaaminen testaus voidaan suorittaa kommunikoimalla ulkopuolisen simulaattorin kanssa DDE-linkin kautta.

FINTOOL-työkaluun kuuluu FLS-viritysmenetelmä, joka pohjautuu pienimmän neliösumman menetelmälle. Menetelmässä sääntöjen parametrit sovitetaan vastaamaan mahdollisimman hyvin annettua mittausdataa. Perusoletuksena käytölle on, että sumean sääntöjoukon rakenne ja tulomuuttujien jäsenyysfunktiot tunnetaan. Menetelmän tarkoitus ei ole automatisoida järjestelmän kehitystä vaan toimia asiantuntijan tukena.

FINTOOL sisältää C++ luokkakirjaston FUZLIB, joka sisältää luokkia sumean logiikan päättelyyn. Ohjelman tuottaman C++ -kielisen koodin ja FUZLIB -kirjaston avulla kehitetty järjestelmä voidaan integroida haluttuun kohdejärjestelmään.

SOM_PAK / LVQ_PAK

Teknisessä korkeakoulussa Otaniemessä kehitetty SOM_PAK (Self-Organizing Map Program Package) on neuroverkkotyökalu, joka käyttää menetelmänään itse-organisointuvaa karttaa. Se muodostuu useista ohjelmista, joiden avulla tehdään tarvittavat toimenpiteet piirrekartan muodostamiseen ja analysointiin. Kaikki ohjelmat toimivat komentopohjaisina ja ainoa graafinen toiminta on muodostetun kartan visualisointi. LVQ_PAK (Learning Vector Quantization Program Package) on neuroverkkotyökalu, joka käyttää menetelmänään oppivaa vektorikvantisaatiota. Se koostuu erillisistä ohjelmista, joiden toimintaa ohjataan lvq-run-ohjelman avulla.

Ohjelmien käyttäminen erikseen on myös mahdollista. Työkalut ovat vapaasti saatavissa ANSI C -koodina tai binäärisenä DOS-järjestelmään.

Toiminta pohjautuu herätedatan käyttöön, jotka on oltava opetus- ja testitiedostoissa ASCII-muodossa SOM_PAKin tarvitsemin tiedoin. Herätetiedostot on käsiteltävä esim. taulukkolaskennan avulla puuttuvien datojen ja muiden häiriöiden poistamiseksi. Muodostettu kartta talletetaan levyllä ASCII-tiedostoon.

TS-SOM

TS-SOM (Tree Structured Self-Organizing Map) on Lappenrannan teknillisessä korkeakoulussa ja Jyväskylän yliopistossa kehitetty työkalu puurakenteisen itseorganisaatiokartan muodostamiseksi. Myös menetelmä on siellä kehitetty. Menetelmä hyödyntää muuttujien välistä ryhmittelyä sekä visualisointia. Työkalu ei ole vapaasti saatavilla.

4.9 KOTIMAISTEN TYÖKALUJEN KAUPALLISTAMINEN

4.9.1 Ohjelmistoliiketoiminnan yleispiirteitä

Suomalainen ohjelmisto-osaaminen on menestynyt kansainvälisesti parhaiten teollisuustuotteissa, joissa sulautettu ohjelmisto toimii tuotteen tai järjestelmän erottamattomana osana. Menestyneitä tuotteita ovat mm. tietoliikenne- ja automaatioalan tuotteet ja yleensäkin sähkö- ja elektroniikkateollisuuden tuotteet. Osa tuotteista on massatuotteita mutta valtaosa vaativia projektitoimituksina myytäviä järjestelmiä.

Tietojenkäsittelytoimiala on perinteisesti painottunut yksittäisratkaisujen kehittämiseen ja projektimyyntiin. Suuret tietojärjestelmäkokonaisuudet ovat aina jossain määrin yksilöllisiä ja siten projektiliiketoiminnan piirissä, vaikka ne enenevästi sisältävätkin valmisosia. Esimerkkejä kansainvälisestikin menestyneistä projektitoimituksina myytävistä ohjelmistoista ovat finanssietojärjestelmät, tehdastietojärjestelmät sekä erilaisia prosesseja opastavat, diagnostisoivat ja monitoroivat järjestelmät.

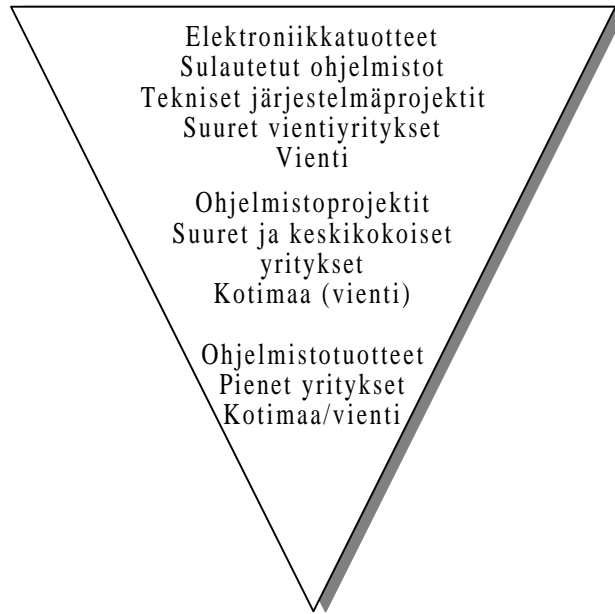
Laitte- ja ohjelmistoalustojen standardoituessa on ohjelmistotuoteliiketoiminta vahvistunut myös Suomessa. Ohjelmistojen tuotteistaminen on välttämätön edellytys kansainväliselle liiketoiminnalle käyttölisenssimyynnin muodossa. Ohjelmistotuoteliiketoiminnan tuotteet voivat olla projektiliiketoiminnan tarvitsemia valmisosia tai -komponentteja. Useimmiten kuitenkin tuote on itsenäisen suppeahkon tehtävälueen työkalu tai kokonaisratkaisu. Menestyneitä suomalaisia esimerkkejä ovat työryhmäohjelmit, suunnitteluohjelmit, tietokantaohjelmit, tietoturva- ja virustorjunta-

ohjelmistot jne. Ohjelmistotuote tarvitsee tuekseen oheispalveluita markkinoinnin, myynnin, koulutuksen ja käyttötuen muodossa.

