

UNIVERSITATEA TEHNICĂ A MOLDOVEI

Cu titlu de manuscris
C.Z.U.: 004.3/004.8

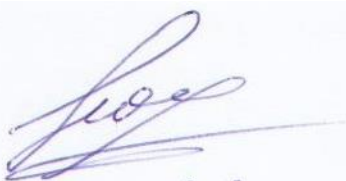
CĂRBUNE VIOREL

**ARHITECTURI HARDWARE ADAPTIVE PENTRU SISTEME NEURO-
FUZZY CU AUTOORGANIZARE**

**232.01. SISTEME DE CONDUCERE, CALCULATOARE ȘI REȚELE
INFORMAȚIONALE**

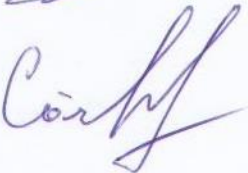
Teză de doctor în științe inginerești

Conducător științific



Zaporozjan Sergiu
dr. în șt. tehnice, conf. univ.

Autor:



Cărbune Viorel

CHIȘINĂU, 2020

©Cărbune Viorel, 2020

CUPRINS

ADNOTĂRI	4
LISTA ABREVIERILOR	7
INTRODUCERE	8
1. ASPECTE GENERALE PRIVIND SISTEMELE INTELIGENTE	15
1.1. Sisteme fuzzy, neuro-fuzzy, rețele neuronale și sisteme cu autoorganizare	15
1.2. Forme de reprezentare și metode de extragere a cunoștințelor	27
1.3. Arhitecturi hardware adaptive	34
1.4. Formularea problemei de cercetare	37
1.5. Concluzii la capitolul 1	43
2. CERCETAREA ȘI DEZVOLTAREA METODELOR ȘI ALGORITMILOR DE LUARE A DECIZIILOR ÎN CONDIȚII DE INCERTITUDINE	44
2.1. Abordarea problemei de identificare a funcțiilor de apartenență	44
2.2. Achiziția, preprocesarea și analiza datelor experimentale	55
2.3. Elaborarea metodelor și algoritmilor de extragere a cunoștințelor expert	66
2.4. Concluzii la capitolul 2	79
3. ELABORAREA ARHITECTURILOR HARDWARE ADAPTIVE	80
3.1. Formularea problemei de implementare a modelelor dezvoltate	80
3.2. Proiectarea arhitecturii pentru sisteme fuzzy cu autoorganizare	80
3.3. Proiectarea arhitecturii pentru sisteme neuronale cu autoorganizare	91
3.4. Simularea și analiza comparativă a soluțiilor arhitectural-structurale propuse	100
3.5. Concluzii la capitolul 3	109
4. APLICAREA TEHNICILOR DE SUPORT DECIZIONAL	110
4.1. Studiu de caz 1: sistem fuzzy pentru procesul de tragere a microfiredelor	110
4.2. Studiu de caz 2: sistem pe bază de rețea neuronală.....	115
4.3. Concluzii la capitolul 4	129
CONCLUZII GENERALE	130
RECOMANDĂRI	131
BIBLIOGRAFIE	132
ANEXE	144
Anexa 1. Descrierea în AHDL a motorului de inferență generic	144
Anexa 2. Descrierea în AHDL a arhitecturilor neuronale generice	147
Anexa 3. Listingul programului de exportare a matricei ponderilor sinaptice	154
Anexa 4. Actul de implementare	155
DECLARAȚIA PRIVIND ASUMAREA RĂSPUNDERII	156
CURRICULUM VITAE	157

Adnotare

la teza „Arhitecturi hardware adaptive pentru sisteme neuro-fuzzy cu autoorganizare” prezentată de către Cărbune Viorel pentru conferirea titlului științific de doctor în științe inginerești, Chișinău, 2020.

Structura tezei. Teza de doctor cuprinde introducerea, patru capitole, concluzii, bibliografia cu 120 titluri, 4 anexe, 131 pagini text de bază, inclusiv 81 figuri și 5 tabele. Rezultatele obținute sunt publicate în 15 lucrări științifice.

Cuvinte cheie: metode de extragere a cunoștințelor, sistem de suport decizional, sistem fuzzy, sistem neuronal, arhitecturi hardware adaptive.

Domeniul de studiu îl constituie studiul aspectelor teoretice și practice ale metodologiilor de extragere a cunoștințelor umane.

Scopul lucrării constă în dezvoltarea unor noi modele, metode și algoritmi de extragere a cunoștințelor expertului uman, elaborarea arhitecturilor hardware adaptive pentru cercetarea proceselor decizionale și construirea sistemelor de suport decizional în aplicații industriale.

Obiectivele cercetării includ analiza aspectelor generale ale sistemelor neuro-fuzzy, metodelor de extragere a cunoștințelor, cercetarea și dezvoltarea metodelor și algoritmilor de suport decizional, proiectarea, simularea și analiza arhitecturilor hardware adaptive.

Noutatea și originalitatea științifică constă în propunerea de noi modele, metode, algoritmi de extragere a cunoștințelor și a structurilor hardware parametrizate. Originalitatea soluțiilor propuse constă în abordarea și îmbinarea tehnicilor inteligente de „machine learning” cu modelul comportamental al operatorului uman.

Problema științifică soluționată constă în elaborarea și cercetarea metodelor originale de preluare a experienței operatorului uman prin colectarea și prelucrarea datelor statistice generate automat, fapt care a condus la un nou mod de abordare în extragerea cunoștințelor.

Semnificația teoretică a lucrării constă în elaborarea și dezvoltarea unor metode originale care pot fi utilizate pentru preluarea experienței operatorului calificat. Abordarea prezentată presupune dezvoltarea unor algoritmi de extragere a cunoștințelor ce rezultă din experiența operatorului uman.

Valoarea aplicativă a lucrării constă în propunerea modelelor, metodelor și algoritmilor de procesare și analiză a datelor referitor la evoluția procesului tehnologic.

Implementarea rezultatelor științifice constă în utilizarea modelelor și sistemelor elaborate în cadrul companiei “Microfir Tehnologii Industriale” S.R.L. Valoarea aplicativă a lucrării se confirmă inclusiv prin actul de implementare a rezultatelor obținute.

Annotation

for the thesis with title “ Adaptive hardware architectures for neuro-fuzzy systems with self-organization”, presented by Cărbune Viorel for conferring a Ph.D. title in engineering sciences, Chişinău, 2020.

Thesis structure. The Ph.D. thesis comprises the introduction, four chapters, conclusions, and bibliography (120 titles), 4 appendixes, 131 pages of main text, 81 figures and 5 tables. The obtained results are published in 15 scientific articles.

Keywords: knowledge extraction methods, decision support system, fuzzy system, neural system, hardware adaptive architectures.

The study domain includes theoretical and practical aspects of human knowledge extraction methodologies.

The purpose of research consists in development of new models, methods and algorithms for extracting human expert knowledge, developing adaptive hardware architectures for researching decision-making processes and building decision support systems in industrial applications.

The research objectives include the analysis of general aspects of neuro-fuzzy systems, knowledge extraction methods, research and development of decision support methods and algorithms, design, simulation and analysis of adaptive hardware architectures.

The scientific novelty consists in proposing new models, methods, algorithms for knowledge extraction and parameterized hardware structures. The originality of the proposed solutions consists in approaching and combining intelligent "machine learning" techniques with the behavioral model of the human operator.

The solved scientific problem resides in elaboration and research of the original methods of human operator experience extraction through collecting and processing of automatically generated statistical data, which led to a new approach in the extraction of knowledge.

The theoretical significance of the work consists in the elaboration and development of original methods that can be used to extract the experience of the qualified operator. The presented approach involves development of algorithms for knowledge extracting from the experience of the human operator.

The applied value consists in: proposing models, methods and algorithms for processing data of technological process evolution.

The implementation of results consists in the use of the elaborated models and systems in “Microfir Tehnologii Industriale” S.R.L. company confirmed by the act of implementation.

Аннотация

диссертации на соискание учёной степени кандидата технических наук с темой
„Адаптивные аппаратные архитектуры для нейро-нечетких систем с
самоорганизацией”, автор Кэрбуне Виорел, Кишинэу 2020

Структура работы. Диссертация состоит из введения, четырёх глав, выводов, библиографии из 120 наименований, 4-х приложений, 131 страниц основного текста, включая 81 рисунков и 5 таблиц. Полученные результаты опубликованы в 15-и работах.

Ключевые слова: методы извлечения знаний, система поддержки принятия решений, нечеткая система, нейронная система, адаптивные аппаратные архитектуры.

Область исследования касается теоретических и практических методов извлечения знаний у оператора.

Целью работы является развитие и разработка новых методов и алгоритмов, позволяющих извлекать и перенимать опыт оператора, разработка адаптивных архитектур для исследования процессов принятия решений и построения систем поддержки принятия решений в промышленных приложениях.

Основные задачи: анализ общих аспектов нечетких и нейронных систем, методов извлечения знаний, исследование и разработка методов и алгоритмов поддержки принятия решений, проектирование, моделирование и анализ адаптивных аппаратных архитектур.

Научная новизна и оригинальность полученных результатов заключается в предложении новых моделей, методов, алгоритмов и параметризованных аппаратных структур. Оригинальность предлагаемых решений состоит в подходе и сочетании интеллектуальных методов «машинного обучения» с поведенческой моделью оператора.

Решённая научная задача заключается в разработке и исследовании оригинальных методов освоения опыта оператора посредством обработки автоматически генерируемых статистических данных, что привело к новому подходу к извлечению знаний.

Теоретическое значение заключается в разработке оригинальных методов, которые могут быть использованы для перенятия опыта квалифицированного оператора. Представленный подход включает разработку алгоритмов извлечения знаний.

Практическая значимость работы заключается в предложении моделей, методов и алгоритмов для обработки и анализа данных об эволюции технологического процесса.

Внедрение научных результатов состоит в использовании разработанных моделей и систем, в компании «Microfir Tehnologii Industriale» S.R.L. и подтверждается актом внедрения.

LISTA ABREVIERILOR

ANFIS – Adaptive Neuro-Fuzzy Inference System
ASIC – Application Specific Integrated Circuit
CPLD – Complex Programmable Logic Device
DSP – Digital Signal Processor
EEPROM – Electrically Erasable Programmable Read-Only Memory
FPGA – Field Programmable Gate Array
IA – Inteligență Artificială
JTAG – Joint Test Action Group
MSE – Mean Squared Error
PTS – presiunea aerului în tubul de sticlă
RAM – Random Access Memory
RN – Rețea Neuronală
RNA – Rețea Neuronală Artificială
RNF – Rețea Neuro-Fuzzy
ROM – Read-Only Memory
RL – rezistența lineară
SBC – Single Board Computer
SNF – Sistem Neuro-Fuzzy
SoC – System on Chip
SRAM – Static Random Access Memory
TSK – Takagi-Sugeno-Kang

INTRODUCERE

Actualitatea temei de cercetare. Pe măsură ce pe piață se manifestă un interes sporit pentru produsele personalizate și cu o varietate bogată, industria tinde să implementeze sisteme de producție care să ofere flexibilitatea necesară, menținând în același timp costurile comparabile cu producția automatizată în serie. Ponderea maximă la formarea costurilor în producția automatizată o constituie forța de muncă umană necesară pentru operațiile de instalare, asistare a proceselor și asigurare a calității. Aceste costuri pot fi recuperate doar în cazul producției în serie foarte mare. În cazul producerii unui lot mic, ca în cazul unui prototip pentru care procesul de producție este foarte personalizat, resursele umane rămân de neînlocuit în ceea ce privește flexibilitatea și costurile sistemelor automate. Prin urmare, obținerea unui nivel de flexibilitate similar cu cel uman poate fi asigurată doar cu ajutorul sistemelor cu un nivel foarte înalt de autonomie [1].

Realitățile sunt de așa natură încât există procese industriale specifice care pot fi caracterizate printr-un grad anumit de ambiguitate (incertitudine) în procesul de luare a deciziilor. Problema de bază în automatizarea proceselor industriale o reprezintă achiziția și structurarea datelor cu scopul antrenării sistemelor inteligente. Procedura de antrenare a sistemelor decizionale necesită un volum destul de mare de date, iar eficiența acestei antrenări depinde direct de calitatea și cantitatea datelor utilizate. Dacă în condițiile procesului continuu de producție cantitatea datelor achiziționate nu reprezintă o problemă, atunci asigurarea calității acestora depinde direct de calificarea și experiența operatorului uman. Trebuie menționat faptul că procesele industriale specifice pot fi caracterizate prin variația parametrilor acestora pe tot parcursul procesului tehnologic. Astfel sarcina sistemelor decizionale inteligente constă în luarea deciziilor care ar asigura precizia parametrilor procesului de producție în limitele specificate de cerințele de calitate. Insuficiența datelor despre starea procesului tehnologic poate provoca scăderea drastică a preciziei parametrilor de calitate. În aceste condiții operatorul uman poate fi ghidat doar de propria experiență, iar îmbunătățirea sau, în cel mai rău caz, menținerea parametrilor procesului tehnologic poate fi asigurată doar datorită specificului decizional al acestuia. Astfel evoluția normală a proceselor tehnologice specifice poate fi asigurată doar datorită calificării înalte a operatorului uman obținute prin acumularea continuă a experienței. În rezultatul aplicării tehnicilor de extragere a cunoștințelor operatorului uman, poate fi preluată experiența acestuia și apoi utilizată pentru antrenarea unor sisteme inteligente cu scopul moștenirii procesului decizional specific.

În anul 2006 a fost fondat la Universitatea Tehnică din Munchen clusterul german de cercetare ”*Cognition for Technical Systems*” cu scopul de a cerceta procesul de cunoaștere umană pentru ca sistemele tehnice să poată prelua performanțele acestuia. În perspectivă sistemele tehnice trebuie să reacționeze în **condiții de incertitudine**, să gestioneze în mod fiabil evenimentele neprevăzute, să se adapteze rapid la schimbarea sarcinilor și capacităților proprii [1, 2]. Elaborarea unor tehnologii inteligente hibride ca bază a realizării sistemelor de luare a deciziilor poate permite sistemului tehnic de a-și planifica propriile acțiuni și de a-și adapta comportamentul în diferite circumstanțe, inclusiv de incertitudine [3-5].

Cercetarea procesului de cunoaștere presupune înțelegerea obiectului cu care se operează. În general, termenul de cunoaștere ține de următoarele aspecte [6]:

- „Natura” cunoașterii care definește ce este cunoașterea și indică dacă aceasta reprezintă un obiect ori un rezultat care poate fi partajat, duplicat și transportat sau este un **proces individual** care este greu de controlat.

- „Disponibilitatea” cunoașterii indică formele în care cunoștințele sunt disponibile și accesibile. Aceasta specifică diferența dintre cunoștințele individuale și cele colective, dintre cunoștințele tacite și cele explicite.

Totodată, se pot evidenția trei perspective ale cunoașterii:

- Perspectiva procesării informațiilor care presupune faptul că cunoștințele și informațiile reprezintă aproximativ același lucru. În acest caz prioritate are viteza de procesare a informației.

- Perspectiva de rețea presupune că cunoștințele sunt rezultatul interacțiunii oamenilor.

- Perspectiva autoreferențială care presupune că cunoașterea este un **proces privat dependent de experiența fiecărui individ** în parte.

Din cele expuse mai sus rezultă existența unui caz particular caracterizat prin faptul că cunoștințele specifice reprezintă rezultatul unui proces individual greu de controlat și dependent de experiența expertului uman. Optimizarea unui proces de producție industrial depinde de activitățile realizate, participanții încadrați și obiectivele scontate [6, 7]. Scopul primar al lucrării constă în extragerea cunoștințelor de la operatorul uman și obținerea modelelor decizionale pentru automatizarea proceselor industriale, iar cel secundar constă în asigurarea unui suport decizional pentru îmbunătățirea indicatorilor de calitate în procesele de producție [8]. În caz general se urmărește scopul cercetării și dezvoltării unor noi metode și modele de extragere a cunoștințelor de la operatorul uman și utilizarea acestora în dezvoltarea sistemelor de suport decizional al proceselor industriale astfel încât algoritmi de luare a deciziilor să poată asigura fie o soluție optimă, fie una acceptabilă practic. Din acest punct de vedere, procesul de extragere

a cunoștințelor cu scopul preluării experienței reprezintă un domeniu specific cu un potențial de cercetare enorm. Totodată, pentru implementarea modelelor sistemelor propuse este importantă proiectarea și elaborarea arhitecturilor hardware adaptive care ar realiza efectiv inferența proceselor decizionale [9].

Domeniul de cercetare. Lucrarea de față are ca domeniu de cercetare studiul aspectelor teoretice și practice ale metodologiilor de extragere a cunoștințelor umane cu scopul utilizării acestora pentru dezvoltarea structurilor cu autoorganizare pentru sisteme de suport decizional.

Ipoteza de cercetare. La baza cercetărilor realizate se află supoziția de utilizare a tehnicilor inteligente în preluarea experienței operatorului uman care pot oferi soluții pentru automatizarea și optimizarea proceselor decizionale în aplicațiile industriale.

Obiectul de cercetare. Modele, metode și algoritmi de extragere a cunoștințelor pentru sisteme de suport decizional în procesele de producție.

Scopul lucrării constă în dezvoltarea unor noi modele, metode și algoritmi de extragere a cunoștințelor expertului uman, elaborarea arhitecturilor hardware adaptive pentru cercetarea proceselor decizionale și construirea sistemelor de suport decizional în aplicații industriale.

Din scopul propus rezultă următoarele **obiective ale cercetării**:

1. Analiza aspectelor generale ale sistemelor neuro-fuzzy, cu autoorganizare, metodelor de extragere a cunoștințelor și arhitecturilor hardware adaptive.
2. Cercetarea, elaborarea și dezvoltarea metodelor și algoritmilor de suport decizional în condiții de incertitudine.
3. Proiectarea arhitecturilor hardware adaptive pentru sisteme hibride de luare a deciziilor.
4. Simularea și analiza comparativă a arhitecturilor hardware adaptive elaborate.
5. Cercetarea posibilităților de realizare practică a structurilor cu autoorganizare pentru sisteme de suport decizional.

Suportul metodologic și teoretico-științific al cercetărilor. Cercetările elaborate sunt bazate pe analiza matematică, metodele numerice, teoria mulțimilor vagi, teoria rețelelor neuronale artificiale, tehnici de achiziție de date și proiectarea circuitelor numerice.

În cadrul lucrărilor de cercetare a fost utilizată metodologia ipotetico-deductivă. Argumentarea utilizării acestei metode reiese din natura experimentală a proceselor studiate și din posibilitatea verificării experimentale a corectitudinii ipotezelor și presupunerilor formulate pe parcursul procesului de cercetare. În cadrul cercetărilor a fost realizată analiza datelor statistice colectate, având ca scop generalizarea procesului studiat.

Noutatea și originalitatea științifică. Au fost elaborate noi modele, metode, algoritmi de extragere a cunoștințelor de la expertul uman, au fost proiectate structuri hardware parametrizate cu posibilitatea de cercetare a proceselor decizionale fuzzy și neuronale, și de construire a sistemelor de suport decizional pentru aplicații industriale. Originalitatea soluțiilor propuse constă în abordarea și îmbinarea tehnicilor inteligente de „machine learning” cu modelul comportamental al operatorului uman.

Problema științifică soluționată constă în elaborarea și cercetarea metodelor originale de preluare a experienței operatorului uman prin colectarea și prelucrarea datelor statistice generate automat de sisteme informaționale de măsurare, fapt care a condus la un nou mod de abordare în extragerea cunoștințelor.

Semnificația teoretică a lucrării o reprezintă elaborarea și dezvoltarea unor metode originale care pot fi utilizate cu succes pentru preluarea experienței operatorului calificat. Experiența operatorului poate fi acumulată într-o bază de cunoștințe. Abordarea prezentată presupune dezvoltarea unor algoritmi de extragere a cunoștințelor ce rezultă din experiența operatorului uman și stocarea acestora. Acest fapt face posibilă cercetarea proceselor decizionale fuzzy și neuronale.

Valoarea aplicativă a lucrării. Modelele, metodele și algoritmi de procesare și analiză a datelor referitor la evoluția procesului tehnologic fac posibilă continuarea cercetărilor în direcția optimizării sistemului de producție în cadrul întreprinderii „Microfir Tehnologii Industriale” SRL din Chișinău.

Rezultatele științifice înaintate spre susținere:

1. Metodă și algoritmi de extragere a cunoștințelor specifice operatorului uman pentru procese decizionale fuzzy și neuronale.
2. Modele Matlab pentru asigurarea proceselor decizionale în condiții de incertitudine.
3. Componente parametrizate și arhitecturi hardware adaptive pentru sisteme fuzzy și neuronale.
4. Sisteme hibride de suport decizional cu autoorganizare pentru aplicații industriale.

Aprobarea rezultatelor lucrării. Rezultatele principale ale lucrării au fost prezentate la 12 foruri științifice (congrese, conferințe, saloane ale cercetării și inovării):

- Zilele Academiei de Științe Tehnice din România, 17-18 Octombrie 2019, Chișinău, Republica Moldova.
- Conferință Internațională "Telecomunicații, Electronică și Informatică", ICTEI 2018, Mai 24-27, 2018, Chișinău, Moldova.
- International Conference on Microelectronics and Computer Science ICMCS 2017, October 19-21, 2017, Chișinău, Moldova.

- International Conference on Microelectronics and Computer Science ICMCS 2014, October 22-25, 2014, Chișinău, Moldova.
- Conferința Națională de Interacțiune Om-Calculator, 2-3 septembrie 2013, Cluj-Napoca, România.
- Salonul Internațional de Inventică PRO INVENT 2012, 27-30 Martie 2012, Cluj-Napoca, România.
- European Exhibition of Creativity and Innovation – EUROINVENT 2012, May 10-12, 2012, Iasi, România.
- International Exhibition of Research, Innovation and Technological Transfer „INVENTICA 2012”, June 13-15, 2012, Iași, România.
- International Salon of Inventions and New Technologies „New Time”, Sep. 27-29, 2012, Sevastopol, Ukraine.
- International Warsaw Invention Show – IWIS 2012, October 16-19, 2012, Warsaw, Poland.
- International Conference on Microelectronics and Computer Science ICMCS-2011, September 22-24, Chișinău, Moldova.
- International Conference on Systems Science ICSS-2010, Sep. 14 - 16, 2010, Wroclaw, Poland
- International Conference on Microelectronics and Computer Science ICMCS-2009, October 1-3, Chișinău, Moldova.

Publicații științifice. La tema tezei au fost publicate 15 lucrări științifice inclusiv două articole în reviste de categoria B+ și C. Au fost obținute 5 medalii de aur, argint și bronz, 1 premiu special, 1 diplomă.

Structura și volumul lucrării. Teza este compusă din introducere, patru capitole, concluzii finale, bibliografie (120 titluri) și 4 anexe. Conținutul de bază al tezei este expus pe 131 pagini, inserează 81 figuri și 5 tabele.

Conținutul de bază a lucrării.

În **Introducere** este prezentată argumentarea și actualitatea temei de cercetare. Sunt formulate scopul și sarcinile cercetării, sunt prezentate domeniul și obiectivele cercetării, elementele de noutate științifică a rezultatelor obținute, este prezentată semnificația teoretică și valoarea aplicativă a domeniului de studiu.

În capitolul I, **Aspecte generale privind sistemele inteligente**, sunt prezentate metode, tehnici și tehnologii moderne de preluare a cunoștințelor, sunt descrise metodologia aplicării acestor tehnici în procesul implementării sistemelor cu inteligență artificială și elementele cheie în aplicarea acestor sisteme pentru automatizarea proceselor tehnologice. Sunt prezentate informații privind domeniile de aplicare ale sistemelor decizionale inteligente și specificul proiectării acestora. Este efectuată analiza impactului factorului uman asupra indicatorilor de

calitate a sistemelor de producție industrială. Este formulată problema și direcția de cercetare. Sunt prezentate tehnologiile disponibile pentru implementarea soluțiilor inteligente în automatizarea sistemelor de producție industrială.

În capitolul II, **Cercetarea și dezvoltarea metodelor și algoritmilor de luare a deciziilor în condiții de incertitudine**, sunt reflectate aspecte metodologice de preluare a experienței umane calificate cu scopul utilizării acesteia la proiectarea sistemelor de suport decizional. Pentru aceasta este propusă și dezvoltată conceptual abordarea inteligentă în luarea deciziilor pentru procese industriale. Ca bază în realizarea cercetărilor experimentale a fost luat procesul tehnologic de turnare a microfiredor. În acest context au fost elaborați algoritmi de preluare a experienței și de extragere a cunoștințelor din datele statistice colectate de la operatori experimentați. A fost abordată problema preluării cunoștințelor de la expertul uman calificat, au fost propuse metode de identificare a funcțiilor de apartenență și a parametrilor specifici acestuia.

Sunt prezentate modele Matlab de suport decizional inteligent care au la bază tehnicile propuse și dezvoltate cu aplicarea logicii fuzzy și a rețelelor neuronale artificiale. Au fost definite variabilele de intrare/ieșire, calificativele lingvistice și identificate funcțiile de apartenență specifice fiecărui operator. Au fost definite regulile de inferență. Totodată au fost prezentate și rezultatele simulărilor modelelor propuse în mediul Matlab.

Este prezentat modelul de dezvoltare al unui sistem de suport decizional, inclusiv al subsistemelor de achiziție, procesare și analiză a datelor privind mersul procesului tehnologic, având la bază tehnologii hardware și software inteligente.

Capitolul III, **Elaborarea arhitecturilor hardware adaptive**, este dedicat proiectării și implementării arhitecturilor hardware reconfigurabile pentru sisteme fuzzy și neuronale.

Sunt descrise arhitecturile hardware proiectate pentru implementarea sistemelor fuzzy și neuronale la nivel de suport decizional în sisteme tehnice cum ar fi, de exemplu, procesul de turnare a microfiredor. Arhitecturile dezvoltate ușurează esențial cercetarea și dezvoltarea sistemelor inteligente de comandă și control. Pentru realizarea proceselor de inferență specifice fuzzy și neuronale au fost proiectate componentele funcțiilor de apartenență definite tabelar, motorului de inferență generic, defuzificatorului și neuronului artificial.

Sunt prezentate relațiile pentru estimarea resurselor care permit calculul cantității și tipului de resurse necesar pentru implementarea arhitecturilor hardware adaptive în dependență de tipul și caracteristicile sistemului.

Este efectuată și prezentată analiza comparativă a soluțiilor arhitectural-structurale propuse și estimată posibilitatea implementării acestor arhitecturi pe circuite reconfigurabile de tip FPGA/CPLD.

În capitolul IV, **Aplicarea tehnicilor de suport decizional**, sunt prezentate unele soluții inovative sub forma unor sisteme hardware încorporate pentru utilizare în cercetarea și dezvoltarea sistemelor inteligente de suport decizional.

Sunt prezentate soluții inteligente, reconfigurabile, de comandă și control în procesul de turnare a microfiredelor, care au fost concepute ca un set flexibil de instrumente, ce poate fi reconfigurat pentru noi condiții sau chiar noi procese industriale.

Este prezentat un model reconfigurabil de sistem neuronal adaptiv cu posibilități de autoorganizare în baza arhitecturii nucleu *Nios II* pe 32 biți pentru cercetarea proceselor decizionale.

În **Concluzii și recomandări** sunt expuse cele mai importante realizări și rezultate ale tezei.

Anexele conțin codurile AHDL ale arhitecturilor hardware dezvoltate, listingul programului de exportare a ponderilor sinaptice și actul de implementare.

1. ASPECTE GENERALE PRIVIND SISTEMELE INTELIGENTE

1.1 Sisteme fuzzy, neuro-fuzzy, rețele neuronale și sisteme cu autoorganizare

Progresul recent în domeniul inteligenței artificiale și optimizarea tehnicilor software de calcul au deschis noi oportunități pentru cercetătorii din domeniu. La baza metodelor de învățare stau astfel de paradigme ale calculului inteligent cum sunt: rețelele neuronale artificiale (RNA), arborii decizionali și Sistemele Neuro-Fuzzy (SNF), care sunt aplicate cu succes pentru soluționarea diverselor probleme din diferite domenii [10-15].

Sistemele fuzzy sunt considerate un caz particular al sistemelor expert bazate pe logica fuzzy. Aceste sisteme reprezintă niște soluții pentru tratarea incertitudinii. Sistemele fuzzy propun o metodă matematică de reprezentare și prelucrare a cunoștințelor subiective ale omului cărora le este caracteristic un anumit grad de incertitudine [16]. În procesul de creare a mulțimilor fuzzy fiecărei variabile de intrare i se atribuie, de obicei, câte 5-7 variabile lingvistice sau calificative. Mecanismul de inferență într-un sistem fuzzy constă din trei etape. La prima etapă, valorile numerice ale intrărilor sunt mapate de o funcție de apartenență conform gradului de apartenență la mulțimile fuzzy respective. Această operație este numită fuzificare. La a doua etapă, sistemul fuzzy evaluează regulile de inferență în conformitate cu ponderile intrărilor. La a treia etapă, valorile fuzzy rezultante sunt transformat din nou în valori numerice. Operația dată este numită defuzificare. Această metodologie oferă posibilitatea operării cu categorii slab definite sau neclare în reprezentarea cunoștințelor umane pentru descrierea procedurii de luare a deciziilor. Comportamentul unor astfel de sisteme este descris printr-un set de reguli fuzzy, cum ar fi:

DACĂ <premisă> ATUNCI <concluzie>

care folosesc variabile lingvistice cu termeni simbolici. Fiecare variabilă reprezintă un set fuzzy.

Inferența este un proces logic care generează o concluzie dintr-o premisă, adică deduce o consecință în dependență de starea curentă. Aceasta este componenta de bază a raționamentului și procesează conținutul informațional disponibil, formulat din propoziții și deduce o informație nouă. În dependență de tipul și corelarea între premise sunt posibile o mulțime de moduri de inferență. Avantajul principal al procesului de inferență este faptul că acesta poate fi realizat atât în spațiul numeric, cât și în cel natural.

Avantajele sistemelor fuzzy:

- Posibilitatea de tratare a incertitudinii specifice cunoștințelor umane cu ajutorul variabilelor lingvistice;

- Simplitatea procesului de comunicare dintre expertul în domeniu și proiectantul sistemului;
- Interpretarea ușoară a rezultatelor, din cauza reprezentării naturale a regulilor;
- Extinderea ușoară a bazei de cunoștințe prin adăugarea de noi reguli;
- Robustețe în raport cu posibilele erori de sistem.

Dezavantajele sistemelor fuzzy:

- Incapabilitatea de generalizare a cunoștințelor, generează răspunsuri doar la interogările din baza sa de reguli;
- Nu sunt robuste în raport cu schimbările topologice ale sistemului și necesită modificări semnificative atât în baza de reguli cât și în procesul redefinire a funcțiilor de apartenență;
- Formularea regulilor de inferență depinde direct de expertul uman și de nivelul de calificare a acestuia.

Specificul sistemelor fuzzy constă în faptul că acestea oferă posibilitate procesării datelor eterogene. Sistemelor fuzzy sunt caracterizate prin faptul că în procesul de inferență pot fi procesate atât date numerice cât și cunoștințe lexicale. Procesul de inferență în sistemele fuzzy reprezintă o transformare neliniară a datelor de intrare. O altă proprietate caracteristică a sistemelor fuzzy o reprezintă faptul că acestea pot oferi o mulțime de soluții obținute prin căi diferite. Aceste proprietăți determină potențialul aplicativ al sistemelor fuzzy în rezolvarea problemelor specifice din diferite domenii în care aplicarea metodelor tradiționale de rezolvare nu este în stare să ofere nici măcar soluții parțiale.

Logica fuzzy este definită ca o generalizare a logicii booleene clasice. Aceasta oferă posibilitatea de a extinde domeniul de valori al variabilelor logice pentru a introduce noțiunea de adevăr parțial. Spre deosebire de logica clasică, care operează cu valorile discrete de 0 și 1 pentru fals și adevărat, logica fuzzy folosește domeniul continuu de valori logice cuprinse în intervalul $[0, 1]$, unde 0 indică falsul absolut, iar 1 indică adevărul absolut. Astfel, dacă în logica clasică un obiect poate aparține (1) sau nu (0) unei mulțimi date, logica fuzzy operează cu gradul de apartenență al obiectului la mulțime și poate lua valori între 0 și 1 . Logica fuzzy oferă instrumentele necesare pentru reprezentarea cunoștințelor și inferența acestora în sistemele inteligente cu ajutorul unor concepte imprecise, numite variabile lingvistice sau variabile fuzzy. Pentru reprezentarea acestora se folosesc mulțimile fuzzy, care tratează cantitativ interpretarea calitativă a cunoștințelor.

Mulțimile vagi reprezintă un concept folosit inițial de Lofti A. Zadeh și de Dieter Klaua în 1965. Paralel, Saliu (1965) definește un tip de structură mai generală numită "L-relation", studiată în context algebric abstract. Relațiile Fuzzy, utilizate în diferite domenii ca lingvistică, luarea deciziilor și clusterizare reprezintă un caz particular a "L – relation" unde L reprezintă intervalul unitar $[0, 1]$. O mulțime fuzzy (vagă) reprezintă o mulțime care nu este definită exact. Sarcina de bază a expertului uman constă în a formula raționamente corecte în baza unor informații incomplete. În matematică, mulțimile vagi sunt o extensie a noțiunii clasice de mulțime elementelor căreia li se atribuie grade de apartenență. Mulțimile vagi operează cu incertitudinea aleatorie subiectivă.

Rețelele neuronale în momentul de față reprezintă una din cele mai active ramuri ale domeniului inteligenței artificiale. Dezvoltarea arhitecturilor de calcul vectoriale a asigurat necesitățile hardware și de performanță pentru optimizarea proceselor de antrenare și simulare a rețelelor neuronale artificiale. Metodele și tehnicile utilizate în dezvoltarea rețelelor neuronale își regăsesc aplicabilitatea la rezolvarea unor clase de probleme specifice din domeniul mai larg al inteligenței artificiale [10, 17]. Rețelele neurale (RN) constituie obiectul principal de cercetare a neuroinformaticii [10, 17]. O rețea neuronală artificială reprezintă o arhitectură de calcul inspirată în special de structura creierului uman, folosită pentru aproximarea și estimarea unor funcții necunoscute [11].

La baza conceptului de rețea neuronală artificială a stat sistemul nervos central al omului. Nu există o definiție general acceptată a RNA. O rețea neuronală artificială, poate fi definită ca o mulțime de entități funcționale (neuroni) care reprezintă niște unități de procesare simple, puternic interconectate ce acționează în paralel și prezintă proprietatea de învățare [10, 11]. Rețelele neuronale artificiale sunt capabile de a efectua operații în mod colectiv și paralel. Termenul "rețele neuronale" se referă la modelele utilizate în inteligență artificială iar modelele de rețele neuronale artificiale reprezintă niște metode computaționale [11]. Funcția rețelei neuronale este determinată de legăturile dintre elementele acesteia. Informația păstrată în rețea este învățată prin ajustarea ponderilor legăturilor dintre unități, conform unui algoritm de învățare [12].

În 1943 Warren McCulloch și Walter Pitts au creat modelul matematic al rețelelor neuronale. Acest model a deschis calea pentru cercetarea și aplicarea rețelelor neuronale artificiale în inteligența artificială [11].

Structural, o rețea neuronală reprezintă o mulțime de neuroni interconectați, iar funcțional o rețea neuronală poate fi definită ca un sistem format dintr-o mulțime de elemente interconectate care cooperează pentru rezolvarea unei probleme anumite. Una din cele mai

importante proprietăți ale rețelelor neuronale o reprezintă faptul că acestea pot fi antrenate prin învățare. Prin învățare rețeaua își poate modifica comportamentul decizional pentru a explica mai bine caracterul unor date. Sistemele care posedă proprietatea de învățare se ciocnesc adesea de problema overfitting-ului (supra-ajustare).

Structurile simple care stau la baza rețelelor neuronale artificiale dispun de o proprietate numită emergență care se manifestă prin creșterea complexității comportamentale a sistemului cauzată de o avalanșă de transformări elementare generate în lanț [11]. Rețelele neuronale artificiale încearcă să modeleze matematic funcțiile cognitive ale creierului uman. Acest lucru duce la interpretarea neuronilor ca unități discrete de procesare distribuită [18]. Conexiunile în interiorul unei rețele neuronale artificiale asigură capacitatea sistemului de asimilare, învățare și luare a deciziilor. W. McCulloch și W. Pitts au descris primul model al rețelei neuronale iar F. Rosenblatt și B. Widrow au dezvoltat primul algoritm de antrenare. Principala caracteristică a rețelelor neuronale este faptul că aceste structuri ar putea învăța după exemple [19]. Rețelele neuronale artificiale sunt în stare să-și modifice structura internă prin modificarea ponderilor conexiunilor dintre neuronii artificiali pentru ajustarea, comportamentului sistemului modelat cu un nivel de eroare acceptabilă.

Avantajele rețelelor neuronale:

- Capacitate de învățare;
- Capacitate de generalizare;
- Robustețe.

Dezavantajele rețelelor neuronale:

- Interpretarea imposibilă a funcționalității;
- Dificultate în determinarea topologiei.

Rețelelor neuronale artificiale le sunt specifice următoarele caracteristici:

- Reprezentarea distribuită a informației: informația în rețeaua neuronală se stochează în mod distribuit prin procesul de ajustare a ponderilor.

- Capacitatea de generalizare a deciziilor în cazul discontinuității datelor de instruire. Această caracteristică depinde direct de arhitectura internă a rețelei neuronale artificiale. De remarcat faptul că odată cu creșterea dimensiunii rețelei crește și capacitatea de memorare a datelor de instruire, însă se observă o scădere a capacității de generalizare. Acest fapt face ca stabilirea arhitecturii optime a rețelei neuronale artificiale să reprezinte etapa esențială în proiectarea acesteia.

- Toleranța la zgomot: RNA pot fi antrenate cu succes, chiar și utilizând seturi de date afectate de zgomot.

- Rezistența la distrugerea parțială: datorită modului distribuit de reprezentare a informației, RNA își poate păstra parțial funcționalitatea chiar și în cazul în care este distrusă o oarecare parte a acesteia.

- Viteza de calcul: procesul de instruire a RNA este unul de durată, însă după antrenare procesul de inferență poate fi realizat destul de rapid cu ajutorul arhitecturilor de calcul vectoriale.

O RNA poate fi antrenată pentru a realiza o anumită funcție prin ajustarea ponderilor dintre neuroni. Acest proces este realizat astfel încât o anumită intrare să sensibilizeze o anumită ieșire. Rețeaua este antrenată pe baza comparării ieșirilor cu o țintă, până ce aceasta nu va asigura precizia necesară. La antrenarea supervizată a RNA se utilizează o mulțime de date de instruire structurate în perechi de tipul intrare-țintă [12]. Rețelele neuronale artificiale sunt caracterizate de 3 elemente: modelul elementului de procesare (neuronul), structura particulară de interconexiuni (arhitectura) și mecanismul de ajustare a legăturilor dintre neuroni (algoritmul de învățare) [20]. Relațiile topologice într-o rețea neuronală artificială pot fi specificate de arhitectura și parametrii interni ai acesteia. Ca parametri interni într-o rețea neuronală artificială servesc ponderile conexiunilor dintre neuroni, împreună cu funcțiile de activare ale acestora. Modelele rețelelor neuronale artificiale manifestă o dinamică de scurtă durată. Regulile locale definesc modul în care neuronii sunt activați ca răspuns la intrările externe sau la ieșirile altor neuroni. De obicei funcția de activare depinde de parametrii rețelei.

Algoritmul de învățare specifică modul în care ponderile rețelei neuronale se schimbă în timp. Procedura de antrenare a rețelelor neuronale artificiale reprezintă un proces de lungă durată în comparație cu dinamica funcției de activare. Algoritmul de învățare depinde de funcția de activare a neuronilor, de valorile țintă furnizate de către asistent și de valoarea curentă a ponderilor conexiunilor. La fel ca și alte metode de învățare, învățarea automată este realizată prin interpretarea datelor de intrare. Rețelele neuronale au fost folosite cu succes pentru soluționarea unei varietăți de probleme, cum ar fi computer vision și recunoașterea vocală, probleme care sunt greu de rezolvat cu ajutorul metodelor clasice de programare.

O rețea neuronală artificială reprezintă o clasă de modele statistice cu următoarele caracteristici:

1. conține seturi de ponderi ajustabile printr-un algoritm de învățare;
2. poate aproxima funcții non-liniare în baza intrărilor acestora.

Ponderile adaptive pot fi privite ca parametri topologici ai rețelei, care sunt implicați direct în procesele de antrenare și inferență.

Abordarea inspirată de biologie în implementările software, moderne ale rețelelor neuronale artificiale, a fost abandonată în folosul unei abordări mai practice, bazate pe statistică și procesare a semnalelor. În sistemele cu inteligență artificială în baza rețelelor neuronale, neuronii artificiali sunt organizați în structuri mult mai complexe, care conțin atât elemente adaptabile cât și neadaptabile. O abordare mai generală a acestui tip de sisteme poate dispune de un potențial aplicativ enorm în rezolvarea problemelor din lumea reală. În practică însă modelele tradiționale de inteligență artificială sunt mai puțin utilizate. Aceste abordări au în comun aceleași principii: non-liniaritate, adaptare, distribuție, procesarea paralelă și locală. Utilizarea modelelor de rețele neuronale artificiale a cauzat schimbarea direcției inteligenței artificiale de la nivel înalt (simbolic), caracterizată prin învățare mecanică, la nivel jos (sub-simbolic), caracterizată prin evoluția dinamică a parametrilor unui sistem.

Tehnicile moderne ale inteligenței artificiale și-au găsit aplicație în aproape toate domeniile cunoașterii umane. Cu toate acestea, un mare accent este pus pe domeniile științelor exacte, iar cel mai mare succes al acestor tehnici este realizat în domeniul ingineriei. Rețelele neuronale și logica fuzzy sunt două tehnici de multe ori aplicate împreună pentru rezolvarea problemelor ingineresti în cazul în care tehnicile clasice nu pot asigura o metodă simplă de rezolvare sau o soluție exactă.

Implementarea sistemelor fuzzy în domeniul industrial a scos la suprafață faptul că dezvoltarea unui sistem fuzzy performant reprezintă o sarcină destul de complexă. Problema identificării funcțiilor de apartenență și a setului de regulilor de inferență corespunzătoare reprezintă un proces monoton de încercări și eșecuri deseori însoțit de erori. Acest fapt a dus la apariția ideii de a aplica algoritmi de învățare în sistemele fuzzy. Pentru automatizarea procesului de ajustare a parametrilor sistemelor fuzzy puteau fi utilizate rețelele neuronale artificiale, care dispun de algoritmi eficienți de învățare. Bazele sistemelor neuro-fuzzy au fost puse la începutul anilor 90. Termenul neuro-fuzzy s-a format prin contopirea acestor două tehnici. Fiecare cercetător utilizează acest termen în felul său, fapt ce a dus la apariția unei oarecare confuzii referitoare la sensul direct al acestui termen. Totuși nu există un consens absolut însă, în general, termenul neuro-fuzzy semnifică un tip de sistem caracterizat printr-o structură similară cu cea a unui sistem fuzzy în care mulțimile fuzzy și regulile de inferență sunt ajustate folosind tehnici iterative specifice rețelelor neuronale. Acest tip de sisteme prezintă două moduri de comportament distincte. În primă fază, numită fază de învățare, acestea se comportă ca rețelele neuronale adaptându-și structura sa prin ajustarea parametrilor interni, ca ulterior, în

faza de execuție, acestea să se comporte ca sisteme logice fuzzy. Primele încercări în această direcție au fost făcute în controlul proceselor. Treptat, aplicația modelelor neuro-fuzzy s-a extins în diferite domenii cum ar fi analiza datelor, clasificarea datelor, identificarea defectelor, suportul decizional etc [8]. Separat, fiecare dintre aceste tehnici dispune atât de avantaje cât și de dezavantaje, însă atunci când sunt utilizate împreună, integrarea acestora oferă rezultate mult mai bune decât cele obținute prin utilizarea individuală a acestora. Rețelele neuronale artificiale și sistemele fuzzy pot fi combinate cu scopul de cumulare a avantajelor totodată eliminând neajunsurile individuale. Această combinație îmbină proprietățile învățării rețelele neuronale cu simplitatea reprezentării sistemelor fuzzy. Astfel, dezavantajele sistemelor fuzzy sunt compensate de proprietățile rețelelor neuronale artificiale. Aceste două tehnici sunt complementare, fapt ce justifică potențialul lor de aplicabilitate.