Oppivat ja älykkäät järjestelmät koostuvat pääosin tietokoneohjelmistoista ja niihin liittyvää liiketoimintaa on kaikissa em. muodoissa. Runsaimmin ohjelmistoja on kehitetty sulautettuihin järjestelmiin, useita osaksi projektitoimitusta ja muutamia ohjelmistotuotteiksi. Myynnin rakenne muistuttaa alaspäin suppenevaa kolmiota, jossa ylhäällä ovat sulautetut järjestelmät, keskellä ohjelmistoprojektit ja alhaalla ohjelmistotuotteet, kuva 13. Luonnollisesti tässä luokittelussa on poikkeuksia ja eri ohjelmistomyynnin alueet ovat osittain päällekkäisiä.

Syitä kuvattuun myynnin rakenteeseen ovat ainakin seuraavat. Koska sulautettujen järjestelmien tuotanto Suomessa on suurta, syntyy luonnostaan eniten oppivien ja älykkäiden järjestelmien sovelluksia, samoin kohtuullinen määrä projektitoimituksiin ja muutamia ohjelmistotuotteiksi. Sovellusten määrä heijastaa myös yritysten kokoa. Suuryritykset tekevät sulautettuja järjestelmiä, keskikokoiset yritykset tuottavat projektimyyntiin ohjelmistoja ja pienyritykset pyrkivät tuotemyyntiin. Kaupallistamiseen vaikuttavat maan teollinen tausta, pääoman saatavuus, saavutettavat markkinat ja osaaminen. Vienti on selvästi suurinta sulautettujen ohjelmistojen ja teollisuuden järjestelmätoimitusten muodossa. Ohjelmistotuotteiden vientipotentiaali on merkittävä, mutta vientiä on ollut monesti vaikea toteuttaa pienyritysten voimavaroin.

Ohjelmistoprojektien vienti on vähäistä ja kasvu edellyttää yritysten laajentumista lähelle projektiviennin markkinoita.



Kuva 13. Ohjelmistoperustaisen liiketoiminnan rakenne Suomessa.

4.9.2 Kaupallistamisen mahdollisuuksia ja vaikeuksia

Oppivien ja älykkäiden järjestelmien markkinat laajentuvat noin 30 %:n vuosivauhdilla (Cognizer Almanac). Näin tarjoutuu hyvä mahdollisuus päästä mukaan kasvuun. Teknologiaohjelman aikana alalle on syntynyt 11 uutta yritystä. Myös jatkossa uusien sovellusten syntyminen noudattaa oletettavasti olemassa olevaa rakennetta eli eniten uutta syntyy sulautetuissa järjestelmissä, kohtuullisesti ohjelmistoprojekteissa ja muutamia uusia ohjelmistotuotteita. Oppivat ja älykkäät järjestelmät tarjoavat runsaasti mahdollisuuksia.

Uusien ohjelmistotuotteiden syntyä jarruttavia tekijöitä ovat kustannukset, joiden voi arvioida ainakin kymmenkertaistuvan edelliseen vaiheeseen verrattuna siirryttäessä vaiheesta toiseen ketjussa tutkimus - tuotekehitys - myynti. Jaruttavaa on myös tuoteperinteen puute, joka näkyy esimerkiksi monimutkaisissa ja epästandardeissa käyttöliittymissä, heikkotasoisissa opaskirjoissa, teknisesti monimutkaisena ratkaisuna, toimintavirheinä, liian kevyenä testuksena, runsaina uusina piirteinä sekä tuotemyynnin ja palvelun osaamattomuutena. Edelleen kotimarkkinat Suomessa ovat pienet, mikä johtaa siihen, että pienyritysten on vaikea kerätä tuotekehitysrahaa kotimarkkinoilta.

Vaiheittain kasvavat kustannukset, tuoteperinteen puutteesta johtuvat syyt ja pienet tulot kotimarkkinoilta aiheuttavat vaikeuksia ohjelmistotuotteita tekeville yrityksille.

4.9.3 Ratkaisumalleja

Ketjussa tutkimus - tuotekehitys - myynti vaiheittain kasvavia kustannuksia voidaan hillitä siten, että kukin vaihe tuottaa suoraan hyödynnettävän tuloksen seuraavalle vaiheelle. Menetelmänä on yritysten ja tutkimuslaitosten yhteistyö ja verkottuminen. Tuoteperinteen puutetta voidaan korjata koulutuksella ja osaamisen lisäämisellä sekä teollisuusstandardien käytöllä ja yhteistyöllä. Tuotteistamisessa onnistuneet yritykset voivat opettaa uusia yrittäjiä. Kotimarkkinoilta hankittu hyvä kokemus ja referenssit ovat tärkeitä viennissä. Kotimarkkinoiden pienuutta korvaavia tekijöitä ovat valtion ja EU:n yritysavustukset ja projektit sekä Tekesin ja Suomen Ulkomaankauppaliiton palvelut ja Kauppa- ja teollisuusministeriön (KTM) vientituki, samoin kuin yritysten yhteistyö ja verkottuminen niin ulkomaisten kuin kotimaisten yritysten kanssa. Vientiin voi päästä myös kotimaisen vientiyrityksen mukana. Edelleen vientiä voi auttaa ulkomaisen riskisijoittajan, omistajan tai myyntikanavan käyttö, mikä tuo osaamista ja yhteyksiä.

Teknologiaohjelmalla on vaikutusta kaikkiin esitettyihin ratkaisuihin. Se tukee tutkimusta ja tuotekehitystä, yhteistyötä ja verkottumista sekä osaamisen lisäämistä ja kansinvälistä yhteistyötä. Teknologiaohjelman tavoite on, että siinä kehitettyjä työvälineitä ja tukipalveluja käytetään ohjelman hankkeissa ja näin luodaan pohjaa työvälineiden kaupallistamiselle. Koulutus sekä tuotteiden myynti ja vienti eivät kuulu teknologiaohjelman alueeseen, mutta koulutukselle voidaan esittää tarpeita yliopisto- ja korkeakoulu yhteistyön kautta ja yrityksiä voidaan aktivoida verkottumaan myyntiä ja vientiä varten.

5. Case-tapaukset

5.1 Meesauunin sumea säätö

Yleistä

Oulun yliopiston prosessitekniikan ja Wisaforest Oy:n (nykyisin Kymmene Oy Pietarsaaren tehtaat) yhteistyönä toteutettiin SUSSU-projektissa (Tekes-projekti) yksi ensimmäisistä sumeista säätimistä selluteollisuudessa Suomessa. Säädön kehityksestä vastasi kolme henkilöä, joiden työnjako oli sovittu etukäteen. Tehtävät olivat tarvittavan tietämyksen keruu sääntökannan muodostamiseksi, sumean säätimen suunnittelu ja toteutus automaatiojärjestelmään sekä tarvittavan käyttöliittymän toteuttaminen. Sovellus integroitiin ALCONT II -järjestelmään.

Prototyypin kehitys

Kehitetty meesauunin sumea säätö perustuu täysin kerättyyn prosessitietämykseen ja käyttäjien kokemuksiin. Tämän tiedon avulla valittiin keskeisimmät mittaukset, joiden perusteella säätötoimet suoritetaan. Absoluuttisten mittausarvojen sijasta käytettiin eräillä mittauksilla derivaattoja kuvaamaan tapahtunutta muutosta. Alkuvaiheessa säädin käsitti n. 50 sääntöä, joissa oli päällekkäisyyksiä. Päällekkäisyydet selvisivät käymällä säännöt läpi mittausdatalla sekä lajittelemalla niitä järjestykseen.

Käytetty päättelymekanismi on perinteinen max-min-menetelmä. Tämä takaa hyvät päivitysmahdollisuudet vaihdettaessa mm. uuteen työkaluun. Sovellusta tehdessä ei ollut valmiita ratkaisuja sumean logiikan toteuttamisessa automaatiojärjestelmään. Myöhemmin tuon automaatiojärjestelmän toimittajan taholta tehdyt sumean logiikan ratkaisut eivät poikkeaa suuresti SUSSU-projektin ratkaisuista.

Prototyypin testaus ja viritys

Testaus suoritettiin kahdessa vaiheessa. Ensin oli staattinen testaus toteutuneiden mittausten ja ohjausten perusteella. Siinä selvitettiin sääntöjen käyttäytymistä sekä niiden mahdollisia päällekkäisyyksiä ja ristiriitaisuuksia. Samalla tutkittiin mittausten suodatusten ja muutosten määrittämisen riittävyttä. Toisessa vaiheessa suoritettiin dynaaminen testaus off-line- ja on-line-käytössä. Tällöin säädin oli integroituna järjestelmään. Suoritetun testauksen ajalta kerättiin jatkuvasti mittauksia mahdollisten virheiden määrittämiseksi ja jäljittämiseksi.

Prototyypin integrointi

Tarvittava laskenta kehitettiin Matlab-ympäristössä ja muutettiin ALCONT II -järjestelmän käyttämälle Pascal-kielelle. Tarvittavasta laskennasta muodostettiin lohkot automaatiojärjestelmän ohjelmointia varten. Lisäksi tehtiin tarvittavat lisäykset ja muutokset käyttöliittymään, jotta säädön valvonta ja ylläpito olisivat mahdollisimman helppoja. Käytettävissä ovat ns. pop-up -ikkunat, jotka tarjosivat miellyttävän tavan esittää mm. käytetyt parametrit ja säännöt. Sääntöikkuna esitti kaikki käytetyt säännöt sanallisessa muodossa sekä kunkin säännön laukeamisasteen. Tämän ikkunan avulla voitiin nähdä suoraan säätimen tekemän toimenpiteen perustelut.

Sovelluksen ylläpito

Tehty sovellus meesauunin lämpötilojen ja happipitoisuuden hallintaan kattoi vain osan prosessin hallintaan tarvittavasta säädöstä. Ensimmäisen vaiheen jälkeen projektia on jatkettu laajentamalla ohjausta koko prosessiin. Syy ensimmäisen vaiheen rajaukseen oli prosessissa käytettyjen erillisten automaatiojärjestelmien välisen tiedonsiirron rajallisuus. Järjestelmämuutosten jälkeen ohjauksen kehitys on jatkunut yhteisprojektina ja keskittynyt prosessin kokonaishallintaan.

5.2 Sumean logiikan soveltaminen selluteollisuuteen

Yleistä

Valmet Automation Oy on kehittänyt ja toteuttanut sumean säädön ratkaisuja vuodesta 1991 alkaen. Ajatus siitä, että mahdollisimman paljon säätimen viritys- ja kehitystyöstä voidaan tehdä suoraan automaatiojärjestelmän näytöiltä, on ollut tavoitteena tarvittavien ohjelmamoduulien ja käyttöliittymien suunnittelussa. Nykyisin säädin voidaan peruskonfigurointia muuttamatta suunnitella Damatic XD -järjestelmän näytöltä.

Selluteollisuuteen tehtyjä tuotteita, joissa hyödynnetään sumeaa logiikkaa, ovat mm. soodakattilan keon ja tulipesän lämpötilaprofiilin hallinta, valkaisun dioksidivaiheen säätö, meesauunin ohjaus ja haihduttamon optimointi sekä jätevesilaitoksen ilmastointi.

Prototyypin kehitys

Perusedellytys sovelluksen rakentamiselle on ongelmakohteen erinomainen perustuntemus. Asiakkaiden prosessit eivät koskaan ole aivan identtisiä jolloin perustietämystä joudutaan kasvattamaan asiakkaan prosessin erityisillä piirteillä. Huolellisen ongelmakartoituksen kautta ryhdytään suunnittelemaan toteutusta.