Sistemele neuro-fuzzy moderne, de obicei, reprezintă rețele neuronale multistrat feedforward. Cu toate acestea, deseori sunt aplicate și alte arhitecturi de rețele neuronale. În rețelele neuro-fuzzy funcțiile de propagare și activare diferă de cele ale rețelelor neuronale clasice. Termenul de sistem neuro-fuzzy este folosit de obicei pentru descrierea sistemelor care posedă următoarele proprietăți [13]:

- Un sistem neuro-fuzzy este bazat pe un sistem fuzzy, antrenat cu ajutorul unui algoritm de învățare provenit din teoria rețelelor neuronale. Procedura de învățare (euristică) determină doar modificări locale în sistemul de bază.
- Un sistem neuro-fuzzy poate fi privit ca o rețea neuronală cu 3 straturi. Primul strat reprezintă variabile de intrare, stratul de mijloc (ascuns) reprezintă reguli de inferență fuzzy iar al treilea strat reprezintă variabilele de ieșire. Mulțimile fuzzy utilizate în aceste sisteme sunt reprezentate prin ponderi. Structura de bază și arhitectura internă a acestor sisteme nu sunt strict definite și pot varia în dependență de cerințele problemei spre rezolvarea căreia sunt orientate. Cu toate acestea, structura menționată oferă posibilitatea reprezentării sugestive a fluxurilor de date la procesarea și învățarea din cadrul modelului. Uneori se poate utiliza o arhitectură cu cinci straturi, unde mulțimile fuzzy sunt reprezentate în straturile doi și patru.
- Un sistem neuro-fuzzy poate fi interpretat ca un sistem cu reguli fuzzy. De asemenea, este posibilă crearea de la zero a sistemului utilizând date de instruire prin inițializarea acestuia cu cunoștințe prelabile sub formă de reguli fuzzy. Nu toate modelele neuro-fuzzy specifică procedurile de învățare pentru crearea regulilor de inferență. Procedura de învățare a unui sistem neuro-fuzzy ia în considerare proprietățile semantice de bază ale sistemului. Acest fapt duce la constrângeri asupra posibilelor modificări aplicabile parametrilor sistemului. Nu toate abordările neuro-fuzzy au această proprietate.

- Un sistem neuro-fuzzy aproximează o funcție necunoscută, n-dimensională, parțial definită cu ajutorul datelor de instruire. Regulile fuzzy din cadrul sistemului pot fi privite ca prototipuri ale datelor de instruire. Un sistem neuro-fuzzy nu trebuie privit ca un tip de sistem expert (fuzzy) și nu are nici o legătură cu logica fuzzy în sensul direct al acesteia.

În general, orice combinație de tehnici bazate pe rețelele neuronale artificiale și pe logica vagă poate fi numită sistem neuro-fuzzy. Aceste combinații de tehnici pot fi împărțite în câteva clase mari [14]:

1. Sisteme neuro-fuzzy cooperative: în sistemele cooperative există o fază de pre-procesare în care mecanismele de învățare a rețelelor neuronale determină unele componente ale sistemului fuzzy conform unor șabloane, *Figura 1.1*. De exemplu, mulțimile sau regulile fuzzy. După ajustarea parametrilor rețeaua neuronală este eliminată, iar procesul de inferență este realizat de către sistemul fuzzy.

2. Sisteme neuro-fuzzy concurente: în sistemele concurente rețeaua neuronală și sistemul fuzzy cooperează permanent și participă împreună la procesul de inferență, *Figura 1.2*. În general, rețelele neuronale preprocesează intrările (sau postprocesează ieșirile) sistemului fuzzy.

3. Sisteme neuro-fuzzy hibride: în această categorie de sisteme rețeaua neuronală este utilizată pentru a învăța unii parametri (mulțimi fuzzy, reguli fuzzy și ponderi ale regulilor) ai sistemului fuzzy în un mod iterativ. Majoritatea cercetătorilor utilizează termenul neuro-fuzzy pentru a descrie doar sistemele neuro-fuzzy hibride.

Într-un sistem cooperativ rețelele neuronale sunt utilizate doar la fază inițială de ajustare a parametrilor sistemului fuzzy. În această fază se determină componentele sistemului fuzzy folosind date de antrenament. În faza următoare, decizională, rețelele neuronale nu mai sunt utilizate, iar procesul de inferență este realizat exclusiv în baza sistemului fuzzy. Structura sistemelor neuro-fuzzy cooperative nu este total interpretabilă fapt ce poate fi considerat ca dezavantaj [14].

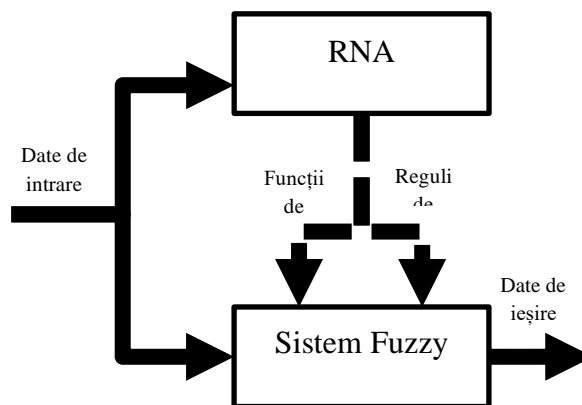


Figura 1.1. Structura sistemelor neuro-fuzzy cooperative.

Un sistem neuro-fuzzy concurrent nu reprezintă un sistem neuro-fuzzy în sensul direct al acestuia, deoarece rețeaua neuronală funcționează în paralel cu sistemul fuzzy. Intrările externe sunt preprocesate de către rețeaua neuronală artificială, apoi sistemul fuzzy procesează ieșirile acesteia. Această metodologie permite abordarea inversă în care preprocesarea inițială este realizată de către sistemul fuzzy iar procesarea propriu-zisă – de către rețeaua neuronală artificială. Dezavantajul sistemelor neuro-fuzzy concurente, constă în faptul că rezultatele nu pot fi complet interpretate [14].

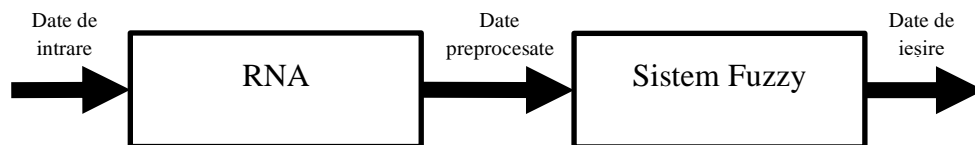


Figura 1.2. Structura sistemelor neuro-fuzzy concurente

În definiția lui Nauck [14]: „Un sistem neuro-fuzzy hibrid este un sistem fuzzy care folosește un algoritm de învățare bazat pe gradienti sau inspirat de teoria rețelelor neuronale (strategii de învățare euristică) pentru determinarea parametrilor săi (seturi fuzzy și reguli fuzzy) prin procesarea paternelor (intrare și ieșire)”. Un sistem neuro-fuzzy poate fi interpretat ca un set de reguli fuzzy. Acest sistem poate fi creat totalmente în baza datelor de intrare, ieșire sau prin inițializare cu cunoștințe reprezentate prin reguli fuzzy. Sistemul rezultat prin fuzionarea sistemelor fuzzy și rețelelor neuronale are ca avantaje învățarea prin paternelor și interpretarea ușoară a funcționalității sale. Există mai multe moduri de dezvoltare a sistemelor neuro-fuzzy hibride. Fiecare cercetător își poate defini modelele proprii. Aceste modele sunt similare în esența sa, dar prezintă o serie de diferențe de bază. Multe tipuri de sisteme neuro-fuzzy sunt implementate cu ajutorul rețelelor neuronale. Reprezentarea prin rețele neuronale este mult mai convenabilă, deoarece permite vizualizarea direcției fluxului de date în sistem și direcției semnalelor folosite pentru actualizarea parametrilor interni. Un beneficiu adițional îl reprezintă posibilitatea comparării diferitelor modele și vizualizarea diferențelor structurale ale acestora.

În rezultatul utilizării tehnicilor de inferență neuro-fuzzy hibride se obțin sisteme inteligente hibride care integrează metodele de raționament uman caracteristice sistemelor fuzzy cu procesul de învățare a rețelelor neuronale artificiale. În literatura de specialitate prin noțiunea de sistem neuro-fuzzy hibrid se subînțeleg o rețea neuronală fuzzy (RNF) sau un sistem neuro-fuzzy (SNF). Sistemele neuro-fuzzy permit realizarea procesului decizional într-o manieră asemănătoare raționamentului uman prin utilizarea mulțimilor fuzzy și a variabilelor lingvistice

la crearea regulilor de inferență. Avantajul principal al sistemelor neuro-fuzzy este posibilitatea exprimării interpretabile a cunoștințelor prin intermediul regulilor de inferență.

Utilizarea sistemelor neuro-fuzzy impune două cerințe contradictorii: interpretabilitatea și acuratețea. În practică una din aceste două proprietăți domină. Domeniul de cercetare a sistemelor neuro-fuzzy este divizat în două ramuri: modelarea fuzzy lingvistică orientată în special spre interpretabilitate, în care se aplică preponderent modelul Mamdani și modelarea fuzzy exactă, în care se aplică preponderent modelul Takagi-Sugeno-Kang (TSK).

Practica arată că direcția de bază în cercetarea sistemelor neuro-fuzzy o constituie sistemele care dispun de posibilitatea de actualizare în timp real în dependență de fluxurile noi de date aplicate la intrările acestora. Posibilitatea de actualizare a sistemului nu presupune doar ajustarea recurentă a parametrilor acestuia, dar și evoluția dinamică cu posibilitatea reducerii componentelor modelului (neuroni, reguli de inferență) pentru gestionarea eficientă și schimbarea dinamică a comportamentului sistemului.

Un sistem de inferență neuro-fuzzy adaptiv sau un sistem de inferență fuzzy implementat cu ajutorul rețelelor neuronale Adaptive Neuro-Fuzzy Inference System (ANFIS) este o rețea neuronală artificială bazată pe modelul de inferențe Takagi-Sugeno. Această tehnică a fost dezvoltată la începutul anilor 90. Deoarece integrează atât rețele neuronale, cât și principiile logicii fuzzy, aceasta posedă potențialul de a cumula beneficiile ambelor metode într-o singură arhitectură. Mecanismul său de inferență este alcătuit dintr-un set de reguli fuzzy cu capacitate de învățare. Din aceste considerente, ANFIS este considerat un estimator universal. Ajustarea eficientă a arhitecturii ANFIS poate fi obținută prin utilizarea algoritmilor genetici pentru optimizarea parametrilor acesteia. Rețeaua neuronală din componența arhitecturii ANFIS poate fi divizată în două părți: premise și consecințe. Această arhitectura este compusă din cinci straturi. Primul strat preia valorile de intrare și determină funcțiile de apartenență ale acestora. Acest strat reprezintă stratul de fuzificare. Gradele de apartenență ale fiecărei funcții sunt calculate prin utilizarea setului de parametri premisă. Al doilea strat este responsabil de generarea ponderilor regulilor de inferență. Al doilea strat reprezintă stratul de inferență. Funcția celui de-al treilea strat este de a normaliza ponderile calculate, prin divizarea fiecărei valori la ponderea totală. Al patrulea strat are ca intrări valorile normalizate și generează parametrii de consecință. Valorile returnate de acest strat sunt defuzificate. Aceste valori sunt aplicate la ultimul strat pentru a returna rezultatul final [15].

Prin definiție autoorganizarea reprezintă proprietatea unui sistem evolutiv de a-și schimba formele de organizare, modelele sau structurile interne. Autoorganizarea sistemelor reprezintă un proces ireversibil în rezultatul cărui se obțin noi structuri mai eficiente. Problema autoorganizării

sistemelor a devenit extrem de actuală odată cu dezvoltarea ciberneticii și a științei calculatoarelor. În Cibernetică apare tendința non-clasică pentru studiul diferitelor sisteme. Sinergia ca disciplină apare ca o nouă direcție foarte promițătoare în știință care extinde semnificativ gama proceselor de auto-organizare, studiate anterior. Se poate observa că auto-organizarea este considerată ca proprietate indispensabilă a sistemului. Introducerea termenului de "sinergie" în știință a dus la apariția a două metode de abordare a cercetării: sinergică și cibernetică. Aceste abordări diferă prin causalitatea comportamentului sistemic al procesului de autoorganizare. Abordarea cibernetică presupune existența unui scop predefinit, spre care tinde sistemul, iar abordarea sinergică nu necesită un scop. Autoorganizarea reprezintă rezultatul unei cooperări între diferitele componente ale sistemului. În sinergie și cibernetică o importanță majoră i se acordă noțiunii de "management", care însă are obiective diferite. Cibernetică oferă modele și metode de control ale sistemelor astfel încât acestea să dispună de un mod de funcționare predefinit. Controlul sinergic nu afectează în mod direct parametrii de control ai sistemului, ci doar declanșează procedurile de autoorganizare internă. În controlul sinergic sistemul alege calea dezvoltării într-o organizație superioară, iar în cel cibernetic, obiectivul este stabilit în prealabil. Ambele mecanisme menționate anterior au la bază comunicații, care apar spontan între elementele sistemului și permit schimbarea structurii și modului de organizare a sistemului fără comenzi de control pentru declanșarea interacțiunilor locale. Tendința unui sistem spre autoorganizare presupune totodată păstrarea integrității acestuia. Pe de altă parte, se vehiculează ideea că dezvoltarea nu poate exista fără instabilitate deoarece aceasta se produce prin tranziție și schimbare a proprietăților sistemului. Declanșarea instantanee a procesului de tranziție a sistemului poate fi cauzată de factori interni sau externi [21].

Sistemele cu autoorganizare au în componența sa un număr mare de componente care interacționează între ele prin bucle de reacție. Arhitectura acestui tip de sisteme presupune existența unui mecanism echilibrat de organizare a structurilor dinamice. Complexitatea internă a acestor sisteme le permite să-și mențină stabilitatea internă, rămânând în același timp deschise pentru interacțiunea cu factorii mediului extern. Haosul reprezintă unul din factorii primordialii în procesul de organizare și de formare a structurilor în cadrul acestor sisteme [22].

Autoorganizarea este cea mai uimitoare proprietate a sistemelor adaptive complexe. Aceasta a fost definită de către F. Heylighen ca „apariție spontană a unei noi structuri coerente globale a sistemelor adaptive complexe plecând de la interacțiunile locale dintre agenți”. Ciberneticianul britanic W. Ross Ashby a formulat principiul autoorganizării conform căruia un sistem dinamic, indiferent de structura acestuia, tinde întotdeauna să evolueze către o stare de echilibru, denumită atractor. Faptul dat este în stare să micșoreze incertitudinea privind starea

sistemului prin reducerea entropiei acestuia, ceea ce și definește autoorganizarea. Echilibrul rezultat reprezintă starea sistemului în care toate părțile componente ale acestuia sunt adaptate reciproc [23]. În anii '60 a început elaborarea de noi aplicații practice, printre care se află și rețelele neuronale artificiale, care reprezintă modele de interacțiune a neuronilor din creierul uman. Acestea au la bază modelul neuronului artificial, propus de W. McCulloch și W. Pitts. Rețelele neuronale nu dispune de o unitate centrală de comandă și control a proceselor de inferență modelate, acestea fiind realizate în baza conexiunilor interne dintre neuroni. Rezultatul simulării proceselor de inferență în aceste sisteme vor reprezenta modele decizionale. Astfel pornind de la un set limitat de reguli simple, se poate obține un comportament mult mai complex la scara întregului sistem. Cercetările acestui tip de sisteme pot explica fenomenele care au loc în sisteme mult mai complexe. Studiul sistemelor adaptive complexe a depistat diferența dintre sistemele cu autoorganizare și sistemele clasice. Condiția necesară pentru ca un sistem să poată fi adaptiv, este ca acesta să dispună de capacitatea de autoorganizare.

Încercările lui J. Holland de a înțelege mecanismele prin care organismele biologice se adaptează la condițiile variabile ale mediului, au dus la fondarea teoriei algoritmilor genetici. Aceasta, utilizând o serie de operații specifice geneticii, cum sunt selecția, mutația, recombinarea, și-a găsit aplicare pentru simularea modului în care pot apărea noi forme de organizare în procesul de antrenare a sistemelor cu inteligență artificială. Astăzi, algoritmi genetici sunt frecvent utilizați în diferite domenii pentru a descrie modul în care evoluează sistemele [23].

Cercetările în domeniu au dus treptat la evidențierea trăsăturilor definitorii ale auto-organizării care o deosebesc de alte proprietăți ale sistemelor adaptive. Într-un sistem declanșarea procesului de organizare are loc în mod instant ca rezultat al interacțiunilor locale între entitățile acestuia. Genul interacțiunilor locale între componentele unui sistem, este determinat de natura sistemului respectiv. Pe parcursul evoluției sistemului, acesta va fi perturbat de influența factorilor mediului extern. Dacă configurația unei anumite părți a sistemului nu depinde de configurația oricărei alte părți, atunci corelația între aceste configurații este zero. În procesul de autoorganizare, corelația dintre diferite părți ale sistemului crește tot mai mult. Interacțiunile locale sunt cauza corelării puternice între configurațiile învecinate. Forța care declanșează procesul de reorganizare prin emergența interacțiunilor locale în sistemele cu autoorganizare este rezultatul acțiunii buclei de reacție prin care cauza produce un efect care acționează reflexiv asupra cauzelor apariției sale. Acesta reprezintă un mecanism de reacție care poate avea ca rezultat amplificarea procesului de reorganizare, sau inhibarea acestuia. Reacția împreună cu evoluția reprezintă condiția necesară a procesului de auto-organizare. Acest fapt

este confirmat de o mulțime de exemple de auto-organizare a biosistemelor. Auto-organizarea cuprinde conceptele de supraviețuire prin adaptare și schimbare prin evoluție.

Un sistem cu o formă de organizare înaltă presupune existența unui agent intern sau extern care controlează sistemul respectiv. În sistemele cu autoorganizatoare procesul de organizare este distribuit în întreg sistemul. Fiecare entitate a acestuia contribuie într-o oarecare măsură la acest proces.

Studiile realizate asupra creierului uman arată că aceasta nu dispune de un mod centralizat de funcționare. Controlul distribuit este implementat deseori și în formele de organizare a sistemelor hardware. Odată cu complexitatea acestor sisteme crește complexitatea mecanismului de interacțiune și propagare a fluxurilor decizionale, și informaționale. Existența unor astfel de rețele nu constituie o condiție necesară a controlului distribuit. Pentru a se realiza un astfel de control entitățile sistemului trebuie să fie capabile să interacționeze pentru atingerea unor obiective comune, trăsătură specifică însă sistemelor adaptive complexe [23].

1.2 Forme de reprezentare și metode de extragere a cunoștințelor

Problematica domeniului cercetat în teză poate fi divizată în două categorii:

1. preluarea experienței operatorului uman și extragerea cunoștințelor cu scopul dezvoltării sistemelor decizionale inteligente pentru comanda și controlul proceselor industriale;

2. dezvoltarea arhitecturilor hardware reconfigurabile pentru comanda și controlul proceselor industriale [24]. Aceste aspecte importante sunt strâns interconectate și ca rezultat necesită abordarea unor tehnologii inteligente care reprezintă o caracteristică esențială a procesului industrial cercetat [25-30]. În cea mai mare parte procesele industriale moderne sunt automatizate. Din cauza complexității proceselor tehnologice moderne, în unele procese industriale specifice, operatorul uman reprezintă o resursă critică. O problemă majoră o constituie proiectarea și implementarea unor tehnici avansate de automatizare și suport decizional care ar contribui esențial la eficientizarea procesului de producție prin excluderea totală sau parțială a operatorului uman din bucla de reacție. În acest context, optimizarea procesului de producție presupune soluționarea unui șir de probleme care ar permite automatizarea proceselor de producție cu îmbunătățirea sau, în cel mai rău caz, cu menținerea caracteristicilor de calitate actuale. Automatizarea proceselor de fabricare utilizând soluții tehnice inovative ar putea permite excluderea totală sau parțială a influenței factorului uman, care prevalează momentan în procesele tehnologice specifice [31]. Utilizarea unui alt mod de abordare și a unor tehnici moderne de procesare a datelor, inclusiv tehnicilor de „machine

learning”, poate duce la obținerea unor noi soluții pentru comanda și controlul proceselor industriale în cazul în care tehnicile clasice nu mai fac față cerințelor de calitate formulate [32].

Problema științifică soluționată rezidă în reconceptualizarea principiilor de dezvoltare a sistemelor cu inteligență artificială prin abordarea specifică a procedeelelor clasice de preluare a experienței operatorului uman, fapt care a demonstrat potențialul aplicativ al acestei abordări în vederea valorificării acestuia.

Ținând cont de creșterea rolului și ponderii sistemelor inteligente în domeniul aplicațiilor industriale devine tot mai actuală problema cercetării, dezvoltării și implementării unor soluții avansate specializate pentru automatizarea proceselor tehnologice specifice atât în faza de producere cât și la etapa de control al calității.

Inteligența artificială emulează comportamentul inteligent prin prelucrarea automată a cunoștințelor disponibile. Sistemele cu inteligență artificială dispun de o careva mulțime de cunoștințe care stă la baza proceselor decizionale ale acestora. Caracteristica de bază a acestor sisteme constă în realizarea activităților decizionale în domenii specifice. Sistemele inteligente bazate pe cunoștințe mimează proprietatea de luare a deciziilor a unui expert uman prin realizarea procesului de inferență a cunoștințelor corespunzătoare [33].

Sistemele expert reprezintă programe capabile să reproducă raționamentul uman prin inducții și deducții [34]. Sistemele expert trebuie să realizeze următoarele obiective:

- preluarea cunoștințelor de la expert;
- stocarea cunoștințelor în baza de cunoștințe;
- posesia unei interfețe de comunicare cu utilizatorul.

Modul de funcționare a unui sistem expert poate fi caracterizat de totalitatea tehnicilor folosite de acesta pentru selectarea cunoștințelor necesare din baza de cunoștințe și realizarea procesului de luare a deciziilor. Baza de cunoștințe se actualizează cu informație obținută de la expertul în domeniu. Această informație necesită a fi prelucrată și structurată corespunzător pentru a putea fi interpretată de către sistemul expert. În baza de cunoștințe informația este stocată sub formă de fapte și reguli. În baza de fapte se stochează toată informația posibilă despre proprietățile unui anumit obiect [35].

Se numește sistem cognitiv totalitatea modelelor, structurilor și metodelor de cunoaștere [36]. Pentru a identifica baza de cunoștințe sunt necesare metode de extragere și de structurare a datelor [33]. Formele de reprezentare și metodele de extragere a cunoștințelor reprezintă factorul primar în proiectarea și implementarea sistemelor inteligente. Eficiența unui sistem inteligent în baza cunoștințelor este determinată de o serie de funcționalități specifice:

1. Achiziția – este caracterizată printr-o colecție de metode și tehnici de achiziția a datelor de intrare care influențează direct calitatea informației culese, care va servi la extragerea cunoștințelor sau ca interogări pentru realizarea raționamentelor;

2. Preprocesarea – este caracterizată prin realizarea unei procesări preventive a datelor de intrare în vederea structurării corespunzătoare a datelor inițiale.

3. Stocare – este caracterizată prin metodele de stocare a cunoștințelor în baza de cunoștințe;

4. Procesarea – este caracterizată prin producerea de cunoștințe noi în urma aplicării procesului de inferență asupra cunoștințelor, interogări către baza de cunoștințe care generează noi cunoștințe pentru completarea bazei existente;

5. Interfațarea cu sistemele care generează sau cele cărora le sunt destinate rezultatele raționamentelor realizate [37].

Calitatea unui sistem cu inteligență artificială depinde direct de calitatea și cantitatea cunoștințelor de care dispune sistemul la momentul realizării raționamentelor. În dependență de sursa din care provin cunoștințele acestea pot fi induse sau deduse. Inteligența artificială simulează raționamentul uman pentru efectuarea proceselor deductive. Prelucrarea cunoștințelor implică definirea unor structuri de stocare și a unor metode de procesare a acestora pentru efectuarea raționamentelor [38].

Deoarece nu există o definiție unanim acceptată a termenilor de informație, date și cunoștințe, iar adesea cei trei termeni se definesc unul prin intermediul celuilalt, de multe ori apar confuzii. În domeniul inteligenței artificiale însă există o diferențiere clară între acești trei termeni care pot fi definiți astfel:

- datele reprezintă premise de intrare pentru procesul de prelucrare a acestora.
- informația reprezintă rezultatul aplicării metodelor de procesare a datelor.
- cunoștințele reprezintă informația despre metodele de procesare a datelor.

Astfel se poate menționa faptul că datele reprezintă o submulțime a cunoștințelor, iar cunoștințele pot fi definite ca o descriere procedurală a unui domeniu de studiu. Forma de reprezentare a cunoștințelor trebuie să țină cont de specificului datelor și să corespundă proprietăților acestora.

Una din caracteristicile specifice ale cunoștințelor o reprezintă faptul că acestea pot fi incomplete. Această proprietate se explică prin faptul că cunoștințele pot fi considerate implicite de către expertul uman, sau pot fi dificil de formulat. Faptul dat presupune identificarea formelor de reprezentare a cunoștințelor și a procedurilor de raționament aplicate asupra acestora în condiții de incompletitudine.

O altă caracteristică a cunoștințelor o reprezintă faptul că acestea pot fi mai puțin sigure. Acest fapt presupune utilizarea unor forme de reprezentare a cunoștințelor cu metode de realizare a raționamentului în condiții de incertitudine [38- 40].

Mecanismul de inferență a cunoștințelor este elementul de bază a unui sistem expert care asigură procesul de cunoaștere prin aplicarea regulilor și strategiilor de raționament asupra bazei de fapte [36, 41]. Cunoașterea se bazează pe trei concepte fundamentale:

1. Fapte – reprezintă informații primare care descriu elementele domeniului;
2. Reguli – descriu modul în care pot fi utilizate faptele;
3. Strategii de raționament – descriu modul în care pot fi folosite regulile.

Procesarea cunoștințelor implică definirea structurilor de stocare și a metodelor de prelucrare, care să asigure realizarea raționamentelor. În rezultat pentru stocarea și utilizarea cunoștințelor se utilizează structuri specifice de reprezentare a acestora [34].

În cazul cunoașterii empirice informația despre obiectul de studiu este achiziționată de către subiectul cunoașterii prin organe senzoriale. Cunoașterea teoretică derivă din cunoașterea empirică, iar informația despre obiectul de studiu este achiziționată de către subiectul cunoașterii prin analiză, sinteză, deducție, inducție, generalizare și particularizare [36]. Procesul de creare a unei baze de cunoștințe este însoțit de procedura de identificare a mulțimii obiectelor semnificative și a relațiilor dintre acestea. Baza de cunoștințe trebuie să conțină suficiente cunoștințe pentru a asigura corectitudinea procesului de inferență la rezolva problemelor din domeniul de studiu. O bază de cunoștințe reprezintă o colecție de obiecte din domeniul de studiu împreună cu relațiile dintre ele. Cerința de bază față de un sistem expert este ca rezultatele procesului de inferență să corespundă observațiilor [41].

Faptul că cunoștințele se pot modifica în rezultatul schimbărilor survenite în domeniul de studiu în care sunt utilizate, sau în rezultatul completării bazei de cunoștințe cu reguli noi, reprezintă o altă caracteristică a cunoștințelor. Din cauza dependenței mari dintre cunoștințele unui domeniu, modificările unora pot cauza invalidarea altora. Această caracteristică presupune identificarea formelor de reprezentare corespunzătoare pentru tratarea excepțiilor [38].

Reprezentarea cunoștințelor constă în identificarea unor relații din lumea reală pentru implementarea acestora în sistemul inteligent proiectat. Cu acest scop se fac observații asupra obiectului de studiu pentru determinarea caracteristicilor de bază a acestuia care pot fi utilizate în procesul decizional. La această etapă este realizată descrierea obiectului cercetat prin intermediul altor obiecte și legăturile dintre acestea [36].

Metodele de reprezentare a cunoștințelor pot fi clasificate în două categorii mari: metode declarative și metode procedurale. Metodele declarative se utilizează la reprezentarea faptelor și

asertiunilor, iar metodele procedurale se utilizează la reprezentarea acțiunilor sau procedurilor. Metodele declarative includ rețele semantice, cadre/obiecte structurate și logica predicatelor. Metodele procedurale includ reguli de producție. Pe lângă acestea mai există și alte metode, utilizate în proiectarea sistemelor expert cum ar fi: scenarii liste, tabele de decizie, arbori decizionali, diagrame de dependență, restricții, rețele Petri stochastice, rețele neuronale [41].

Reprezentarea cunoașterii are ca scop descrierea domeniului de studiu în care sistemul realizează procesul decizional. Există diferite metode de reprezentare a cunoștințelor în domeniul sistemelor expert: metode bazate pe logica formală, metode bazate pe partajarea proprietăților (reprezentări succesoriale) [35]. Pot fi distinse trei clase de metode de reprezentare a cunoștințelor:

1) metode de reprezentare logică (reguli de inferență) - utilizează expresii specifice logicii formale pentru reprezentarea cunoștințelor;

2) metode de reprezentare relaționale (grafuri și rețele) - stochează cunoștințele sub formă de graf în care nodurile reprezintă obiectele domeniului de studiu iar arcele reprezintă relațiile dintre acestea.

3) metode de reprezentare procedurală (proceduri) - tratează cunoștințele ca pe un set de instrucțiuni pentru rezolvarea unei probleme [36, 41].

Formele de reprezentare a cunoștințelor definesc structurile de reprezentare a acestora care trebuie să satisfacă o serie de cerințe:

- integritate reprezentatională – posibilitatea de acoperire reprezentatională, completă a tuturor categoriilor de cunoștințe din cadrul domeniului respectiv;
- acoperire achiziționară – forma de reprezentare trebuie să asigure procesul de achiziționare a cunoștințelor;
- structurare informațională – formele de reprezentare trebuie să definească proceduri de prelucrare a cunoștințelor;
- eficacitatea procesării – formele de reprezentare trebuie să asigure eficiența procesului de inferență a cunoștințelor.

Formele de reprezentare a cunoștințelor pot fi caracterizate cu ajutorul următoarelor proprietăți:

- gradul de granularitate al reprezentării determinat de nivelul de detaliere al primitivelor reprezentationale;
- gradul de modularitate al structurilor de reprezentare ce exprimă nivelul de independență relativă a acestor structuri;

- gradul de compilare al reprezentării, exprimă măsura în care reprezentarea favorizează anumite scheme de utilizare a cunoștințelor.

- gradul de nedeterminare se referă la numărul de soluții alternative de reprezentare din care se poate face reprezentarea unei anumite părți a cunoașterii.

Metodele și tehnicile de reprezentare a cunoștințelor pot fi clasificate, în funcție de modul de prezentare a acestora, în trei categorii mari [33]:

1. forme de reprezentare simbolică (calculul predicatelor, reguli de producție, cadre, frame-uri, grafuri etc.);
2. forme de reprezentare neuronală;
3. forme de reprezentare genetică.

La moment există mai multe metode de reprezentare a cunoștințelor, care asigură mai mult sau mai puțin soluționarea problemelor specifice din domeniul inteligenței artificiale.

Din aceste metode se pot menționa cele mai frecvent utilizate:

1. logica propozițiilor și calculul predicatelor;
2. reguli de producție;
3. rețele semantice;
4. grafuri conceptuale.

Regulile de producție reprezintă una din modalitățile clasice de reprezentare a cunoștințelor într-un sistem expert. Termenul de reguli de producție determină o metodă de reprezentare a cunoștințelor bazată pe logica propozițiilor, în care faptele și regulile reprezintă niște entități invariabile. Metoda regulilor de producție este eficientă pentru reprezentarea recomandărilor sau strategiilor cunoașterii care se referă la soluționarea unor probleme specifice dintr-un domeniu particular. Reprezentarea cunoștințelor prin reguli de producție se face prin două tipuri de structuri: fapte și reguli. Faptele constituie aserțiuni care descriu un anumit aspect din realitate. Ele reprezintă entitățile elementare ale cunoașterii. Faptele nu permit deducerea de noi cunoștințe iar regulile se utilizează pentru a descrie legăturile dintre faptele pe baza cărora se va efectua procesul de deducție. Într-o regulă de producție, atât premisele cât și concluziile sunt fapte [35].

Reprezentarea cunoștințelor cu ajutorul regulilor de producție face parte din categoria metodelor procedurale. Metodele date provin din teoria logicii formale, în care sunt cunoscute ca reguli de inferență. Un sistem de producție reprezintă un sistem în care procesul de generare a noilor cunoștințe este asigurat de reguli de producție predefinite [38].

În logică, propozițiile reprezintă aserțiuni care pot fi adevărate sau false. Logica propozițiilor oferă o metodă formală de calcul a valorii de adevăr a propozițiilor [35].

Cunoștințele procedurale în sistemele de producție sunt reprezentate prin reguli de producție sub forma unor perechi condiție-acțiune. O regulă de producție are forma generală: *“Dacă condiție atunci acțiune”* în care, acțiunea reprezintă un concept, iar condiția reprezintă entitatea de evaluarea căreia depinde declanșarea regulii. Organizarea cunoștințelor, în metoda de reprezentare prin reguli de producție, se realizează pe trei nivele: declarativ, procedural și strategic.

1. Nivelul declarativ presupune reprezentarea cunoștințelor sub forma unei colecții denumite bază de fapte.

2. Nivelul procedural presupune reprezentarea cunoștințelor sub forma unor perechi premisa-concluzie, stocate în baza de reguli.

3. Nivelul strategic presupune reprezentarea cunoștințelor sub formă de reguli ce tratează procesul decizional ca pe o secvență de decizii aplicate asupra secvențelor de acțiuni.

Astfel structura unei baze de cunoștințe în sistemele de producție constă din: baza de fapte, baza de reguli și mulțimea strategiilor de control.

Sistemele de producție au la bază reguli de producție și mecanisme inferențiale. Regulile de producție au forma condiție-acțiune, unde partea condiție precizează condițiile care trebuie îndeplinite pentru a declanșa o acțiune. Această parte reprezintă un declanșator. Dacă condiția este îndeplinită de regula respectivă, aceasta este selectată pentru declanșare. Regulile selectate pentru declanșare intră în mecanismul inferențial în care se realizează filtrarea, în urma căreia în dependență de prioritatea acestora este selectată pentru declanșare o singură regulă. Procesul inferențial se desfășoară conform următoarelor etape: evaluare și execuție.

La etapa de evaluare se determină dacă există în bază reguli de declanșat și dacă acestea au condiția îndeplinită.

La etapa de execuție se declanșează regulile validate la etapa de evaluare.

Mecanismul poate fi oprit la etapa de evaluare în cazul absenței regulilor declanșatoare, iar oprirea în etapa de execuție poate fi cauzată de declanșarea unei reguli.

În unele cazuri specifice informația cu care operează un sistem cu inteligență artificială poate avea un caracter incert. Există cazuri în care veridicitatea unei aserțiuni nu poate fi asigurată decât parțial. Unele concepte implicit conțin în sine un oarecare grad de incertitudine sau o oarecare formă de inexactitate. Procesul decizional al expertului uman adesea este determinat de informații parțiale sau incomplete. Inexactitatea datelor în sistemele expert poate fi cauzată de surse nesigure de informație sau de ipoteze formulate în baza unor date incomplete. Ca surse de incertitudine pot servi: interfața de comunicare cu utilizatorul sistemului expert,

nivelul de calificare al utilizatorului, sau competențele acestuia. Sursa principală de incertitudine o reprezintă imprecizia informației care stă la baza dezvoltării sistemului expert [38-40].

Pentru operarea cu concepte inexacte sau incerte este necesară o metodă de inferență specifică care să asigure acest proces. Aceasta presupune utilizarea unei logici inexacte și a unui proces de inferență specific. Necesitatea utilizării metodelor de inferență inexactă se datorează faptului că deseori sunt disponibile doar niște cunoștințe sau date inexacte care trebuie procesate. Reprezentarea informației inexacte în sistemele expert poate fi realizată prin trei metode de bază: probabilități subiective, factori de certitudine și logică fuzzy. Trebuie remarcat faptul că există o oarecare confuzie în utilizarea celor trei termeni. Adesea factorul de certitudine este confundat cu probabilitatea subiectivă, iar termenul de logică fuzzy era utilizat pentru descrierea oricărei metode de operare cu raționamente inexacte [42, 43].

Cea mai simplă metodă de reprezentare a cunoștințelor inexacte constă în atribuirea unei probabilități fiecărei propoziții. Pentru determinarea probabilităților de apariție a fiecărui eveniment particular se folosește Teorema lui Bayes. Această teoremă provine din teoria clasică a probabilităților și presupune o distribuție aleatoare a valorilor. Cu ajutorul acestei teoreme se pot calcula probabilitățile tuturor regulilor. Avantajul acestei metode constă în faptul că fiecare propoziție este caracterizată printr-o singură valoare probabilă. Simplitatea în utilizare a acestei metode impune de asemenea și anumite restricții [38]. Probabilitatea subiectivă caracterizează cât de adevărată poate fi o afirmație.

În rezultatul analizei făcute se poate concluda faptul că metoda de extragere și forma de reprezentare a cunoștințelor utilizate la dezvoltarea unui sistem cu inteligență artificială depinde de tipul sistemului care va realiza inferența acestor cunoștințe pentru a asigura suportul procesului decizional cu ajutorul cunoștințelor, raționamentelor și concluziilor noi.

1.3 Arhitecturi hardware adaptive

Problema preluării experienței umane niciodată nu a făcut parte din categoria problemelor triviale, dar la momentul de față a trecut în categoria provocărilor ingineresti soluționarea cărora necesită cunoștințe în domeniu și destulă ingeniozitate din partea cercetătorilor. Eficiența unui sistem inteligent depinde direct de natura și calitatea datelor utilizate pentru antrenare care se răsfrânge în final asupra procesului de inferență. Creșterea performanței sistemelor cu inteligență artificială poate fi obținută prin creșterea frecvenței de lucru, utilizarea paralelă a mai multor unități de procesare (arhitectură super-scalară) sau prin utilizarea arhitecturilor hardware specializate. La etapa actuală, creșterea vitezei de procesare nu mai poate fi asigurată doar în rezultatul utilizării tehnologiilor de integrare. Un impact semnificativ în dezvoltarea sistemelor

cu inteligență artificială a fost asigurat de soluțiile arhitecturale bazate pe diferite tipuri de paralelism. Acest fapt demonstrează încă odată oportunitatea și necesitatea abordării inovative a domeniului arhitecturilor de calcul moderne. Disponibilitatea tehnicilor moderne de procesare a datelor oferă posibilități noi de implementare a metodologiilor de extragere a cunoștințelor [5].

Termenul de sistem adaptiv se referă la un sistem interdependent compus din entități cuplate între ele care cooperează pentru a se adapta și auto-organiza la condițiile mediului înconjurător. Studiile în domeniul arhitecturilor hardware adaptive oferă o varietate de noi soluții și mijloace de abordare a diferitor probleme legate de metodele de organizare și funcționare eficientă a sistemelor. Baza sistemelor hardware adaptive o constituie componentele sau entitățile funcționale colaborative [21-24].

Procesul de adaptare a sistemelor poate fi realizat pe cale software sau hardware. Adaptarea software poate fi realizată în general printr-o schimbare funcțională la nivelul aplicației. Adaptarea hardware, spre deosebire de cea software, reprezintă o modificare mai profundă în organizarea internă a arhitecturii de calcul a unui sistem încorporat. Acest fapt permite o adaptare mai puternică, dar care, în schimb este limitată de specificul procesului de reconfigurare a dispozitivelor Field Programmable Gate Array (FPGA) [44].

Arhitecturile hardware auto-adaptive bioinspirate se bazează pe o serie de entități cu capacitate de prelucrare paralelă, care implementează în mod distribuit capabilități de autoadaptare, cum ar fi autorutare, autoplasare și autoconfigurare în timpul rulării [45].

În prezent sistemele de calcul se bazează pe diferite platforme arhitecturale care se deosebesc din punct de vedere al eficienței, flexibilității și complexității de proiectare [46-50]. Din acestea pot fi menționate:

- Circuite specializate (ASIC – Application Specific Integrated Circuits);
- Microcontrolere;
- Procesoare de prelucrare a semnalelor (DSP – Digital Signal Processor);
- Circuitele logice reconfigurabile (CPLD – Complex Programmable Logic Device, FPGA – Field Programmable Gate Array);
- Procesoare dedicate (SoC – System on Chip).

Din cauză că circuitele specializate sunt destul de rapide acestea sunt și foarte eficiente în aplicații dedicate, însă dispun de o flexibilitate limitată deoarece acestea nu pot fi readaptate pentru rezolvarea altor probleme. Din cauza costurilor mari de proiectare a circuitelor specializate, se dovedește mai eficientă utilizarea ASIC-urilor în cazul în care este necesară o producere în masă. În cazurile producției unei serii limitate, costurile unui circuit specializat cresc esențial. Din această cauză utilizarea acestora se limitează la aplicațiile specializate.

Microcontrolerele moderne oferă performanța necesară la preț mic. Din cauza numărului mare de producători și a concurenței prețurile acestor sisteme sunt în scădere continuă. Pe piața modernă există o gamă destul de largă de microcontrolere. Acest fapt le face destul de accesibile și permite dezvoltarea sistemelor la costuri destul de mici care sunt determinate de costurile de proiectare a cablajului imprimat, costul mijloacelor de proiectare și costul microcontrolerului.

Circuitele logice programabile – CPLD/FPGA dispun de performanță superioară în aplicațiile care necesită procesare paralelă. Circuitele reconfigurabile îmbină avantajele de performanță ale logicii cablate cu flexibilitatea sistemelor programabile [51]. În cazul aplicațiilor specializate cu un număr relativ mic de produse, utilizarea circuitelor reconfigurabile devine destul de eficientă din cauză că producerea circuitelor specializate ASIC necesită investiții destul de mari.

După tipul conexiunilor programabile, circuitele FPGA pot fi clasificate ca: circuite cu memorie Static Random Access Memory (SRAM) și circuite cu antifuzibile. Logica circuitelor FPGA este implementată cu ajutorul "tabelelor de adevăr" realizate cu ajutorul celulelor de memorie statică, unde intrările funcțiilor sunt aplicate la liniile de adresă a acestor tabele. Un tabel de $2 \times n$ celule de memorie poate implementa orice funcție logică cu n intrări. Tabele de adevăr împreună cu bistabilele, formează blocuri logice configurabile. Aceste blocuri logice sunt interconectate între ele și cu pinii de intrare/ieșire. Programarea circuitelor cu memorie SRAM se realizează prin înscrierea configurației în celule de memorie statică. Programul de configurare a circuitului reconfigurabil se încarcă serial prin intermediul interfeței Joint Test Action Group (JTAG), sau circuitul îl poate citi serial dintr-o memorie de configurare, la conectarea circuitului la alimentare. Biții cuvântului de configurare setează valorile celulelor de memorie de configurare din circuit, setând valorile tabelelor de adevăr. Circuitele cu memorie SRAM sunt reprogramabile, fapt ce le face foarte flexibile. Aceste circuite pot fi reactualizate nemijlocit în interiorul sistemului proiectat, înlăturând erorile comise la etapa de proiectare [45]. Din această categorie de circuite FPGA fac parte circuitele următorilor producători: Altera, Xilinx, Atmel.

Circuite reconfigurabile cu antifuzibile pot fi programate doar o singură dată. Din cauză că antifuzibilele au dimensiuni reduse, arhitecturile bazate pe acestea pot conține milioane de elemente logice. Un alt avantaj al acestor circuite îl reprezintă nivelul foarte înalt de securizare, care împiedică copierea algoritmilor de funcționare și a structurilor hardware implementate. Circuitelor FPGA cu antifuzibile sunt produse de următorii producători: Actel, Quicklogic, Cypress.

Pe lângă circuitele logice programabile cu memorie SRAM și cu antifuzibile se mai folosesc și circuitele CPLD/FPGA care păstrează programul de configurare într-o memorie

Electrically Erasable Programmable Read-Only Memory (EEPROM). Aceste circuite nonvolatile pot fi reconfigurate prin încărcarea unui nou cuvânt de configurare în memoria EEPROM.

Dezvoltarea industriei dispozitivelor mobile a contribuit la dezvoltarea sistemelor SoC. Aceste soluții conțin practic tot necesarul pentru proiectarea sistemelor de calcul performante: unitate centrală de procesare cu unul sau câteva nuclee, interfețe de comunicare cu echipamentele periferice, coprocesoare și interfețe grafice, etc. Datorită interfețelor de comunicare încorporate, aceste platforme au creat o nouă clasă de sisteme de calcul – numită Single Board Computer (SBC) [52, 53]. Apariția noilor tehnologii prin dezvoltarea procesoarelor neuronale ar putea soluționa o mulțime de probleme ingineresti din domeniul procesării datelor. Creșterea performanței sistemelor cu inteligență artificială depinde direct de performanța procesoarelor specializate în realizarea procesului de inferență. Antrenarea eficientă a RNA este realizată adesea cu ajutorul acceleratoarelor grafice. Recent Intel a dezvoltat noi procesoare specializate în inteligența artificială, utilizând tehnologii inspirate de arhitectura creierului. În aceste procesoare neuromorfice informația este prezentată prin impulsuri, asemănător ca în neuronii reali.

Utilizarea acestor soluții permite extinderea funcționalităților sistemelor tradiționale în domeniile de consum și industrial.

1.4 Formularea problemei de cercetare

Cercetarea procesului de cunoaștere umană pentru ca sistemele tehnice să poată prelua performanțele acestuia este în vizorul mai multor centre de cercetare. În perspectivă sistemele tehnice trebuie să reacționeze în **condiții de incertitudine** sporită, să gestioneze în mod fiabil evenimentele neprevăzute, să se adapteze rapid la schimbarea sarcinilor și capacităților proprii. Tehnologia care permite realizarea sistemelor de luare a deciziilor o constituie planificarea automatizată combinată cu autodiagnosticul și autoevaluarea. Capacitățile respective îi permit sistemului să-și planifice propriile acțiuni, să reacționeze la defecte sau alte evenimente neprevăzute și să-și poată adapta comportamentul în circumstanțe variabile în timp [1-4].