Kehitystyössä suunnitellaan ensin huolellisesti paperille. Suunnittelijan on päätettävä mm. sääntökannan tulomuuttujien maksimimäärä, niiden positiot ja taulukoitava mittaukset. Sen jälkeen voidaan sovellusmoduulit asentaa järjestelmään ja aloittaa suunnittelutyö näytöltä. Tulomuuttujia voidaan valita ylimäärin lopullisen sovelluksen siitä kärsimättä. Yleensä suunnittelutyö etenee suurinpiirtein seuraavasti:

- Kaikkien tulomuuttujien eli mittausten valinta ja taulukointi
- Tulomuuttujien nimeäminen
- Tulomuuttujien jäsenyysfunktioiden karkea määrittely
- Tulomuuttujien jäsenyysfunktioiden nimeäminen
- Ohjauksen jäsenyysfunktioiden karkea määrittely
- Ohjauksen jäsenyysfunktioiden nimeäminen
- Tulomuuttujien valinta sääntökantaan
- Sääntökannan teko.

Sääntökannan muodostaminen tapahtuu täyttämällä taulukkoa Damatic XD -järjestelmän operointinäytöltä. Sääntö muodostuu taulukosta valituista mittauksista ja niiden jäsenyysfunktioista. Säännön ei tarvitse sisältää kaikkia muuttujia, vaan tarpeettomat jätetään pois. Ohjelmamoduulit käyttävät perinteisiä kolmio- tai trapetsoidi-jäsenyysfunktioita sekä Mamdanin min-max-päättelemekanismia.

Prototyypin testaus ja viritys

Viritystyössä hyödynnetään virittäjän oman kokemuksen lisäksi operaattoreiden ja muun käyttöhenkilökunnan kokemuksia prosessista. Kehitettyä säädintä on mahdollista testata ja virittää off- ja on-line-toiminnassa. Testaaja tai operaattori voi valita off- tai on-line-toiminnon suoraan operointipäätteeltä. Muuttujien käyttäytymistä hän seuraa konfiguroitavista trendi-ikkunoista, joihin halutut muuttujat voidaan asettaa.

Nykyään on ollut pyrkimyksenä nopeuttaa säätimien hienoviritystä ja käyttöönottoa. Toimivaksi ratkaisuksi on osoittautunut neuroverkkoanalyysi, jota varten on kehitetty erillinen neuroverkkotyökalu. Tällä työkalulla voidaan prosessianalyysin lisäksi luoda integroituja neuroverkkosovelluksia Damatic XD -järjestelmään.

Prototyypin integrointi

Damatic XD:ssä kehitettyä järjestelmää ei tarvitse integroida, koska toteutus on tehty suoraan integrointiympäristöön. Kehitetty sovellus on avoin siinä mielessä, että kaikki sovelluksen parametrit ovat näkyviä niitä haluavalle. Tällöin voidaan esim. kopioida parametrit toiseen työkaluun ja suorittaa vastaavat laskennat myös siinä.

Sovelluksen ylläpito

Sovellus on toteutettu ja konfiguroitu suoraan automaatiojärjestelmään. Sen ylläpidon suorittaa instrumentointihenkilöstö kuten muunkin automaation. Erillistä koulutusta sovelluksesta annetaan niille henkilöille, joiden vastuualueesta on kyse.

5.3 Neuroverkkojen soveltaminen paperiteollisuuteen

Yleistä

Lappeenrannassa sijaitseva Corintec Oy on kehittänyt neuroverkkosovelluksia paperi- ja selluteollisuuteen. Se käyttää yhdessä Taipale Engineering Oy:n kanssa kehittämäänsä Model-CC työkalua, joka pohjautuu kappaleessa 4.8.1 esitettyyn Q-Opt-työkaluun. Model-CC toimii tehokkaassa Unix-työasemassa ja sillä on graafinen käyttöliittymä. Järjestelmä liitetään tehtaan muihin automaatio- ja tietojärjestelmiin.

Prototyypin kehitys

Model-CC kykenee hyödyntämään useanlaista tietämystä prosessista. Tästä syystä prosessin tunteminen ja oikeiden mittausten valitseminen on hyvin keskeisellä sijalla sovellusta kehitettäessä. Hyödynnettäviin tietoihin kuuluvat kerätty mittaustietäminen, prosessin rakenteen ymmärtäminen, fysikaaliset ominaisuudet ja mittausten väliset riippuvuudet sääntöjen muodossa. Näitä tietoja yhdessä käyttämällä voidaan muodostaa prosessista neuroverkkomalli.

Esikäsittelyllä on keskeinen sija ennen neuroverkon opettamista. Esikäsittelyyn kuuluu mm. mittausten keskinäisten lineaaristen ja epälineaaristen riippuvuuksien testaaminen, antureiden kohinan ja ryöminnän tarkastelu, puuttuvan ja väärän datan hallinta sekä mittausten keskinäisen ajoituksen kohdistaminen. Esikäsittelyllä aineistolla opetetaan off-line-malli, jonka keskeinen analyysitoiminto on herkkyysanalyysi. Siitä saadaan hyvä graafinen esitys kahden valitun muuttujan välille.

Prototyypin testaus ja viritys

Kun kehitetty off-line-malli on saatu muodostettua, se voidaan siirtää on-line-toimintaan tehtaalle. Tällöin joudutaan rakentamaan liittynät prosessiautomaatioon tarvittavien mittausten saamiseksi reaaliaikaisesti. On-line-toiminta tarkoittaa, että malli ennustaa haluttuja tietoja prosessista ja esittää ne käyttäjälle graafisissa kuvissa. Nämä tiedot ovat usein sellaisia, jotka on mahdollista mitata vasta tuotteen valmistuttua. Viritukseen kuuluu oleellisena osana tehdyn neuroverkkomallin päivittäminen. Päivitys tapahtuu

automaattisesti uusilla prosessi- ja laboratoriotiedoilla sekä kerätyn mittaushistorian avulla. Tällä tavalla malli mukautuu prosessin muutoksiin.

Integrointi kohdejärjestelmään

Tehtyä mallia ei tarvitse erikseen integroida, koska kehitysympäristönä toimiva työasema soveltuu suoraan annettuun tehtävään. Integroinnilla voidaankin käsittää enemmän itse sovelluksen käyttökoulutuksen ja -ohjauksen antamista kuin itse laitetekniikkaa.

6. Yhteenveto

Oppivan ja älykkään sovelluksen kehitykseen sopivan työkalun valinta on yleensä useiden asioiden summa ja kompromissi. Keskeisimpinä tekijöinä valinnassa ovat kehittäjien aiemmat kokemukset työkaluista, sovelluksen lopullinen integrointiympäristö sekä sovelluksen ratkaisuun käytetyt menetelmät. Näistä tekijöistä kokemuksen määrittäminen on helpoin ja sillä on yleensä hyvin suuri merkitys työkalun valinnassa. Vaikutus voi olla positiivinen tai negatiivinen. Yhä useammalla laitevalmistajalla on tarjolla omia ratkaisuja sumean logiikan ja neuroverkkojen liittämiseen omiin järjestelmiinsä. Tämä tieto on usein arvokasta ja saattaa helpottaa ja nopeuttaa kehitystä oleellisesti. Laitekohtaiset työkaluratkaisut on rajattu tämän selvityksen ulkopuolelle, mutta niitä kannattaa tiedustella laitevalmistajilta. Projektin alkuvaiheessa on mietittävä menetelmän sopivuutta ongelman ratkaisuun, mutta missään vaiheessa ei ole syytä unohtaa muita menetelmiä tai osaratkaisuja. Työkalua valittaessa ei sovi unohtaa, että kehitystyössä voidaan käyttää useaa eri työkalua eri vaiheissa. Tehdyn sovelluksen siirtäminen työkalusta toiseen ei useinkaan ole kovin suuri työ. Tällöin voidaan jokaisesta työkalusta hyödyntää sen ominaisuudet maksimaalisesti.

Katsaukseen valitut työkalut edustavat tämän hetken tilannetta kaupallisten ohjelmistojen osalta. Selvä trendi on, että työkalut keskittyvät Windows-ympäristöön joitain poikkeuksia lukuunottamatta. Samalla valmistajat panostavat yhä enemmän käytettävyyteen. Joistain neuroverkkotyökaluista on edelleen tarjolla työasemaversioita parempaan laskentakapasiteettiin pyrittäessä. Myöskään client/server -tyyppisiä ratkaisuja ei tule unohtaa tietoverkkojen kasvaessa.

Katsauksen jokaisen ohjelman käyttö vaatii opettelua ja perehtymistä ennen kun kaikki ohjelman hyödyt saadaan esiin. Yhteen menetelmään erikoistuneissa ohjelmissa on käyttöliittymät kehitetty helppokäyttöisiksi verrattuna ohjelmiin, joissa menetelmiä on useita. Näitä erikoistuneita ohjelmia katsauksessa ovat fuzzyTECH, TILShell ja BrainMaker. Niiden eduksi voidaan laskea kyky hyödyntää Windows-ohjelmien tyypillistä toimintaa ja ulkoasua.

Kehitettäessä uutta automaatiota joudutaan keräämään runsaasti mittauksia kehitystä ja testausta varten. Jo sovelluskehityksen alkuvaiheessa datan esikäsittely korostuu tärkeäksi osaksi projektia. Nykyiset sumean logiikan ja neuroverkkojen työkalut eivät sisällä riittäviä ominaisuuksia esikäsittelyyn, vaan joudutaan hankkimaan työkaluihin liitettäviä lisämoduuleja tai erillisiä ohjelmia. Tämä lienee keskeisin puute nykyisissä työkaluissa. Kartoituksen ohjelmista olivat vahvimmat esikäsittelyssä DataEngine ja BrainMaker. fuzzyTech ja NeuralWorks nousevat vahvoiksi, jos niihin hankitaan tarjolla olevat lisämoduulit ja apuohjelmat.

Sovelluksen rakentaminen on verrattain erityyppistä kehitettäessä sumean logiikan tai neuroverkkosovellusta. Sumean logiikan kehittäminen vaatii hyvät ja visuaaliset sääntö- ja muuttujaeditorit, neuroverkkokehitys on enemmänkin kiinni käytetyistä menetelmistä ja parametreista. FuzzyTechissä ja TILShellissä sumean logiikan visuaalisuus ja editointiominaisuudet ovat viety pisimmälle. Neuroverkkotyökaluissa laajin menetelmäkirjo on NeuralWorksissä. Valikoima saattaa tuntua jopa liian laajalta ja vaatiikin hieman totuttelua ja tutustumista kaikkien parametrien ja toimintojen selvittämiseen.

Sovelluksen kehittämisessä oleellinen osa on saadun ratkaisun analysointi. Käytetyimmät menetelmät analysoinnissa ovat herkkyysanalyysi ja simulointi. Hyvin harvasta ongelmasta on olemassa matemaattista mallia ja siksi simulointiominaisuuksien käyttö rajoittuu pieneksi simuloinnin varsinaisessa mielessä. Yleensä se korvataan vertaamalla toteutuneita ja laskettuja tuloksia mittausdatan avulla. Kaikki katsauksen työkalut soveltuvat hyvin tähän tehtävään. Herkkyysanalyysi tehdään sumean logiikan ohjelmissa yleensä käyttämällä 3D-pintakuvaa ja neuroverkkohjelmissa numeerista raporttia.

Kehitetyn sovelluksen sulautus suoritetaan lähes aina C-kielellä tai vastaavalla ohjelmointikielellä. Useaan järjestelmään on jo saatavissa valmiit apuvälineet laskennan toteuttamiseksi jolloin jäljelle jää parametrien siirtäminen kohdejärjestelmään. Tämä helpottaa suuresti varsinkin ylläpitoa.