Pornind de la problema studiului sistemelor tehnologice se propune cercetarea clasei de sisteme în care procesul de comandă și control este asigurat total sau parțial de către operatorul uman, iar comportamentul decizional al acestuia poartă deseori un **caracter intuitiv** și este realizat în condiții de incertitudine [16]. Astfel presupunând că pentru această clasă de probleme nu pot fi obținute soluții clasice generale de automatizare, iar procesele tehnologice studiate pot fi desfășurate doar cu implicarea directă a expertului uman, rezultă supoziția că în procesul

decizional expertul se poate baza adesea doar pe intuiție și propria experiență acumulată. Problema automatizării acestor procese este una foarte complicată pentru soluționarea căreia se recurge la utilizarea diverselor metode alternative cum ar fi utilizarea inteligenței artificiale. Cu scopul soluționării acestei probleme se propune cercetarea și elaborarea unui sistem decizional inteligent hibrid, schema de structură a căruia este prezentată în *Figura 1.3*.

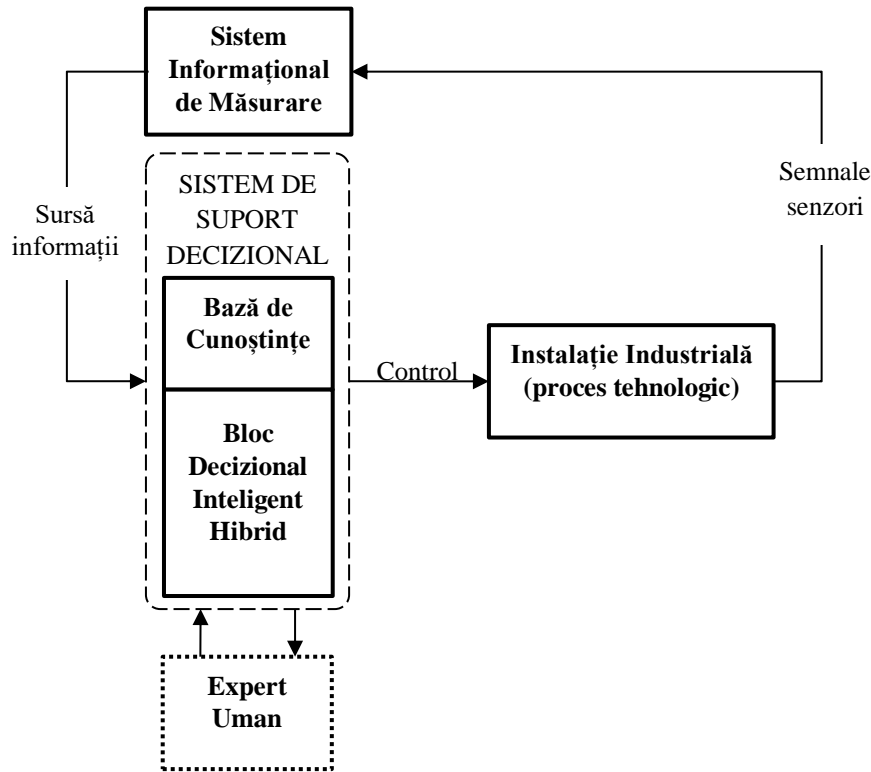


Figura 1.3. Structura sistemului de suport decizional inteligent pentru aplicații industriale.

Din structura sistemului prezentat în *Figura 1.3* face parte sistemul informațional de măsurare, care asigură cu date despre dinamica procesului sistemul de suport decizional la baza căruia se află o bază de cunoștințe structurate extrase de la expertul uman și blocul decizional inteligent hibrid. Acesta poate realiza suportul decizional al expertului uman în cazul controlului asistat, sau poate realiza controlul automat al procesului tehnologic.

În aceste condiții se conturează problema extragerii cunoștințelor expertului uman cu scopul preluării experienței acestuia și construirea sistemelor decizionale în baza cunoștințelor extrase. Astfel apare necesitatea dezvoltării metodelor și algoritmilor de extragere și structurare a cunoștințelor cu scopul aplicării acestora în proiectarea modelelor sistemelor decizionale. Reieșind din specificul de calcul al proceselor de inferență apare necesitatea asigurării suportului hardware a acestora prin proiectarea de arhitecturi specializate.

Pentru cercetarea posibilității extragerii cunoștințelor expertului uman și preluării experienței acestuia este necesar de apelat la un proces de producție sau un sistem tehnologic caracterizat prin:

- prezența operatorului uman în bucla de reacție a procesului de producție;
- procesul de producție este caracterizat de un anumit grad de ambiguitate;
- procesul de producție se desfășoară în condiții de incertitudine.

Pentru asigurarea bazei tehnico-experimentale a cercetărilor științifice a fost selectat procesul industrial de turnare a microfiredelor prin metoda Taylor-Ulitovsky [54, 55], care îndeplinește toate cerințele enumerate mai sus, iar acțiunile operatorului uman au un caracter intuitiv și sunt realizate în condiții de incertitudine.

Ca dovadă a complexității aplicării metodelor clasice în automatizarea procesului de turnare a microfiredelor poate servi faptul că până în prezent cele mai utilizate metode de turnare sunt cea manuală sau cea semiautomată, în bucla de reacție a căroră se află operatorul uman. Acest fapt dovedește încă odată necesitatea unui sistem decizional inteligent care ar asigura suportul decizional și automatizarea unor astfel de procese. Afirmatia făcută reprezintă o constatare a unui fapt real și nicidecum nu tinde să nege în general posibilitatea soluționării acestei probleme prin metodele clasice. Complexitatea și neliniaritatea procesului studiat, numărul foarte mare de variabile care îl descriu sau lipsa de informație despre variabilele caracteristice ale procesului și despre dependențele dintre acestea, toate acestea pot constitui cauza eșecului aplicării metodelor clasice.

Prezența operatorului uman în procesul de turnare a microfiredelor confirmă faptul că această problemă în general poate fi soluționată și chiar indică direct una din soluțiile existente – operatorul uman. Acesta, fiind ghidat de experiența și intuiția sa în procesul decizional, oferă, poate nu o soluție optimă, dar totuși o soluție acceptabilă din punct de vedere al cerințelor de calitate. În situația în care nu se dispune de nici o soluție optimă, generală, este binevenită chiar și soluția aproximativă oferită de către operatorul uman, împreună cu avantajele și dezavantajele acesteia. În aceste noi condiții, problema se reduce la extragerea experienței operatorului uman și replicarea algoritmului decizional al acestuia. În caz ideal se propune generalizarea algoritmului decizional și eliminarea influenței factorului uman pentru a obține un model decizional eficient.

Influența factorului uman asupra procesului tehnologic este caracterizată de ambiguitatea indusă în procesul decizional prin luarea deciziilor incorecte, prin luarea deciziilor diferite în condiții inițiale similare sau prin luarea deciziilor identice în condiții inițiale diferite. Se poate crea impresia că utilizarea experienței operatorului uman, procesul decizional al căruia nu este bine definit, poate cauza soluții eronate, însă specificul acestui proces constă nu în excluderea

deciziilor incorecte din procesul decizional, ci în minimizarea influenței acestora asupra rezultatului final. Astfel alternând decizii corecte și eronate, conform unei tactici cunoscute doar de către operatorul uman, acesta evită efectul cumulativ al erorilor produse pe parcursul întregului proces tehnologic, asigurând integral calitatea necesară a microfirului.

În urma studiului procesului industrial de turnare a microfiredelor condus de către un operator experimentat a fost înaintată ideea dezvoltării și implementării unui sistem inteligent fuzzy de comandă și control al procesului de turnare [54]. Pentru a estima posibilitatea realizării controlului procesului de turnare, eficienței procesului de comandă și a control în timp real a parametrilor microfiredelor a fost propusă abordarea care ar utiliza tehnicile inteligente [26, 27, 30, 56]. Aplicarea tehnicilor moderne de inteligență artificială în procesele industriale oferă un potențial aplicativ enorm la comanda și controlul acestora. Metodele și algoritmi logicii fuzzy [43, 57-66] sunt implementate în procesele de comandă și control fie separat [67-71], fie combinat cu alte metode și tehnici inteligente [56, 72, 73].

Structura conceptuală a sistemului inteligent fuzzy de comandă și control al procesului de turnare a microfiredelor este prezentată în *Figura 1.4*. Sistemul poate avea în componența sa un bloc de control inteligent fuzzy (sau neuronal). Legătura dintre blocul de control și sistemul tehnologic se realizează cu ajutorul blocului de achiziție. La etapele inițiale de colectare a bazei de cunoștințe și de testare este necesară prezența operatorului uman în bucla de luare a deciziilor pentru controlul procesului de turnare și asigurarea parametrilor de calitate a microfiredelor.

Inițial au fost determinate cele mai importante variabilele de sistem care pot fi utilizate în procesul de control:

RL – rezistența lineară;

PTS – presiunea în tubul de sticlă.

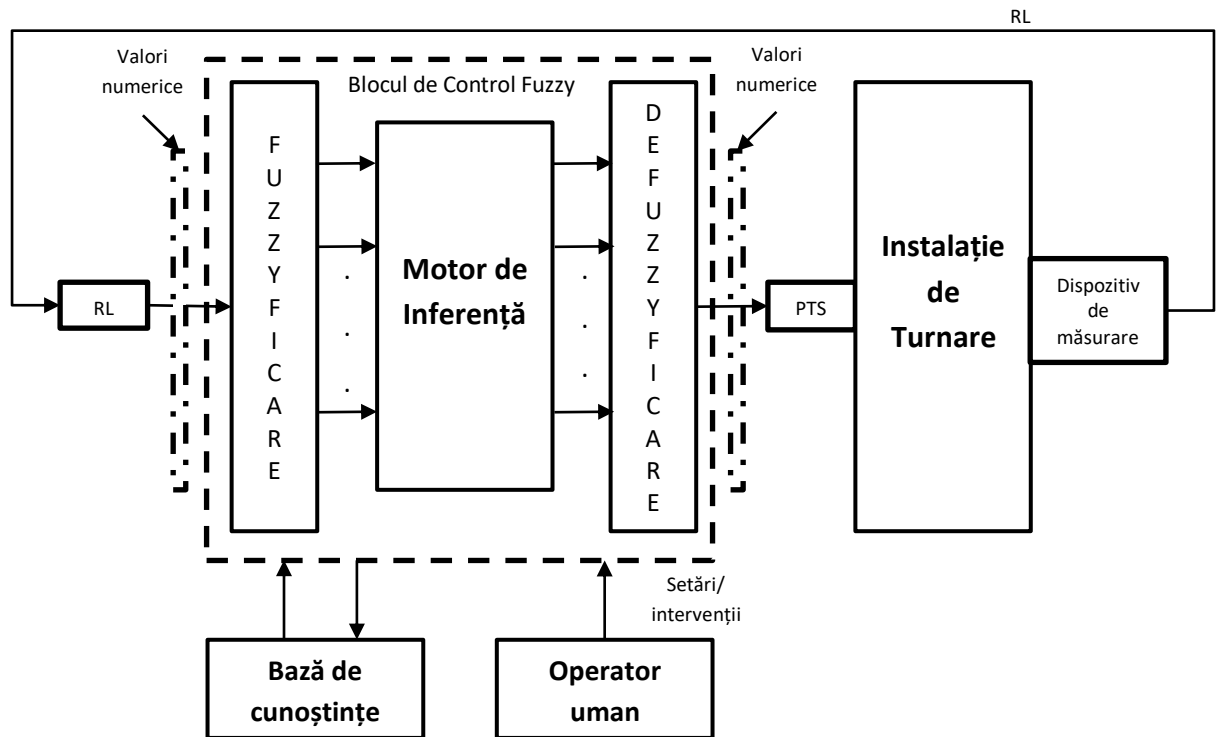


Figura 1.4. Structura sistemului inteligent de comandă și control.

Variabila de control PTS este una esențială în conducerea procesului tehnologic. În *Figura 1.4* subsistemul de control inteligent este reprezentat cu un bloc de tip fuzzy [55].

O problemă specială în construcția sistemelor fuzzy ține de alegerea funcțiilor de apartenență. În [54] variabilelor lingvistice li s-au atribuit următoarele funcții de apartenență:

Funcția triunghiulară:

$$\mu_1(x, a, b, c) = \begin{cases} 0, & \text{if } x \leq a; \\ \frac{x-a}{c-a}, & \text{if } a < x \leq c; \\ \frac{b-x}{b-c}, & \text{if } c < x < b; \\ 0, & \text{if } x \geq b \end{cases} \quad (1.1)$$

Funcția gaussiană:

$$\mu_2(x, a, b) = \exp\left[-\frac{(x-a)^2}{2b^2}\right] \quad (1.2)$$

Funcția sigmoidă:

$$\mu_3(x, a, b) = \{1 + \exp[-a(x-b)]\}^{-1} \quad (1.3)$$

Funcțiile de apartenență prezentate mai sus au fost propuse reieșind din specificul teoretico-practic al procesului tehnologic cercetat. Pentru asigurarea procesului decizional a fost

definit și setul de reguli de inferență [54, 55]. Reieșind din cele expuse pot fi formulate următoarele întrebări, respectiv probleme de cercetare:

1. În ce măsură funcțiile de apartenență propuse corespund cerințelor înaintate față de sistem și dacă acestea pot asigura aplicabilitatea practică a sistemului decizional din care fac parte. Aceste întrebări își au originea inclusiv în faptul că operatorul uman reacționează intuitiv și deseori nu este în stare să formuleze exact cunoștințele pentru a transmite cât mai eficient experiența sa. Din aceste motive apare problema captării experienței operatorului uman, extragerii cunoștințelor și reprezentarea acestora într-o formă sau alta pentru a le utiliza ulterior în construirea sistemelor decizionale inteligente.

2. Care sunt tehnologiile ce pot fi aplicate pentru implementarea sistemului decizional inteligent, dacă pot fi aplicate alte metode de proiectare a sistemelor inteligente în cazul dat și care este aplicabilitatea acestor metode în situația respectivă. Din cauza specificului inferențial al fiecărei tehnici de dezvoltare a sistemelor inteligente apare problema suportului hardware al procesului de implementare a sistemului de control inteligent. Astfel se pune problema proiectării arhitecturilor de calcul specializate pentru realizarea proceselor de inferență specifice fuzzy și neuronale.

3. Care ar fi arhitecturile hardware potrivite pentru implementarea proprietăților de adaptabilitate și de autoorganizare ale metodelor și modelelor propuse. Din aceste motive se propune identificarea și proiectarea unor arhitecturi hardware care ar fi în stare să valorifice potențialul sistemelor adaptive și al celor cu autoorganizare, fiind totodată ușor reconfigurate pe parcursul procesului de cercetare și implementare a sistemelor decizionale hibride.

Cercetările teoretice și experimentale în direcția extragerii cunoștințelor operatorilor de înaltă calificare urmează a fi realizate în paralel cu dezvoltarea structurii sistemului de comandă și control. Ca metode de soluționare a clasei de probleme menționate anterior și a problemei captării experienței operatorului uman în special, se propun metodele bazate pe logica fuzzy și cele bazate pe calculul neuronal. La etapa de analiză și studiu prealabil a fost luată decizia utilizării anume a acestor metode din cauza specificului lor: logica fuzzy oferă posibilitatea unui răspuns flexibil la intrări vagi și imprecise, iar rețelele neuronale artificiale oferă flexibilitate prin acceptarea de posibile intrări imprecise [17, 43, 74].

Pentru aplicarea și implementarea cu succes a metodelor de reprezentare a cunoștințelor prin reguli de inferență este necesar de a realiza următoarele activități:

- identificarea parametrilor și variabilelor caracteristice procesului tehnologic de turnare a microfirelor;

- colectarea datelor experimentale care caracterizează reacția operatorului uman la condițiile de desfășurare a procesului tehnologic cu scopul acumulării datelor statistice brute;
- validarea datelor colectate pentru confirmarea corespunderii parametrilor microfiredorurnate cu cerințele standardizate de calitate, realizată la etapa de control al calității;
- preprocesarea datelor brute prin filtrarea datelor cu scopul de a minimiza influența negativă a deciziilor incorecte asupra modelului decizional final;
- determinarea calificativelor variabilelor logice fuzzy cu care expertul uman operează intuitiv în procesul de luare a deciziilor;
- identificarea funcțiilor de apartenență a calificativelor variabilelor caracteristice și a parametrilor arhitecturali ai rețelei neuronale destinate pentru descrierea procesului;
- extragerea cunoștințelor din experiența operatorului uman colectată, prin identificarea regulilor de inferență de care se ghidează acesta și prin ajustarea ponderilor rețelei neuronale artificiale;
- implementarea modelelor de simulare a proceselor decizionale obținute;

1.5 Concluzii la capitolul 1

În rezultatul cercetărilor efectuate în domeniul sistemelor inteligente au fost analizate metodele de extragere și formele de reprezentare a cunoștințelor pentru a putea fi utilizate în cercetarea și dezvoltarea sistemelor de producție inteligente. A fost evidențiată importanța automatizării proceselor de producție industrială și au fost determinate aspectele cu impact major asupra eficienței implementării sistemelor inteligente de producție. În contextul proceselor tehnologice specifice au fost analizate diverse metode de automatizare și estimată aplicabilitatea acestora. S-a constatat relevanța practică a acestor metode.

În contextul suportului arhitectural au fost analizate posibilitățile și limitele arhitecturilor/sistemelor de comandă și control în procesele de producție, în special a celor inteligente. Au fost analizate tehnologiile și mijloacele disponibile pentru elaborarea și dezvoltarea soluțiilor inteligente și de suport decizional în sisteme de producție industrială cu scopul implementării unor abordări inovative în procesele de producție specifice la care participă direct operatorul uman. Analiza efectuată a permis formularea problemei științifice și a direcției de cercetare.

2. CERCETAREA ȘI DEZVOLTAREA METODELOR ȘI ALGORITMILOR DE LUARE A DECIZIILOR ÎN CONDIȚII DE INCERTITUDINE

2.1 Abordarea problemei de identificare a funcțiilor de apartenență

Ținând cont de creșterea rolului și a ponderii sistemelor inteligente în domeniul aplicațiilor industriale devine tot mai actuală problema cercetării, dezvoltării și implementării unor soluții avansate, specializate de preluare a experienței expertului uman.

Procesele industriale specifice pot fi caracterizate printr-un grad anumit de ambiguitate în procesul de luare a deciziilor [75]. Problema de bază în procesul de extragere a cunoștințelor o reprezintă structurarea datelor cu scopul utilizării acestora în continuare pentru construirea sistemelor de luare a deciziilor. În particular, problema determinării funcțiilor de apartenență la etapa de proiectare a sistemelor fuzzy, este una foarte actuală în condițiile creșterii cerințelor față de automatizarea proceselor industriale dictate de concurența economică și eficientizarea proceselor tehnologice utilizate [7, 76]. În aceste condiții dezvoltarea sistemelor fuzzy depinde direct de calificarea expertului uman și de capacitatea acestuia de a formula caracteristicile sistemului. Astfel precizia sistemului proiectat depinde de posibilitatea expertului uman de a formula regulile de inferență, de a specifica numărul variabilelor fuzzy și a funcțiilor de apartenență ale acestora [77, 78]. Pe lângă faptul că proceselor caracterizate de condiții de incertitudine le este specific și un anumit factor de ambiguitate, o influență negativă asupra proprietăților sistemului poate avea inclusiv și procedura de identificare a numărului de variabile fuzzy și a funcțiilor de apartenență ale acestora din cauza că nici chiar teoria care se află la baza acestei metodologii nu oferă un răspuns concret la întrebările care trebuie să fie numărul de variabile și care trebuie să fie funcția de apartenență a fiecărei variabile în diferite condiții specifice. Soluționarea acestor momente destul de importante rămân pe seama expertului uman și adesea acestea pot fi soluționate doar pe cale experimentală, iar eficiența soluțiilor obținute depinde doar de experiența expertului. În condițiile date formalizarea și standardizarea acestor proceduri devine problema principală pentru inginerii care se ocupă de proiectarea sistemelor fuzzy. Soluționarea acestei probleme poate duce la eficientizarea lucrului inginerilor prin oferirea acestora posibilității de implementare a unui algoritm pentru automatizarea proceselor respective.

După cum se știe, alegerea corectă a formei funcției de apartenență nu este deloc o sarcină trivială iar o funcție de apartenență unică pentru fiecare concept vag nu poate fi definită

datorită incertitudinii, diversității și diferențelor individuale ale datelor. Din aceste motive se consideră că funcțiile de apartenență pot fi selectate în mod arbitrar, sau intuitiv, fără a urma careva proceduri prestabilite. Cu toate acestea trebuie să se țină cont cât de natura datelor utilizate pentru identificarea funcțiilor de apartenență, atât și de natura procesului pe care aceste îl caracterizează [77-79].

Procesul decizional în cadrul procesului de producție, ca regulă, poate fi caracterizat de un grad oarecare de incertitudine, generat de incompletitudinea, imprecizia, fragmentarea, validitatea, neclaritatea sau contradicția informației [75, 77]. În special, este posibil ca expertul uman să nu poată descrie exact naturală procesului în ceea ce privește starea sau dinamica acestuia, sau să nu poată stabili legătura cauză efect; doi experți pot percepe și interpreta diferit informația punând accent pe diferite aspecte ale acesteia; acuratețea achizițiilor de date, care poate fi realizată cu un anumit grad de eroare; efectele negative ale zgomotului sau particularitățile individuale ale procesului [76, 77]. În cazul proceselor industriale specifice, în care realizarea experimentelor poate scoate procesul din starea de bună funcționare și respectiv poate cauza pierderi enorme ca rezultat al distrugerii parțiale sau totale a sistemului, apare nevoia de a identifica cât mai precis funcțiile de apartenență iar precizia și exactitatea acestora poate influența negativ evoluția acestor procese specifice, deoarece atât achiziția datelor valide în timp real, cât și aplicarea promptă a planului strategic reprezintă condițiile necesare pentru reducerea costurilor și menținerea procesului pe traiectoria dorită [80, 81].

Pentru creșterea calității produsului finit, apare necesitatea aplicării sistemelor de suport decizional sau de luare a deciziilor, pentru a îmbunătăți performanța operatorului uman sau pentru automatizarea completă a procesului de producție [75, 76]. Sistemele decizionale propuse în literatură se bazează pe modele matematice clasice de rezolvare a problemelor de automatizare și de suport decizional [82-86]. Cu toate acestea, soluțiile respective nu țin cont de incertitudinea și ambiguitatea datelor de intrare și nici nu reproduc procesul de luare a deciziilor al experților aplicat într-un domeniu vag. De fapt, modelul decizional pe care îl utilizează fiecare operator uman în procesul de producție este adesea perceput cu incertitudine.

Aplicarea logicii fuzzy pentru soluționarea problemelor de luare a deciziilor argumentează odată în plus proprietatea acestei metodologi de a soluționa problema gestionării cunoștințelor imprecise și incerte fapt care asigură suportul procesului de luare a deciziilor [75, 79]. Mai mult decât atât, logica fuzzy este destul de aproape de limbajul natural și permite prezentarea cunoștințelor într-o formă naturală. Acest lucru face ca sistemele decizionale bazate pe logica vagă să fie mai preferabile pentru utilizatorii umani decât alte sisteme de suport

decizional din cauza inteligibilității raționamentului semantic utilizat pentru definirea variabilelor lingvistice și a regulilor de inferență.

În literatura de specialitate există mai multe metode propuse pentru identificarea funcțiilor de apartenență [82-86]. Unele din ele au o abordare bazată pe distribuția de probabilitate sau pe distribuția de posibilități în cazul în care datele în baza cărora este aplicată metoda de identificare sunt ambigui sau incerte [77]. Deseori, datele statistice care descriu procesul de studiu pot fi destul de ambigui și pot avea un oarecare factor de incertitudine. Astfel apartenența acestora la anumite categorii poate fi calculată în baza unor clustere [82, 83].

O soluție asemănătoare soluției propuse este descrisă în [82], dar care dispune și de o serie de diferențe conceptuale specifice între cele două abordări. Această abordare se referă la ideea calculului probabilității apartenenței valorilor fluxurilor de intrare la categoriile descrise de către variabilele fuzzy. Mai mult decât atât, în abordarea propusă se oferă o metodă specifică de estimare a unor indici caracteristici ai procesului de producție care pot fi utilizați pentru o descriere mai exactă a procesului decizional al operatorului uman și a entităților cu care acesta interacționează în procesul de comandă și control.

În acest paragraf, este prezentată și dezvoltată o metodă de calcul a funcțiilor de apartenență care pot fi definite pentru dezvoltarea sistemelor de luare a deciziilor în condiții de incertitudine în procesele de producție care la moment includ operatorul uman în bucla de reacție [79].

La baza sistemelor fuzzy se află funcțiile de apartenență utilizate pentru reprezentarea valorilor de intrare neclare. Acestea pot fi generate prin diferite moduri, unul dintre care este utilizarea expertului pentru a defini funcțiile. Această metodă însă nu este întotdeauna aplicabilă. Astfel definirea automată a funcțiilor de apartenență reprezintă o problemă destul de actuală. În aceste condiții au fost dezvoltate mai multe metode de identificare a funcțiilor de apartenență din seturi de date.

Metoda de generare automată a funcțiilor de apartenență prezentată în [82, 83] divizează setul inițial de date în clase care pot fi utilizate pentru obținerea funcțiilor de apartenență. Această metodă constă în gruparea datelor în clase, apoi pentru clasele obținute sunt generate funcțiile de apartenență [82-86]. Aplicarea metodei prezentate în [82] pentru identificarea funcțiilor de apartenență în baza datelor incerte achiziționate de la expertul uman poate duce la apariția unui număr destul de mare de clase și de reguli de inferență, fapt ce poate cauza creșterea complexității motorului de inferență. Un alt dezavantaj al metodei analizate îl constituie faptul că aceasta este orientată doar la identificarea funcțiilor de apartenență clasice și nu

prevede posibilitatea dezvoltării unor funcții personalizate care ar putea moșteni caracterul comportamental al expertului.

Metoda propusă este una bazată pe procesarea fluxurilor de date și are unele asemănări cu metoda descrisă în [82], dar reprezintă o abordare nouă care are drept scop eliminarea neajunsurilor menționate mai anterior. Pentru a atinge acest obiectiv se propune, pe de o parte, să se micșoreze numărul maxim al variabilelor lingvistice. Pe de altă parte, se propune posibilitatea definirii personalizate a funcțiilor de apartenență. În acest context a fost elaborat un algoritm de identificare a funcțiilor de apartenență în baza procesării și interpretării fluxurilor de date intrare/ieșire diagrama de activitate a căruia este prezentată în *Figura 2.1*.

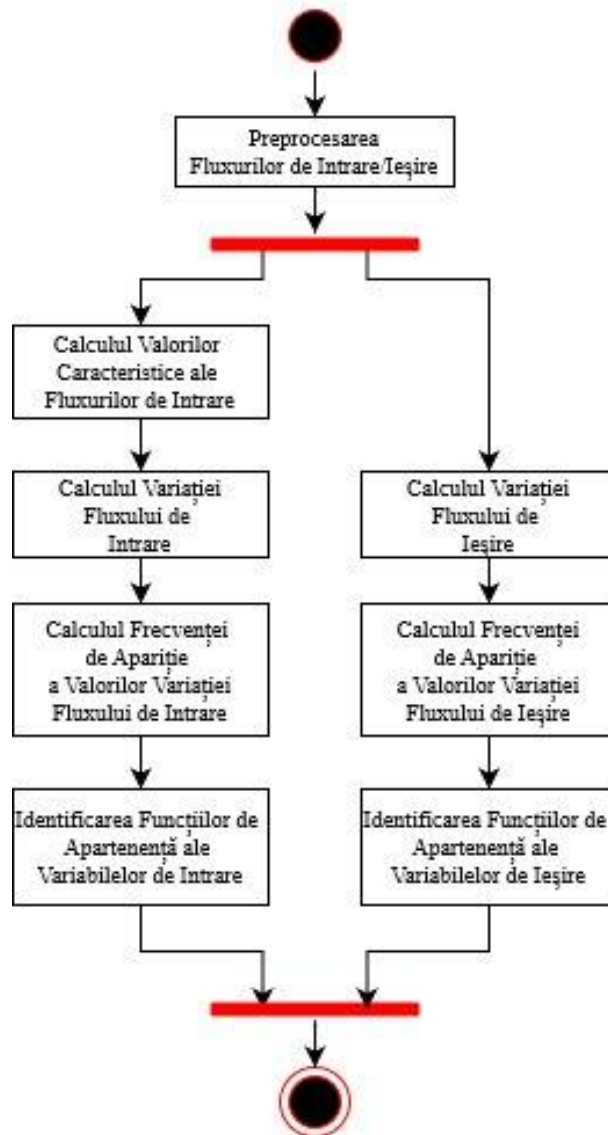


Figura 2.1 Diagrama de activitate a metodei de identificare a funcțiilor de apartenență.

În condițiile în care procesul tehnologic este controlat de către operatorul uman care asigură calitatea produsului finit, pot fi înaintate ipotezele că operatorul uman este unul

experimentat și datele achiziționate în urma monitorizării procesului de producție sunt valide. Pornind de la aceste ipoteze se pot trage concluziile că evoluția procesului este una acceptabilă iar procesul se află în starea de bună funcționare (a se vedea *Figura 2.5*).

Pentru a identifica funcțiile de apartenență a fost dezvoltat un algoritm de extragere a acestora, care se bazează pe procesarea și interpretarea fluxurilor de date de intrare/ieșire. Procedura de extragere a funcțiilor de apartenență poate fi împărțită în cinci etape: preprocesarea fluxurilor de intrare/ieșire, calculul valorilor caracteristice pentru fluxurile de intrare, calcularea variației fluxurilor de intrare/ieșire, calculul frecvenței de apariție a valorilor variației fluxurilor de intrare/ieșire obținute la etapa anterioară și identificarea funcțiilor de apartenență, *Figura 2.1*. Aceste etape sunt descrise pe scurt în continuare.

Preprocesarea fluxurilor de intrare/ieșire. În funcție de setul de date disponibile, poate fi necesar să se efectueze preprocesarea datelor de intrare/ieșire pe măsură ce procesul este modelat. La această etapă, datele sunt filtrate în baza valorilor care reprezintă parametrii de interes obținuți ca urmare a verificării calității produsului final. Datele ar trebui să fie procesate cu scopul de a exclude eșantioanele care nu corespund cerințelor de calitate. Așadar, filtrarea datelor este utilă pentru eliminarea eșantioanelor eronate, inconsistente sau incerte. Această procedură își propune să reducă la minim influența negativă a acestor sample-uri asupra modelului procesului decizional final. Această etapă poate fi considerată opțională în cazul în care se asigură calitatea și validarea datelor experimentale colectate. Condiția dată poate fi îndeplinită doar dacă procesul se desfășoară într-un mediu ideal sub controlul unui operator foarte experimentat. Pentru majoritatea aplicațiilor reale, etapa respectivă devine obligatorie.

Procedura de filtrare poate fi realizată paralel asupra fluxurilor de intrare/ieșire pentru identificarea în fluxurile de intrare a valorilor care nu corespund criteriilor prestabilite de cerințele impuse procesului. Prin urmare, filtrarea constă în eliminarea sample-urilor de date identificate din fluxurile de intrare împreună cu sample-urile corespunzătoare acestora din fluxurile de ieșire. Această etapă va afecta atât fluxurile de intrare cât și cele de ieșire conform relațiilor (2.1) și (2.2):

$$\varphi_{out}^f = \{x | x \in \varphi_{out}^{nf}, Min \leq x \leq Max\} \quad (2.1)$$

$$\varphi_{in}^f = \{x | x \rightarrow y, x \in \varphi_{in}^{nf}, y \in \varphi_{out}^f\} \quad (2.2)$$

unde φ_{out}^f reprezintă fluxul de ieșire filtrat, φ_{out}^{nf} - fluxul de ieșire nefiltrat, φ_{in}^f reprezintă fluxul de intrare filtrat iar φ_{in}^{nf} - fluxul de intrare nefiltrat.

Ca urmare a preprocesării, vor fi obținute noi fluxuri de date intrare/ieșire, care vor fi construite pe baza fluxurilor inițiale (a se vedea *Figura 2.6*). Datele obținute în urma prelucrării pot fi utilizate direct la etapa următoare.

Calculul valorilor caracteristice. Calculul valorilor caracteristice ale fluxurilor de intrare este una dintre cele mai importante etape deoarece acești parametri caracterizează numeric procesul de luare a deciziilor al operatorul uman în timp ce sistemul este supravegheat de către acesta. Valorile caracteristice ale variabilelor de intrare caracterizează numeric reperele de care se ghidează operatorul în procesul decizional. Identificarea acestor valori se face prin calcularea numărului de apariții pentru fiecare valoare din fiecare flux de intrare în parte pentru a determina valoarea cu cea mai mare frecvență de apariție. Astfel valoarea cu cel mai mare număr de apariții din fiecare flux de intrare și reprezintă valoarea caracteristică a acestei variabile de intrare. Cu alte cuvinte, sunt determinate o serie de caracteristici relevante cu privire la procesul de luare a deciziilor. Acești parametri indică operatorului că procesul tehnologic se desfășoară normal și nu necesită intervenții.

Calcularea variației fluxurilor de intrare/ieșire. Din etapa anterioară, putem concluziona că operatorul în procesul de luare a deciziilor este ghidat de valorile caracteristice ale variabilelor de intrare, iar toate intervențiile sale sunt efectuate relativ la acele valori. Pentru a prelua modelul decizional cât mai exact, se determină variația parametrilor de intrare în raport cu valorile de referință ale variabilelor de intrare. Astfel, pentru fiecare valoare din fluxurile de intrare, se determină abaterea, care este definită ca diferența dintre valoarea curentă a variabilei de intrare și valoarea sa caracteristică (de referință) conformitate cu (2.17). Pentru fluxurile de ieșire, variația este calculată în raport cu valoarea anterioară a variabilei de ieșire din același flux de date (2.18).

Urmând procedura de mai sus, se determină domeniile de valori pentru fiecare variabilă de intrare/ieșire, care ilustrează modul și viteza de reacție a operatorului în procesul de luare a deciziilor. Acest fapt oferă posibilitatea de a înțelege modul în care procesul decizional al operatorului uman variază în funcție de variația parametrilor de intrare.

Calculul frecvenței de apariție a valorilor variației fluxurilor de intrare/ieșire. Această etapă este comună atât pentru fluxurile de intrare cât și cele de ieșire. Ea constă în calcularea numărului de apariții pentru fiecare valoare din fluxurile de intrare/ieșire obținute la pasul precedent. Ca urmare a acestei etape, pentru fiecare variabilă de intrare/ieșire, se identifică valoarea cu cel mai mare număr de apariții în flux. Astfel, se identifică cele mai frecvente valori discrete din fluxurile de intrare/ieșire. Acest număr, asociat unei variabile de intrare/ieșire și notat ca N_{Max} va fi utilizat în faza finală.

Identificarea funcțiilor de apartenență. La această etapă, este calculată mulțimea coeficienților de apartenență prin utilizarea rezultatelor obținute la etapa anterioară. Pentru aceasta este necesar să se calculeze numărul de repetări ale valorilor discrete în raport cu numărul de apariții ale valorii cu frecvența de apariție maximă în flux. Pentru a obține funcția de apartenență normalizată, numărul de apariții N_x al valorii discrete x , este împărțit la număr de apariții a celei mai frecvente valori N_{Max} din fluxul considerat, conform relației (2.3):

$$Mf(x)_{in/out} = \frac{N_x}{N_{Max}} \quad (2.3)$$

În acest mod, se obțin funcțiile de apartenență normalizate pentru fiecare variabilă de intrare/ieșire (a se vedea *Figura 2.21*). Ca urmare a acestei proceduri, perechile de date „valoarea variației variabilei de intrare/ieșire - coeficient de apartenență” sunt sortate în ordine crescătoare în funcție de valoarea variației variabilei de intrare/ieșire, relația (2.4).

$$\varphi_{in/out}^s = \varphi_{in/out}^f \uparrow \quad (2.4)$$

Având în vedere natura discretă a coeficienților de apartenență, pentru a obține o funcție de apartenență continuă, definită pe întreaga domeniul de valori, este necesară aplicarea unei metode clasice de interpolare a funcțiilor pe intervale. Pentru a construi funcțiile de apartenență, este necesar să se stabilească forma și punctele de interes ale acestora.

Analizând funcția de apartenență astfel obținută poate fi estimată vizual forma acesteia. Estimarea vizuală poate servi ca indiciu al corectitudinii selectării formei funcției de apartenență doar pentru expertul uman și nu are nici o semnificație pentru sistemele de calcul. Astfel apare problema determinării formei funcției continue de apartenență din funcția de apartenență discretă. Această problemă poate fi soluționată destul de simplu în cazul unor forme simple cum ar fi triunghiulară, (1.1), dreptunghiulară sau trapez, (2.5), însă devine destul de greu de soluționat în cazul formei sigmoide (1.3). Deoarece metodologia respectivă nu oferă un răspuns clar la această întrebare, o soluție parțială poate oferi gruparea tuturor formelor funcțiilor de apartenență în 2 categorii:

- trapez, (2.5), (caz particular triunghi (a se vedea *Figura 2.16, 2.17*), (1.1) – un singur maxim, dreptunghi – câteva maxime) – mai multe maxime;
- personalizată – funcția de apartenență poate repeta exact valorile coeficienților de apartenență sau poate constitui o aproximare a acestora (a se vedea *Figura 2.20, 2.22*).

$$\mu_F(x, a, b, c, d) = \begin{cases} 0, & x < a; \\ \frac{x-a}{b-a}, & a \leq x \leq b; \\ 1, & b < x < c; \\ \frac{d-x}{d-c}, & c \leq x \leq d; \\ 0, & x > d \end{cases} \quad (2.5)$$

Trebuie menționat faptul că în urma calculului coeficienților de apartenență pentru fiecare valoare din flux, pot exista valori din domeniul de valori coeficientul de apartenență al cărora este foarte mic sau chiar egal cu zero (a se vedea *Figura 2.21*). Acest fapt poate fi cauzat de natura procesului, de dinamica acestuia, de proprietățile echipamentului sau de comportamentul și reacția operatorului uman. În rezultat poate fi observat fenomenul în care una sau câteva valori ale coeficientului de apartenență reprezintă valori ale unor minime locale, fapt care poate indica absența valorii respective în fluxul analizat, ceea ce nu justifică corectitudinea acestei valori. În cazul formei trapezoidale a funcției de apartenență această problemă poate fi soluționată destul de simplu prin identificarea punctelor de interes ale trapezului și interpolarea liniară a valorilor dintre acestea (2.6), (2.7), (2.8), (2.9).

$$a = \text{MAX}(x) | Mf(x) = 0, x < y, Mf(y) = 1 \quad (2.6)$$

$$b = \text{MIN}(x) | Mf(x) = 1 \quad (2.7)$$

$$c = \text{MAX}(x) | Mf(x) = 1 \quad (2.8)$$

$$d = \text{MIN}(x) | Mf(x) = 0, x > y, Mf(y) = 1 \quad (2.9)$$

În caz particular pentru $b = c$ se obține funcția de apartenență de formă triunghiulară, (1.1), (a se vedea *Figura 2.16, 2.17*).

În cazul unei forme personalizate a funcției de apartenență soluționarea acestei probleme poate fi obținută fie prin implementarea unei metode de interpolare care ar ignora valorile respective sau prin asigurarea fluxului de date cu tot spectrul de valori din domeniul de valori, (a se vedea *Figura 2.20, 2.22*).

După determinarea formei funcției de apartenență și a metodei de interpolare în cazul formei personalizate de reprezentare pot fi determinate funcțiile de apartenență ale celorlalte variabile fuzzy. Poate fi observat faptul că funcția de apartenență a stării de bună funcționare separă domeniul de valori în două domenii de funcționare proastă: domeniul din partea stângă și cea dreaptă. Aceste domenii pot reprezenta variabile fuzzy funcțiile de apartenență ale cărora pot fi calculate elementar din funcția de apartenență a stării de bună funcționare. Pentru aceasta este destul ca din funcția de apartenență a stării de bună funcționare să se scadă valoarea 1 iar forma funcției obținute în așa mod va reprezenta negația stării de bună funcționare, adică starea de proastă funcționare. Această funcție poate fi separată în două calificative de stânga și de dreapta

în raport cu valorile minime a acesteia din stânga, (2.10) și respectiv dreapta, (2.11), fapt ce va oferi posibilitatea de a opera cu mai multe variabile fuzzy pentru controlul mai exact al procesului.

$$Mf_L^b(x) = \begin{cases} 1 - Mf_C^g(x), & x < y, Mf_C^g(y) = 1; \\ 0, & x \geq y, Mf_C^g(y) = 1. \end{cases} \quad (2.10)$$

$$Mf_R^b(x) = \begin{cases} 0, & x \geq y, Mf_C^g(y) = 1; \\ 1 - Mf_C^g(x), & x < y, Mf_C^g(y) = 1. \end{cases} \quad (2.11)$$

Această metodă asigură suprapunerea parțială a funcțiilor de apartenență alăturate, caracteristică utilă în unele cazuri în care se dorește generalizarea procesului decizional și una nedorită în cazurile în care se dorește preluarea cât mai exactă a comportamentului operatorului.

În cazul în care nu se dorește suprapunerea funcțiilor de apartenență ale variabilelor logice (în cazurile fără ambiguitate), punctele de interes ale formelor funcțiilor de apartenență pot fi definite astfel încât acestea să nu se suprapună decât pe domenii de valori în care doar una dintre acestea poate lua valori diferite de zero.

Pentru cercetarea și aplicarea metodei de extragere a cunoștințelor expertului uman, propusă în paragraful 2.1 și dezvoltarea algoritmului de identificare a funcțiilor de apartenență din fluxurile de date intrare/ieșire, este necesară formarea unei baze de date experimentale. Trebuie menționată importanța influenței comportamentului decizional al operatorului asupra calității produsului final în procesele de producție specifice în condiții de incertitudine [7, 39, 40]. Astfel analiza aplicabilității abordării propuse poate fi realizată doar în baza unui proces de producție specific care îndeplinește anumite cerințe stipulate în paragraful 1.4.

În acest scop, ținând cont de posibilitățile existente, a fost luată decizia de a apela la procesul tehnologic de turnare a microfiredelor pentru utilizarea acestuia ca bază experimentală pentru cercetare, *Figura 2.2* [54, 55].

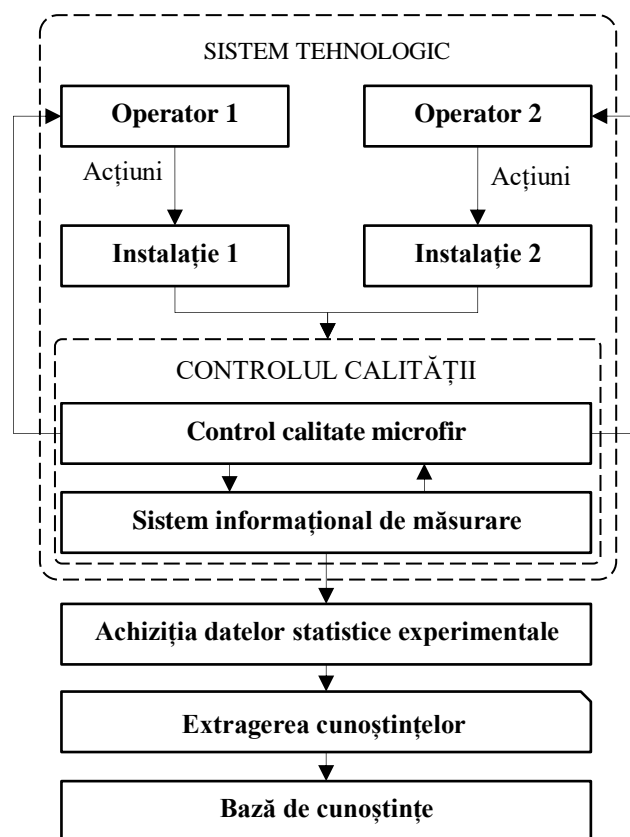


Figura 2.2. Schema lucrărilor experimentale de cercetare.

Pentru formarea bazei de date experimentale au fost identificate resurse necesare pentru demararea procesului de achiziție:

- operatori experimentați care ar menține sub control procesul de producție al microfiredelor și ar asigura totodată calitatea acestora;
- instalații de turnare la care vor fi desfășurate diverse procese tehnologice de turnare a microfiredelor cu proprietăți diferite;
- echipament de măsurare a parametrilor fizici ai microfiredelor;
- echipament de control al calității produsului finit pentru validarea datelor experimentale;
- instrumente software specializate pentru categorizarea și stocarea datelor experimentale într-un format comod pentru utilizare și interpretare.