Kotimaisten sumean logiikan ja neuroverkkotyökalujen kehitys lähti liikkeelle 1990-luvun alussa samalla kun kiinnostus muutenkin kasvoi tutkijoiden piirissä. Työkalut ovat peräisin yliopistoissa ja korkeakouluissa tehdyistä tutkimuksista, joissa tarkoitus on ollut tutustua aiheeseen. Tutkimuksen edetessä työkaluihin on kertynyt menetelmiä tutkijoiden omien kiinnostusten mukaan. Kaikissa Suomessa kehitetyissä sumean logiikan työkaluissa on jokin dataan perustuva viritysmenetelmä, jolla pyritään helpottamaan ja parantamaan sovelluksen kehitystä.

Sumean logiikan ja neuroverkkosovellutusten kehittäminen jonkin yleisesti standardoidun kehitysprosessin mukaan on vielä toteutumatta. Toteutuessaan se helpottaa sovellusprojektin jakamista osiin eri kehitysorganisaatioiden kesken. Samalla se voi nopeuttaa kehitystä ja saada yhä useampia tahoja kiinnostuneeksi aiheesta.

Lähdeluettelo

Berg, P. 1996. Toteutuskokemuksia Oppivien ja älykkäiden järjestelmien sovellukset -ohjelman hankkeista. Espoo: Valtion teknillinen tutkimuskeskus. 28 s. + liitt. 12 s. VTT Tiedotteita - Meddelanden - Research Notes 1746, ISBN 951-38-4901-5.

Dayhoff, J. E. 1990. Neural Network Architectures: An Introduction. New York: Van Nostrand Reinhold. 259 s. ISBN 0-442-20744-1.

Driankov, D., Hellendoorn, H. & Reinfrank, M. 1993. An Introduction to Fuzzy Control. Berlin: Springer-Verlag. 316 s. ISBN 3-540-56362-8.

Haykin, S. 1994. Neural Networks: a Comprehensive Foundation. New York: Macmillan. 696 s. ISBN 0-02-352761-7.

Hellendoorn, H. 1993. Design and Development of Fuzzy Systems at Siemens R&D. Proceedings of FUZZ-IEEE'93, San Francisco, CA, USA, March 28 - April 1, 1993, pp. 1365-1370.

Inform 1996. NeuroFuzzy & FuzzyCluster Module. <http://www.fuzzytech.com/>

Isomursu, P., Niskanen, V., Carlsson, C. ja Eklund, P. 1993. Sumean logiikan mahdollisuudet. Helsinki: Teknologian kehittämiskeskus. 101 s. TEKES-julkaisu 34/93, ISBN 951-47-1930-1.

Isomursu, P. 1995. A software engineering approach to the development of fuzzy control systems. Espoo: Valtion teknillinen tutkimuskeskus. 79 s. + app. 55 s. VTT Publications 230, ISBN 951-38-4768-3.

Koikkalainen, P. (toim.) 1994. Neurolaskennan mahdollisuudet. Helsinki: Teknologian kehittämiskeskus. 151 s. TEKES-julkaisu 43/94, ISBN 951-47-1950-6

Kosko, B. 1992. Neural Networks and Fuzzy Systems. Prentice-Hall International. 449 s. ISBN 0-13-611435-0.

Zimmerman, H-J. 1991. Fuzzy Set Theory - and Its Applications. 2nd, rev. ed. The Netherlands: Kluwer Academic Publishers. 399 s. ISBN 0-7923-9075-X.







Liite A: Yhteystietoja Suomessa


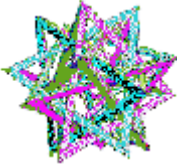







Matlab	Computer Solutions Europe Oy Tekniikantie 4, 02150 Espoo Puh: (09) 455 0055 Fax: (09) 455 0051 WWW: http://www.comsol.fi E-Mail: info@comsol.fi
Matlab	Control Cad Oy Tekniikantie 21, 02150 ESPOO Puh: (09) 437 5504 Fax: (09) 437 5506 WWW: http://www.ccad.fi/ E-Mail: control.cad@ccad.fi
BrainMaker	TVC ohjelmistotalo Kaustintie 1, 69600 Kaustinen Puh: (06) 8612 424 Fax: (06) 8612 425 E-Mail: tvk@kaustinen.kpnet.fi
G2	Control Software Oy Veneentekijäntie 4, 00210 Helsinki Puh: (09) 67 6744 Fax: (09) 67 0077 E-Mail: control.software@csoy.fi
AboaFuzz	FF-Automation Oy Kierretie 20, 01650 Vantaa Puh: (09) 852 4700 Fax: (09) 4707 WWW: http://www.ff-automation.com/ E-Mail: info@ffauto.inet.fi
Q-Opt	Taipale Engineering Oy Laserkatu 6, 53850 Lappeenranta Puh: (05) 624 3114 tai (049) 752 605 Fax: (05) 412 0949 WWW: http://www.kareltek.fi/taipale E-Mail: ossi.taipale@lut.fi tai ossi.taipale@kareltek.fi
Neural-MILL	Control Express Finland Ltd. Simasalonkatu 4, PL 37, 57200 Savonlinna Puh: (015) 520 620 Fax: (015) 520 629

Liite B: Hintatietoja





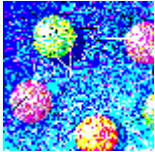

Ohjelmisto	Versio	Hinta
fuzzyTECH	Precompiler Edition	DM 4 600
	Online Edition (sis. NF & FC Module)	DM 12 5000
	NeuroFuzzy & FuzzyCluster Module	DM 2 000
	DataAnalyzer Module	DM 2 900
TILShell	TILShell Basic Edition	US\$ 550
	Standard Edition	US\$ 1 650
	Professional Edition	US\$ 2 650
Matlab	Matlab -perusohjelmisto	FIM 15 000
	Fuzzy Logic Toolbox	FIM 8 000
	Neural Network Toolbox	FIM 8 000
	Simulink	FIM 16 000
	Compiler	FIM 10 000
DataEngine	DataEngine -perusohjelmisto	DM 6000
	V.i. add-on for LabView	DM 2500
	ADL Library	DM 1000
NeuralWorks	Professional II/Plus	US\$ 2 300
	Designer Pack	US\$ 2 400
	DataSculptor	US\$ 600
BrainMaker	BrainMaker -perusohjelmisto	US\$ 200
	Professional	US\$ 800
G2	G2 -peruspaketti	FIM 200 000
	G2 + NeurOn-Line	FIM 300 000