Experimentele vor fi realizate conform următorului plan:

1. specificarea procesului decizional studiat cu scopul extragerii de cunoștințe;
2. identificarea variabilelor de interes ale procesului de studiu;
3. stabilirea preciziei dispozitivelor de măsurare a parametrilor procesului cercetat;
4. identificarea variabilelor de control și de reacție ale procesului decizional;

5. specificarea cerințelor de calitate ale produsului finit conform fișei cerințelor tehnice;
6. documentarea experimentelor – include toate combinațiile între operatori, instalații de turnare și tipurile de microfir turnat;
7. realizarea experimentului conform planului de activități, analiza comportamentului operatorului uman pe parcursul procesului de producție cu scopul de a evidenția caracteristicile specifice ale acestuia (reguli de inferență, valori caracteristice ale parametrilor de intrare/ieșire);
8. analiza datelor obținute.

La etapa de cercetare a procesului tehnologic de turnare a microfiredelor a fost remarcat faptul că din mulțimea de parametri măsurați la conducerea procesului de turnare, prin metoda Taylor-Ulitovsky, operatorul este ghidat doar de un număr foarte redus de parametri [54, 55]:

P_v – vidul (presiunea) în tubul de sticlă;

R_p – rezistența liniară nominală (diametrul nominal al microfiredului);

iar în cazul în care sistemul de măsurare a rezistenței liniare nominale nu poate asigura precizia necesară a măsurărilor, doar de: vidul în tubul de sticlă, forma picăturii de aliaj, culoarea picăturii de aliaj.

Analizând comportamentul operatorului uman pe parcursul procesului de turnare a microfiredelor se pot evidenția cu certitudine următoarele particularități:

- operatorul reacționează în timp util la schimbările de stare a sistemului;
- operatorul nu ia decizii bazate pe algoritmi sofisticăți sau pe modele matematice;
- operatorul utilizează un număr mic de calificative în procesul decizional;
- operatorul folosește un set redus de date de intrare/ieșire.

După identificarea tuturor parametrilor de control ai procesului de turnare, care reprezintă intrări și ieșiri pentru procesul decizional al operatorului uman, se pot selecta acei parametri utilizarea cărora, în procesul de dezvoltare a sistemului automat de comandă, va furniza un model decizional satisfăcător. În acest sens a fost luată decizia de a utiliza ca parametri de control mulțimea parametrilor măsurați: P_v – presiunea în tubul de sticlă și R_p – rezistența liniară nominală. Se cere specificat faptul că presiunea în tubul de sticlă reprezintă una din intrările de control pentru sistemului de turnare și respectiv una din ieșirile procesului decizional pentru operatorul uman, iar rezistența liniară nominală reprezintă una din ieșirile sistemului de turnare și respectiv una din intrările procesului decizional al operatorului uman. Ceilalți parametri măsurabili nu participă în procesul decizional iar valorile acestora sunt specificate în fișa tehnică în dependență de tipul și caracteristicile microfiredului turnat [54, 55].

Pentru crearea bazei de date experimentale s-a decis de a se colecta doar date valide de la diferite stații de turnare, de la cei mai experimentați operatori, pentru microfibre cu diferiți parametri. Pentru aceasta au fost grupate datele după categoriile menționate și au fost selectate seturile de date colectate de la cei mai experimentați operatori care turnau microfibră cu rezistența de 5-10k Ω pentru care sistemul de măsurare a rezistenței liniare nominale asigură precizia necesară. Aceste proceduri au fost efectuate cu scopul de a compara procesele decizionale ale diferitor operatori în aceleași condiții și în condiții diferite.

2.2 Achiziția, preprocesarea și analiza datelor experimentale

Toate experimentele în domeniu au fost petrecute în incinta companiei „Microfir Tehnologii Industriale” S.R.L. care a dezvoltat un sistem informațional de măsurare [87] și colectare a parametrilor procesului tehnologic de turnare a microfibrelor cu izolație din sticlă pentru controlul calității acestora în regim de fabricație [88]. Toate datele achiziționate în regim de fabricație sunt categorizate și stocate într-o bază de date, accesul la care se realizează printr-un software specializat.

După achiziția în regim de fabricație și stocarea datelor caracteristice procesului tehnologic de turnare a microfibrelor, datele trec prin etapa de validare în care microfibrul trece prin procesul de control al calității în regim postfabricație. Special pentru procesul de control al calității microfibrelor în regim postfabricație a fost proiectat și implementat, de către Universitatea Tehnică a Moldovei în colaborare cu „Microfir Tehnologii Industriale” S.R.L., un sistem informațional de măsurare a parametrilor microfibrelor cu izolație din sticlă, *Figura 2.3*. Sistemul dezvoltat asigură monitorizarea procesului de control al calității prin rebobinarea microfibrului și măsurarea în regim dinamic a parametrilor acestuia *Figura 2.4*.

Sistemul este compus din următoarele componente:

- calculator personal, pe care rulează aplicația;
- subsistem de măsurare a forței coercitive și a diametrului microfibrului;
- subsistem electromecanic de rebobinare a microfibrului în faza de verificare.



Figura 2.3 Sistemul informațional de control al calității microfibrelor cu izolație din sticlă.

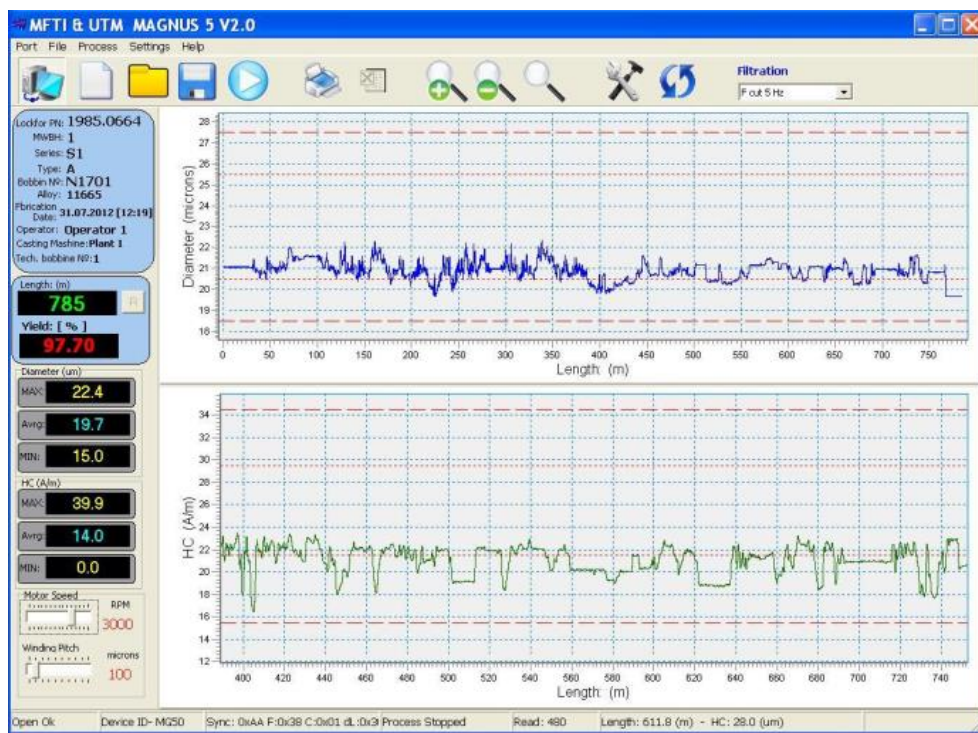


Figura 2.4 Interfața ”operator – sistem tehnologic”.

Utilizând baza de date statistice obținută în procesul de fabricație au fost selectate date care au fost validate la etapa de control al calității. Astfel au fost selectate date statistice de la doi, cei mai experimentați operatori, care au turnat microfir de 5–10kΩ, la două instalații diferite. Pentru analiza acestor date au fost construite graficele prezentate în *Figura 2.5 (a)* și *(b)*.

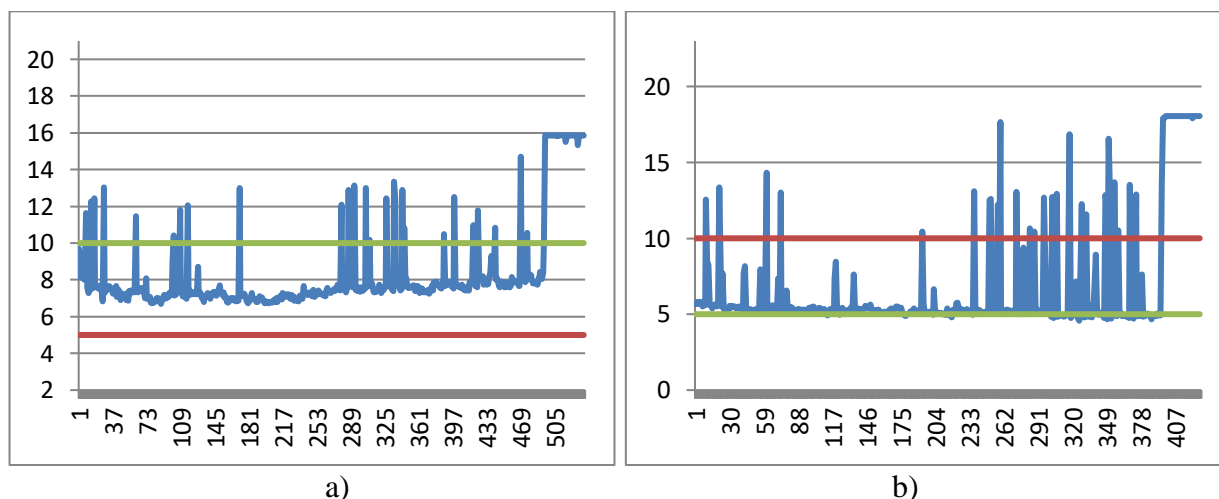


Figura 2.5 Graficul variației în timp a rezistenței liniare a microfirului de 5 – 10kΩ, turnat de către operatorul 1 la instalația 1 (a) și de către operatorul 2 la instalația 2 (b).

Comparând graficele prezentate în *Figura 2.5 (a)* și *(b)* pot fi remarcate trăsăturile caracteristice ale operatorilor. Spre exemplu din *Figura 2.5 (a)* se poate remarca tendința

operatorului de a menține valoarea rezistenței liniare în zona de mijloc a intervalului, fapt ce poate fi explicat prin dorința operatorului de a menține procesul de turnare într-o zonă egal depărtată de limitele de jos și de sus cu scopul de a dispune de timp de reacție, în ambele direcții, în cazul în care procesul se va abate de la mersul normal.

Procesul decizional al celui de-al doilea operator prezentat în *Figura 2.5 (b)* poate fi caracterizat prin tendința acestuia de a menține valoarea rezistenței liniare aproape de limita de jos a intervalului, fapt ce la prima vedere contrazice logicii prin faptul că operatorul își limitează timpul și spațiul de reacție în cazul abaterii procesului de la mersul normal prin variația rezistenței liniare în direcția scăderii acesteia. Explicația utilizării acestei metodologii individuale este justificată pe deplin de faptul că procesului de turnare îi este specifică depășirea preponderent a limitei de sus a intervalului. În acest caz operatorul, prin utilizarea acestei strategii de turnare, asigură minimizarea numărului de depășiri ale intervalului de calitate și stabilește procesul în zona decurgerii normale a acestuia. Astfel utilizarea acestei strategii de turnare caracterizează comportamentul acestui operator ca pe unul care cunoaște în esență specificul procesului de turnare a acestui tip de microfire.

Pentru a exclude influența negativă a sample-urilor care depășesc limitele cerințelor de calitate asupra modelului decizional, datele achiziționate au fost filtrate obținându-se graficele prezentate în *Figura 2.6 (a) și (b)*.

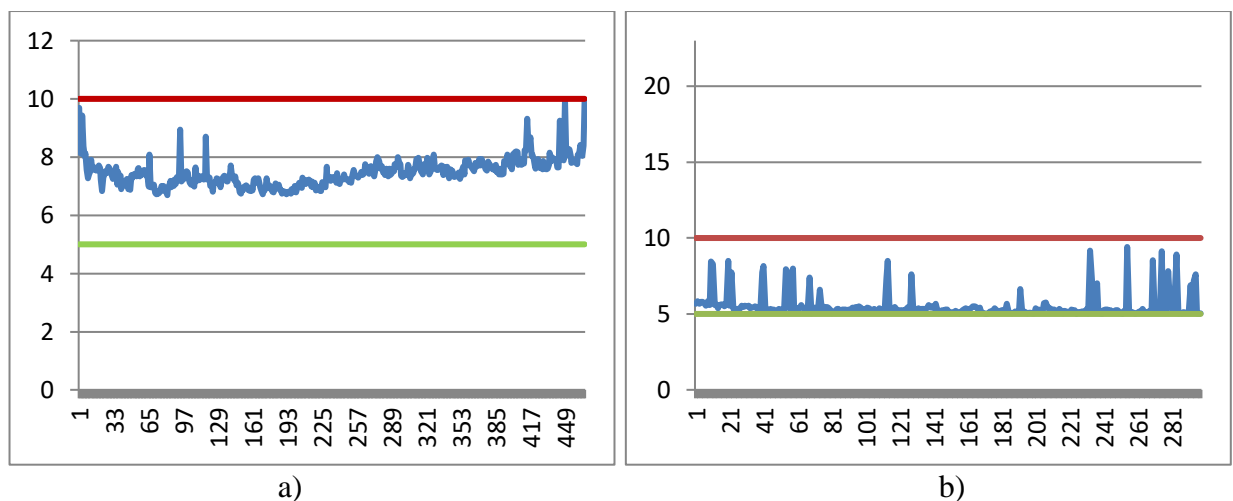


Figura 2.6 Graficul variației în timp a rezistenței liniare a microfîrului de 5 – 10k Ω , turnat de (a) operatorul 1 la instalația 1 și (b) operatorul 2 la instalația 2, după filtrare.

În rezultatul analizei datelor statistice colectate de la primul operator din 536 de sample-uri 467 satisfac cerințele de calitate, ceea ce caracterizează microfîrului turnat ca având o calitate de aproximativ 87%. În cazul celui de-al doilea operator din 428 de sample-uri 298 satisfac

cerințele de calitate. Astfel microfirul turnat de către al dilea operator satisface cerințele de calitate în măsură de proximativ 70%.

Pentru identificarea domeniilor de valori utilizate de către operatori au fost construite graficele $R_p = f(P_v)$ pentru fiecare operator în parte, *Figura 2.7 (a) și (b)*.

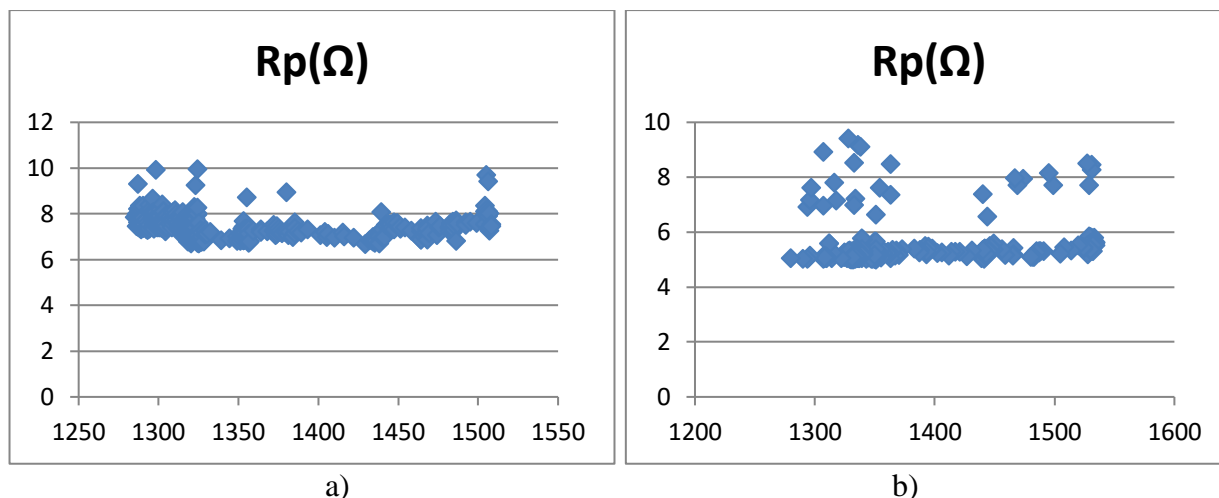


Figura 2.7 Graficul repartiției valorilor rezistenței liniare a microfirului de 5 – 10kΩ, turnat de (a) operatorul 1 la instalația 1 și (b) operatorul 2 la instalația 2, în funcție de valoarea vidului în tubul de sticlă.

Din graficele prezentate în *Figura 2.7* se poate face concluzia că *operatorul 1* pentru a menține rezistența în limitele de $[6.69k\Omega - 9.96k\Omega]$, operează cu valori ale vidului în intervalul $[1285 - 1508]$, iar *operatorul 2* menține rezistența în limitele de $[5.05k\Omega - 5.64k\Omega]$, operând cu valori ale vidului în intervalul $[1280 - 1534]$. Din natura și repartiția densității datelor pe tot intervalul poate fi făcută concluzia că aceștia utilizează preponderent valori situate în partea stângă a domeniului de valori al vidului pentru a menține rezistența liniară a microfirului exclusiv în partea dreaptă a domeniului său de valori.

Comparând strategiile de turnare a celor doi operatori se poate menționa faptul că strategiile decizionale sunt asemănătoare însă metodologiile de turnare aplicate diferă deoarece primul utilizează intervalul de centru-dreapta iar celălalt – de centru-stânga, rezistența microfirului turnat de primul operator variază mai mult decât rezistența microfirului turnat de cel de-al doilea care asigură un diametru mai omogen pe toata lungimea acestuia, însă se caracterizează și de o valoare absolută mai mare a variației rezistenței liniare. Cu toate acestea, doi operatori care utilizează metodologii de turnare diferite, totuși toarnă microfir care corespunde cerințelor de calitate. Ca potențial al acestor observații se poate remarca posibilitatea de a cumula mai multe metode diferite de turnare cu scopul de a obține un model decizional universal.

Pentru a putea analiza repartizarea valorilor parametrilor microfirului în intervalele de lucru ale operatorilor au fost construite graficele acestora sortate crescător, *Figura 2.8 (a), (b)* și *Figura 2.9 (a), (b)*. Din graficele prezentate în *Figura 2.8* se poate evidenția dependența preponderent liniară a rezistenței.

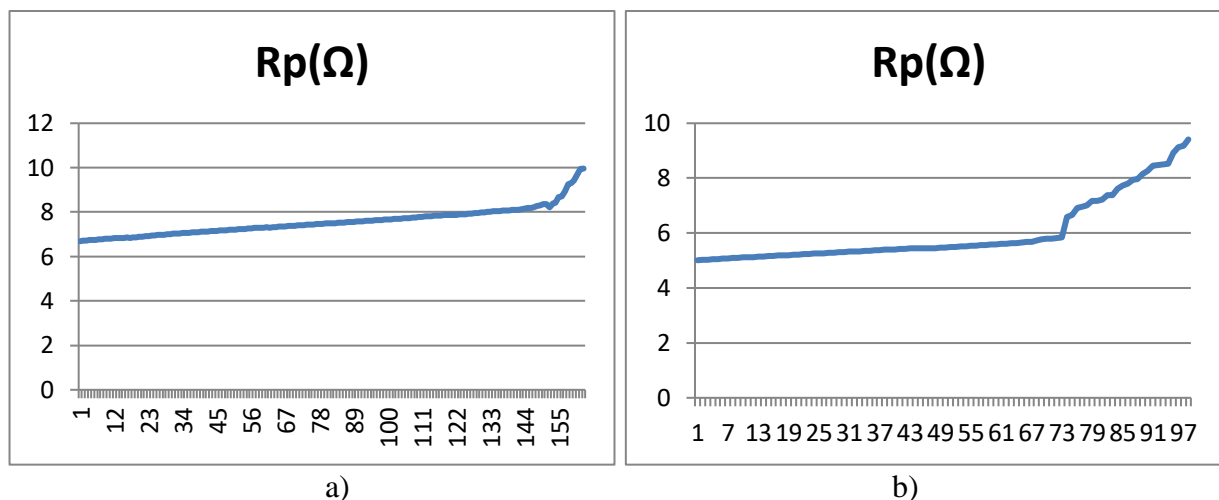


Figura 2.8 Graficul repartizării domeniului de valori în creștere pentru rezistența liniară a microfirului, turnat de (a) operatorul 1 la instalația 1 și (b) operatorul 2 la instalația 2, în funcție de numărul de ordine a valorii.

Graficele din *Figura 2.8* sunt asemănătoare și pot fi caracterizate prin două pante aproape liniare, dar se observă că pantele acestor funcții sunt diferite. Acest parametru poate fi utilizat pentru a caracteriza specificul metodei de turnare a operatorului cât și pentru caracterizarea individuală a instalației de turnare. Divizarea graficelor în două zone liniare cu pante diferite poate indica faptul că operatorii separă intervalul de valori în două zone: zona de confort relativ, în care procesul de turnare decurge stabil iar operatorii se implică foarte puțin în cursul procesului efectuând lent corecțiile necesare și zona critică, în care procesul de turnare decurge mai agresiv iar operatorii se implică prompt și ferm. Poate fi observat faptul că operatorii încearcă să mențină procesul într-un domeniu de siguranță relativă în care acesta poate fi dirijat eficient, cu intervenții minore și tinde să evite cât mai mult zona critică, în care sunt necesare acțiuni ferme. Fiecare operator, în dependență de experiența sa, își definește individual intervalele și strategia de acționare în interiorul acestora.

Din *Figura 2.9* se poate remarca clar tendința de repartizare liniară a valorilor parametrului de conducere P_V utilizat de către operatori la controlul procesului de turnare. Astfel pot fi interpretate ca caracteristici specifice ansamblului operator-instalație de turnare diferența pantelor graficelor valorilor P_V sortate crescător specifice fiecărui operator și diferența dintre domeniile de valori ale parametrului dat.

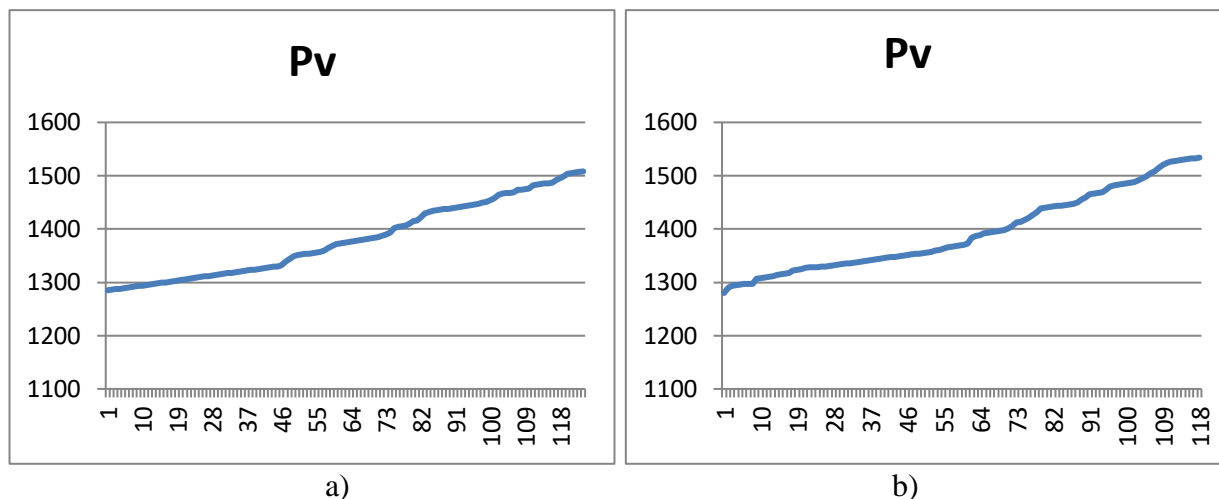


Figura 2.9 Graficul repartizării domeniului de valori în creștere pentru vidul în tubul de sticlă a microfirului, turnat de (a) operatorul 1 la instalația 1 și (b) operatorul 2 la instalația 2, în funcție de numărul de ordine a valorii.

Din cauza utilizării de către operator preponderent a presiunii ca parametru de reglare, comportamentul acestuia, poate fi descris ca un sistem cu o intrare – R_p și o ieșire – P_v , (2.12):

$$P_v = f(R_p); \quad (2.12)$$

unde:

$$P_v \sim R_p; \quad (2.13)$$

iar comportamentul instalației de turnare poate fi descris cu ajutorul relației (2.14):

$$R_p = f(P_v); \quad (2.14)$$

unde:

$$P_v \sim 1/R_p. \quad (2.15)$$

Astfel instalația de turnare a microfirului, la creșterea vidului în tubul de sticlă, reacționează cu răcirea picăturii de aliaj cauzată de îndepărtarea acesteia de la inductor, fapt ce duce la scăderea valorii rezistenței liniare a microfirului și respectiv la creșterea diametrului acestuia. Operatorul uman în acest proces însă are o reacție inversă. Acesta reacționează, la mărirea rezistenței liniare în raport cu rezistența nominală, mărind valoarea vidului în tubul de sticlă iar la micșorarea rezistenței liniare în raport cu rezistența nominală – micșorând valorii area vidului în tubul de sticlă. Astfel, la etapa de preluare a experienței operatorului, pentru descrierea procesului decizional, se va duce cont de reacția acestuia la condițiile variabile ale procesului tehnologic [81, 89-92]. Deoarece abilitățile decizionale și experiența fiecărui operator sunt strâns legate de instalația la care au fost căpătate și perfecționate, s-a decis de a se cerceta aceste entități în ansamblu, (operator-stație de turnare) cu scopul valorificării potențialului cumulativ al acestui cuplu indispensabil. Acest fapt oferă un potențial de generalizare a

influenței instalației de turnare și a experienței operatorului uman asupra calității produsului final.

Observațiile realizate pe parcursul studiului procesului tehnologic evidențiază faptul că operatorul uman nu operează cu valori numerice directe ale vidului din tubul de sticlă în procesul decizional, ci cu mărimi relative ale acestuia. În același timp, pentru variabila de ieșire (rezistența liniară a microfirului), mai comodă este reprezentarea numerică relativă la valoarea rezistenței caracteristice pentru fiecare operator în parte. Reprezentarea variației acestor parametri poate fi ușor realizată prin intermediul variabilelor lingvistice. Astfel poate fi afirmat faptul că operatorul controlează procesul de turnare prin variația parametrului de intrare a sistemului condus, *Figura 2.10*. În acest caz operatorul poate fi descris cu ajutorul relației (2.16):

$$\Delta P_{V_i} = f(\Delta R p_i); \quad (2.16)$$

unde:

$$\Delta R p_i = R p_i - R p^{Characteristic}; \quad (2.17)$$

$$\Delta P_{V_i} = P_{V_{i+1}} - P_{V_i}. \quad (2.18)$$

În relația (2.17) $R p^{Characteristic}$ reprezintă valoarea rezistenței liniare cu cea mai mare frecvență de apariție pe parcursul întregului proces de turnare a microfirului, iar în (2.18) P_{V_i} și $P_{V_{i+1}}$ sunt două măsurări consecutive ale vidului efectuate în procesul de turnare.

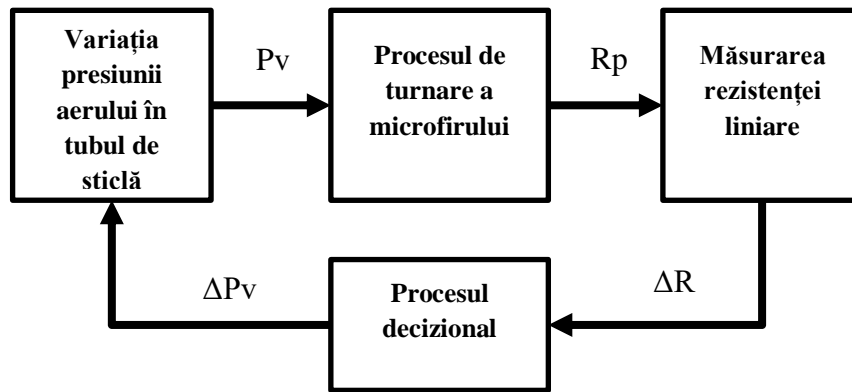


Figura 2.10 Schema de structură a modelului decizional în procesul de turnare a microfirilor.

Pentru calculul parametrului $\Delta R p_i$ ordinea datelor nu contează însă este nevoie de a determina valoarea parametrului $R p^{Characteristic}$ care reprezintă o valoare de referință pentru operator. Valoarea acestui parametru de referință se definește conștient sau inconștient de către fiecare operator și poate depinde de instalația de turnare, tipul microfirului turnat, cerințele de calitate sau experiența acumulată. Astfel conducerea procesului de turnare a microfirului de 5–

$10k\Omega$ este caracterizată prin menținerea valorii rezistenței liniare în vecinătatea valorii de referință caracteristice $Rp^{Characteristic}$ ce aparține intervalului dat.

Pentru determinarea rezistenței de referință caracteristice $Rp^{Characteristic}$ pentru fiecare operator a fost construit graficul repartizării frecvenței de apariție a valorilor rezistenței liniare pe parcursul procesului de turnare, *Figura 2.11*.

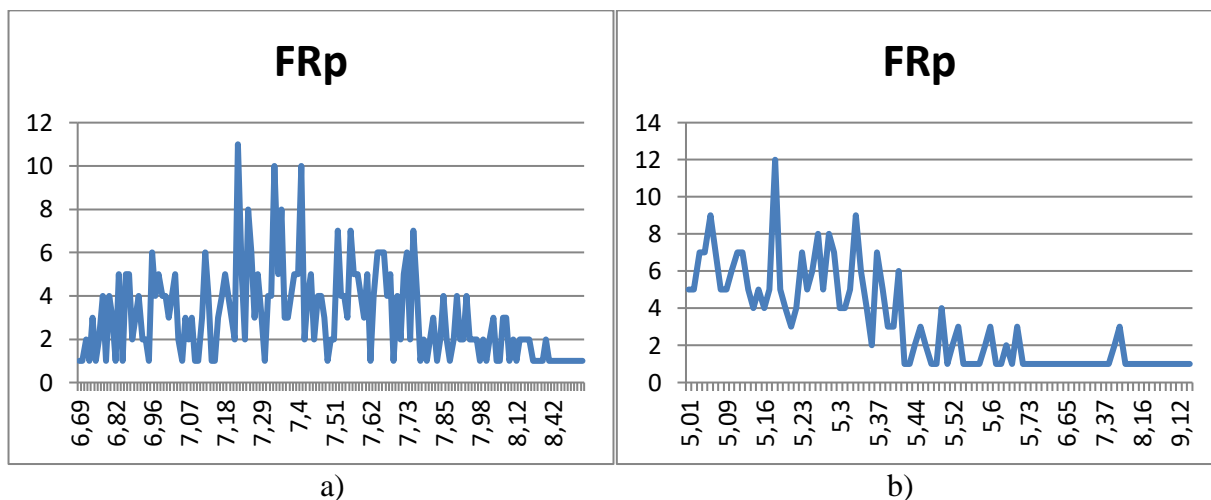


Figura 2.11 Graficul repartizării frecvenței de apariție a valorilor rezistenței liniare pe durata procesului de turnare a microfirului turnat de către (a) operatorul 1 la instalația 1 și (b) operatorul 2 la instalația 2.

Analizând datele prezentate în *Figura 2.11* se pot determina valorile cu frecvența maximă de apariții (11) este $Rp_i=7.22k\Omega$ pentru primul operator și (12), $Rp_i=5.18k\Omega$, pentru al doilea. Acest fapt dovedește că operatorii mențin rezistența liniară a microfirului în apropierea valorilor $Rp^{Characteristic}=7.22k\Omega$ și $Rp^{Characteristic}=5.18k\Omega$ respectiv.

Cercetările realizate au dat posibilitatea de a valorifica importanța parametrilor caracteristici pentru îmbunătățirea calității microfirului turnat fără cheltuieli suplimentare doar prin ajustarea valorilor de referință optimale ale parametrilor de bază. Astfel în cazul primului operator optimizarea valorii caracteristice a rezistenței microfirului $Rp^{Characteristic}=6.32k\Omega$ v-a aduce un aport nesemnificativ la calitatea finală a microfirului care va constitui aceleași 87%. În cazul celui de-al doilea operator modificarea valorii caracteristice a rezistenței microfirului $Rp^{Characteristic}=5.84k\Omega$ v-a aduce un aport destul de semnificativ la calitatea finală a microfirului care va constitui aproximativ 86%. Astfel calitatea microfirului turnat de către cel de-al doilea operator poate crește cu 16% în rezultatul schimbării valorii de referință a parametrului de intrare cu condiția păstrării specificului decizional propriu.

Utilizând valoarea de referință a rezistenței liniare caracteristice a fiecărui operator cercetat se calculează variația relativă a rezistenței liniare *Figura 2.12* după formula (2.18).

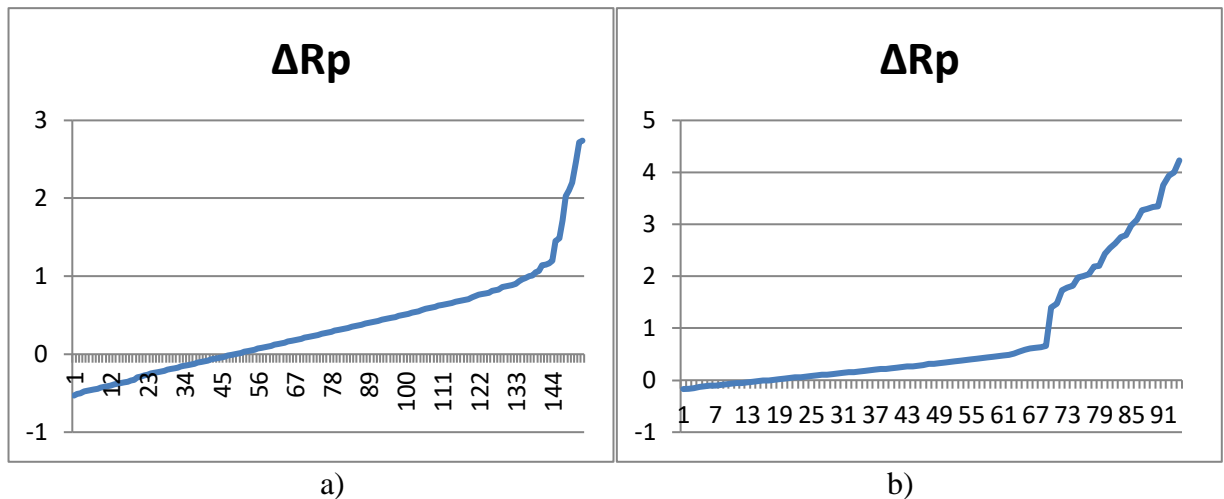


Figura 2.12 Graficul domeniului de valori pentru variația relativă a rezistenței liniare calculate pentru (a) operatorul 1 și (b) operatorul 2.

Graficul din *Figura 2.12* caracterizează dependența variației rezistenței liniare de proprietatea decizională individuală a operatorului specificul căreia este reflectat de două porțiuni liniare ale graficului și de domeniile de valori ale acestor curbe. Primul interval liniar este mai larg și manifestă o creștere relativ lentă față de valoarea de referință a rezistenței liniare caracteristice, iar cel de-al doilea interval este caracterizat printr-o creștere mai rapidă. Caracteristica specifică a proceselor decizionale este prevalarea acțiunilor în rezultatul cărora crește lent rezistența liniară a microfirului. Domeniile de valori specifice fiecărui operator sunt definite de intervalele $[-0.53k\Omega, 2.74k\Omega]$ și $[-0.17k\Omega, 4.23k\Omega]$ respectiv, iar valorile de $1.15k\Omega$ și $1.4k\Omega$ reprezintă punctele de schimb ale pantelor creșterii rezistenței liniare. Diferențele între parametri cum ar fi lățimea intervalelor, domeniile de valori și pantele variațiilor rezistenței liniare reprezintă caracteristicile individuale ale modelelor decizionale specifice fiecărui operator.

În *Figura 2.13* sunt prezentate graficele repartizării frecvențelor de apariție a valorilor variației rezistenței liniare pe durata procesului de turnare. Cu ajutorul acestor grafice se poate determina cea mai frecventă valoare a variației rezistenței liniare ($0k\Omega$) pentru fiecare operator, relativ la rezistența liniară caracteristică, care are valoarea numărul maxim de apariții pe durata procesului de turnare egală cu 11 pentru primul operator și 12, pentru al doilea.

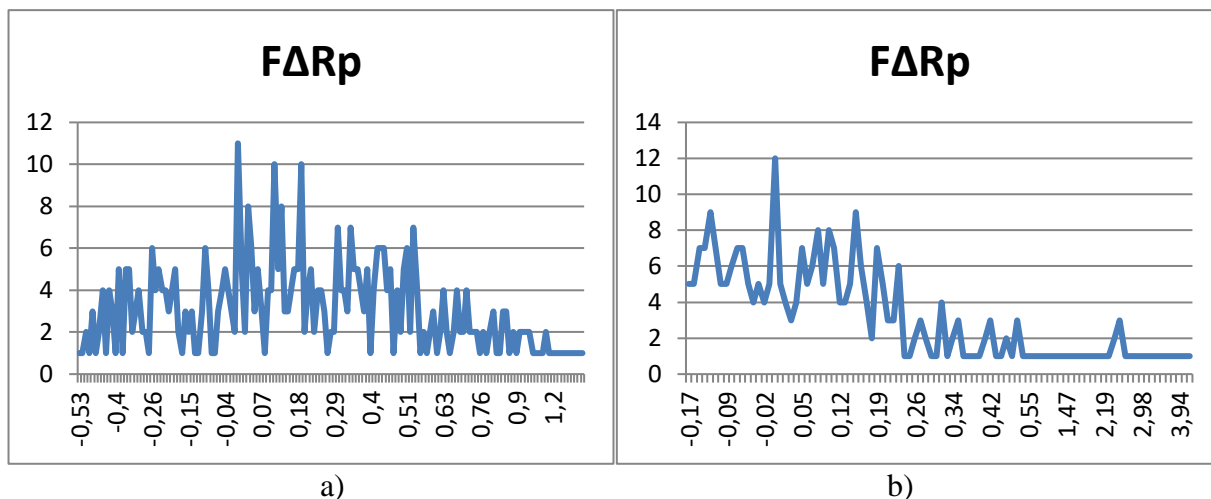


Figura 2.13 Graficul repartizării frecvenței de apariție a valorilor variației rezistenței liniare pe durata procesului de turnare a microfirului turnat de către (a) operatorul 1 la instalația 1și (b) operatorul 2 la instalația 2.

Valoare de $0k\Omega$, fiind cea mai frecventă valoare a variației rezistenței liniare pentru ambii operatori, caracterizează în ansamblu procesele decizionale ale operatorilor și procesul tehnologic în care participă aceștia ca fiind un proces mai mult sau mai puțin stabil, cu o dinamică cât de cât constantă. Comparând domeniile de valori ale variației rezistenței liniare pentru ambii operatori poate fi observată asimetria acestora, însă ambele au aceeași valoare cu frecvența de apariție maximă.

Procesul de calcul al parametrului de control ΔP_{V_i} este realizat păstrând ordinea și natura datelor experimentale colectate, care au fost supuse procedurii de filtrare pentru a se păstra caracteristica și proprietatea reacție relative față de valoarea curentă a vidului în tubul de sticlă. Nici chiar cel mai experimentat operator nu poate răspunde la întrebarea care este valoarea presiunii ce trebuie setată pentru a micșora o abatere anumită a rezistenței liniare de la valoarea caracteristică. Operatorul poate să nu utilizeze nici valoarea curentă a vidului, deoarece acesta intuiește care poate fi reacția sa, cât de repede și în ce direcție trebuie modificat vidul în tubul de sticlă pentru egalarea valorii curente a rezistenței liniare cu valoarea caracteristică [82].

Analizând graficele din *Figura 2.14* pot fi identificate următoarele deosebiri comportamentale în procesele decizionale ale celor doi operatori cercetați: operatorul 1 conduce procesul de turnare utilizând o metodă decizională prin care oferă o dependență aproape liniară a variației vidului în tubul de sticlă și stabilește domeniul de valori pentru variația vidului în intervalul $[-10, 11]$, pe când operatorul 2 conduce procesul utilizând o metodă decizională care manifestă o dependență neliniară a variației vidului și are domeniul de valori pentru variația vidului în intervalul $[-21, 16]$.

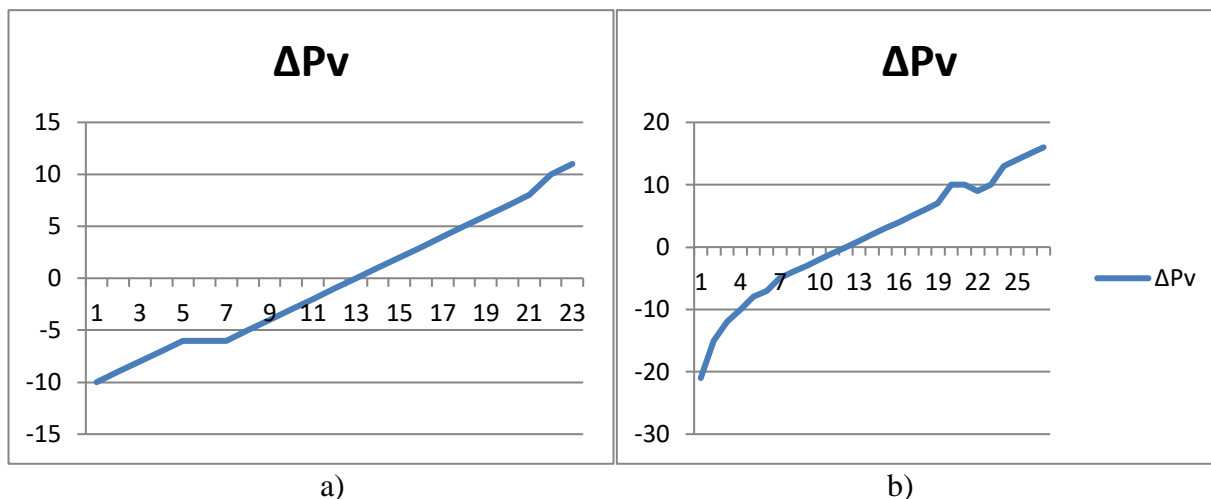


Figura 2.14 Graficul repartizării domeniului de valori în creștere pentru variația vidului în tubul de sticlă a microfirului, turnat de (a) operatorul 1 la instalația 1 și (b) operatorul 2 la instalația 2, în funcție de numărul de ordine a valorii.

În *Figura 2.15* sunt prezentate graficele repartizării frecvențelor de apariție a valorilor variației vidului în tubul de sticlă pe durata procesului de turnare. Cu ajutorul acestor grafice se pot determina cele mai frecvente valori ale variației vidului care reprezintă -1 pentru primul operator și 1 pentru al doilea operator, și are frecvența maximă de apariții pe durata procesului de turnare egală cu 93 și 57 respectiv.

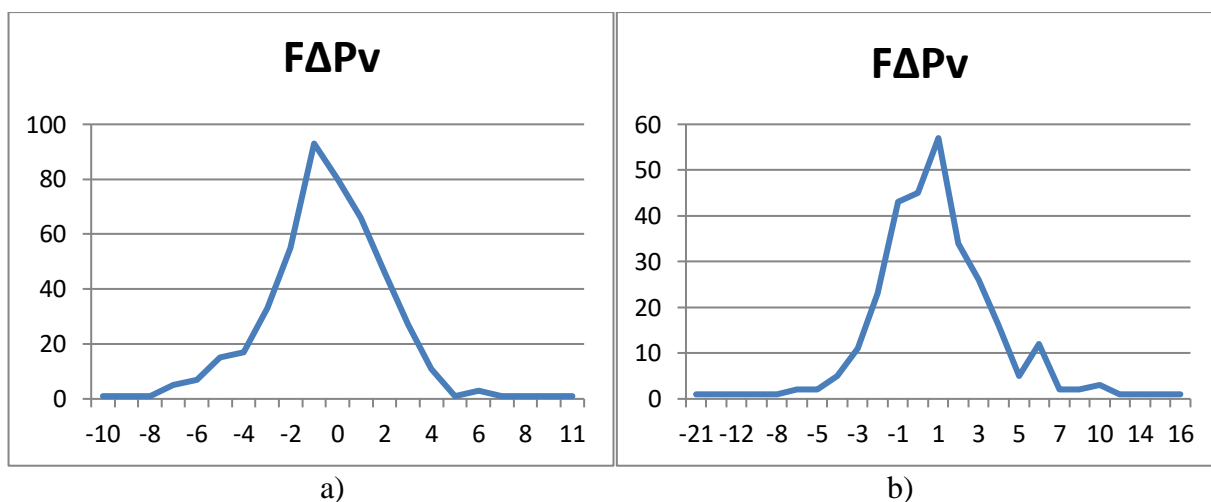


Figura 2.15 Graficul repartizării frecvenței de apariție a valorilor variației vidului în tubul de sticlă pe durata procesului de turnare a microfirului turnat de către (a) operatorul 1 la instalația 1 și (b) operatorul 2 la instalația 2.

Comparând domeniile de valori ale variației vidului în tubul de sticlă ale ambilor operatori se poate observa faptul că domeniul de valori utilizat de al doilea operator include total intervalul utilizat de primul. Acest specific poate semnifica faptul că al doilea operator posedă o

strategie de turnare mai universală argumentată de specificul procesului tehnologic, pe când primul operator manifestă un comportament diferit orientat spre asigurarea unor proprietăți specifice ale microfirului turnat.

În rezultatul etapei de preprocesare a datelor statistice au fost identificați o serie de indicatori individuali cum ar fi domeniul de valori al parametrilor de conducere care poate caracteriza numeric metodologia decizională ce reflectă direct influența experienței operatorului uman asupra evoluției procesului tehnologic. Indicatorii determinați pentru fiecare operator în parte pot fi generalizați sau mixați cu scopul de a obține metodologii decizionale universale.

2.3 Elaborarea metodelor și algoritmilor de extragere a cunoștințelor expert

Parametrii caracteristici identificați au fost utilizați pentru elaborarea metodologiilor decizionale, care ar fi în stare să preia specificul decizional individual al fiecărui operator. Pentru crearea modelelor decizionale a fost utilizat mediul grafic de dezvoltare a sistemelor fuzzy din pachetul de programe Matlab [93]. În urma analizei datelor colectate de la fiecare operator, a fost calculat un set de parametri care au servit la construirea funcțiilor de apartenență caracteristice. În *Figura 2.16 (a)* și *(b)* sunt prezentate funcțiile de apartenență pentru calificativele $-\Delta Rp$ și $+\Delta Rp$ definite, pentru fiecare operator individual, pe intervalele $[-0.53 -0.53 0]$, $[0 2.74 2.74]$ și $[-0.17 -0.17 0]$, $[0 4.23 4.23]$ respectiv.

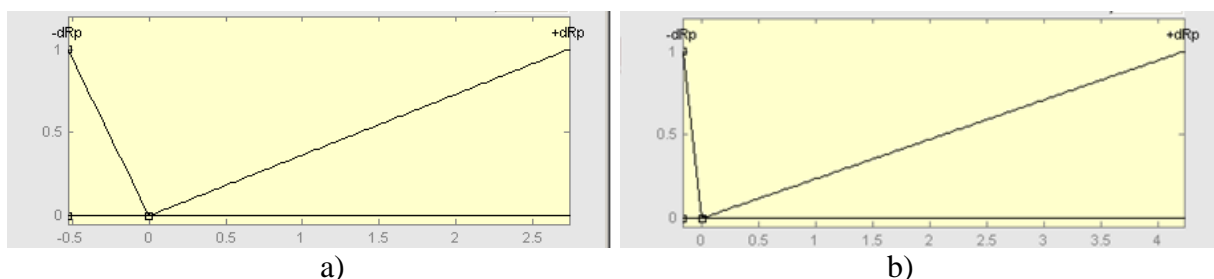


Figura 2.16 Funcțiile de apartenență triunghiulare ale calificativelor $-\Delta Rp$ și $+\Delta Rp$ adaptate la comportamentul decizional al (a) operatorului 1 și (b) operatorului 2.

În urma analizei datelor despre evoluția presiunii în tubul de sticlă pe parcursul procesului de turnare au fost calculați parametrii caracteristici în baza cărora au fost construite funcțiile de apartenență ale variabilelor fuzzy $-\Delta Pv$ și $+\Delta Pv$ definite individual pentru fiecare operator pe intervalele $[-10 -10 -1]$, $[-1 11 11]$ și $[-21 -21 1]$, $[1 16 16]$ respectiv [94].

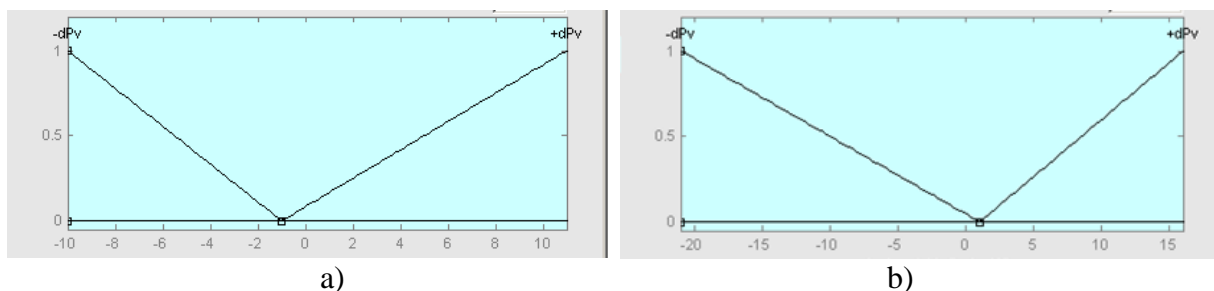


Figura 2.17 Funcțiile de apartenență triunghiulare clasice ale calificativelor $-APv$ și $+APv$ adaptate la comportamentul decizional al (a) operatorului 1 și (b) operatorului 2.

Pentru defuzificarea rezultatelor obținute în urma procesului de inferență a fost aplicată o mulțime de metode standard de defuzificare. Dintre acestea a fost selectată metoda „som” care aproximează destul de bine graficul funcției de ieșire pentru ΔPv , *Figura 2.18*, păstrând totodată domeniul de valori $[-10\ 11]$ caracteristic pentru primul operator și domeniul $[-21\ 16]$ caracteristic pentru cel de-al doilea, *Figura 2.18*.

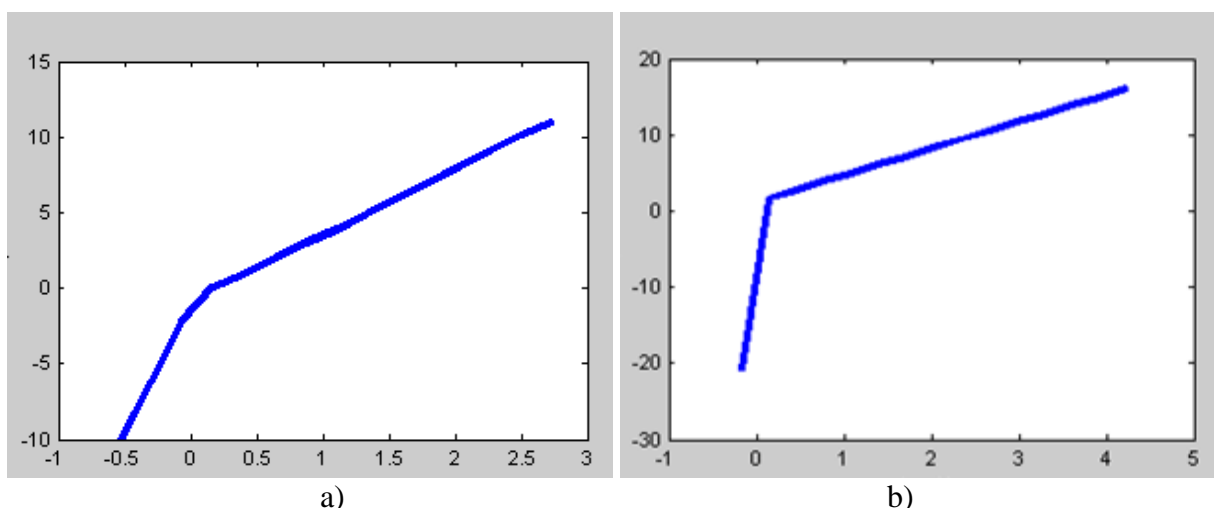


Figura 2.18 Graficul dependenței ΔPv de ΔRp obținut după procesul de defuzificare prin metoda „som” caracteristic (a) operatorului 1 și (b) operatorului 2.

În rezultatul aplicării procedurilor de adaptare a modelului teoretic de extragere a cunoștințelor, au fost obținute o serie de modele decizionale generalizate pentru fiecare entitate individuală operator – instalație de turnare, care sunt în stare să preia experiența operatorului uman, specificul decizional al acestuia și trăsăturile tehnologice individuale ale instalației. Ca rezultat al combinării modelelor teoretice cu parametrii caracteristici ai operatorilor și instalațiilor de turnare au fost obținute modele decizionale care preiau doar parțial experiența acestora. Acest fapt nu poate fi considerat ca dezavantaj deoarece fiecare operator este caracterizat de un grad înalt de ambiguitate. Faptul dat poate fi explicat fie prin viteza mică de

reacție, fie prin latența sistemului de comandă și control. Din aceste motive comportamentul operatorului uman capătă mai mult un caracter probabilistic, iar metodele dezvoltate pot exclude influența negativă a acestor trăsături asupra procesului tehnologic.

Pentru explorarea potențialul experienței operatorului uman, în vederea preluării cât mai exacte a experienței acestuia, au fost propuse o serie de modele decizionale în care funcțiile de apartenență și-ar păstra cât mai exact forma moștenită de la operator. Astfel se pune problema utilizării funcțiilor de apartenență extrase nemijlocit din datele colectate de la operatorul uman. Devine evident faptul că operatorul nu operează cu întreg spectrul de valori, iar funcția de apartenență nu poate fi definită pe întreg domeniul de valori. Din această cauză a fost luată decizia de a interpola liniar funcția de apartenență pe aceste intervale.

Utilizând valorile parametrului ΔR_p se poate construi funcția caracteristică de apartenență a calificativului $0\Delta R_p$ care reprezintă variația rezistenței liniare a microfirului relativ la rezistența liniară caracteristică, *Figura 2.19*. Funcțiile de apartenență ale variabilelor fuzzy $-\Delta R_p$ și $+\Delta R_p$ se determină prin inversarea funcției de apartenență a variabilei logice $0\Delta R_p$, *Figura 2.20*.

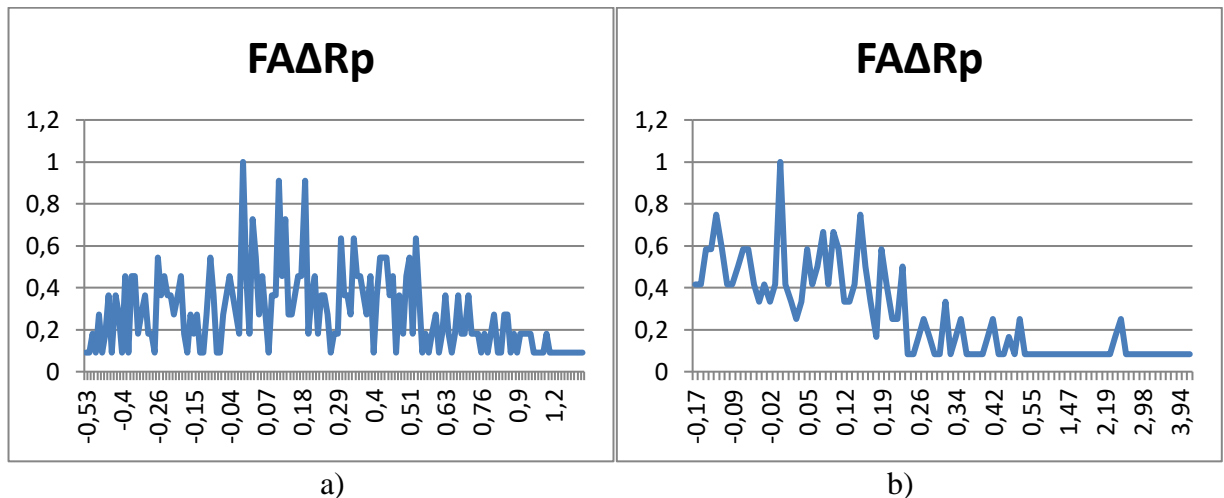


Figura 2.19 Graficul funcției de apartenență a valorii variației rezistenței liniare pe domeniul de valori al microfirului turnat de către (a) operatorului 1 și (b) operatorului 2.

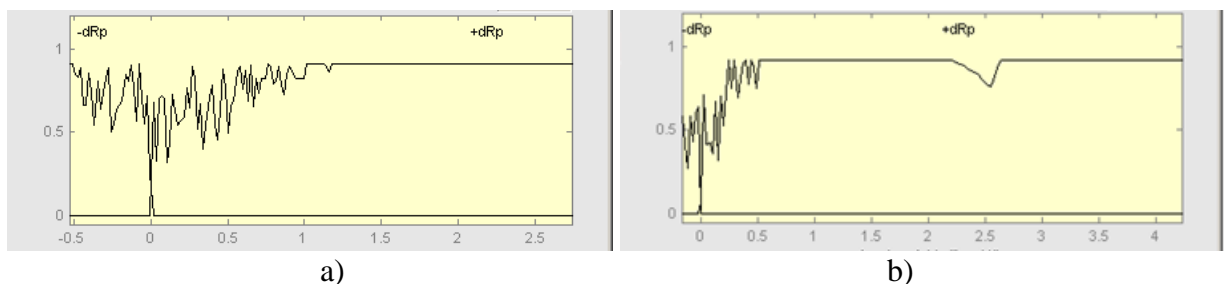


Figura 2.20 Funcțiile de apartenență ale calificativelor $-\Delta R_p$ și $+\Delta R_p$ specifice comportamentul decizional (a) al operatorului 1 și (b) al operatorului 2.

Procedând analogic pentru ΔP_v au fost construite funcțiile caracteristice de apartenență ale variabilelor fuzzy $0\Delta P_v$, Figura 2.21 și $+\Delta P_v$, Figura 2.21.

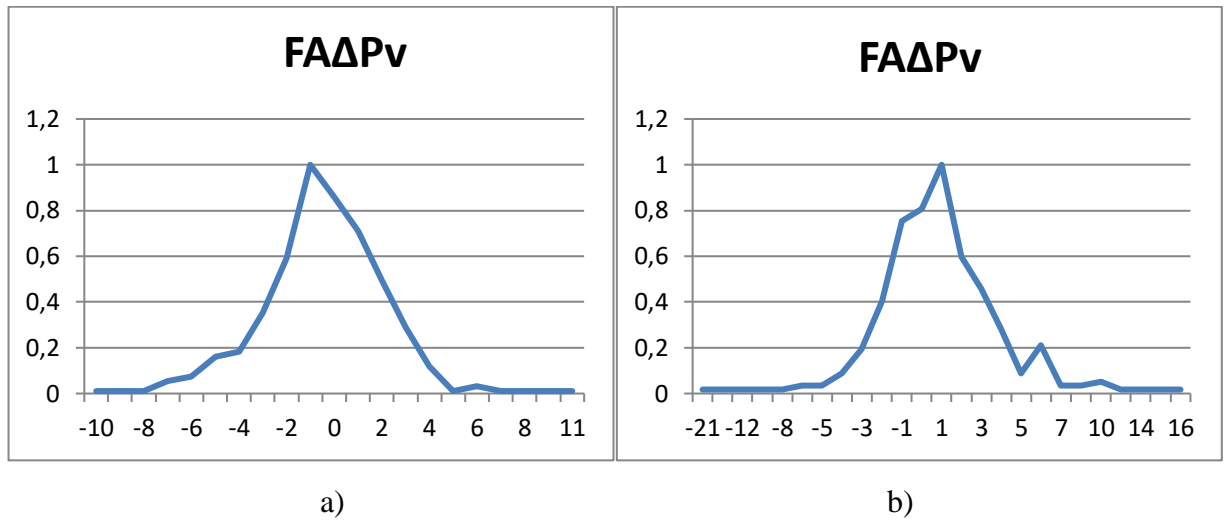


Figura 2.21 Graficul funcției de apartenență a valorii variației vidului în tubul de sticlă pe domeniul de valori al microfirului turnat de către (a) operatorului 1, la instalația 1 și (b) operatorului 2, la instalația 2.

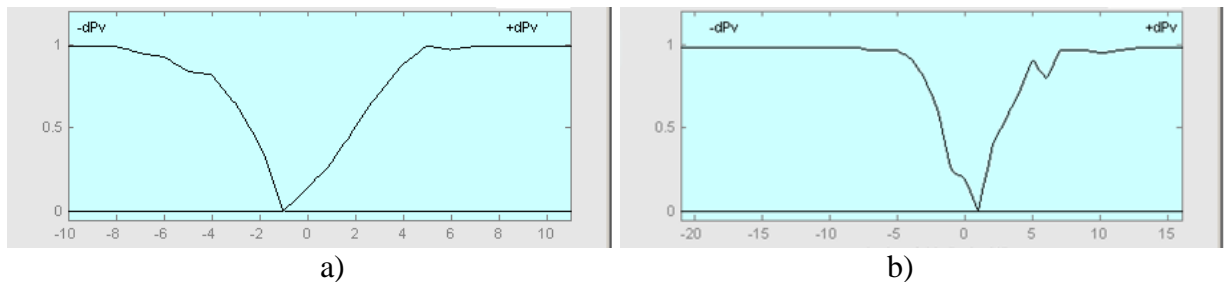


Figura 2.22 Funcțiile de apartenență ale calificativelor $-\Delta P_v$ și $+\Delta P_v$ specifice comportamentul decizional (a) al operatorului 1, la instalația 1 și (b) al operatorului 2, la instalația 2.

În urma experimentărilor cu metodele de inferență, a fost selectată metoda „lom” de defuzificare, Figura 2.23, ca fiind cea care generalizează cel mai bine funcția de ieșire a procesului decizional caracteristic operatorului uman, păstrând totodată domeniu specific de valori al fiecărui operator.

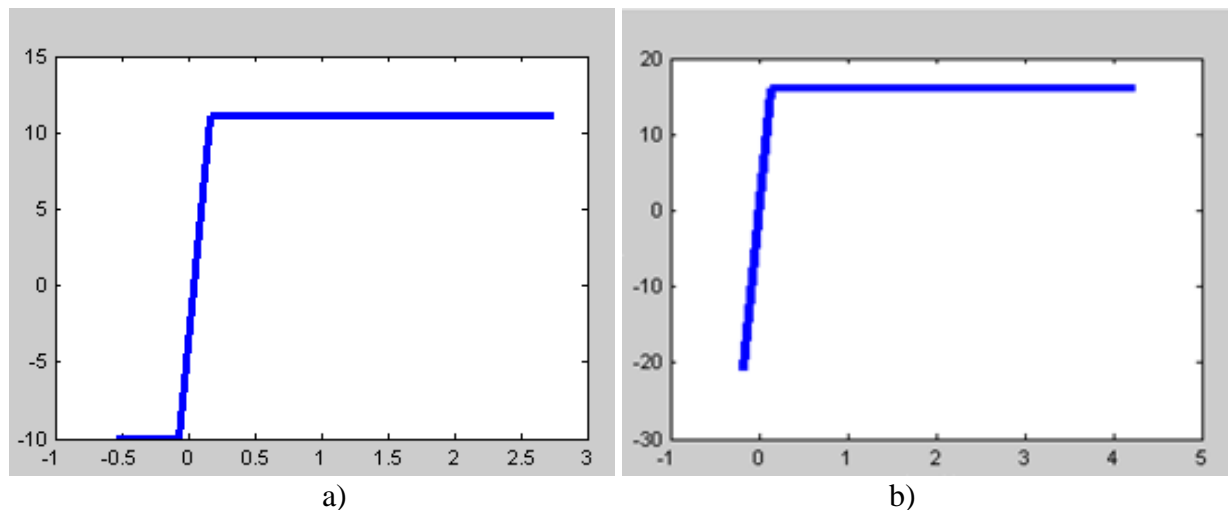


Figura 2.23 Graficul dependenței ΔP_v de ΔR_p obținut după defuzificare prin metoda „lom” caracteristice (a) operatorului 1 și (b) operatorului 2.

Analizând graficele dependenței variației vidului în dependență de variația rezistenței liniare relative obținute pentru ambii operatori, *Figura 2.18*, *Figura 2.23*, pot fi confirmate afirmațiile enunțate anterior cu un șir de specificări esențiale printre care se poate menționa faptul că modelele decizionale obținute pentru ambii operatori manifestă din plin proprietățile individuale ale caracteristicilor specifice comportamentului acestora, totodată marcând și trăsăturile comune ale proceselor decizionale.

Metodele de preluare a cunoștințelor umane prezentate au dreptul la viață iar posibilitatea și succesul implementării acestora depinde de mai mulți factori cum ar fi numărul de variabile fuzzy, tipul și forma funcțiilor de apartenență, numărul regulilor de inferență și metoda de defuzificare. Luând în considerație toate aceste nuanțe împreună cu cerințele și condițiile inițiale ale problemei, rezolvării căreia trebuie să servească aceste modele, poate fi afirmat faptul că modelele și metodele descrise pot reprezenta niște soluții parțiale pentru clasa de probleme specifică caracterizată de ambiguitate sau imprecizie a datelor. În rezultatul acestor cercetări au fost elaborate mai multe modele, descrise metode și proceduri de preluare a experienței operatorului uman [95]. Aceste proceduri pot fi automatizate pentru simplificarea procesului de dezvoltare a sistemelor de suport decizional fuzzy.

Funcțiile caracteristice ΔR_p și ΔP_v , identificate anterior, au fost utilizate pentru elaborarea modelelor decizionale în baza rețelelor neuronale artificiale, care ar fi în stare să moștenească particularitățile decizionale ale operatorilor. Pentru crearea și antrenarea rețelelor neuronale artificiale a fost utilizat Toolbox-ul „*Neural Network*” din pachetul de programe Matlab. Utilizând aceste funcții caracteristice în procesul de antrenare a rețelelor neuronale

artificiale se pot obține, la fel ca și în cazul sistemelor decizionale fuzzy, modele decizionale specifice care ar păstra proprietățile individuale ale experienței fiecărui operator. În baza vectorului ΔRp calculat anterior a fost proiectată o rețea neuronală artificială care ar aproxima funcția caracteristică conform relației (2.16). Cu acest scop a fost selectată opțiunea „Fitting Tool” din fereastra de start a aplicației ”Neural Network Toolkit” din pachetul de programe Matlab. Această opțiune permite crearea unei rețele neuronale artificiale capabile să aproximeze funcții de mai multe variabile.

Ulterior sunt selectat seturile de date de intrare și ieșire din spațiul de lucru Matlab. Datele caracteristice sunt încărcate, din fișiere Excel, în variabilele vectoriale notate dRp și dPv din spațiul de lucru al pachetului de programe Matlab. Vectorul de intrare notat dRp și vectorul de ieșire notat dPv reprezintă intrările și respectiv ieșirile rețelei neuronale artificiale proiectate pentru automatizarea procesului tehnologic de turnare a microfiredelor. Opțiunea “Matrix rows” indică faptul că datele din vectorii coloană notați dRp și respectiv dPv trebuie interpretate ca 467 de sample-uri a câte un element din setul de date colectat de la primul operator și respectiv ca 298 de sample-uri a câte un element din setul de date colectat de la al doilea operator.

La etapa de configurare a rețelei neuronale sunt realizate setările arhitecturale individuale ale rețelei cum ar fi numărul de neuroni (10) din stratul ascuns [96]. A fost ales un număr relativ mic de neuroni, deoarece numărul mare de neuroni poate cauza creșterea semnificativă a timpului de simulare, proces care necesită resurse mari de calcul, pe când procesul decizional trebuie realizat în timp real. În cazul în care soluția obținută nu oferă precizia necesară, această etapă poate fi repetată cu scopul de a identifica pe cale experimentală numărul optimal de neuroni din arhitectura rețelei.

Pentru antrenarea rețelei neuronale proiectate prin metoda ”Levenberg-Marquardt backpropagation”, au fost utilizate datele caracteristice ale celor doi operatori. După finalizarea procesului de antrenare, care consumă ceva timp, sunt prezentate detaliile despre starea acestui proces și precizia algoritmului de funcționare a rețelei neuronale. În rezultatul acestei etape sunt generate grafice care descriu modul de funcționare și performanța rețelei neuronale artificiale elaborate.

Figura 2.24 prezintă fereastra de antrenare a rețelelor neuronale cu două straturi (stratul ascuns alcătuit din 10 neuroni și stratul de ieșire alcătuit dintr-un singur neuron). Din *Figura 2.24* poate fi făcută concluzia că antrenarea rețelei neuronale, în baza datelor caracteristice colectate de la primul operator, a fost realizată pe parcursul a 13 epoci, iar antrenarea rețelei pentru al doilea operator a fost realizată pe parcursul a 8 epoci. De asemeni sunt prezentați o

serie de parametri ca: timpul de antrenare, performanța, gradientul și validarea care caracterizează procesul de antrenare și testare a rețelei neuronale feed-forward.

Pentru crearea rețelei neuronale artificiale de tip feed-forward cu doua straturi a fost utilizată funcția *feedforwardnet*. Rețeaua este alcătuită dintr-un strat ascuns alcătuit din 10 neuroni și din stratul de ieșire, alcătuit dintr-un singur neuron [97].

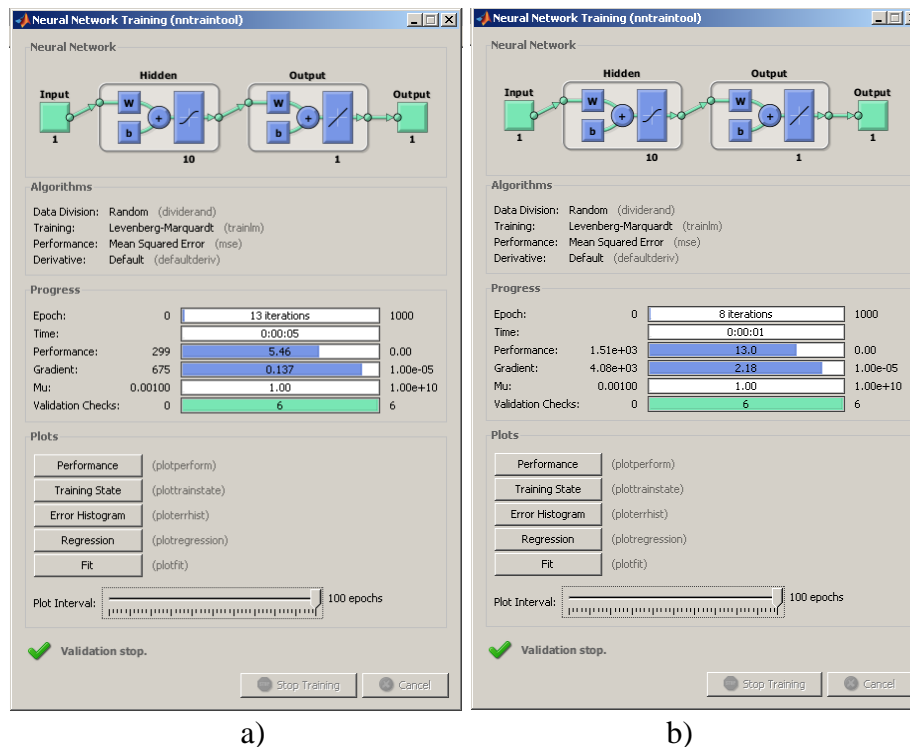


Figura 2.24 Fereastra de antrenare a rețelelor neuronale în baza datelor colectate de la (a) operatorul 1 și (b) operatorul 2.

În *Figura 2.25* este reprezentat graficul de performanță a rețelelor neuronale artificiale antrenate și testate în baza seturilor de date colectate de la operatorii cercetații. Pe acest grafic sunt prezentate curbele de antrenare, de validare și de testare [98]. Graficul respectiv arată descreșterea erorii pătratice pe parcursul procesului de antrenare (curba albastră). Curba verde reprezintă dinamica erorii în cadrul procesului de validare. Procesul de antrenare a rețelei neuronale artificiale se stopează în momentul în care valoarea erorii la etapa de validare nu mai descrește. Curba roșie reprezintă dinamica erorii pe parcursul procesului de testare și indică modul în care rețeaua neuronală artificială antrenată generalizează seturile noi de date. Performanța maximă în procesul de validare (5.2366) a fost obținută în epoca a 7-a în cazul primului operator și (8.5402) în epoca a 2-a în cazul celui de-al doilea operator, *Figura 2.25*.

Diagramele de performanță a celor două modele indică faptul că eroarea medie pătratică descrește pe măsura creșterii numărului de epoci, *Figura 2.25*. Erorile setului de testare și

validare au caracteristici comparabile și nu se observă devieri majore în apropierea epocilor 7 și 2, (timpul în care a fost obținută performanța maximă).

Parametrul *Mean Squared Error* (MSE) reprezintă diferența medie pătratică între valorile reale și cele corecte. Sunt preferabile valori cât mai mici ale parametrului MSE, iar valoarea 0 a acestuia semnifică funcționarea perfectă a modelului.

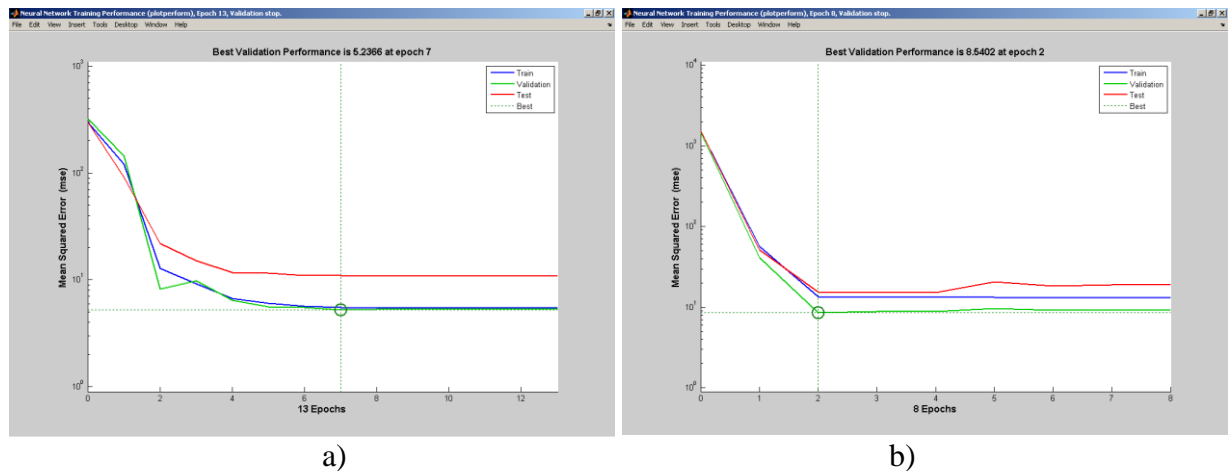
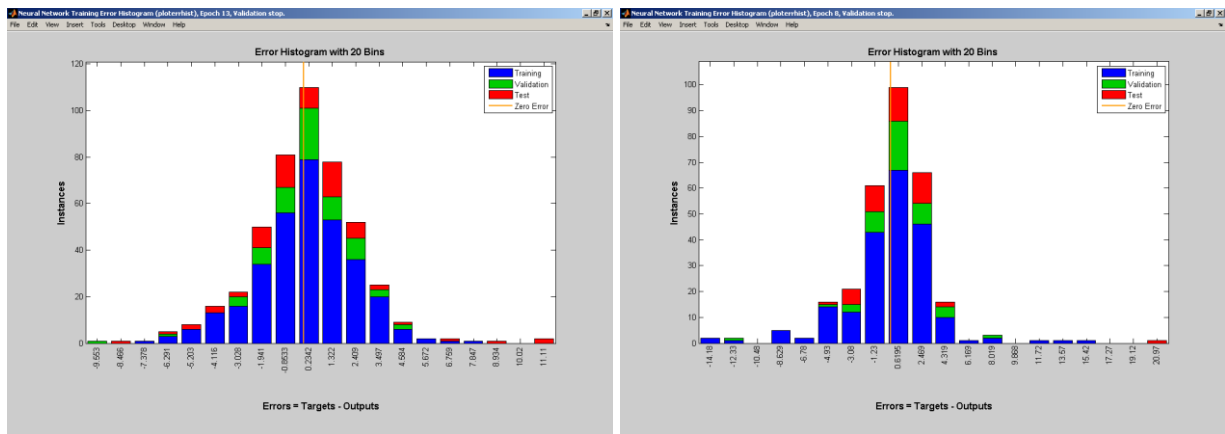


Figura 2.25 Graficul performantei rețelei neuronale artificiale antrenate în baza datelor colectate de la (a) operatorul 1 și (b) operatorul 2.

În rezultatul procesului de antrenare a rețelelor neuronale au fost obținute histogramele erorilor, *Figura 2.26*, în care cu bare albastre este notat setul de date utilizat pentru antrenarea rețelei, cu bare de culoare verde - setul de date utilizat pentru validarea rețelei iar cu bare de culoare roșie - setul de date utilizat pentru testarea rețelei neuronale. Histogramele respective pot servi pentru a obține o viziune suplimentară asupra performanței modelelor dezvoltate. Cea mai mare parte a datelor este repartizată în apropierea liniei care indică valoarea zero a erorii. Graficul dat constituie o metodă vizuală de estimare a omogenității datelor. Această metodă de reprezentare indică dacă careva date manifestă un caracter diferit față de restul datelor.

Histogramele repartizării erorilor pe parcursul etapelor de instruire, validare și testare, sunt prezentate în *Figura 2.26*. Aceste histograme indică faptul că valorile erorii la aproximarea datelor sunt distribuite într-un interval destul de îngust în preajma valorii 0 [99].



a)

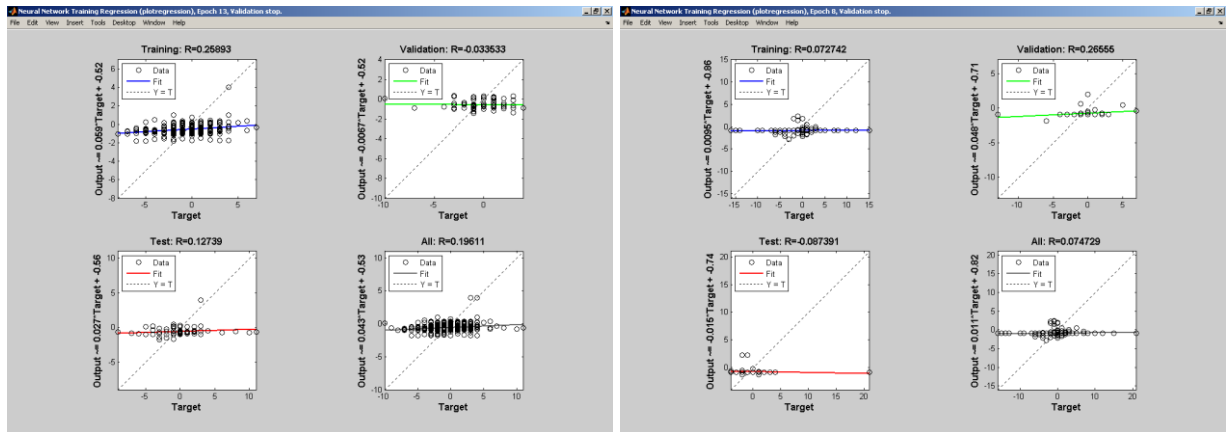
b)

Figura 2.26 Histogramele de repartizare a erorilor în procesul antrenării, validării și testării rețelelor neuronale antrenate în baza datelor colectate de la (a) operatorul 1 și (b) operatorul 2.

Graficele regresiei prezintă ieșirile din punct de vedere al obiectivelor de antrenare, validare și testare. Distribuția de-a lungul drepte de 45 de grade, care indică faptul că ieșirile rețelei sunt egale cu ieșirile țintă, denotă potrivirea perfectă a modelului dezvoltat cu procesul decizional cercetat. Valoarea regresiei (R) indică nivelul corelației între ieșirile reale și cele corecte. Valoare 1 a parametrului R semnifică o corelație perfectă iar 0 – una aleatorie. Deoarece valorile parametrilor MSE și R sunt foarte apropiate de zero se poate face afirmația că natura datelor poartă un caracter aliator și ambiguu [100].

În cazul modelelor proiectate, valoarea parametrului R este egală cu 0.19611 în cazul modelului decizional al primului operator și cu 0.074729, în cazul celui de-al doilea operator. Aceste valori au fost obținute în urma utilizării întregului set de date. Valorile acestui parametru dovedesc odată în plus comportamentul preponderent aleator al modelelor decizionale dezvoltate în baza experienței operatorului uman. Poate fi menționat faptul că modelele dezvoltate nu aproximează pe deplin procesele decizionale reale, în schimb acestea au moștenit comportamentul ambiguu de la operatorii umani [101].

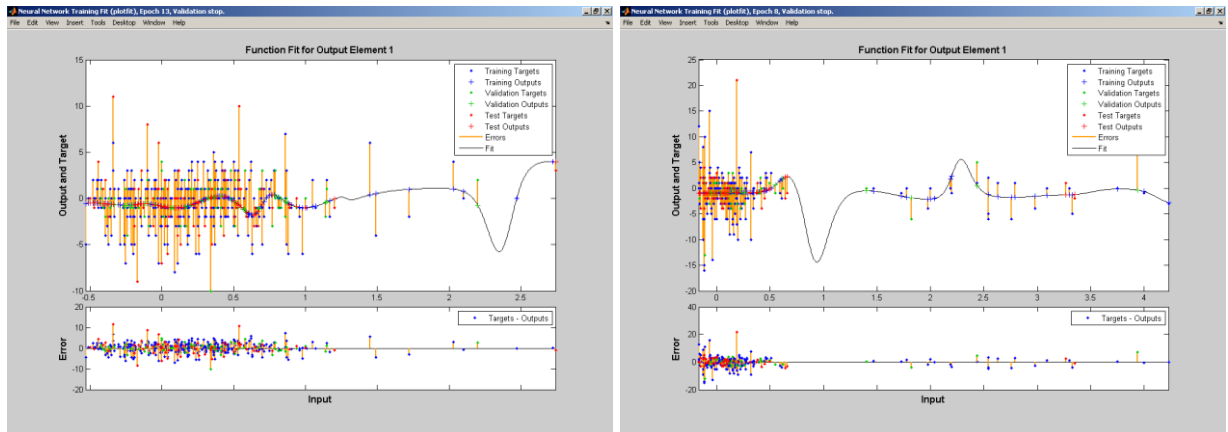
În final se poate menționa faptul că în cazul acestei clase de probleme, metodele clasice de estimare a performanței nu sunt capabile să reflecte corect calitatea modelelor dezvoltate din cauza caracterului și naturii aparent aleatorii a deciziilor operatorilor. Faptul că comportamentul operatorilor reprezintă o funcție matematică multivaloare, face ca semnificația actuală a acestui parametru să-și piardă sensul și pune accent pe necesitatea unei noi metode de calcul a acestui indicator.



a) b)
Figura 2.27 Graficele regresiei în procesul de antrenare, validare și testare a rețelelor neuronale antrenate în baza datelor colectate de la (a) operatorul 1 și (b) operatorul 2.

În graficul din *Figura 2.28* este prezentat modul în care rețeaua neuronală aproximează procesul decizional al operatorului. Pe axa orizontală sunt plasate datele de intrare dRp iar pe axa verticală sunt prezentate datele experimentale dPv^{exp} marcate cu simbolul + și datele reale de la ieșirea rețelei neuronale artificiale dPv^{net} marcate cu simbolul \circ . După cum se observă din graficul de mai jos rețeaua neuronală în procesul de funcționare are o eroare relativă destul de mare fapt ce poate pune la îndoială aplicabilitatea acestor modele la automatizarea deciziilor în procesul de turnare a microfiredelor. Ținând însă cont atât de sursa și natura datelor cât și de specificul problemei, poate fi menționat faptul că toți acești factori influențează într-o oarecare măsură asupra procesului de antrenare a rețelei neuronale. Asupra eficienței procesul de antrenare poate influența chiar și ordinea datelor. Rețelele neuronale proiectate și antrenate generalizează procesul decizional al operatorilor însă nu sunt în stare de a-l prelua complet. În comparație cu operatorul uman, rețelele neuronale nu sunt în stare să moștenească proprietățile ambigue ale acestuia deoarece acestea se comportă determinist fiind descrise prin funcții surjective. Același operator pus în aceleași condiții nu poate asigura o marjă de eroare suficient de mică fapt însă ce nu poate nega eficiența procesului decizional al acestuia.

Problema de bază a modelelor obținute o constituie faptul că acestea dispun de o eroare medie pătratică destul de mare în pofida faptului că acestea aproximează destul de bine comportamentul sistemelor decizionale cercetate. Valoarea mică a parametrului MSE indică faptul că sistemul modelat este aproximat destul de bine. Valoarea erorii medii pătratice trebuie interpretată în funcție de mărimea valorilor pe care le aproximează. Poate fi observat faptul că domeniul de valori și MSE au același ordin, 10^1 . Acest fapt semnifică că erorile calculate reprezintă valori de ordinul $10^{0.5}$ [102].



a)

b)

Figura 2.28 Graficul aproximării proceselor decizionale ale operatorilor de către rețelele neuronale antrenate în baza datelor colectate de la (a) operatorul 1 și (b) operatorul 2.

Analizând datele colectate de la ambii operatori se poate observa faptul că datorită comportamentului ambiguu eroarea obținută în rezultatul acțiunilor întreprinse de către aceștia în condiții asemănătoare manifestă valori destul de mari a erorii, fapt ce nu influențează critic asupra procesului de turnare. Faptul că valorile mari ale erorii nu influențează procesul, în măsura în care acesta să cauzeze abateri semnificative de la cerințele de calitate, poate fi explicat prin aceea că operatorii mențin procesul în zona de siguranță în care abaterile decizionale nu sunt atât de critice. O argumentare a acestui fenomen poate fi construită în jurul faptului că valorile mari ale erorilor decizionale apar doar în situațiile când parametrii tehnologici se află în apropierea valorilor caracteristice ale parametrilor de calitate. Acest fapt permite operatorului să mențină procesul stabil în intervalul în care acesta evoluează destul de inert. În cazul în care procesul se depărtează de zona de siguranță, operatorul nu-și mai permite să facă erori decizionale atât de grave luând decizii ferme și cât mai corecte. Această observație semnifică faptul că doar valoarea erorii nu poate caracteriza pe deplin caracteristicile de performanță și de calitate a modelelor obținute. Parametrii respectivi trebuie evaluați în ansamblu cu valoarea parametrilor de intrare și ieșire a sistemului decizional. Astfel operatorul uman în procesul de conducere poate lua atât decizii referitoare la conducerea procesului de turnare cât și referitor la acuratețea acestor decizii. În cazul în care operatorul ține procesul sub controlul, acesta își poate permite să întârzie cu decizia, să ia o decizie mai puțin optimă sau chiar să ezite să interacționeze cu sistemul dacă această acțiune nu poate influența negativ asupra calității microfirului turnat, iar în cazul în care operatorul intuiește că procesul de turnare necesită acțiuni ferme și momentane, acesta execută acțiunile corespunzătoare pentru a asigura calitatea microfirului. Astfel experiența

îi poate sugera operatorului uman când acesta se poate comporta mai lejer și când acesta trebuie să-și mobilizeze toate forțele în procesul decizional.

În rezultatul etapei de sinteză a modelelor de preluare a cunoștințelor operatorului uman în baza rețelelor neuronale artificiale se poate menționa faptul că au fost obținute modele decizionale individuale pentru fiecare operator în parte. Modelele implementate manifestă trăsături comportamentale specifice operatorilor umani, datele experimentale ale cărora au fost utilizate pentru antrenare, însă odată cu experiența acestora au fost preluate și o serie de trăsături specifice, determinate de gradul de incertitudine al datelor, fapt ce nu poate fi clasificat ca trăsătură negativă în condițiile în care această particularitate constituie chintesența individuală a fiecărui expert uman. Această proprietate poate fi tratată ca dovadă a faptului că modelele decizionale obținute au un comportament foarte asemănător celui manifestat de către operatorii umani.

Dezvoltarea sistemelor inteligente, performante este însoțită de o mulțime de probleme destul de complexe printre care pot fi menționate problema identificării funcțiilor de apartenență, a setului de calificative și reguli de inferență corespunzătoare. Acest fapt a dus la apariția ideii de a aplica algoritmi la baza cărora s-ar fi aflat cât logica fuzzy atât și rețelele neuronale artificiale. Reieșind din specificul metodei, aceasta presupune aplicarea combinată a tehnicilor neuronale și fuzzy la soluționarea problemelor ingineresti în cazul în care aplicarea fiecărei tehnici în parte nu este în stare să rezolve problema în cauză. Această abordare ar oferi posibilități extinse de aplicabilitate prin cumulara părților forte ale acestor două metodologii și îmbunătățirea rezultatelor fiecărei metode în parte prin excluderea reciprocă a neajunsurilor acestora. Utilizarea acestor două metode la soluționarea clasei de probleme specifice din domeniul automatizării proceselor industriale prin extragerea experienței operatorilor umani și la automatizarea procesului de turnare a microfiredelor în special, lărgeste domeniul de cercetare și în rezultat poate extinde semnificativ potențialul aplicativ al acestei abordări. În caz particular sistemele neuronale pot fi utilizate în domeniul automatizării procesului de turnare a microfiredelor la automatizarea etapelor de identificare și ajustare a parametrilor arhitecturilor hardware reconfigurabile din componența sistemului fuzzy pentru comanda și controlul parametrilor de producție. Această abordare presupune prezența a două moduri de comportament distincte. În faza inițială sistemul trebuie antrenat, prin alimentarea acestuia cu seturi de date colectate de la operatorul uman cu scopul de a prelua experiența acestuia prin identificarea numărului și a valorilor parametrilor specifici fiecărui operator și fiecărui proces tehnologic, pentru a putea adapta structura sistemului fuzzy prin ajustarea parametrilor arhitecturilor hardware interne, ca

ulterior, în faza decizională, sistemul să manifeste un comportament inferențial caracteristic sistemelor fuzzy clasice.