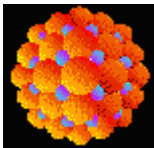



Liite C: Sumean logiikan työkaluvalmistajien yhteystietoja

<p>CubiCalc</p> 	<p>HyperLogic Corporation 456 East Grand Avenue, Suite 304 P.O.Box 300010, Escondido, CA 92030-0010, USA Puh: (619) 746-2765 Fax: (619) 746-4089 WWW: http://www.electriciti.com/hl/ E-Mail: prodinfo@hyperlogic.com</p>
<p>DataEngine</p> 	<p>MIT GmbH Promenade 9, 52076 Aachen, Germany Puh: +49/24 08/9 45 80 Fax: +49/24 08/9 45 82 WWW: http://ss-m3.mitgmbh.de/ E-Mail: info@mitgmbh.de</p>
<p>Fuzzy Control Package</p> 	<p>Bytronic International Ltd. The Courtyard, Reddicap Trading Estate, Sutton Coldfield, West Midlands, B75 7BU, England Puh: +44 (0)121 378 0613 Fax: +44 (0)121 311 1774 WWW: http://www.bytronic.co.uk E-Mail: sales@bytronic.co.uk</p>
<p>Fide</p>	<p>Aptronicx, Inc. 2150 North First Street, Ste. 300 San Jose, CA 95131, USA Puh: (408) 428-1888 Fax: (408) 428-1884 E-mail: parish@cactus.org</p>
<p>FLINT</p> 	<p>Logic Programming Associates Ltd. Studio 4, Royal Victoria Patriotic Building Trinity Road, London, SW18 3SX, UK Puh: +44 181 871 2016 Fax: +44 181 874 0449 WWW: http://www.lpa.co.uk E-Mail: lpa@cix.compulink.co.uk</p>
<p>FuziCalc</p>	<p>FuziWare, Inc. P.O.Box 11287, Knoxville, TN 37939-1287, USA Puh: (615) 588-4144 Fax: (615) 588-9487</p>
<p>FUZZLE</p> 	<p>MODiCo, Inc. P.O.Box 8485, Knoxville, TN 37996-0002, USA Puh: (423) 584-4915 Fax: (423) 584-4934 WWW: http://users.aol.com/fuzzle20 E-Mail: modico@aol.com</p>
<p>FuzzyShell</p> 	<p>ICCT Technologies, Inc. 500 Alden Road, Suite 205, Markham, Ontario, L3R 5H5, Canada Puh: (905) 475-9921 Fax: (905) 475-5350 WWW: http://www.stria.ca/ICCT/ E-Mail: anton@inforamp.net</p>
<p>FuzzySoft</p>	<p>GTS Trautzl GmbH Gottlieb-Daimler-Str.9, W-2358 Kaltenkirchen/Hamburg, Germany Puh: (49) 4191 88665 Fax: (49) 4191 8711</p>

<p>fuzzyTECH</p> 	<p>INFORM GmbH Pascalstrasse 23, D-52076 Aachen, Germany Puh: +49-2408-9456-80 Fax: +49-2408-9456-85 WWW: http://www.fuzzytech.com E-Mail: hotline@inform-ac.com</p>
<p>Mathematica Fuzzy Logic Pack</p> 	<p>Wolfram Research, Inc. P.O.Box 6059, Champaign, IL 61826-6059, USA Puh: (217) 398-0700 Fax: (217) 398-0747 WWW: http://www.wri.com/ E-Mail: info@wolfram.com</p>
<p>Matlab Fuzzy Logic Toolbox</p> 	<p>MathWorks, Inc. 24 Prime Park Way, Natick, MA 01760-1500, USA Puh: (508) 653-1415 Fax: (508) 653-6284 WWW: http://www.mathworks.com/ E-Mail: info@mathworks.com</p>
<p>METUS Fuzzy Logic</p> 	<p>The Metus Systems Group 1 Griggs Lane, Chappaqua, NY 10514, USA Puh: (914) 238-0647 Fax: (914) 238-0837 WWW: http://www.paltech.com/metus/metus.htm E-Mail: ecox@paltech.com</p>
<p>NeuFuz4</p> 	<p>National Semiconductor Corp. 2900 Semiconductor Drive, P.O.Box 58090, Santa Clara, CA 95052-8090, USA Puh: 1(800) 272-9959 Fax: (910) 339-9240 WWW: http://www.nsc.com/ E-Mail:</p>
<p>O'INCA Design Framework</p> 	<p>Intelligent Machines, Inc. 1153 Bordeaux Drive, Sunnyvale, CA 94089, USA Puh: (408) 745-0881 Fax: (408) 745-6408 WWW: http://www.ppgsoft.com/ppgsoft/oi_inc.html E-Mail: mail@ppgsoft.com</p>
<p>FUZZYSTUDIO</p> 	<p>SGS-THOMSON Microelectronics WWW: http://www.st.com/ E-Mail:</p>
<p>SieFuzzy</p> 	<p>Siemens AG ANL A441-VE Wolfgang Keim, P.O.Box 3220, 91052 Erlangen, Germany Puh: +49 9131 7 22885 Fax: +49 9131 7 32127 WWW: http://www.siemens.de/research/siefuzzy/ E-Mail: siefuzzy@zfe.siemens.de</p>
<p>TILShell</p> 	<p>Togai InfraLogic, Inc. 18 Technology Drive, Suite 146, Irvine, CA 92718, USA Puh: (714) 588-3800 Fax: (714) 588-3808 WWW: http://www.ortech-engr.com/fuzzy/togai.html E-Mail: togai@aol.com</p>