Poate fi propusă de asemenea și o altă abordare diametral opusă caracterizată de faptul că procesul decizional ar fi realizat de o mulțime de rețele neuronale artificiale cu o specializare îngustă și care ar fi orientate spre asigurarea calității microfiredelor cu parametri specifici, iar logica fuzzy ar putea fi aplicată punctat la implementarea procesului decizional de arbitraj care în dependență de cerințele de calitate și cele tehnologice ar realiza prioritizarea algoritmilor specifici de luare a deciziilor. Neajunsul acestei abordări constă în necesitatea dezvoltării unui număr mare de rețele neuronale artificiale care depinde direct de nomenclatorul de producție. Acest fapt duce la creșterea complexității arhitecturale a sistemului decizional cât și la creșterea esențială a bazei de cunoștințe care în loc să realizeze stocarea unui volum minimal de date asigurat prin generalizarea experienței acumulate, este nevoită să stocheze date de antrenare a fiecărei rețele neuronale pentru a asigura diversitatea produsului final. Acest fapt reprezintă, în unele cazuri specifice, un compromis acceptabil deoarece permite eliminarea problemei overfitting-ului, problemă specifică care poate apărea în procesul de antrenare a rețelelor neuronale în cazul în care acestea încearcă să acumuleze și să generalizeze un volum mare de date nestructurate. În aceste cazuri cu cât rețeaua neuronală mai mult este antrenată să îndeplinească noi sarcini, cu atât mai rău aceasta îndeplinește sarcinile vechi. Ocolirea acestui neajuns este asigurată prin adăugarea de noi modele decizionale care manifestă același neajuns, însă pe domeniul său specific de funcționare asigură performanța și precizia necesară. În rezultat, în locul unui sistem inteligent general se obține un sistem mult mai complex format din subsisteme imperfecte, care în careva limite caracteristice pot oferi soluții parțiale satisfăcătoare. Astfel prin aplicarea unei metode de cumulare a acestor soluții parțiale poate fi obținut un sistem inteligent care ar funcționa conform cerințelor și specificațiilor necesare asigurând totodată și precizia necesară. În acest caz sarcina de agregare a tuturor soluțiilor parțiale oferite de rețelele neuronale artificiale din componența sistemului îi revine sistemului decizional strategic realizat în baza logicii fuzzy. Astfel sistemul decizional fuzzy poate selecta și aplica strategia potrivită de comandă și control în dependență de intrările curente ale sistemului de producție sau în dependență de cerințele și specificațiile tehnice față de produsul finit. Generalizarea procesului decizional în această abordare poate fi asigurată la nivelul ierarhic superior de către sistemul fuzzy și se limitează doar la numărul de reguli din componența motorului de inferență. Astfel procesul de turnare a microfiredelor cu parametri noi, pentru care nu există o strategie concretă în baza de cunoștințe și o rețea neuronală special antrenată, poate fi asigurat prin aplicarea parțială a strategiilor existente în dependență de parametrii de intrare ai sistemului. Trebuie menționat

faptul că această abordare va putea asigura cerințele de calitate ale produsului finit doar în cazul în care totalitatea soluțiilor parțiale din cadrul sistemului va asigura acoperirea domeniului respectiv. Pentru a asigura această acoperire va fi necesar ca sistemul să conțină nu doar un număr cât mai mare de soluții parțiale, dar ca acestea să caracterizeze dacă nu întreg spectrul de produse, atunci un domeniu cât se poate de larg și cât se poate de divers.

Utilizarea combinată a sistemelor fuzzy, neuronale și clasice necesită extinderea domeniului de cercetare în vederea aplicării soluțiilor mixte în comanda și controlul proceselor tehnologice industriale. Astfel adăugarea unui procesor clasic în arhitectura sistemelor inteligente menționate poate extinde esențial posibilitățile de implementare a algoritmilor de automatizare a procesului de filtrare a datelor, de calcul și ajustare a parametrilor specifici și de configurare a arhitecturilor hardware. Acest fapt deschide perspectiva implementării noilor algoritmi de luare a deciziilor sau de configurare și adaptare în baza diferitor metodologii cum ar fi algoritmi genetici.

2.4 Concluzii la capitolul 2

În rezultatul procesului de cercetare au fost obținute o serie de modele cu un comportament intuitiv, care demonstrează odată în plus posibilitatea preluării experienței expertului uman. Modelele dezvoltate oferă soluții pentru argumentarea deciziilor în precese de producție și în special în procesul de turnare a microfiredelor. Aceste modele și metode inteligente permit extragerea cunoștințelor și utilizarea acestora pentru suportul decizional al operatorului uman în procesul de producție. Aplicabilitatea acestor metode poate fi confirmată sau infirmată doar practic, iar interoperabilitatea acestor modele se poate estima doar în ansamblu. Poate fi evidențiat potențialul aplicării acestor modele la rezolvarea unor clase de probleme specifice în care o importanță destul de mare o are viteza de reacție și nu precizia acesteia.

3. ELABORAREA ARHITECTURILOR HARDWARE ADAPTIVE

3.1 Formularea problemei de implementare a modelelor dezvoltate

În rezultatul etapei de elaborare și testare a modelelor matematice au fost identificate acele modele care pot fi utilizate cât mai rezultativ pentru controlul automat al procesului industrial de turnare a microfirelor. Pornind de la specificul problemei expuse și ținând cont de metodele și modelele propuse pentru rezolvarea acesteia au fost propuse și metodele corespunzătoare de implementare a acestor modele matematice în hardware. Astfel, reieșind din specificul modelelor de conducere în care este utilizată logica fuzzy și rețelele neuronale artificiale, se pot defini două direcții diferite de implementare și anume implementarea software sau hardware a acestor modele. Deoarece procesul de conducere la turnarea microfirelor trebuie realizat în timp real iar deciziile pentru reglare trebuie luate prompt, aceste caracteristici implică necesitatea utilizării metodei hardware de implementare a algoritmilor de comandă și control. La etapa testării preventive a algoritmilor de comandă și control a procesului de turnare a microfirelor are sens implementarea software a algoritmilor decizionali care nu necesită resurse hardware suplimentare și exclude etapa de proiectare și implementare a hardware-ului specializat din ciclul de dezvoltare. Deoarece modelele și metodele propuse spre implementare, pe parcursul cercetărilor, pot suferi modificări și îmbunătățiri frecvente, implementarea de la bun început a acestora în hardware poate cauza dificultăți manifestate prin repetarea etapei de proiectare.

Reieșind din natura procesului de cercetare a problemei, din start, a fost formulată problema proiectării și implementării unor astfel de arhitecturi hardware, care ar exclude total sau cel puțin ar minimiza semnificativ, etapa de proiectare și implementare repetată a versiunilor îmbunătățite ale algoritmilor de comandă și control. Cu acest scop a fost formulată sarcina de a proiecta arhitecturi hardware adaptive și reconfigurabile care ar fi capabile să se adapteze, în timp real, la noi condiții și cerințe inițiale [103, 104]. De asemenea a fost argumentată, prin minimizarea timpului de reproiectare și implementare a algoritmilor de comandă și control, necesitatea proiectării unor arhitecturi hardware parametrizate.

3.2 Proiectarea arhitecturii pentru sisteme fuzzy cu autoorganizare

Pentru implementarea modelelor matematice elaborate și prezentate în capitolul 2 este necesar de a proiecta în parte fiecare componentă din structura unui sistem fuzzy: fuzificator, motor de inferență și defuzificator. Soluția propusă pentru implementarea sistemului decizional are la bază structura clasică a unui sistem fuzzy cu unele modificări specifice. Arhitecturile

propușe implementează fuzificatorul în baza funcțiilor de apartenență definite tabelar prin intermediul blocurilor de memorie [96]. Capacitatea fiecărui bloc de memorie depinde direct de domeniul de valori al parametrului pe care îl definește. Pentru definirea funcțiilor de apartenență și a formelor acestora a fost utilizată variația relativă a rezistenței firului și variația presiunii în tubul de sticlă. Implementarea acestei metode de fuzificare presupune utilizarea circuitelor de memorie, care oferă posibilitatea definirii unor funcții de apartenență neliniare, diferite de cele clasice, fapt care poate fi destul de convenabil în cazul unor probleme mai specifice.

Avantajul principal al fuzificatorului implementat cu ajutorul acestei metode constă în posibilitatea redării unor funcții de apartenență neliniare destul de complexe. Un alt avantaj îl reprezintă utilizarea blocurilor de memorie cu ajutorul cărora pot fi definite funcții de apartenență care pot fi ulterior modificate dinamic. Acest fapt oferă o flexibilitate enormă sistemului proiectat în raport cu sistemele clasice. De asemenea ca avantaj poate fi considerat timpul mic de calcul al valorii funcției de apartenență.

Dezavantajul principal al abordării date îl reprezintă utilizarea unei capacități mari de memorie pentru definirea tuturor funcțiilor de apartenență ale calificativelor variabilelor fuzzy. Capacitatea memoriei utilizate poate fi micșorată prin micșorarea domeniilor de valori.

La etapa de identificare a variabilelor fuzzy și a funcțiilor de apartenență a acestora trebuie ținut cont de cerințele față de precizia modelului proiectat. Definirea funcțiilor de apartenență este realizată pentru fiecare calificativ al fiecărei variabile de intrare. Pentru simplificarea arhitecturii sistemului, la etapa de proiectare, este utilizată reprezentarea valorilor probabilității cu ajutorul numerelor întregi de la 0 la 2^n . Acest interval este dictat de necesitatea asigurării preciziei dorite. Funcțiile de apartenență sunt implementate cu ajutorul blocurilor de memorie RAM/ROM realizate cu ajutorul megafuncțiilor *lpm_ram* sau *lpm_rom* din Quartus.

Pentru a defini tabelară funcțiile de apartenență și pentru a le înscrie în blocurile de memorie RAM/ROM a fost necesar de a realiza procedura de ajustare a acestora. Procedura dată constă în translarea domeniului de valori în partea dreaptă sau în partea stângă a axei x . Această procedură trebuie realizată pentru ca valoarea variabilei de intrare să reprezinte adresa celulei de memorie în care se află valoarea funcției de apartenență a calificativului respectiv.

Astfel celulele de memorie vor conține valorile probabilităților pentru funcțiile de apartenență deja deplasate pe axa x și nu valorile reale ale parametrilor de intrare.

Pentru implementarea modelului motorului de inferență cu arhitectură reconfigurabilă au fost analizate mai multe de soluții clasice apoi utilizate cele mai potrivite pentru soluționarea problemei în cauză. Pe lângă aceasta au fost propuse soluții ingineresti pentru rezolvarea

problemelor apărute la îmbunătățirea și adaptarea soluțiilor clasice la specificul problemei abordate.

Conceptul motorului de inferență generic se cere a fi definit în cazul soluționării unor probleme specifice de luare automată a deciziilor, algoritmi cărora se pot schimba în timp. De obicei aceste sisteme sunt caracterizate prin capacitatea de autoorganizare a procesului decizional fapt ce face dificilă proiectarea unui asemenea sistem, cât și implementarea unui astfel de algoritm decizional. Arhitecturile propuse pentru soluționarea acestor probleme pot fi utilizate atât pentru implementarea algoritmilor generalizați de luare a deciziilor cât și pentru asigurarea posibilității de reconfigurare dinamică [105].

Implementarea algoritmilor de luare a deciziilor în baza structurilor hardware generice poate simplifica drastic etapa de proiectare sau poate reduce semnificativ durata acesteia. Avantajele acestei abordări pot fi valorificate pe deplin la etapele de testare și racordare a modelelor dezvoltate în procesul ajustării acestora la noi cerințe tehnologice. Eficiența utilizării arhitecturilor proiectate poate fi reflectată în procesul proiectării sistemelor decizionale în baza motorului de inferență generic.

La proiectarea motorului de inferență generic a fost utilizată metodologia dezvoltării sistemelor fuzzy la baza căreia se află logica vagă, utilizarea căreia a suferit o răspândire largă în domeniile în care metodele clasice de soluționare a problemelor nu pot asigura rezultate scontate [106, 107]. Această metodologie este utilizată pentru descrierea proceselor decizionale neliniare sau probabilistice. Utilizarea regulilor de inferență configurabile în structura motorului de inferență oferă posibilitatea modificării în timp a acestora. Utilizarea arhitecturilor configurabile în componența structurii motorului de inferență, spre deosebire de procesoarele Fuzzy specializate, exclude etapa de reproiectare a circuitelor integrate prin utilizarea circuitelor FPGA care dispun de un număr mare de porturi de intrare/ieșire și necesită doar reconfigurarea circuitului [108]. Chiar și etapa de proiectare a unui nou nucleu fuzzy poate fi simplificată semnificativ prin utilizarea bibliotecilor de elemente logice fuzzy.

Motorul de inferență generic poate servi pentru dezvoltarea unor algoritmi decizionali implementați inițial în condițiile insuficienței de date [108]. Prin urmare, utilizarea motoarelor de inferență reconfigurabile oferă posibilitatea implementării diferitor algoritmi decizionali în sistemele fuzzy.

Pentru implementarea în hardware, a regulilor de inferență, a fost proiectată o librărie de elemente logice fuzzy după analogia celor din logica clasică, *Figura 3.1* [107].

A	B	min(A,B)
0	0	0
0	1	0
1	0	0
1	1	1

AND

A	B	max(A,B)
0	0	0
0	1	1
1	0	1
1	1	1

OR

Figura 3.1 Tabelele de adevăr a funcțiilor logice fuzzy AND și OR [107].

Cu acest scop a fost implementată porta logică fuzzy, parametrizată, *FUZZYAND*, *Figura 3.2*, realizată în baza unui comparator și a unui multiplexor cu două intrări și o ieșire. Parametrul *Width* a componentei grafice *FUZZYAND* reprezintă numărul de biți pe care sunt reprezentate intrările și ieșirile acesteia. Ieșirea multiplexorului generează valoarea minimă dintre cele două intrări. Codul AHDL al acestui element este prezentat în *Anexa 1.1*.

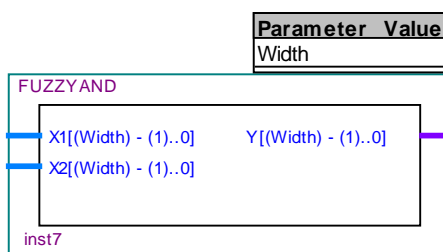


Figura 3.2 Reprezentarea grafică a componentei parametrizate FUZZYAND.

Pentru exemplificarea procedurii de sinteză a arhitecturii porții *FUZZAND* a fost prezentată schema obținută în rezultatul implementării unei porți logice pe 4 biți, *Figura 3.3*. Poate fi observat faptul că la sinteza acestei porți logice fuzzy au fost folosite sumatoarele binare pe 4 biți pentru implementarea comparatorului și o mulțime de porți logice în baza cărora a fost implementat multiplexorul cu două intrări și o ieșire pe 4 biți.

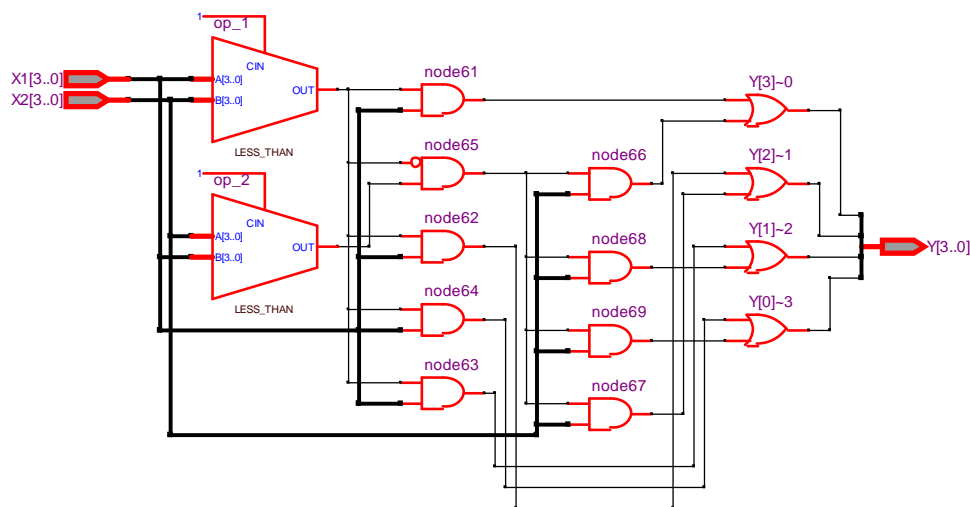


Figura 3.3 Structura internă a porții logice FUZZYAND pe 4 biți.

Pentru testarea porții logice *FUZZYAND* a fost generată diagrama de timp prezentată în *Figura 3.4*. Această diagramă de timp oferă posibilitatea de vizualizare a modului de funcționare a porții logice. Din diagramă se poate observa faptul că în cazul aplicării la cele două intrări a unor valori arbitrare, circuitul furnizează la ieșire valoarea minimă a acestora.

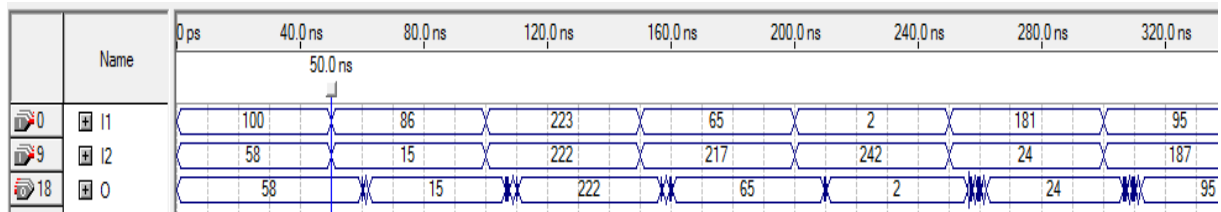


Figura 3.4 Diagrama de timp a porții logice **FUZZYAND**.

Pentru exemplificarea procedurii de sinteză a arhitecturii porții logice *FUZZYOR* a fost realizată sinteza unei porți logice pe 4 biți, *Figura 3.5*. Se poate observa faptul la sinteza acestei porți logice fuzzy au fost utilizate sumatoarele binare pe 4 biți pentru implementarea comparatorului și multiplexorul cu două intrări și o ieșire pe 4 biți implementat în baza porților logice [107]. Codul AHDL al acestui element este prezentat în *Anexa 1.2*.

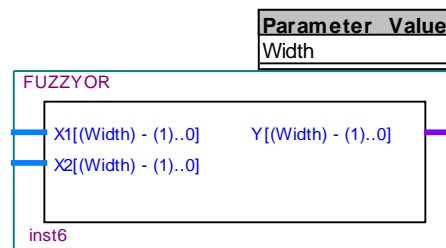


Figura 3.5 Reprezentarea grafică a componentei parametrizate **FUZZYOR**.

Structura internă a porții logice *FUZZYOR* este prezentată în *Figura 3.6*. Poate fi observat faptul că pentru implementarea comparatorului au fost folosite sumatoarele binare pe 4 biți iar pentru implementarea multiplexorului, cu două intrări și o ieșire pe 4 biți, a fost utilizată o mulțime de porți logice.

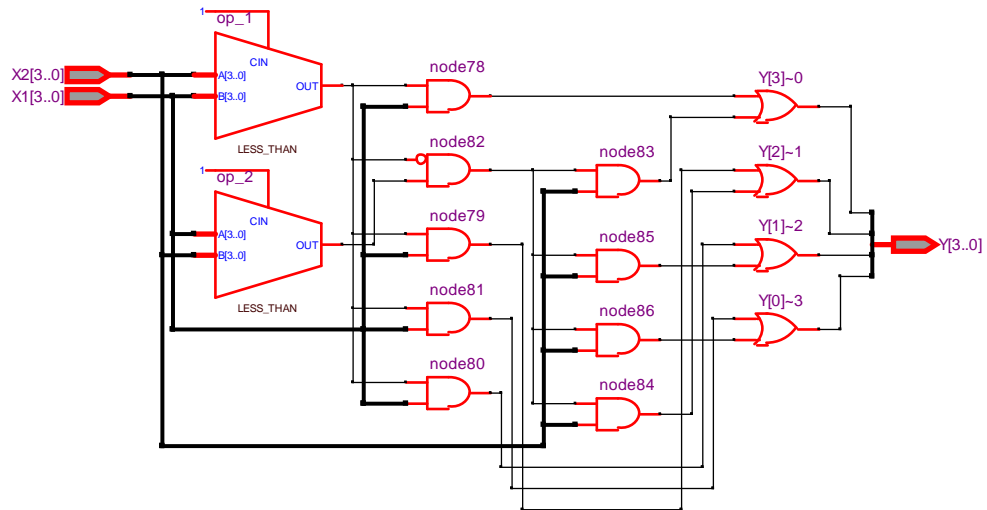


Figura 3.6 Structura internă a porții logice FUZZYOR pe 4 biți.

În urma testării porții logice *FUZZYOR* a fost generată diagrama de timp prezentată în *Figura 3.7*. Diagrama de timp oferă viziunea asupra modului de funcționare a acestei porți logice care constă în asigurarea la ieșire a valorii maxime dintre cele două valori aplicate la intrările componente.

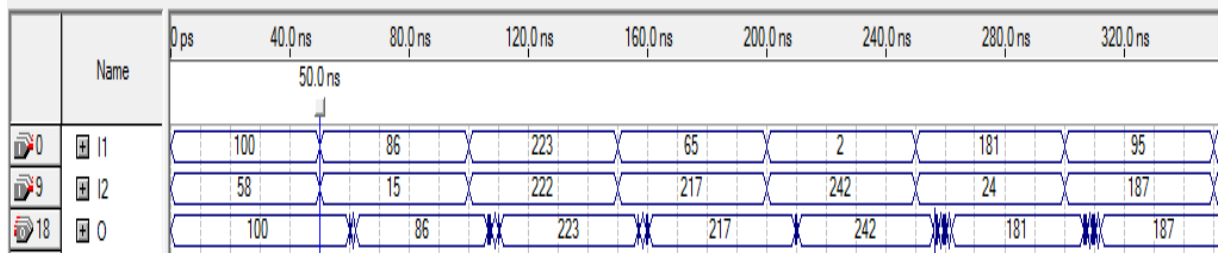


Figura 3.7 Diagrama de timp a porții logice FUZZYOR.

Utilizând librăria de elemente logice fuzzy, dezvoltată anterior, poate fi implementată hardware orice funcție logică fuzzy. Utilizarea setului dat de componente oferă avantaje enorme în procesul proiectării și implementării motoarelor de inferență din cadrul sistemelor decizionale fuzzy prin standardizarea și simplificarea maximă a procesul de sinteză a acestor sisteme.

După analogia cu matricele logice programabile a fost proiectată structura motorului de inferență reconfigurabil sau generic, care include în structura acestuia toate combinațiile conjunctive posibile dintre calificativele variabilelor de intrare. Ulterior acestea sunt aplicate la intrările unui bloc de multiplexoare pentru a realiza mulțimea disjuncțiilor definite de tabelul de adevăr. Reprezentarea fiecărei reguli din componența motorului de inferență este realizată în formă disjunctivă normală. Această formă de reprezentare oferă posibilitatea exprimării oricărei funcții logice fuzzy. Pentru a varia numărul de conjuncții din forma disjunctivă normală a regulii

de inferență a fost introdus semnalul *MIN* care are funcția de a anula întreaga construcție conjunctivă. Implementarea funcțiilor constante poate fi realizată prin aplicarea semnalelor *MIN* la toate conjuncțiile sau prin aplicarea semnalului *MAX* la prima conjuncție, fapt ce va duce la ignorarea valorilor celorlalte conjuncții.

Utilizarea acestei metode de reprezentare a funcțiilor logice oferă posibilitatea de a exprima orice regulă de inferență în forma disjunctivă normală (3.1):

$$C_{Cal\ i} = \sum_{k=1}^n Op_k; \quad (3.1)$$

unde $C_{Cal\ i}$ este variabila C determinată de calificativul i (3.2)

$$Op_i = \begin{cases} MIN, & PR[1..0] = '00' \\ MAX, & PR[1..0] = '01' \\ A_{Cal\ i} \ \& \ B_{Cal\ i}, & PR[1..0] = '10' \end{cases} \quad (3.2)$$

iar operanzii Op_i sunt descriși de relația (3.3)

$$\begin{aligned} Op_1 &= Var_1 Cal_1 \ \& \ Var_2 Cal_1 \ \& \ \dots \ \& \ Var_{N-1} Cal_1 \ \& \ Var_N Cal_1; \\ Op_2 &= Var_1 Cal_1 \ \& \ Var_2 Cal_1 \ \& \ \dots \ \& \ Var_{N-1} Cal_1 \ \& \ Var_N Cal_2; \\ &\vdots \\ Op_M &= Var_1 Cal_1 \ \& \ Var_2 Cal_1 \ \& \ \dots \ \& \ Var_{N-1} Cal_1 \ \& \ Var_N Cal_M; \\ &\vdots \\ Op_{M*N-1} &= Var_1 Cal_M \ \& \ Var_2 Cal_M \ \& \ \dots \ \& \ Var_{N-1} Cal_M \ \& \ Var_N Cal_{M-1}; \\ Op_{M*N} &= Var_1 Cal_M \ \& \ Var_2 Cal_M \ \& \ \dots \ \& \ Var_{N-1} Cal_M \ \& \ Var_N Cal_M; \end{aligned} \quad (3.3)$$

în care intrările $PR[n-1..0]$ reprezintă intrări de configurare ale celule logice fuzzy care permite configurarea dinamică a funcției acesteia.

Pentru cazul cu o variabilă de intrare și o variabilă de ieșire determinate de două calificative fiecare, regulile de inferență capătă forma:

$$C_{Cal\ i} = Op_1 \ V \ Op_2 \quad (3.4)$$

unde:

$$Op_1 = \begin{cases} MIN, & PR[1..0] = '00' \\ MAX, & PR[1..0] = '01' \\ A_{Cal\ 1}, & PR[1..0] = '10' \end{cases}, \quad (3.5)$$

$$Op_2 = \begin{cases} MIN, & PR[3..2] = '00' \\ MAX, & PR[3..2] = '01' \\ A_{Cal\ 2}, & PR[3..2] = '10' \end{cases}. \quad (3.6)$$

Aplicând această formă de reprezentare la implementarea algoritmilor decizionali de automatizare a procesului de turnare a microfiredelor, aceste reguli din componența motorului de inferență suferă o serie de transformări:

regula *if dRp is -dRp then dPv is -dPv*

se exprimă în forma $-dPv = Op_1 \vee Op_2$

unde:

$$Op_1 = -dRp \quad PR[1..0] = '10';$$

$$Op_2 = MIN \quad PR[3..2] = '00';$$

iar regula *if dRp is +dRp then dPv is +dPv*

se exprimă în forma $+dPv = Op_1 \vee Op_2$

unde:

$$Op_1 = MIN \quad PR[1..0] = '00';$$

$$Op_2 = +dRp \quad PR[3..2] = '10'.$$

Pentru testarea algoritmilor decizionali elaborați în baza unui motor de inferență generic au fost definite tabelar funcțiile de apartenență, pentru calificativele variabilei de intrare, apoi au fost încărcate în blocurile de RAM/ROM respective, astfel realizându-se blocul de fuzificare [107]. Ieșirile fuzzificatorului se aplică la intrările blocului de elemente logice *FUZZYAND* care realizează toate combinațiile conjunctive posibile ale calificativelor. Blocul de multiplexoare reprezintă însuși modulul de reconfigurare care în dependență de microprogramul înscris în memoria de program (vectorul *PR*) poate asigura accesul conjuncțiilor la blocul de elemente logice *FUZZYOR* în dependență de logica decizională elaborată. Astfel modificând microprogramul de configurare se poate ușor implementa orice regulă de inferență care ar asigura logica procesului decizional. Avantajul acestei metode constă în faptul că regulile de inferență și funcțiile de apartenență pot fi modificate în timp real doar prin modificarea datelor de configurare sau prin rescrierea funcțiilor de apartenență, fapt ce nu necesită reproiectarea arhitecturii sistemului fuzzy dezvoltat.

Cu scopul generalizării procesului de proiectare a sistemelor fuzzy a fost implementată componenta parametrizată *FUZZYCEL*, *Figura 3.8*. Aceasta reprezintă o celulă logică fuzzy, reconfigurabilă care în dependență de valoarea intrărilor de configurare poate realiza funcțiile *FUZZYOR*, *FUZZYAND* sau *MUX*, *Anexa 1.3*. Utilizarea acestei componente oferă omogenitate arhitecturii sistemului proiectat fiind implementat în baza aceluiași tip de elemente. Pe lângă avantajul menționat anterior se mai poate remarca faptul că odată cu utilizarea acestui tip de

componente la sinteza arhitecturii sistemelor decizionale fuzzy acestea capătă proprietatea de reconfigurare prin intermediul interfeței de scriere a datelor.

Celula parametrizată a motorului de inferență reconfigurabil este alcătuită din următoarele părți componente:

- $A[Width-1..0], B[Width-1..0]$ – intrări logice fuzzy reprezentate pe $Width$ biți;
- $OP[1..0]$ – intrare de configurare a funcției logice fuzzy a celulei parametrizate;
- $C[Width-1..0]$ – ieșire pe $Width$ biți a celulei parametrizate;
- $Width$ – parametrul care definește lățimea magistralelor de intrare și ieșire a celulei parametrizate.

În dependență de valoarea aplicată la intrarea de configurare, celula logică fuzzy poate fi configurată pentru a realiza una din următoarele funcții logice fuzzy binare sau unare:

- $OP[1..0] = '00'$ $FUZZYCEL(A, B) \equiv A$ – repetor a magistralei de intrare A;
- $OP[1..0] = '01'$ $FUZZYCEL(A, B) \equiv B$ – repetor a magistralei de intrare B;
- $OP[1..0] = '10'$ $FUZZYCEL(A, B) \equiv FUZZYAND(A, B)$;
- $OP[1..0] = '11'$ $FUZZYCEL(A, B) \equiv FUZZYOR(A, B)$;

unde A și B reprezintă variabile fuzzy reprezentate ca numere întregi, fără semn, pe $Width$ biți.

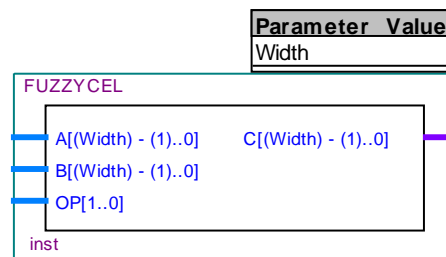


Figura 3.8 Reprezentarea grafică a celulei parametrizate **FUZZYCEL** a motorului de inferență reconfigurabil.

Se poate remarca faptul că componenta *FUZZYCEL* nu implementează în arhitectura internă valorile constante *MIN* și *MAX*. Aceste valori pot fi însă simplu obținute prin aplicarea constantelor *GND* și *VCC* la una din intrările componente cu aplicarea intrărilor de configurare respective pentru multiplexarea acestor valori la ieșire.

În rezultatul sintezei componente *FUZZYCEL* pe 4 biți a fost obținută arhitectura prezentată în *Figura 3.9*. Poate fi observat faptul că pentru implementarea funcționalităților necesare în arhitectura respectivă sunt incluse toate celelalte arhitecturi specificate anterior.

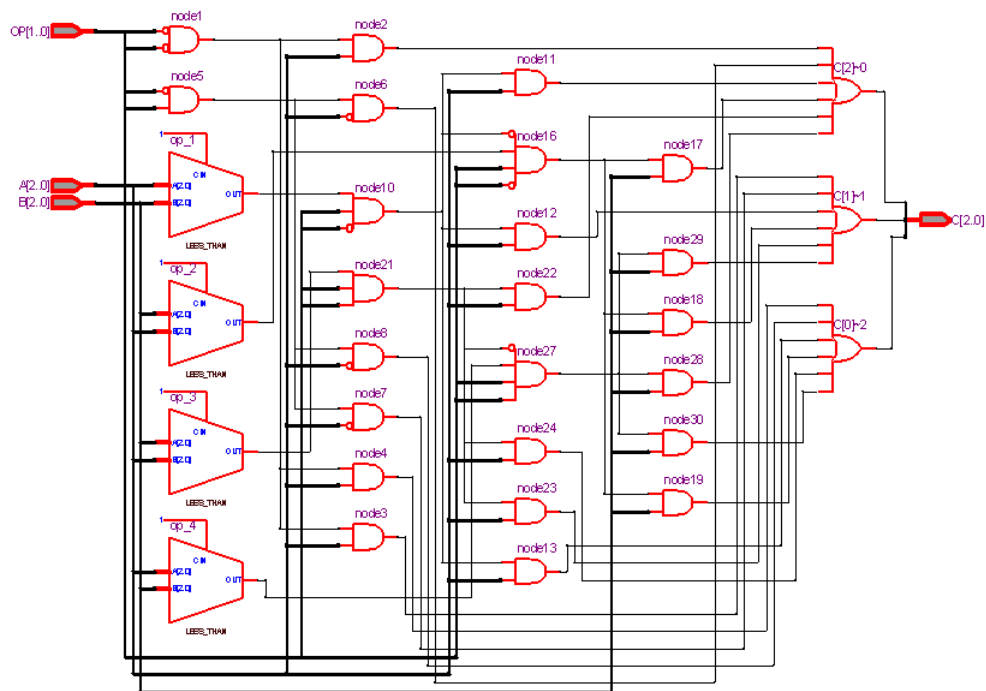


Figura 3.9 Structura internă a celei parametrizate a motorului de inferență reconfigurabil.

În concluzie poate fi menționat faptul că utilizarea componentei *FUZZYCEL* cauzează creșterea inevitabilă a complexității sistemului proiectat în așa fel, din cauză că acesta înglobează o funcționalitate mai extinsă, în schimb procesul de proiectare se simplifică esențial. Utilizarea practică a acestui tip de motoare de inferență poate fi valorificată în procesul cercetării sistemelor de luare a deciziilor automate în cazul în care regulile de inferență pot necesita modificări frecvente pe parcursul etapelor de proiectare, implementare și testare a algoritmilor decizionali. În acest caz algoritmul decizional al sistemului fuzzy poate fi ușor modificat fără modificarea arhitecturii interne. Sistemul decizional proiectat prin această metodă devine foarte flexibil, destul de universal și capătă un grad anumit de independență arhitecturală în raport cu schimbările algoritmului decizional, desigur în limitele resurselor utilizate și a problemei specificate.

Alegerea metodei de defuzificare în sistemele fuzzy depinde de caracteristicile specifice ale sistemului cercetat. Defuzificatorul în sistemele fuzzy reprezintă un bloc funcțional care are funcția de defuzificare a variabilelor de ieșire, procedură caracterizată prin conversia ieșirilor fuzzy din valori probabilistice în valori numerice specifice parametrilor procesului cercetat. În cazul în care cerințele față de sistem pun accent pe precizia procesului decizional, în detrimentul vitezei de calcul și a simplității sistemului, pot fi utilizate următoarele metode de defuzificare:

Mamdani, Larsen sau a “*Centrului de Masă*”, care în caz particular poate fi redusă la metoda *Sugeno de ordin 0*, în care ieșirea fiecărei reguli de inferență reprezintă o constantă. În cazul în care este necesar un compromis între viteza de calcul și complexitatea algoritmului de calcul, pot fi utilizate aceleași metode de defuzificare a funcțiilor discrete. În cazul sistemelor care nu specifică cerințe suplimentare față precizia procesului decizional, în procesul de defuzificare poate fi utilizată una din metodele: *Sugeno* sau *Metoda Maximului* [109].

Pentru simplificarea algoritmului decizional și pentru minimizarea utilizării resurselor hardware, a fost identificată pe cale experimentală cea mai potrivită metodă de defuzificare care este metoda *Maximului*. Implementarea acestei metode de defuzificare nu necesită proiectare de noi structuri arhitecturale și poate fi realizată în baza structurilor deja disponibile cum ar fi blocurile de memorie implementate la etapa de fuzificare și comparatoarele disponibile în librăria standard de componente din pachetului de programe *Quartus*. Pentru implementarea hardware a metodei *Maximului* de defuzificare a fost sintetizat un comparator pe 12 biți cu ajutorul megafuncției parametrice *lpm_compare*, *Figura 3.10*.

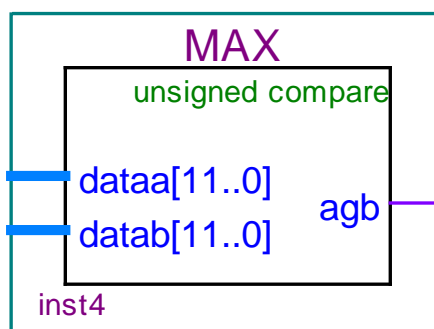


Figura. 3.10 Reprezentarea grafică a comparatorului din componența defuzicatorului la baza căruia se află metoda *Maximului* de defuzificare.

În urma utilizării megafuncției *lpm_compare* pentru sinteza comparatorului dispăre necesitatea descrierii hardware a acestui bloc funcțional, comportamentul său fiind descris automat în unul din limbajele de descriere hardware selectate de către utilizator. Blocul *Max* realizează defuzificarea prin selectarea valorii probabilității maxime din valorile probabilităților tuturor regulilor de inferență din componența motorului de inferență.

În rezultat se poate afirma cu certitudine faptul că în cazul problemei specificate eficiența maximă, în ceea ce privește utilizarea resurselor hardware cât și optimizarea resurselor de timp, se atinge prin aplicarea metodei *Maximului* de defuzificare. În final a fost elaborată baza completă de elemente logice fuzzy necesare pentru sinteza unui sistem decizional fuzzy integral. De asemeni a fost proiectată și implementată structura hardware care extinde potențialul

funcțional și arhitectural al sistemelor dezvoltate în baza acestor componente. Acest fapt a dat posibilitatea de a dezvolta arhitectura generică a motorului de inferență care oferă posibilitatea modificării în timp real a regulilor de inferență din componența acestuia fără intervenții arhitecturale în structura sistemului decizional. Modificarea regulilor de inferență poate fi realizată prin rescrierea datelor de configurare, iar extinderea lățimii magistralelor de date se realizează prin modificarea unui singur parametru.

3.3 Proiectarea arhitecturii pentru sisteme neuronale cu autoorganizare

Necesitatea dezvoltării unei arhitecturi hardware generice pentru implementarea rețelelor neuronale artificiale este determinată de neajunsurile și complexitatea metodelor clasice de implementare. Arhitectura generică a rețelelor neuronale artificiale a fost dezvoltată și implementată în cadrul cercetării procesului de turnare a microfiredelor cu scopul implementării modelelor neuronale de automatizare a acestuia.

Reieșind din specificul caracteristic al problemei vizate, a fost identificată cea mai potrivită arhitectură pentru implementarea rețelelor neuronale cu diverse arhitecturi clasice. Astfel pentru implementarea arhitecturilor clasice a rețelelor neuronale artificiale a fost utilizată arhitectura de tip feed-back – arhitectură care este reprezentată printr-un graf orientat, *Figura 3.11*.

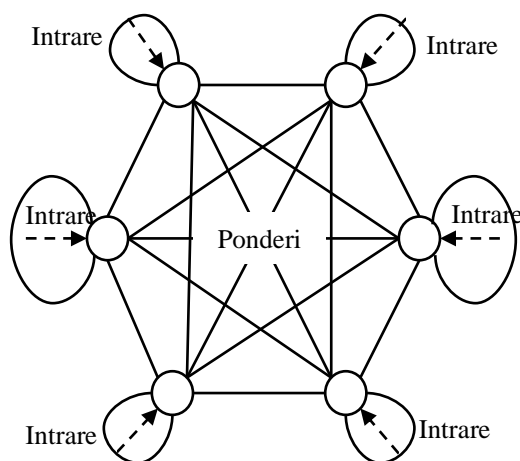


Figura 3.11 Rețea neuronală artificială cu arhitectură *Hopfield* [110].

Mulțimea arcelor acestui graf reprezintă conexiunile sinaptice între neuronii rețelei caracterizate prin ponderi sinaptice. Valorile acestora sunt calculate în dependență de metoda de învățare utilizată la etapa de antrenare a rețelei. Pentru simplificarea, generalizarea și optimizarea procesului de inferență, arhitectura rețelei neuronale e fost reprezentată printr-o matrice de

adiacență. Această metodă de reprezentare oferă posibilitatea implementării paralele a algoritmului de inferență a RNA pe o arhitecturi SIMD unde:

I – vectorul intrărilor externe și interne a RNA, (3.7), *Tabelul 3.1*;

O – vectorul ieșirilor/intrărilor interne a RNA, (3.8);

W – matricea ponderilor sinaptice;

Φ – funcție de activare.

$$I_{1 \times 2n} = I_{1 \times n} \hat{\wedge} O_{1 \times n} \quad (3.7)$$

$$O_{1 \times n} = \phi(I_{1 \times 2n} \times W_{2n \times n}) \quad (3.8)$$

Tabelul 3.1 – Structura vectorului de intrare a RNA generice.

I_1	I_2	...	I_n	O_1	O_2	...	O_n
-------	-------	-----	-------	-------	-------	-----	-------

Utilizarea modelului *Hopfield* pentru implementarea rețelelor neuronale artificiale generice poate contribui semnificativ la optimizarea etapelor de cercetare în domeniul rețelelor neuronale artificiale. Specificul procesului de inferență a rețelelor neuronale artificiale cu arhitectură de tip *Hopfield* poate fi caracterizat prin efectuarea succesivă a calcului până la stabilizarea rezultatului. În *Figura 3.11* este prezentat un model de rețea *Hopfield* în care fiecare neuron este conectat cu toți ceilalți. Ca rezultat se poate menționa faptul că modelul *Hopfield* reprezintă cea mai potrivită arhitectură pentru proiectarea RNA generice.

În procesul de cercetare accentul a fost pus pe procesul de generalizare a arhitecturii RNA prin realizarea unui număr cât mai mare de conexiuni între neuroni. Ieșirea fiecărui neuron a fost aplicată la intrările celorlalți neuroni cu scopul de a obține o rețea neuronală artificială deplin conectată. Legătura de feedback a fost realizată prin conectarea ieșirii fiecărui neuron la una din intrările acestuia. Intrările externe $I_{1 \times n}$ și interne/ieșirile $I_{(n+1) \times 2n} / (O_{1 \times n})$, reprezintă vectori de dimensiuni $1 \times n$ unde n reprezintă numărul de intrări externe, numărul de neuroni și numărul de ieșiri ale rețelei neuronale generice. Ponderile intrărilor externe și interne sunt stocate în matricea de ponderi sinaptice W de dimensiune $2 \times 2n$, *Tabelul 3.2*.

Tabelul 3.2 – Structura matricei ponderilor sinaptice a rețelei neuronale artificiale generice.

	N_1	N_2	\dots	N_n
I_1	$W_{1,1}$	$W_{1,2}$	\dots	$W_{1,n}$
I_2	$W_{2,1}$	$W_{2,2}$	\dots	$W_{2,n}$
\cdot	\cdot	\cdot	\cdot	\cdot
\cdot	\cdot	\cdot	\cdot	\cdot
\cdot	\cdot	\cdot	\cdot	\cdot
I_n	$W_{n,1}$	$W_{n,2}$	\dots	$W_{n,n}$
N_1	$W_{n+1,1}$	$W_{n+1,2}$	\dots	$W_{n+1,n}$
N_2	$W_{n+2,1}$	$W_{n+2,2}$	\dots	$W_{n+2,n}$
\cdot	\cdot	\cdot	\cdot	\cdot
\cdot	\cdot	\cdot	\cdot	\cdot
\cdot	\cdot	\cdot	\cdot	\cdot
N_n	$W_{2n,1}$	$W_{2n,1}$	\dots	$W_{2n,n}$

Utilizarea arhitecturii *Hopfield* la proiectarea rețelei neuronale artificiale generice face posibilă descrierea acesteia printr-o singură matrice sinaptică (de adiacență) cu dimensiunea $2 \times 2n$. Fiecare linie din această matrice reprezintă vectorul de ponderi al neuronului respectiv. Ponderile intrărilor externe sunt amplasate începând cu linia 1 și până la linia n a matricei $W_{2n \times n}$. Elementele, începând cu linia $n+1$ și până la linia $2n$ inclusiv, reprezintă ponderile conexiunilor interne ale rețelei neuronale. Utilizarea acestui model permite implementarea majorității topologiilor clasice ale rețelelor neuronale artificiale. Valoarea 0 în matricea ponderilor sinaptice indică lipsa conexiunii între intrarea externă și neuron sau între doi neuroni.

Specificul rețelei neuronale artificiale generice constă în posibilitatea modificării topologiei interne a acesteia fără modificarea arhitecturii hardware. Acest fapt oferă o mulțime de avantaje în procesul implementării arhitecturilor clasice ale rețele neuronale artificiale. În limitele resurselor existente rețeaua neuronală poate fi reconfigurată prin rescrierea matricei ponderilor sinaptice. Astfel devine posibilă implementarea oricărei arhitecturi neuronale ce nu depășește numărul de neuroni implementați inițial în hardware.

Pentru implementarea modelelor neuronale sunt necesare arhitecturi hardware specializate care ar fi în stare să realizeze procesul inferențial [105]. Implementarea în hardware a modelelor decizionale neuronale necesită implementarea arhitecturii rețele neuronale generice în baza circuitelor reconfigurabile [103, 111]. Deoarece la baza oricărei rețele neuronale se află modelul clasic al neuronului, sarcina se reduce la implementarea modelului parametrizat al

neuronului artificial. Pentru simplificarea procesului de dezvoltare a neuronului artificial a fost realizată divizarea arhitecturii acestuia în următoarele componente structurale și funcționale:

- sinapse prezentate prin blocul de multiplicare;
- corpul neuronului reprezentat prin blocul de sumare;
- funcție de activare reprezentată printr-un bloc aritmetic.

În rezultat principiul de funcționare a unui neuron poate fi redus la sumarea intrărilor amplificate și aplicarea unei funcții de activare asupra sumei obținute.

Modelul arhitectural al neuronului trebuie să dispună de un număr de intrări egal cu suma dintre numărul de intrări externe și numărul de neuroni al rețelei din care acesta face parte. Pentru generalizarea modelului rețelei neuronale artificiale a fost pusă condiția ca numărul de intrări externe să fie egal cu numărul de neuroni al acesteia. În rezultat a fost obținut numărul total de intrări al unui neuron care este egal cu $2n$, unde n este numărul de neuroni la rețelei. Astfel fiecare neuron dispune de n intrări externe și n intrări interne la care sunt conectate ieșirile celorlalți neuroni din rețea.

Modelul neuronului artificial a fost divizat în trei componente funcționale diferite: P , S și T , *Figura 3.12*, unde:

- P – reprezintă blocul de amplificarea sinaptică;
- S – reprezintă blocul de sumare;
- T – reprezintă funcție de activare.

Structura modulară a modelului parametric al neuronului artificial va oferi o flexibilitate sporită în procesul dezvoltării arhitecturilor neuronale, oferind dezvoltatorilor posibilitatea de a implementa arhitecturile proprii specializate pentru rezolvarea anumitor probleme. Astfel în procesul de dezvoltare inginerii pot selecta spre exemplu funcția de activare a fiecărui neuron în parte, sau își pot dezvolta funcția proprie de activare.

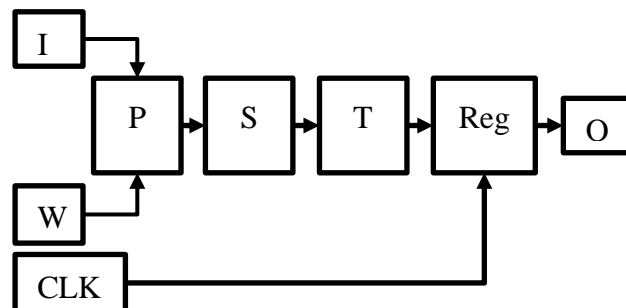


Figura 3.12 Schema de structură a neuronului digital.

Blocul de amplificare sinaptică, notat P , realizează înmulțirea fiecărui semnal de intrare cu ponderea corespunzătoare a acestuia, *Figura 3.11*. Componenta parametrizată de amplificare sinaptică, P , din structura neuronului este elaborată în *AHDL, Anexa 2.1*, cu ajutorul multiplicatoarelor care calculează paralel produsele dintre intrările și ponderile respective. Componenta parametrizată de amplificare sinaptică conține următorii parametri: *Width* – lățimea în biți a intrărilor și a ponderilor, și *Inputs* – numărul de intrări. La intrările acestei componente se aplică vectorul de intrare a RNA și vectorul ponderilor, iar semnalul de ieșire reprezintă vectorul de produse obținut prin înmulțirea fiecărei intrări cu ponderea respectivă.

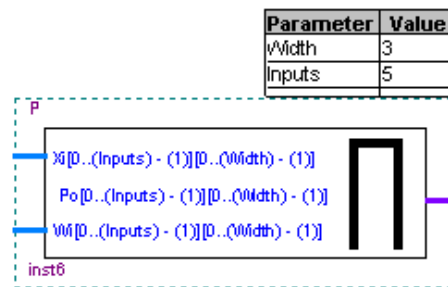


Figura 3.13 Reprezentarea grafică a componente de amplificare sinaptică.

În rezultatul sintezei componente de amplificare cu 5 intrări de date, 5 intrări pentru ponderi și 5 ieșiri pe 3 biți fiecare, au fost obținute 5 structuri asemănătoare cu cea prezentată în *Figura 3.14* în care se poate observa tipul componentelor utilizate.

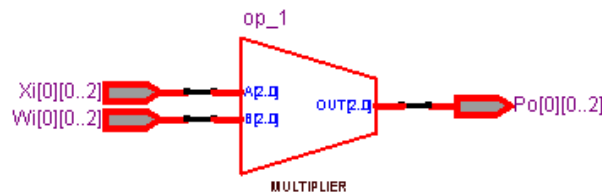


Figura 3.14 Structura internă a componente de amplificare sinaptică.

Poate fi observat faptul că în cazul dat pentru fiecare intrare a fost rezervat câte un multiplicator care are funcția amplificării acesteia. În cazul în care va crește lățimea magistralelor de intrare aceasta va cauza creșterea numărului resurselor utilizate.

Blocul de sumare, notat S , realizează cumularea tuturor intrărilor, *Figura 3.15*. Componenta parametrizată de sumare din structura neuronului este elaborată în *AHDL, Anexa 2.2*, cu ajutorul sumatoarelor care, în baza unui arbore *Wallace*, calculează suma respectivă. La intrările acestei componente se aplică vectorul produselor sinaptice iar ieșirea reprezintă un scalar.

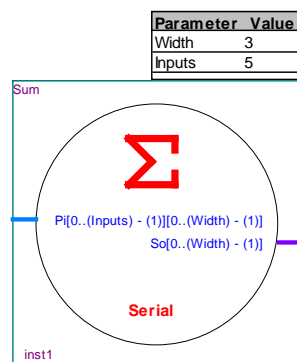


Figura 3.15 Reprezentarea grafică a componentei de acumulare paralelă.

În rezultatul sintezei componentei parametrizate de sumare cu 5 intrări și o ieșire pe 3 biți, a fost obținută structura prezentată în *Figura 3.16* în care se poate observa arhitectura arborelui *Wallace* la baza căreia se află o mulțime de sumatoare interconectate. Pentru a elimina posibilitatea apariției depășirii domeniului de valori în procesul de calcul este necesar de calculat numărul de biți al sumatorului în dependență de domeniul de valori al datelor de intrare.

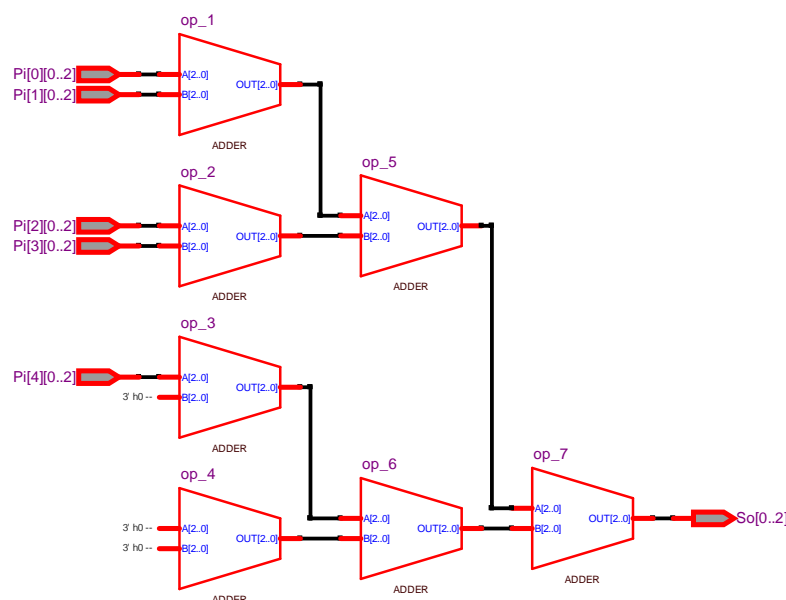


Figura 3.16 Structura internă a componentei de acumulare paralelă

Pentru calculul sumei tuturor produselor sinaptice acestea se adună două câte două obținându-se sume parțiale pentru care se repetă aceeași procedură până când nu rămâne doar o singură sumă care este generată la ieșire. Se poate observa faptul că utilizarea acestei metode de sumare, în raport cu sumarea secvențială, oferă o întârziere aproximativ de două ori mai mică, în schimb implică utilizarea unui număr mult mai mare de sumatoare. Trebuie menționat faptul că din cauză că sinteza blocului de sumare este efectuată în baza unei componente parametrizate, întârzierile care pot apărea în arhitectura finală pot varia în dependență de valorile parametrilor, iar faptul respectiv trebuie luat în calcul în procesul de sincronizare a blocurilor.

Blocul parametrizat pentru implementarea funcției de activare, notat T , realizează calculul valorii funcției de activare în dependență de valoarea sumei totale a produselor sinaptice aplicate la intrarea acestui bloc, *Figura 3.17*. Pentru implementarea diferitor funcții de activare au fost elaborate în *AHDL*, *Anexa 2.3*, câteva blocuri parametrizate pentru o serie de funcții clasice cum ar fi: *treaptă*, *liniară*, *sigmoidă*, etc. În dependență de funcția de activare implementată la sinteza arhitecturii acesteia poate fi utilizat un număr mai mare sau mai mic de resurse.

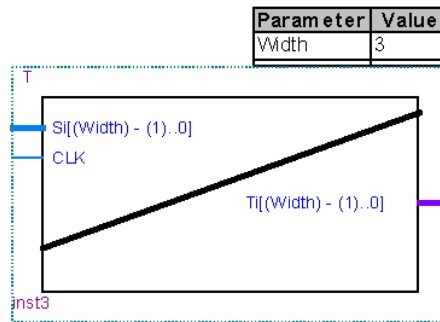


Figura 3.17 Reprezentarea grafică a componentei pentru funcția de activare liniară.

Spre deosebire de celelalte componente, blocul funcției de activare conține un singur parametru, *Width*. Ca exemplu clasic de implementare a funcțiilor de activare în baza circuitelor FPGA a fost prezentată funcția *liniară*, *Figura 3.17*.

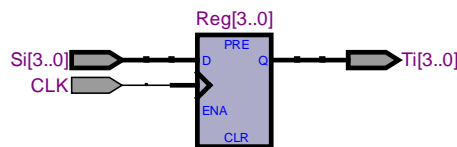


Figura 3.18 Structura internă a componentei pentru funcția de activare liniară.

Componenta pentru calculul funcției de activare conține un registru de *Width* biți pentru păstrarea rezultatului, *Figura 3.18*. Acest fapt introduce necesitatea sincronizării componentei respective prin adăugarea semnalului de tact, *CLK*.

Structura ierarhică a modelului neuronului artificial permite dezvoltatorilor să adapteze structura internă a acestuia prin utilizarea diferitor funcții de activare. Modelul neuronului parametrizat face ca intrările și ieșirile sale să poată lua valori binare sau întregi cu semn. El dispune de intrări externe și intrări ale ponderilor sinaptice. Codul *AHDL* al neuronului parametrizat, *Anexa 2.4*, este implementat modular cu ajutorul librărilor de componente parametrizate P , S și T care sunt incluse în programul de bază. Procesul de funcționare al fiecărui neuron este sincronizat pe frontul crescător al semnalului de ceas, *Figura 3.18*.

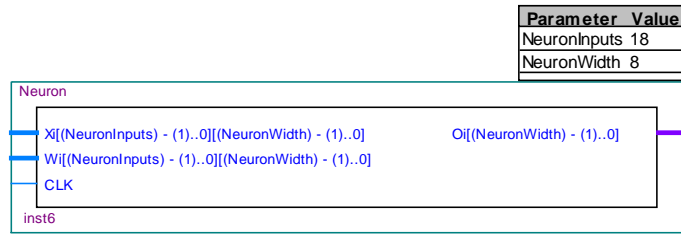


Figura 3.18 Modelul parametrizat a neuronului artificial.

În rezultatul sintezei arhitecturii neuronului parametrizat a fost obținută structura prezentată în *Figura 3.19* în care poate fi remarcată asemănarea conceptuală cu modelul neuronului artificial prezentat în *Figura 3.11*.

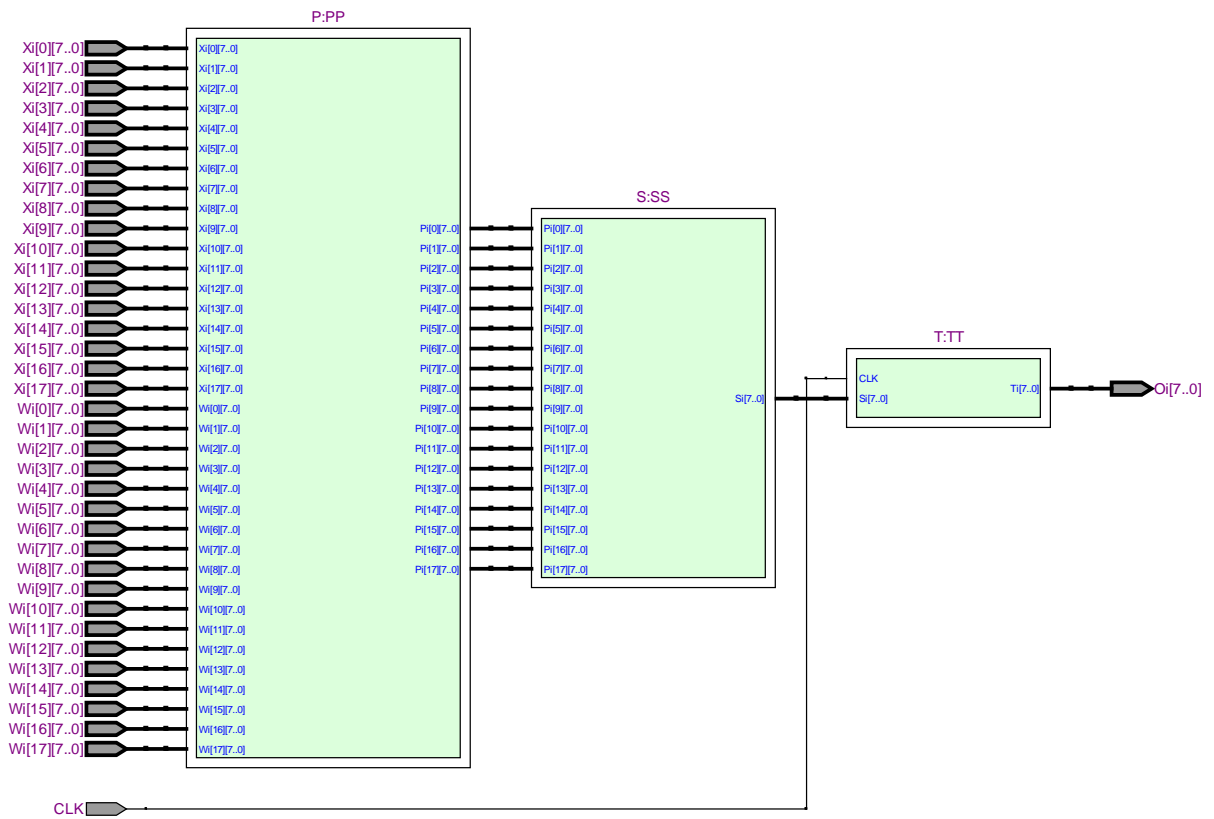


Figura 3.19 Modelul parametrizat a neuronului artificial.

Utilizarea librăriei de componente specializate și a modelului neuronului parametrizat în special, la etapa de proiectare a rețelei neuronale artificiale, oferă posibilitatea de ajustare a parametrilor acestuia în dependență de necesitățile sistemului proiectat. Utilizarea în blocul de sumare a arborelui Wallace duce la optimizarea procesului de calcul inferențial, iar buferizarea ieșirii neuronului face posibilă implementarea regimului de funcționare pipeline al acestei arhitecturi [112].

Utilizarea modelului parametrizat al neuronului artificial simplifică semnificativ procesul de proiectare a rețelelor neuronale artificiale, însă totuși presupune efectuarea repetată a procedurilor monotone de plasare manuală a componentelor și de trasare a conexiunilor între acestea. Pentru a simplifica și mai mult acest proces, în baza modelului neuronului parametrizat, a fost conceput modelul parametrizat al unei rețele neuronale artificiale generice *Figura 3.20*. Modelul dezvoltat conține doi parametri:

NetWidth – numărul de biți pe care vor fi reprezentate toate intrările și ieșirile neuronului;

NetNeurons – numărul total de neuroni din componența rețelei neuronale artificiale.

Componenta *Network* dispune de următoarele intrări:

I – vectorul intrărilor externe;

W – matricea ponderilor sinaptice;

CLK – semnal de sincronizare;

O – vectorul ieșirilor.

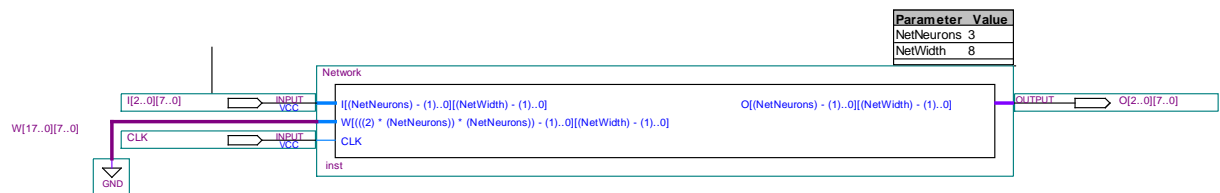


Figura 3.20 Componenta parametrizată a rețelei neuronale artificiale generice.

Pentru exemplificare a fost implementată o rețea neuronală artificială generică, *Anexa 2.5*, de tip Hopfield schema de structură a căreia a fost obținută în rezultatul sintezei arhitecturii hardware corespunzătoare prezentată în *Figura 3.21*. Această structură simplifică perceperea vizuală a direcției de propagare a datelor în procesul de inferență ciclică.

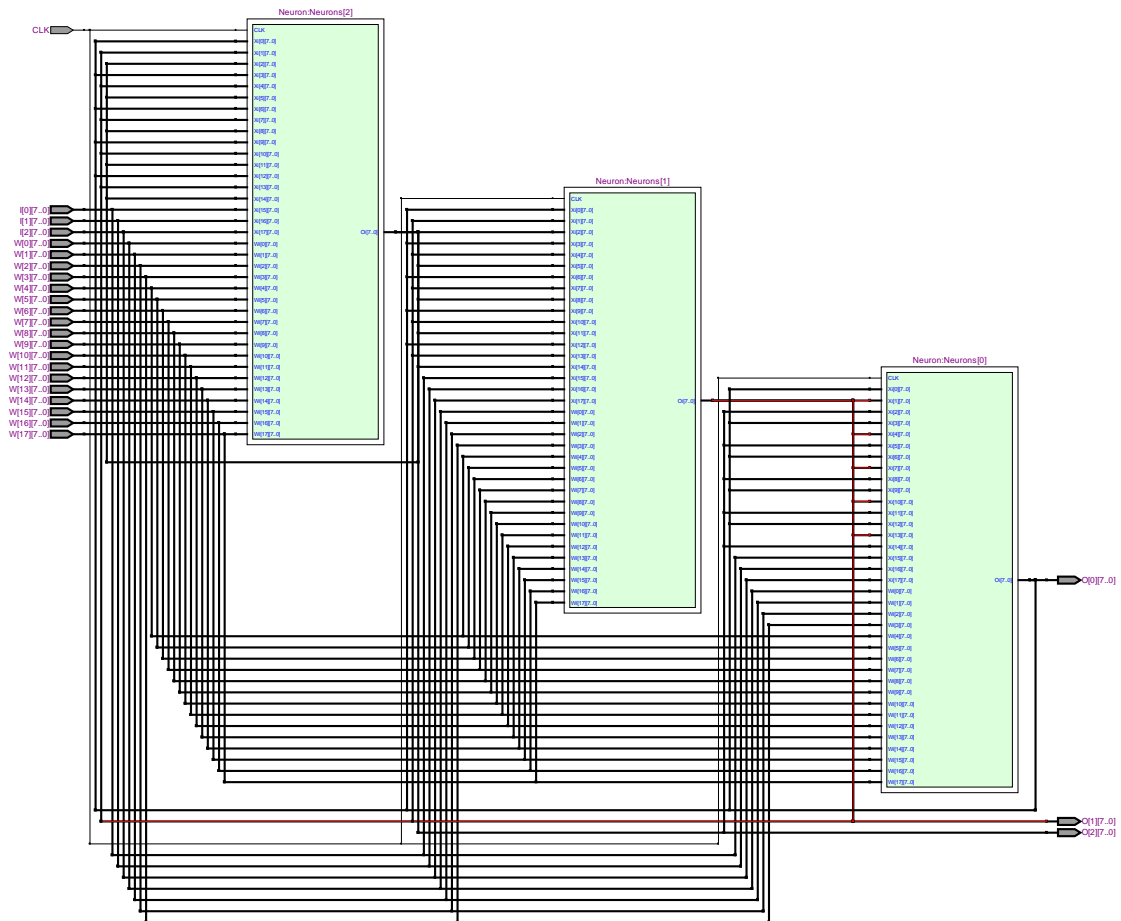


Figura 3.21 Schema de structură a rețelei neuronale artificiale generice.

Pentru implementarea modelelor neuronale de automatizare a procesului de turnare a microfiredelor au fost dezvoltate și implementate metode și instrumente de sinteză a rețelelor neuronale artificiale în baza circuitelor FPGA. Utilizarea circuitelor reconfigurabile pentru implementarea metodelor de extragere a cunoștințelor umane nu reprezintă avantaje funcționale remarcabile deoarece principiul de funcționare a sistemului în general rămâne neschimbat. Cu toate acestea, la etapa de proiectare, pot fi optimizate careva funcționalități cu scopul de a spori viteza de calcul. Un avantaj esențial al metodei respective de implementare arhitecturală îl reprezintă posibilitatea optimizării arhitecturale caracterizată prin faptul că întregul sistem poate fi proiectat pe un singur circuit pentru a reduce esențial cheltuielile de hardware și dimensiunile sistemului în ansamblu. Avantajul principal al implementării rețelelor neuronale în baza circuitelor reconfigurabile poate fi valorificat doar în cazul utilizării circuitelor FPGA, fapt ce oferă posibilitatea paralelizării procesul inferențial [103].

3.4 Simularea și analiza comparativă a soluțiilor arhitectural – structurale propuse

În rezultatul implementării sistemelor decizionale în baza algoritmilor fuzzy și neuronali a apărut problema utilizării cât mai eficiente a resurselor hardware ale circuitelor FPGA. Pentru a

descoperi dependențele dintre parametrii de interes ai sistemelor cercetate a fost realizată o serie de simulări cu diferite valori de intrare ale acestora. În dependență de tipul sistemului fuzzy sau neuronal pot varia și parametrii caracteristici ai acestora. Spre exemplu din natura procesului de inferență fuzzy se poate remarca dependența sporită a acestuia de capacitatea zonei de memorie și lungimea cuvântului prin care sunt reprezentate funcțiile de apartenență și dependența mult mai mică de numărul elementelor logice ale circuitului cauzate de simplitatea relativă a procesului de inferență, pe când specificul arhitectural al rețelei neuronale generice proiectate presupune atât utilizarea unui spațiu destul de larg de memorie pentru stocarea ponderilor sinaptice, a intrărilor și ieșirilor, cât și utilizarea elementelor logice pentru asigurarea conexiunilor între entitățile participante la procesul de inferență. Pentru a se putea remarca direct dependența dintre parametrii de interes și parametrii arhitecturali ai sistemului a fost decis de a o estima în baza sistemelor decizionale autonome, fără includerea unităților centrale de procesare și a interfețelor de comunicare. Această restricție va oferi o viziune mai profundă asupra caracteristicilor individuale ale arhitecturilor decizionale fără a induce și alte dependențe. Pentru a realiza analiza respectivă este necesar de a stabili mai întâi de toate lista parametrilor arhitecturai de intrare și a parametrilor de interes pentru sistemele cercetate. În cazul arhitecturii decizionale fuzzy ca parametri de intrare pot servi numărul de intrări, numărul de calificative lingvistice și lățimea magistralelor de date. Reieșind din specificul sistemului decizional fuzzy proiectat careva parametri pot fi considerați constanți și predefiniți. În rezultat ca parametru de intrare a arhitecturii fuzzy rămâne doar lățimea magistralelor de date. În cazul arhitecturii decizionale neuronale, din cauza corelației parametrilor arhitecturali prevăzute la etapa de proiectare a acestora, ca parametri de intrare pot fi considerați numărul de neuroni și lățimea magistralelor de date. Din cauza specificului algoritmului decizional implementat în baza a 10 neuroni artificiali, în final, ca parametru de intrare rămâne doar lățimea magistralelor de date. Pentru identificarea parametrilor de interes a arhitecturilor hardware proiectate și implementate în baza circuitelor FPGA este necesar de a analiza mulțimea parametrilor disponibili din raportul de simulare generat de pachetul de programe *Quartus*. Astfel ca parametri de interes pot servi: timpul total de simulare a proiectului, familia și tipul circuitului utilizat, frecvența minimă de lucru a arhitecturii, numărul elementelor logice utilizate pentru implementarea arhitecturii, puterea maximă de consum, memoria utilizată și numărul de regiștri. Astfel a fost realizată simularea sistemului decizional fuzzy cu lungimea variabilă a cuvântului de date, *Tabelul 3.3*.

Tabelul 3.3 Rezultatele simulării sistemului decizional fuzzy

Familia	Circuitul	Lungimea cuvântului (biți)	Timp de simulare (minute)	CLK (MHz)	Elemente logice	Memorie (biți)
Cyclone II	EP2C70F896C6	8	2:43	260.01	33/68,416 (<1%)	8192/1152000
		12	2:00	161.29	5/33,216 (<1%)	195,608/1152000

În rezultatul simulării sistemului decizional fuzzy în dependență de lungimea cuvântului au fost depistate unele relații neevidente dintre parametrii de bază ai arhitecturii și parametrii de interes cum ar fi: odată cu creșterea lungimii cuvântului durata de compilare a proiectului scade în pofida așteptărilor, în condițiile respective poate fi observată scăderea frecvenței maxime admisibile a semnalului de sincronizare care poate fi cauzată atât de numărul și modul de amplasare a componentelor structurale ale arhitecturii hardware, cât și de lungimea traseelor dintre acestea. În pofida așteptărilor, odată cu creșterea lungimii cuvântului se observă o scădere a resurselor logice utilizate, împreună cu creșterea memoriei utilizate. Toate aceste fenomene pot fi cauzate de caracteristicile individuale ale algoritmului de plasare a structurilor dezvoltate pe resursele hardware ale circuitului FPGA. Capacitatea totală a memoriei utilizate pentru sinteza arhitecturii decizionale fuzzy poate fi calculată ca suma dintre capacitatea memoriei utilizate la implementarea fuzzificatorului pentru definirea tabelară a funcțiilor de apartenență și capacitatea memoriei utilizate la implementarea defuzzificatorului:

$$M_{totală} = M_{fuzzyficare} + M_{defuzzyficare} \quad (3.9)$$

unde:

$$M_{fuzzyficare} = \sum_{i=1}^I 2^{b_i} \times b_i \times N_i^v \quad (3.10)$$

și

$$M_{defuzzyficare} = \sum_{j=1}^O 2^{b_j} \times b_j \times N_j^v \quad (3.11)$$

în care I reprezintă numărul intrărilor sistemului fuzzy, O – numărul ieșirilor, b – lungimea cuvântului de reprezentare a valorii variabilei lingvistice de intrare cu indicele i sau de ieșire cu indicele j , N^v – numărul variabilelor lingvistice de intrare cu indicele i sau de ieșire cu indicele j .

Având în vedere specificul procesului decizional implementat în care lungimile de reprezentare a tuturor variabilelor lingvistice de intrare și de ieșire sunt indicate de același parametru cu o valoare prestabilită, relația de mai sus capătă forma:

$$M_{totală} = I \times 2^b \times b \times N^v + O \times 2^b \times b \times N^v \quad (3.12)$$

de unde

$$M_{totală} = 2^b \times b \times N^v \times (I + O) \quad (3.13)$$

iar reieșind din condiția că numărul variabilelor lingvistice de intrare trebuie să fie egal cu numărul variabilelor lingvistice de ieșire $O = I$ rezultă relația:

$$M_{totală} = 2 \times I \times 2^b \times b \times N^v \quad (3.14)$$

din care este obținută relația finală:

$$M_{totală} = I \times 2^{b+1} \times b \times N^v. \quad (3.15)$$

Înlocuind parametrii arhitecturali în relația anterioară pot fi argumentate valorile capacității memoriei obținute experimental în rezultatul simulării arhitecturii decizionale fuzzy și prezentate în *Tabelul 3.3*: pentru lungimea cuvântului de 8 biți $M_{totală} = 1 \times 2^{8+1} \times 8 \times 2 = 8192$ și respectiv $M_{totală} = 1 \times 2^{12+1} \times 12 \times 2 = 196608$ pentru 12 biți.

În rezultatul simulării arhitecturii decizionale fuzzy proiectate pot fi menționate o serie de caracteristici specifice cum ar fi: utilizarea preponderent a resurselor de memorie disponibile pe circuitul FPGA pentru definirea funcțiilor de apartenență și utilizarea moderată a resurselor logice pentru realizarea procesului de inferență. Aceste dependențe neliniare dintre creșterea volumului resurselor de memorie și a volumului de resurse logice duce la o serie de limitări arhitecturale specifice cum ar fi imposibilitatea scalării proporționale a arhitecturii proiectate în baza circuitului cercetat din cauza creșterii exponențiale a resurselor de memorie necesare pentru implementarea acesteia. Din aceste considerente devine imposibilă implementarea pe acest circuit a arhitecturii decizionale fuzzy cu lungimea cuvântului de 16 biți, deoarece arhitectura respectivă depășește aproximativ de 4 ori limitele resurselor de memorie de 1,152,000 biți disponibile pe circuitul EP2C70F896C6, iar arhitectura respectivă necesită cel puțin 4,194,304 biți. Informația respectivă poate fi utilă pentru selectarea cu destulă precizie a familiei și circuitului individual în dependență de parametrii de bază a sistemului fuzzy care trebuie implementat.

În rezultatul analizei datelor colectate în urma simulării arhitecturii decizionale neuronale cu diferiți parametri arhitecturali pe familii diferite de circuite reconfigurabile a fost completat *Tabelul 3.4* în baza căruia au fost construite și o mulțime de grafice prezintă o serie de dependențe dintre parametrii de interes ai arhitecturii respective.

Tabelul 3.4 Rezultatele simulării sistemului decizional neuronal

Familia	Circuitul	Lungime a cuvântului (biți)	Timp de simulare	CLK (MHz)	Elemente logice	Registre
MAX10	10M50SAE144I7G	8	3:36	46.01	4,046 / 49,760 (8 %)	1765
		16	8:07	37.22	16,449 / 49,760 (33 %)	3525
		32	–	–	–	–
Cyclone IV	EP4CE115F29I7	8	02:33	46.85	3,962 / 114,480 (3 %)	1765
		16	04:09	36.91	7,743 / 114,480 (7 %)	3525
		32	24:40	34.14	59,560 / 114,480 (52 %)	7045
Cyclone 10	10CL120YF780I7G	8	2:38	47.13	3,996 / 119,088 (3 %)	1765
		16	4:11	37.63	7,727 / 119,088 (6 %)	3525
		32	23:59	33.15	57,041 / 119,088 (48 %)	7045

Analizând parametrii de interes în cazul simulărilor realizate în baza circuitelor din familia MAX 10, Cyclone IV și Cyclone 10 au fost construite graficele din *Figura 3.22*, *Figura 3.23* și *Figura 3.24* în care sunt prezentate dependențele dintre lungimea cuvântului, numărul elementelor logice utilizate pentru sinteza arhitecturii corespunzătoare, frecvența maximă a semnalului de sincronizare și numărul regiștrilor utilizați.

În cazul circuitelor analizate din familiile MAX și Cyclone poate fi remarcată creșterea exponențială a numărului elementelor logice, relativ la creșterea lungimii cuvântului, utilizate pentru sinteza arhitecturii decizionale neuronale.

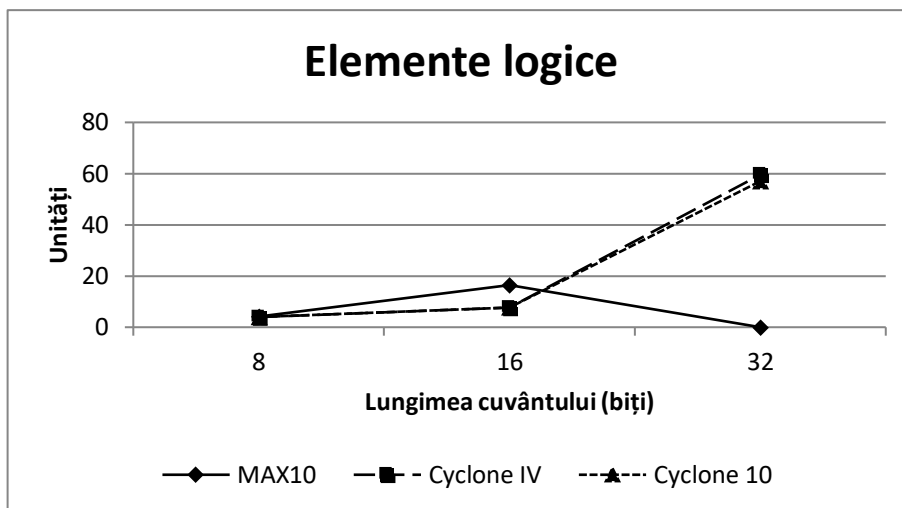


Figura 3.22 Graficul dependenței dintre lățimea cuvântului și numărul elementelor pentru circuitele din familiile MAX și Cyclone.

În cazul celui mai performant circuit din familia MAX 10, 10M50SAE144I7G poate fi menționat faptul că devine imposibilă sinteza arhitecturii decizionale cu lungimea cuvântului pe 32 biți cauzată de numărul insuficient al elementelor logice, proprietate specifică a familiei respective. Pentru toate circuitele studiate poate fi observată scăderea frecvenței maxime a semnalului de sincronizare care poate fi cauzată de creșterea numărului componentelor logice din arhitectura sistemului, sau de modul de mapare a arhitecturii sistemului decizional pe resursele hardware ale circuitului FPGA [113]. Toate aceste premize pot duce la creșterea întârzierilor apărute ca rezultat al asigurării propagării uniforme a semnalului de sincronizare spre toate componentele arhitecturale.

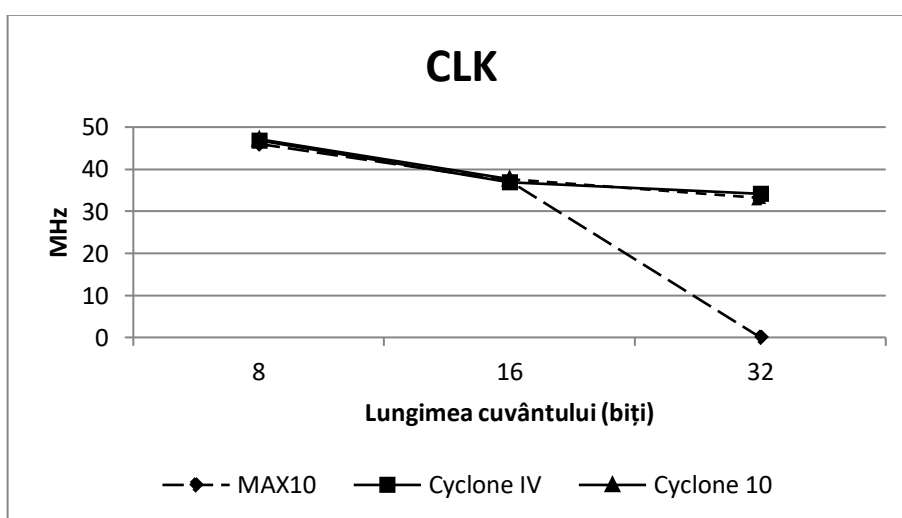


Figura 3.23 Graficul dependenței dintre lățimea cuvântului și frecvența semnalului de sincronizare pentru circuitele din familiile MAX și Cyclone.

Pe parcursului cercetărilor realizate a fost soluționată cu succes problema selectării circuitului FPGA care ar dispune de resursele hardware suficiente pentru implementarea arhitecturii decizionale cu parametrii arhitecturali specificați. Această problemă la rândul său a dat posibilitatea de a formula problema inversă: care pot fi combinațiile parametrilor arhitecturali (numărul maxim al neuronilor din arhitectura rețelei neuronale și lungimea cuvântului) ai sistemului decizional care poate fi implementat pe un circuit FPGA specific. Pentru a estima posibilitățile implementării arhitecturilor decizionale neuronale pe circuitele FPGA a fost realizată încercarea nereușită de identificare empirică a parametrilor arhitecturali pentru fiecare familie de circuite cu diferiți parametri de intrare. Analizând datele prezentate în *Figura 4.3* poate fi identificată cauza care a dus la abandonarea acestei metode de estimare a parametrilor arhitecturali ai sistemului decizional neuronal și anume faptul că creșterea lungimii cuvântului ca și creșterea numărului de neuroni din componența rețelei neuronale cauzează creșterea enormă a timpului de compilare și simulare a proiectului, iar simulări respectiv trebuie realizate destul de multe pentru a identifica concret parametrii arhitecturali. Un alt neajuns al acestei metode constă în dependența timpului de simulare de parametrii sistemului de calcul pe care rulează algoritmul de mapare. Astfel a fost observat faptul că în unele cazuri maparea cu succes pe circuitul specificat nu este posibilă din cauza cantității insuficiente de memorie a sistemului de calcul și nu de lipsa resurselor hardware a circuitului, sau maparea cu succes a unei arhitecturi specifice, care inițial nu a putut fi realizată, poate fi obținută prin mapări repetate crescând parametrii de intrare până la valorile respective. Aceste fenomene pot ușor compromite datele obținute cu atâta complexitate astfel ducând la concluzii greșite și pot cauza invalidarea rezultatelor obținute.

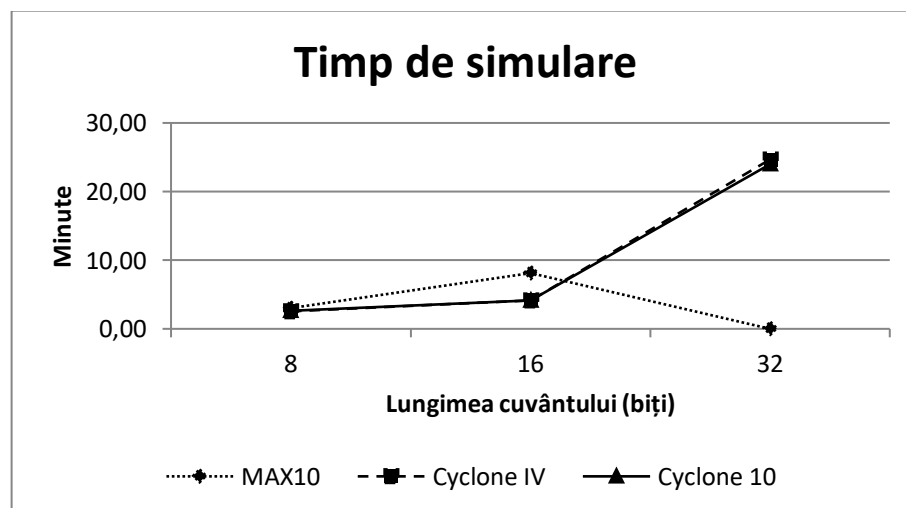


Figura 3.24 Graficul dependenței dintre lățimea cuvântului și timpul de analiză și sinteză a arhitecturii neuronale cu 10 neuroni pe circuitele din familiile MAX 10, Cyclone IV și Cyclone 10.

Aceleași simulări însă au putut servi indirect pentru rezolvarea acestei probleme. În urma simulărilor realizate au fost obținute date despre numărul de regiștri utilizați la sinteza arhitecturilor specifice, parametru ulterior utilizat pentru calculul resurselor necesare pentru implementarea rețelelor neuronale cu parametri arhitecturali specifici.

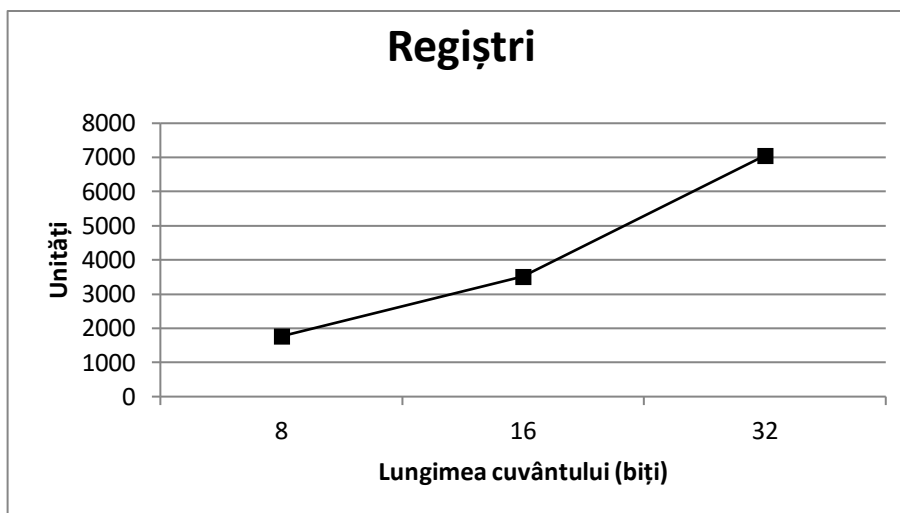


Figura 3.25 Graficul dependenței dintre lățimea cuvântului și numărul de regiștri utilizați la implementarea rețelei neuronale pe circuite din familia Cyclone.

În urma unor observații asupra parametrului respectiv a fost remarcat faptul că acesta nu se modifică de la familie la familie în condiții inițiale de simulare (numărul egal de neuroni și lungimea cuvântului) și coincide ca valoare în condiții arhitecturale similare, fapt ce a dus la ideea estimării acestor resurse în baza valorii parametrului cercetat, *Figura 3.25*.

Astfel memoria totală (numărul de regiștri) utilizată la implementarea sistemului poate fi exprimată cu ajutorul relației:

$$M_{totală} = M_{intrări} + M_{ponderi} + M_{ieșiri} + M_{ct} \quad (3.16)$$

unde $M_{intrări}$ reprezintă capacitatea în biți a memoriei necesare pentru stocarea intrărilor rețelei neuronale, $M_{ponderi}$ – capacitatea în biți a memoriei necesare pentru stocarea ponderilor sinaptice ale rețelei neuronale, $M_{ieșiri}$ – capacitatea în biți a memoriei necesare pentru stocarea ieșirilor rețelei neuronale, iar M_{ct} reprezintă memoria necesară pentru sinteza contorului utilizat pentru stabilizarea rezultatului procesului de inferență a rețelei neuronale artificiale și poate depinde de arhitectura internă a acesteia, care în cazul de față reprezintă valoarea $3N$, unde N reprezintă numărul de neuroni din arhitectura rețelei neuronale, iar parametrul b reprezintă lungimea cuvântului în biți.

$$M_{intrări} = N \times b \quad (3.17)$$

$$M_{ponderi} = 2 \times N^2 \times b \quad (3.18)$$

$$M_{ieșiri} = N \times b \quad (3.19)$$

$$M_{ct} = [\log_2(3 \times N)] \uparrow \quad (3.20)$$

Înlocuind în relația (3.16) valorile pentru memorie din relațiile: (3.17), (3.18), (3.19) și (3.20), a fost obținută relația (3.21):

$$M_{totală} = b \times (2 \times N^2 + 2 \times N) + [\log_2(3 \times N)] \uparrow \quad (3.21)$$

care utilizată pentru sistemul decizional cu 10 neuroni pe 8, 16 și 32 biți confirmă valorile obținute experimental:

$$M_{totală} = 8 \times (2 \times 10^2 + 2 \times 10) + 5 = 1765,$$

$$M_{totală} = 16 \times (2 \times 10^2 + 2 \times 10) + 5 = 3525,$$

$$M_{totală} = 32 \times (2 \times 10^2 + 2 \times 10) + 5 = 7045,$$

iar pentru estimarea posibilității implementării unui sistem decizional cu 10 neuroni pe 64 biți în baza circuitului 10L120YF780I7G, pentru care procesul de analiză, sinteză și mapare ar putea dura ore în șir, *Figura 3.26*, a fost calculat numărul total de regiștri și apoi justificată experimental valoarea acestui parametru:

$$M_{totală} = 64 \times (2 \times 10^2 + 2 \times 10) + 5 = 14085.$$

Flow Summary	
<input type="text" value="Filter"/>	
Flow Status	In progress - Thu Jan 30 16:21:54 2020
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	ANN
Top-level Entity Name	Interface
Family	Cyclone 10 LP
Device	10CL120YF780I7G
Timing Models	Final
Total logic elements	537,539
Total registers	14085
Total pins	75
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	576
Total PLLs	0

Figura 3.26 Rezultatul compilării și sintezei parțiale a arhitecturii neuronale cu 10 neuroni pe 64 biți.