Liite D: Neuroverkkotyökaluvalmistajien yhteystietoja

Brain Maker 	California Scientific Software 10024 Newton Road, Nevada City, CA 95959, USA Puh: (800) 284-8112, (916) 478-9040 Fax: (916) 478-9041 WWW: http://www.calsci.com/ E-Mail: sales@calsci.com
EasyNet	Software Frontiers Systems, Inc. P.O.Box 8524, Mesa, AZ 85214, USA Puh: (602) 985-8550 Fax: (602) 985-8550
FleFTool 	Flexible Intelligence Group, LLC. Box 1477, Tuscaloosa, AL 35486-1477, USA Puh: (205) 345-5166 Fax: (205) 345-5095 WWW: http://www.flextool.com/ E-Mail: order@flextool.com
HNeT	ITANIS International, Inc. 1737 Holly Lane, Pittsburgh, PA 15216-1151, USA Puh: (412) 344-4553 Fax: (412) 344-3541
Matlab Neural Network Toolbox 	The MathWorks, Inc. 24 Prime Park Way, Natick, MA 01760-1500, USA Puh: (508) 653-1415 Fax: (508) 653-6284 WWW: http://www.mathworks.com/ E-Mail: info@mathworks.com
NeuFrame 	Neural Computer Sciences Unit 3, Lulworth Business Centre Nutwood Way, Totton, Southampton SO40 3WW, UK Puh: +44 (0) 1703 667775 Fax: +44 (0) 1703 663730 WWW: http://www.demon.co.uk/skylake/
Neural Intelligence	Cogito Software, Inc. P.O.Box 451, Pine River, MN 56474, USA Puh: (218) 587-3120 Fax: (218) 587-3414
Neural Network Utility 	IBM 3605 Highway 52 North Rochester, MN 55901-7829, USA Puh: 1-800-IBM-CALL Fax: 1-800-426-4329 WWW: http://www.research.ibm.com/ E-Mail: nninfo@vnet.ibm.com
NeuralWorks 	NeuralWare, Inc. 202 Park West Drive, Pittsburgh, PA 15275, USA Puh: (412) 787-8222 Fax: (412) 787-8220 WWW: http://www.neuralware.com/ E-Mail: saleswp@neuralware.com
Neuralyst	Cheshire Engineering Corporation 650 Sierra Madre Villa, Suite 201 Pasadena, CA 91107, USA Puh: (818) 351-0209 Fax: (818) 351-8645

<p>NeuroForecaster</p> 	<p>NIBS, Inc. 62 Fowlie Road, Republic of Singapore 1542 Puh: (+65)-344-2357 Fax: (+65)-344-2130 WWW: http://www.singapore.com/products/nfga/</p>
<p>NeuroGenetic Optimizer</p> 	<p>BioComp Systems, Inc. 2871 152nd Avenue N.E., Redmond, WA 98052, USA Puh: (206) 869-6770 Fax: (206) 869-6850 WWW: http://www.bio-comp.com/ E-Mail: biocomp@biocomp.seanet.com</p>
<p>NeuroLab</p> 	<p>Mikuni Berkley R & D Corp. 4000 Lakeside Dr., Richmond, CA 94806, USA Puh: (510) 222-9880 Fax: (510) 222-9884 WWW: http://www.mikuni.com/ E-Mail: neurolab-info@mikuni.com</p>
<p>G2 NeurOn-Line</p> 	<p>Genesym Corporation 125 CambridgePark Drive, Cambridge, MA 02140, USA Puh: (617) 547-2500 Fax: (617) 547-1962 WWW: http://www.gensym.com/</p>
<p>NeuroShell</p> 	<p>Ward Systems Group, Inc. Executive Park West, 5 Hillcrest Dr., Frederick, MD 21702, USA Puh: (301) 622-7950 Fax: (301) 622-5666 WWW: http://www.wardsystems.com/ E-Mail: WardSystems@msn.com</p>
<p>NeuroSolutions</p> 	<p>NeuroDimension, Inc. 1800 N. Main Street, Suite D4, Gainseville, FL 32609, USA Puh: (352) 377-5144 Fax: (352) 377-9009 WWW: http://www.nd.com/ E-Mail: info@nd.com</p>
<p>nn/xnn</p> 	<p>Neureka ANS Klaus Hansens vei 31b, 5037 Solheimsvik, Norway Puh: +47 55 544163 WWW: http://www.bgif.no/neureka/ E-Mail: neureka@bgif.no</p>
<p>O'INCA Design Framework</p> 	<p>Intelligent Machines, Inc. 1153 Bordeaux Drive, Sunnyvale, CA 94089, USA Puh: (408) 745-0881 Fax: (408) 745-6408 WWW: http://www.ppgsoft.com/ppgsoft/oi_inc.html E-Mail: mail@ppgsoft.com</p>
<p>The OWL Neural Network Library</p> 	<p>HyperLogic Corporation PO Box 300010, Escondido, CA 92030-0010, USA Puh: (619) 746-2765 Fax: (619) 746-4089 WWW: http://www.electriciti.com/hl E-Mail: prodinfo@hyperlogic.com</p>

<p style="text-align: center;">Partek</p> <p style="text-align: center;">PARTEK</p>	<p>Partek, Inc. 5988 Mid Rivers Mall Dr, St. Charles, Missouri 63304, USA Tel: (314) 926-2329 Fax: (314) 441-6881 WWW: http://www.partek.com/</p>
<p style="text-align: center;">Qnet</p> 	<p>Vesta Services, Inc. 1001 Green Bay Rd., Winnetka, IL 60093, USA Puh: (847) 446-1655 Fax: (847) 446-1674 WWW: http://members.aol.com/vestaserv/index.html E-Mail: VestaServ@aol.com</p>