Metoda de calcul a potențialului de implementare a arhitecturii neuronale pe circuite individuale, practic a dat posibilitatea de a elimina din procesul de cercetare etapele de compilare și sinteză repetată a sistemelor decizionale cu parametri arhitecturali specifici, fapt ce a dus la optimizarea și eficientizarea procesului de cercetare, dezvoltare și implementare a acestor sisteme în baza rețelelor neuronale.

3.5 Concluzii la capitolul 3

În urma analizei efectuate a fost posibilă identificarea caracteristicilor individuale ale metodelor de proiectare a sistemelor de luare a deciziilor propuse. A fost determinat dezavantajul esențial al implementării sistemelor de suport decizional în baza arhitecturii generice fuzzy. Această metodă de dezvoltare necesită capacități relativ mari de memorie, în schimb, manifestă un consum redus de elementele logice, fapt ce duce la o utilizare irațională și neproportională a resurselor hardware ale circuitelor reconfigurabile [103]. Soluția acestei probleme o poate reprezenta implementarea funcțiilor de apartenență clasice în baza blocurilor aritmetice, fapt ce va duce la diminuarea semnificativă a aplicabilității și a universalității sistemului proiectat, în schimbul utilizării eficiente a resurselor hardware, minimizării întârzierilor, specializării circuitului în rezolvarea unei clase mai restrânse de probleme și creșterii frecvenței de operare. Această abordare permite echilibrarea utilizării resurselor de memorie și a celor logice.

În cazul implementării sistemelor de suport decizional în baza arhitecturii neuronale generice a fost identificată problema utilizării preponderent a elemente logice, fapt ce cauzează epuizarea rapidă a acestor resurse și reduce semnificativ din potențialul aplicativ al circuitului. Soluționarea acestui neajuns constă în utilizarea memoriei disponibile pentru stocarea intrărilor, ponderilor sinaptice și a ieșirilor prin utilizarea megafuncțiilor *lpm_ram*. Aplicarea acestei metode de dezvoltare va duce la utilizarea unei capacități duble de memorie, proprietate arhitecturală specifică acestei megafuncții. Neajunsul dat poate fi evitat în cazul implementării unui sistem de suport decizional cu configurație dublă utilizată pentru sporirea siguranței sistemului, cu două configurații distincte care pot implementa funcționalități diferite sau în cazul implementării algoritmilor de antrenare a RNA care presupun păstrarea ponderilor sinaptice curente, a ponderilor noi calculate, a ieșirilor reale și a celor destinate pentru antrenare.

În rezultatul simulărilor efectuate pe diferite familii de circuite au fost identificate trasăturile specifice ale arhitecturilor propuse și stabilite relațiile dintre utilizarea diferitor tipuri de resurse hardware. Aceste realizări au oferit posibilitatea implementării arhitecturilor propuse pe circuitele respective reieșind din capacitățile resurselor interne ale acestora, astfel micșorând semnificativ numărul total de simulări și timpul necesar pentru dezvoltarea sistemului.

4. APLICAREA TEHNICILOR DE SUPORT DECIZIONAL

4.1 Studiu de caz: sistem fuzzy pentru procesul de tragere a microfiredor

La etapa de definire a funcțiilor de apartenență, pentru fiecare calificativ, se definesc valorile probabilităților valorilor variabilelor de intrare. Pentru a simplifica sistemul la etapa de proiectare se utilizează exprimarea valorii probabilităților cu ajutorul numerelor întregi în intervalul de la 0 la 4095. Acest interval este ales reșind din specificul problemei care necesită o precizie destul de mare. Intervalul se extinde până la valoarea 10^n determinată de numărul de biți cu ajutorul cărora sunt reprezentate valorile variabilelor de intrare și ieșire. Funcțiile de apartenență sunt realizate prin intermediul blocurilor de memorie ROM implementate cu ajutorul megafuncțiilor din *Quartus 7.2*. După această etapă poate fi construit graficul funcțiilor de apartenență, *Figura 4.1*.

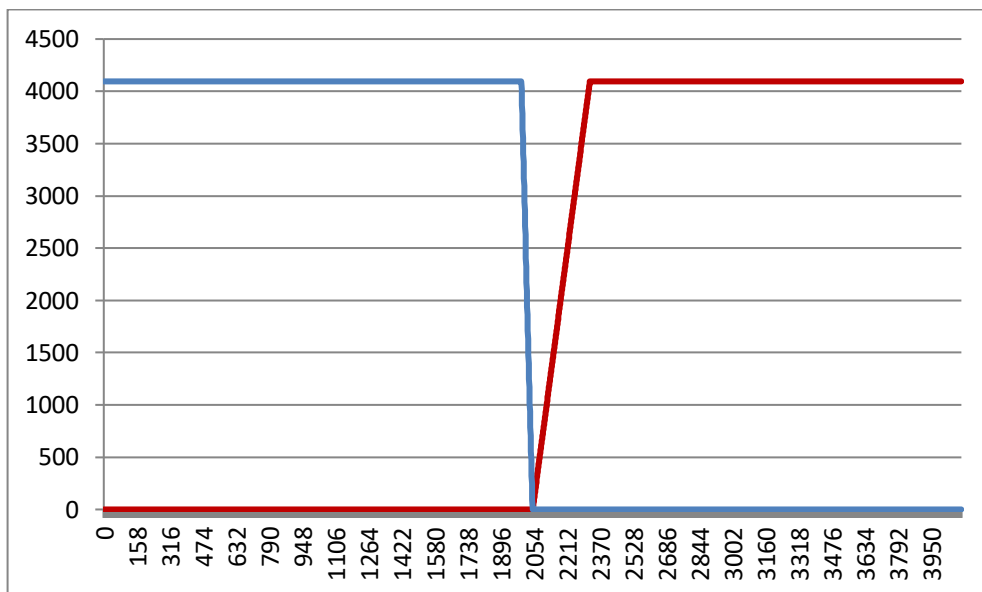


Figura 4.1 Funcțiile de apartenență definite tabelar pentru fiecare calificativ al parametrului de intrare dRp .

Pentru a putea defini tabelar funcțiile de apartenență și pentru a le înscrie în blocurile de memorie ROM destinate acestora, este necesar de a realiza procedura de ajustare, care constă în translarea intervalului în partea dreaptă sau în cea stângă pe axa x , în dependență de mărimea intervalului pe care este definită funcția. Astfel valorile variabilelor de intrare vor reprezenta direct adresa celulei de memorie în care se află valoarea funcției de apartenență pentru calificativul dat, *Figura 4.1*.

Notă: Celulele de memorie conțin valorile probabilităților pentru funcțiile de apartenență deja deplasate pe axa x și nu valorile reale ale parametrilor de intrare.

La etapa de sinteză a motorului de inferență generic, reconfigurabil, care ar putea implementa algoritmi decizionali elaborați anterior, s-a decis de a nu utiliza blocurile *FUZZYCELL* deoarece acestea îngreunează citirea și înțelegerea schemei de funcționare a motorului de inferență din cauza omogenității acestora. Din aceste motive au fost create blocurile reconfigurabile în baza *Matricelor de Porți Logice Fuzzy Programabile* după analogia cu arhitecturile clasice (*PLA*) utilizând porți *FUZZYAND*, *FUZZYOR* și un set de multiplexoare pentru crearea componentei *FUZZYRULE* care reprezintă implementarea hardware a unei singure reguli de inferență din componența motorului de inferență generic.

Astfel blocul reconfigurabil *FUZZYRULE* utilizat pentru implementarea algoritmului de comandă și control al procesului de turnare a microfiredorilor cu un parametru de intrare caracterizat de două calificative pentru acesta va avea două intrări pe 12 biți care vor reprezenta intrările calificativelor variabilei dRp : $-dRp$ și $+dRp$. Acestea se aplică fiecare la câte o poartă *FUZZYAND*. Blocul *FUZZYRULE* mai dispune de o intrare de configurare sau de programare pe 3 biți $PR[2..0]$ cu ajutorul căreia se selectează sursa operanzilor aplicați la poarta *FUZZYOR*.

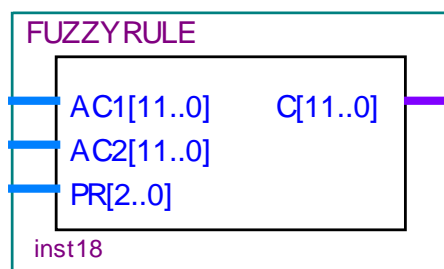


Figura 4.2 Blocul configurabil de reprezentare a regulilor de inferență *FUZZYRULE*.

După cum se poate observa din schema internă a blocului *FUZZYRULE*, *Figura 4.3*, a doua intrare a porților logice *FUZZYAND* este conectată la VCC fapt ce semnifică că ieșirea acestor porți depinde de o singură intrare și anume de calificativul $-dRp$ sau $+dRp$. În cazul altor probleme cu un număr mai mare de variabile de intrare și cu un număr mai mare de calificative această componentă va avea numărul de intrări egal cu numărul total al calificativelor tuturor variabilelor de intrare. Evident acest fapt va duce la creșterea complexității regulilor de inferență prin mărirea numărului de operanzi, dar în pofida acestui fapt se păstrează posibilitatea proiectării unui motor de inferență generic reconfigurabil capabil să implementeze orice regulă de inferență construită din combinațiile conjunctive și disjunctive ale calificativelor variabilelor de intrare.

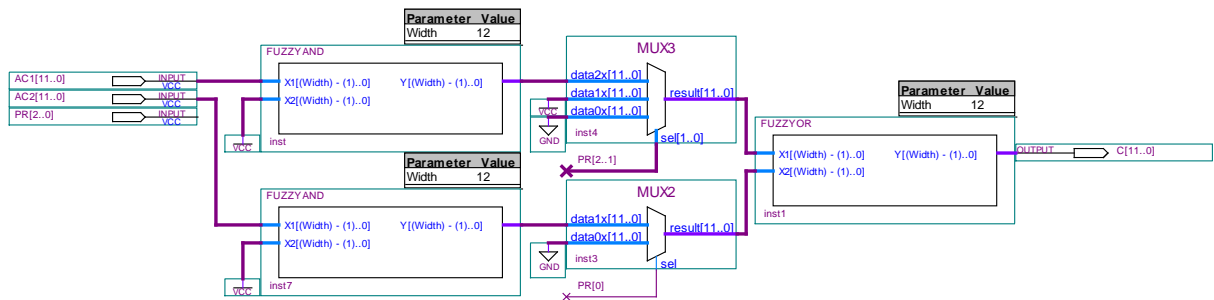


Figura 4.3 Schema internă a blocului configurabil *FUZZYRULE*.

Utilizarea multiplexoarelor în structura internă a componentei *FUZZURULE* are scopul de a oferi utilizatorului la etapa de testare a sistemului, sau proiectantului la etapa de proiectare, posibilitatea de a aplica la porțile *FUZZYOR* acele ieșiri ale conjuncțiilor care sunt prevăzute de regula de inferență ce stă la baza algoritmului decizional realizat de această arhitectură, *Figura 4.4*. Aceste multiplexoare oferă posibilitatea de a asigura la intrarea porții *FUZZYOR* a valorii fuzzy logice '1' prezentată pe 12 biți care reprezintă valoarea întregă fără semn $2^{12}-1$ notată *MAX*, sau a valorii logice '0' care reprezintă valoarea 0 prezentată pe 12 biți – valoare întregă fără semn notată *MIN*. Valoarea *MIN* aplicată la una din intrări face ca ieșirea porții *FUZZYOR* să depindă doar de cealaltă intrare, iar aplicată la ambele multiplexoare forțează ca ieșirea porții *FUZZYOR* să ia valoarea *MIN* sau '0'. Valoarea *MAX* aplicată cel puțin la unul din multiplexoarele interne (în cazul cercetat la primul multiplexor) forțează ieșirea porții logice *FUZZYOR* să ia valoarea *MAX* sau '1'. Astfel aceste multiplexoare pot configura regula de inferență inclusiv să asigure valorile constante '1' sau '0'.

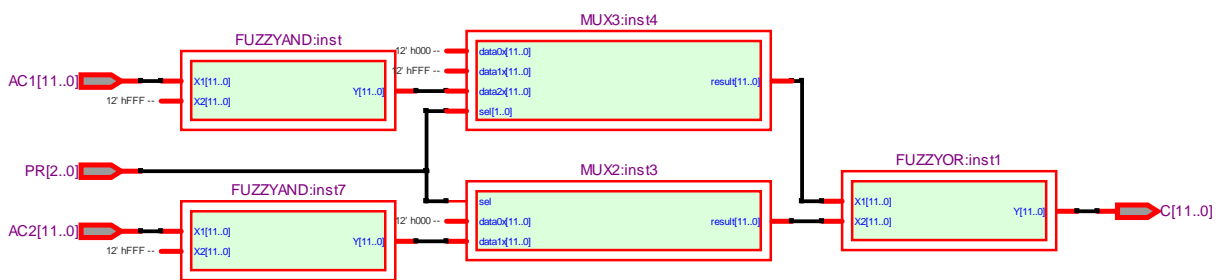


Figura 4.4 Schema internă obținută în rezultatul sintezei componentei configurabile **FUZZYRULE**.

În rezultat se poate menționa faptul că structura sistemului integrat devine inevitabil mai complexă deoarece înglobează o funcționalitate extinsă însă procesul de proiectare se simplifică esențial prin utilizarea componentelor parametrizate și reconfigurabile. Utilizarea practică a

acestui tip de motoare de inferență poate fi apreciată în procesul de implementare a sistemelor de preluare automată a cunoștințelor de la operatorul uman, în cazul în care regulile de inferență necesită modificări frecvente pe parcursul etapelor de proiectare, implementare și ajustare a algoritmului decizional. În aceste condiții algoritmul decizional poate fi ușor modificat fără re-proiectarea și modificarea structurii interne a sistemului fuzzy, făcându-l astfel destul de flexibil, universal și relativ independent de algoritmul decizional în limitele aceleiași clase de probleme.

La realizarea procesului de defuzificare au fost utilizate pentru calificativele variabilei de ieșire, blocuri de memorie ROM, implementate cu ajutorul megafuncției parametrice *lpm_rom*. În blocurile de memorie ROM au fost stocate tabelar valorile funcțiilor de defuzificare astfel încât fiind aplicate la intrările de adresă ale acestora să genereze la ieșire valorile deja defuzificate ale calificativelor variabilei de ieșire, *Figura 4.5*.

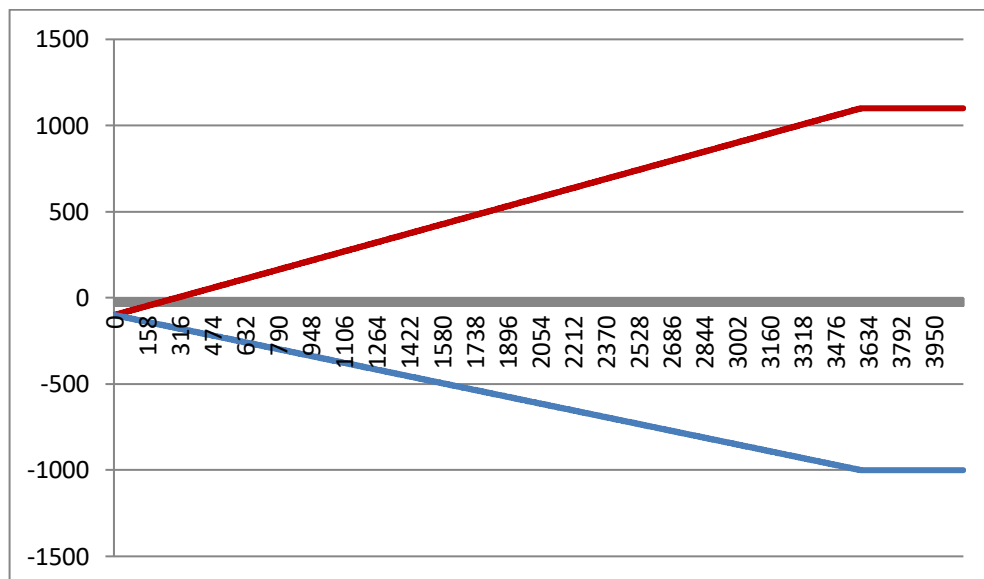


Figura 4.5 Funcțiile de defuzificare definite tabelar pentru fiecare calificativ al parametrului de ieșire *dPv*.

Pentru sinteza defuzicatorului care ar putea implementa metoda de defuzificare selectată anterior a fost decis ca la etapa de testare a arhitecturilor hardware reconfigurabile să fie utilizat un comparator. Pentru aceasta a fost utilizată megafuncția parametrică *lpm_compare*, *Figura 4.6*, care implementează funcția *Maxim* de defuzificare pe 12 biți, fără semn. Astfel modul de funcționare al defuzicatorului se reduce la compararea fără semn a celor două valori de ieșire ale funcțiilor de apartenență definite tabelar ale calificativelor variabilei de ieșire. Dintre aceste două ieșiri se selectează, în dependență de ieșirea comparatorului, valoarea respectivă.

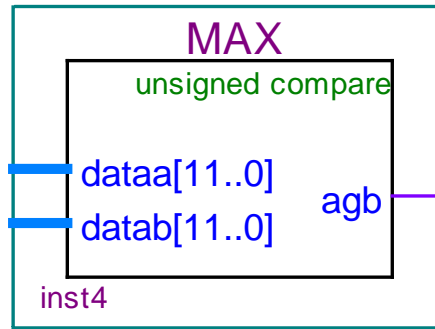


Figura 4.6 Defuzzificator prin metoda maximului în baza comparatorului.

Pentru testarea sistemului inteligent de automatizare a procesului de turnare a microfiredelor în baza experienței captate de la operatorul uman a fost implementată arhitectura motorului de inferență în baza blocurilor *FUZZYRULE* care implementează regulile de inferență din cadrul algoritmului decizional, *Figura 4.3*.

Pentru testarea algoritmului decizional a fost implementată schema caracteristică de reglare diferențială în care ca intrare servește variația rezistenței liniare relativ la valoarea caracteristică a acesteia. Această schemă oferă posibilitatea evaluării corectitudinii funcționării modelului decizional elaborat, *Figura 4.7*.

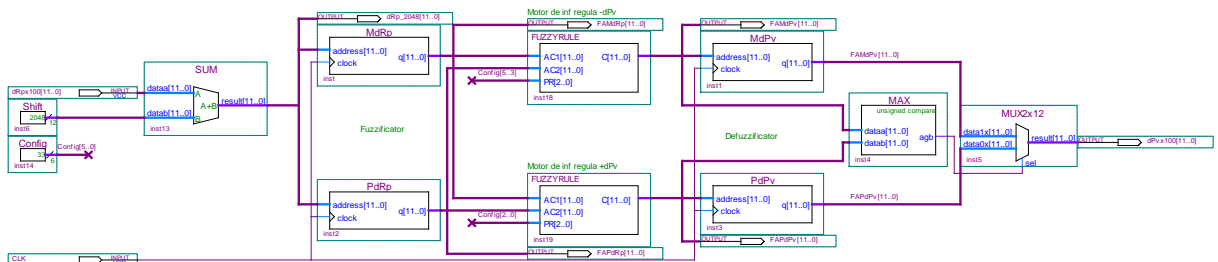


Figura 4.7 Schema caracteristică de reglare diferențială.

Ca dovadă a bunei funcționări a algoritmului elaborat în p.2.3.2 poate servi diagrama de timp prezentată în *Figura 4.8*. Este necesar de menționat faptul că asupra valorilor variabilelor de intrare și ieșire sunt aplicați coeficienți de amplificare – 100, care transformă domeniul de valori admisibile din domeniul numerelor reale în domeniul numerelor întregi [114, 115]. Această procedură simplifică implementarea în hardware a arhitecturii algoritmului decizional.

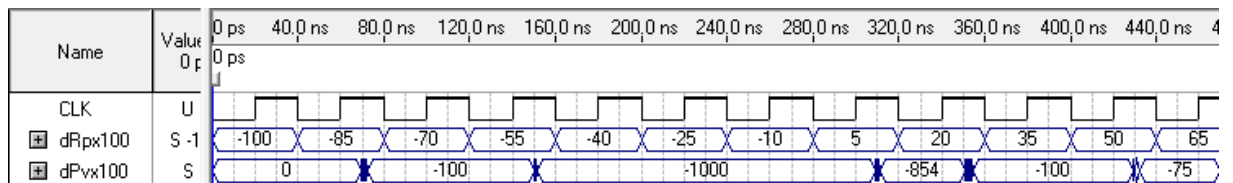


Figura 4.8. Diagrama de timp a algoritmului de reglare diferențială.

Pentru aplicarea sistemului decizional în automatizarea procesului de producție a microfiredelor a fost elaborat sistemul integral de comandă și control automat, la care se poate aplica valoarea directă a variabilei de intrare și care generează la ieșire valoarea integrală a variabilei de ieșire. Aplicarea variabilei de ieșire la sistemul de producție va iniția controlul procesului de turnare a microfiredelor, *Figura 4.9*.

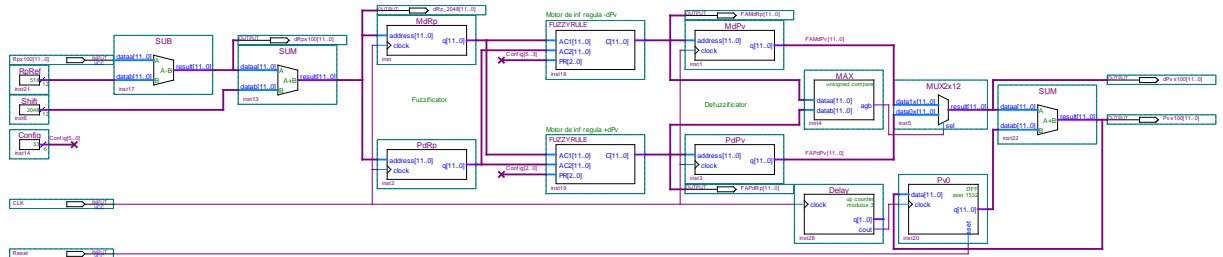


Figura 4.9. Schema integrală a sistemului de control automat a procesului de turnare a microfiredelor.

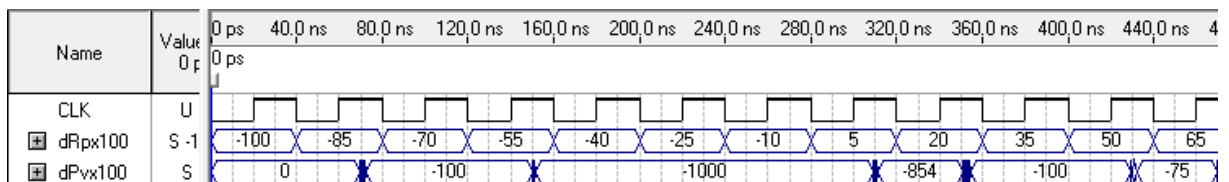


Figura 4.10. Diagrama de timp a algoritmului decizional integral de control automat a procesului de turnare a microfiredelor.

Utilizarea blocurilor și a porților logice fuzzy parametrizate a făcut posibilă implementarea în hardware a sistemului decizional fuzzy bazat pe experiența operatorului uman pentru comanda și controlul procesului de turnare a microfiredelor.

4.2 Studiu de caz 2: sistem pe bază de rețea neuronală

În cazul implementării sistemului decizional în baza rețelelor neuronale artificiale este abordată problema mapei algoritmului de comandă și control pe structuri hardware. În acest context este propusă o variantă de mapare a modelelor elaborate în pachetul de programe *Matlab* pe dispozitive reconfigurabile. Cu acest scop a fost implementată hardware o rețea neuronală artificială generică și aplicată o metodă clasică de mapare a modelelor dezvoltate pe circuite reconfigurabile [111, 116]. Abordarea propusă permite minimizarea duratei etapelor de proiectare și implementare a sistemelor decizionale în baza rețelelor neuronale artificiale. Astfel accentul a fost pus pe etapele cheie de proiectare și dezvoltare a sistemelor cu inteligență artificială la baza cărora se află o rețea neuronală artificială reprezentată printr-un digraf

complet. O atenție deosebită a fost acordată eficientizării etapelor de proiectare și implementare a rețelelor neuronale artificiale.

Pentru testarea sistemului neuronal de comandă și control automat a procesului de turnare a microfiredelor au fost implementate arhitecturile componentelor neuronale: multiplicator, sumator și funcție de activare. Arhitectura care implementează funcția amplificării sinaptice constă dintr-o mulțime de multiplicatoare care efectuează înmulțirea intrărilor neuronului cu ponderile acestora, *Figura 4.11*.

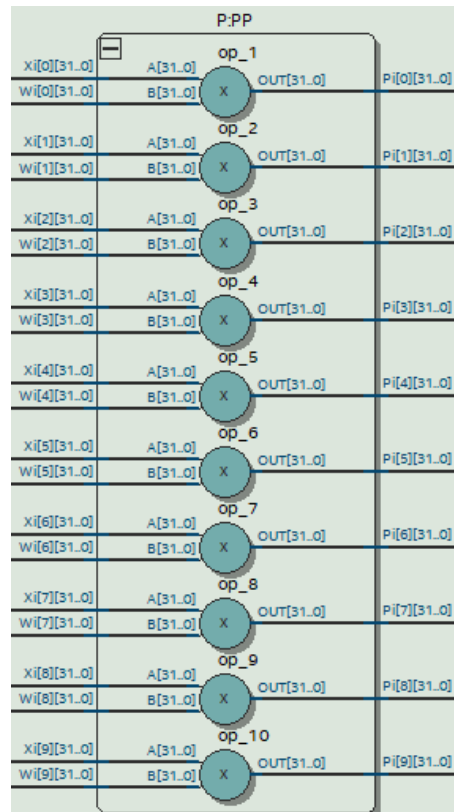


Figura 4.11 Arhitectura amplificatorului sinaptic a neuronului artificial.

Componenta parametrizată a amplificatorului sinaptic – *P* din structura neuronului este elaborată în *AHDL, Anexa 2.1* și asigură generarea produselor dintre intrările neuronului și intrările ponderilor sinaptice. În dependență de valoarea parametrului *Inputs* poate fi modificat numărul de intrări a neuronului iar cu ajutorul parametrului *Width* poate fi variată lățimea magistralelor de intrare și ieșire a componentei, fapt ce asigură flexibilitatea necesară pentru utilizarea cu succes a acestei componente la proiectarea arhitecturilor neuronale.

Pentru testarea funcționării sumatorului, funcția căruia se reduce la adunarea tuturor produselor obținute în urma amplificării sinaptice, a fost implementată arhitectura de sumare cumulativă cu 10 intrări pe 32 biți, *Figura 4.12*, obținută în baza componentei parametrizate de sumare realizate în *AHDL, Anexa 2.2*.

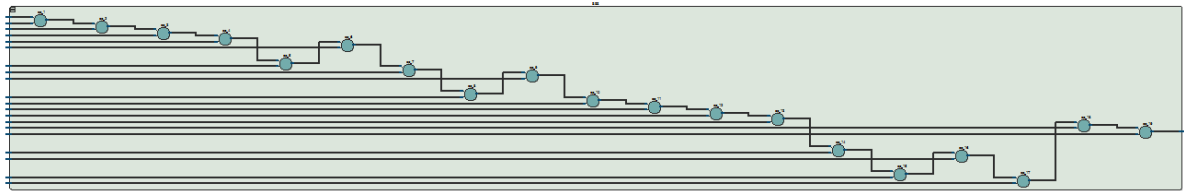


Figura 4.12 Arhitectura modului de sumare cumulativă cu 10 intrări pe 32 biți fiecare.

Sumarea valorilor de intrare amplificate se face cu ajutorul sumatoarelor binare din componența blocului parametrizat – *S*, iar asupra rezultatului acestei sume este aplicată funcția de activare realizată de componenta hardware parametrizată – *T*.

Pentru implementarea modelului neuronului utilizat la dezvoltarea arhitecturii sistemului inteligent de comandă și control automat a procesului de turnare a microfîrelor a fost implementată una din cele mai simple funcții de transfer și anume funcția de transfer liniară, *Figura 4.13*. Această funcție de transfer a fost implementată cu ajutorul componente parametrizate – *T* descrise în *AHDL, Anexa 2.3*, care conține un singur parametru – *Width* de care depinde lățimea magistrelor de date de intrare și ieșire.

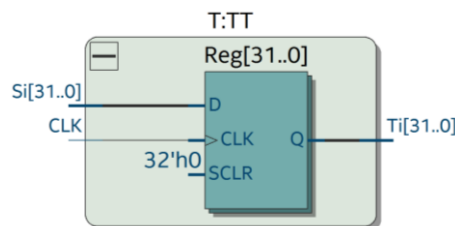


Figura 4.13 Arhitectura funcției de activare liniare.

În componența blocului funcției de activare intră un registru pe 32 biți destinat pentru păstrarea ieșirii neuronului. Structura ierarhică a modelului neuronului realizată cu ajutorul componentelor discrete *P*, *S* și *T* permite dezvoltatorilor să utilizeze diferite funcții de activare în dependență de condițiile problemei formulate, [117]. În cazul proiectului de cercetare a procesului industrial de turnare a microfîrelor a fost selectată funcția de activare liniară.

Pentru implementarea rețelelor neuronale artificiale generice este necesar de a proiecta arhitectura hardware parametrizată a unui singur neuron artificial în baza componentelor parametrizate proiectate anterior: *P*, *S* și *T*.

Modelul neuronului utilizat la proiectarea sistemului decizional elaborat are numărul de intrări egal cu suma dintre numărul de intrări externe și numărul de neuroni a rețelei neuronale artificiale. Pentru simplificarea modelului rețelei neuronale artificiale generice a fost pusă condiția ca numărul de intrări externe să fie egal cu numărul de neuroni a acesteia. În rezultat a

fost stabilit numărul total de intrări a unui neuron egal cu $2n$ unde n este numărul de neuroni din arhitectura rețelei, *Figura 4.14*. Astfel fiecare neuron dispune de n intrări externe și n intrări interne la care sunt aplicate ieșirile neuronilor din rețea.

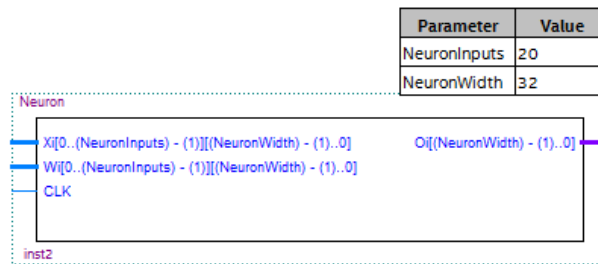


Figura 4.14. Componenta parametrizată a unui neuron artificial.

Parametrul *NeuronWidth* al modelului neuronului parametrizat, *Figura 4.14*, oferă posibilitatea ca intrările și ieșirile rețelei neuronale proiectate cu ajutorul acestei componente să poată lua valori cât binare atît și întregi cu semn. Acest lucru poate fi modificat la etapa de proiectare a rețelei neuronale artificiale prin modificarea parametrului *Width* a componenteii ierarhic superioare *Network*. Valoarea parametrului *NeuronWidth* este atribuită parametrului *Width* a blocurilor ierarhice inferioare – *P*, *S* și *T*, *Figura 4.15*. Astfel toate intrările și ieșirile fiecărui neuron din întreaga rețea neuronală vor avea aceeași reprezentare. Modelul neuronului artificial pe lângă intrările interne și externe mai conține și intrările ponderilor sinaptice – *W* reprezentate de asemenea cu ajutorul numerelor întregi cu semn. Modul de funcționare a fiecărui neuron este sincronizat pe frontul crescător al semnalului de ceas.

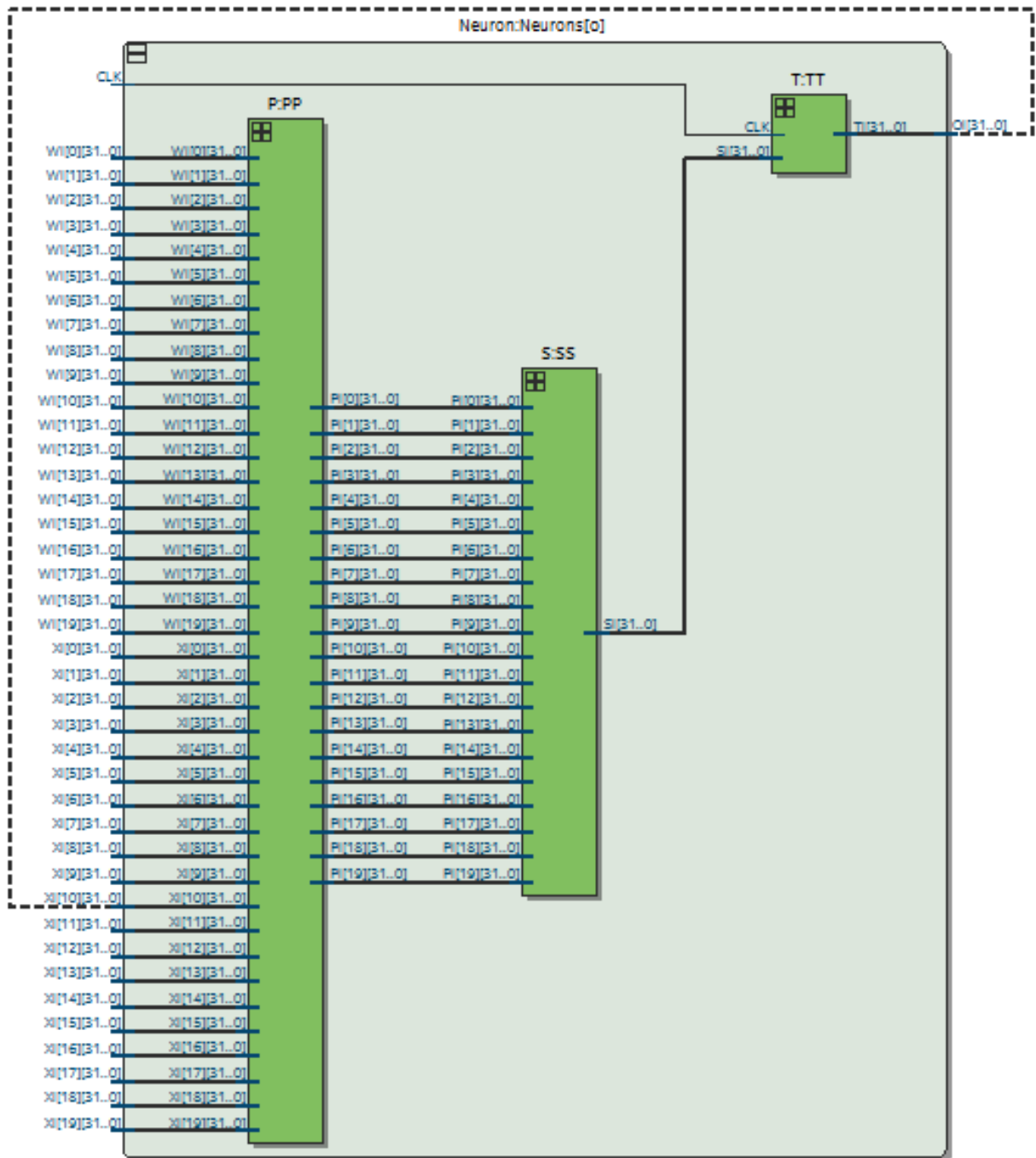


Figura 4.15. Arhitectura unui neuron artificial cu 10 intrări externe și 10 conexiuni interne pe 32 biți.

Utilizarea acestui component la etapa de proiectare a rețelei neuronale artificiale oferă posibilitatea de a ajusta parametri de bază a neuronului în dependență de necesitățile sistemului proiectat. Utilizarea unui număr mare de sumatoare binare în blocul de sumare paralelă a neuronului artificial duce la optimizarea funcționării arhitecturii respective cauzată de minimizarea latenței acestei structuri, iar buferizarea ieșirii neuronului face posibilă funcționarea acestei arhitecturi în regim pipeline.

Utilizând arhitecturile dezvoltate anterior, a fost implementat modulul unui neuron parametrizat, *Anexa 2.4*. Acesta la rândul său a fost utilizat la dezvoltarea modelului parametrizat al rețelei neuronale artificiale generice, *Anexa 2.5* [118]. Parametrul *NetNeurons* al arhitecturii rețelei neuronale generice reprezintă numărul de neuroni din componența acesteia, iar al doilea parametru – *NetWidth* reprezintă lățimea magistralelor de intrare și ieșire, *Figura 4.16*. Pentru testarea modelelor decizionale propuse a fost implementată o rețea neuronală generică destinată pentru automatizarea procesului de turnare a microfiredelor. Cu acest scop a fost proiectată și antrenată o rețea neuronală artificială cu arhitectură de tip *Hopfield* în baza componentei parametrizate *Network* cu 10 neuroni pe 32 biți. Se poate observa faptul că numărul ponderilor poate fi calculat după formula (4.1) unde N_w reprezintă numărul total de ponderi, N_{wii} – numărul de ponderi ale intrărilor interne și N_{Ie} – numărul de ponderi ale intrărilor externe.

$$N_w = N_{ii}^2 + N_{Ie}^2 \quad (4.1)$$

Din specificul arhitecturii rețelei neuronale de tip *Hopfield* rezultă egalitatea:

$$N_{ii} = N_{Ie} \quad (4.2)$$

Această valoare se datorează faptului că fiecare neuron este conectat cu toți ceilalți neuroni, dar pe lângă conexiunile interne mai sunt prezente și intrările externe iar numărul acestora este egal cu numărul intrărilor interne.

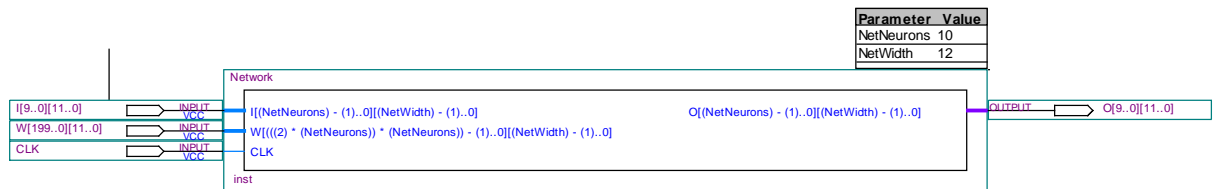


Figura 4.16. Arhitectura parametrizată a rețelei neuronale artificiale de tip *Hopfield* pentru automatizarea procesului de turnare a microfiredelor.

În urma sintezei arhitecturii rețelei neuronale artificiale generice cu 10 neuroni pe 32 biți a fost obținută structura de calcul prezentată în *Figura 4.17*.

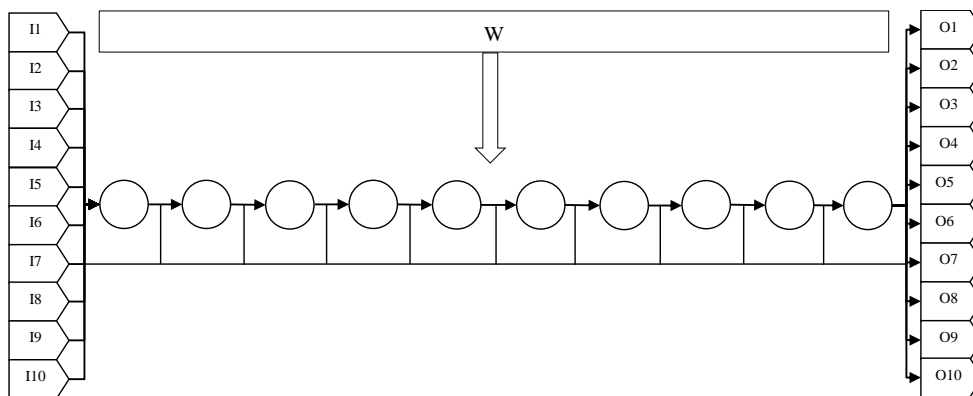


Figura 4.17 Schema de structură a rețelei neuronale artificiale generice cu *10* intrări externe, *10* neuroni și *10* ieșiri

Din cauza complexității arhitecturii obținute în urma procesului de sinteză și a numărului foarte mare de conexiuni între componentele sistemului nu poate fi prezentată arhitectura respectivă, în schimb pentru o percepere vizuală de principiu a fost prezentată schema de structură a arhitecturii obținute, *Figura 4.17* care reprezintă un digraf complet cu *10* intrări externe și *10* ieșiri la care se aplică mulțimea ponderilor sinaptice – W .

În rezultatul proiectării rețelei neuronale generice a fost obținută arhitectura prezentată în *Figura 4.17*, iar în rezultatul compilării proiectului elaborat a fost obținută arhitectura unei rețele neuronale la baza căreia se află modelul neuronului artificial prezentat în *Figura 4.15*. Poate fi remarcat faptul că modelul obținut pentru neuronul artificial conține *10* intrări externe, *10* intrări interne și *200* de intrări destinate aplicării ponderilor sinaptice. Pentru păstrarea rezultatului simulării neuronale, în componența fiecărui neuron este inclus un registru pe *32* biți, care asigură păstrarea valorii ieșirii fiecărui neuron pentru ca aceasta să poată fi aplicată la intrările celorlalți neuroni la următorul ciclu de calcul fără a cauza pierderi de date.

În rezultatul simulării procesului de inferență al rețelei neuronale generice a fost obținută diagrama de timp din *Figura 4.18*. Din diagramă se poate observa faptul că modelul rețelei neuronale artificiale funcționează în regim de pipeline. Arhitectura de tip *Hopfield*, împreună cu regimul pipeline de funcționare a rețelei neuronale, impune o serie de restricții și anume faptul că rezultatul simulării pentru intrările curente va fi disponibil cel mult după $2n$ tacte de ceas, unde n reprezintă numărul neuronilor din structura rețelei neuronale artificiale simulate. Deoarece rețeaua neuronală artificială are posibilitatea de a fi reconfigurată chiar în timpul regimului de funcționare, latența pipeline-ului de asemenea poate varia de la o configurație la alta [111, 116]. Pentru a generaliza modul de funcționare a arhitecturii date au fost stabilite anumite restricții față de valoarea latenței arhitecturii neuronale care constituie N tacte, unde N reprezintă numărul de

neuroni, iar pentru a asigura recurența (feed-back-ul) acestei arhitecturi, latența maximă trebuie dublată [119].

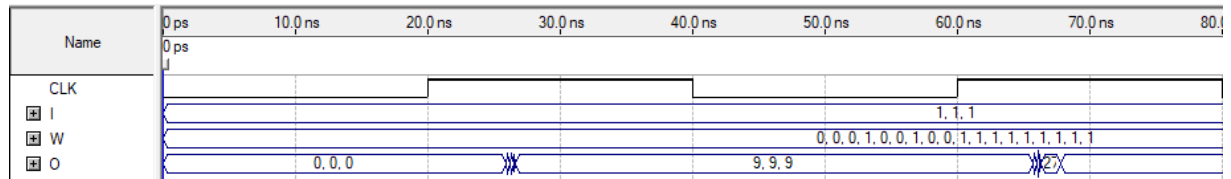


Figura 4.18. Diagrama de timp a procesului de inferență neuronal.

În rezultat se cere specificat faptul că pentru a nu altera propagarea datelor în interiorul arhitecturii neuronale, valoarea intrărilor trebuie păstrată pe tot parcursul procesului de inferență, adică pe parcursul a $2N$ tacte. Această restricție arhitecturală nu indică neapărat faptul că răspunsul corect va fi disponibil anume după expirarea acestui interval maxim ci asigură faptul că după expirarea acestui interval de timp sistemul va genera date de ieșire valide indiferent de arhitectura rețelei neuronale artificiale simulate pe arhitectura generică. În cazul unei arhitecturi de tip „single layer perceptron” rezultatul procesului de inferență poate fi obținut chiar și după un singur tact. Acest fapt permite utilizatorului sistemului de a organiza procesul de inferență și schimbul de date cu arhitectura de calcul în așa mod încât aceasta să ofere o performanță maximă de calcul prin minimizarea întârzierilor în pipeline prin eficientizarea procesului de comunicare cu interfața de intrare/ieșire.

Arhitectura RNA generice în baza modelului *Hopfield* a fost aplicată la proiectarea sistemului de automatizare a procesului de turnare a microfiredelor. Cu acest scop au fost setați parametrii componente hardware conform cerințelor și anume pe cale experimentală a fost determinat numărul necesar de neuroni – 10. Reprezentarea intrărilor, ponderilor sinaptice și a ieșirilor a fost făcută pe 32 biți pentru a dispune de o rezervă în procesul de cercetare, cu atât mai mult că și variabilele din mediul *Matlab* pot fi convertite pe 32 biți. Deoarece sistemul de automatizare a procesului de turnare a microfiredelor conține o singură variabilă de ieșire – variația vidului în tubul de sticlă ΔP , celelalte ieșiri vor fi ignorate. Pentru sistemul de comandă și control sunt necesare două intrări: una pentru variabila de intrare R - rezistența liniară a microfiredului, iar cealaltă intrare va fi conectată la valoarea constantă I și va fi utilizată ca intrare *bias*. Celelalte intrări pot fi ignorate sau pot fi conectate forțat la GND , adică la aceste intrări va fi aplicată valoarea 0 [95, 96].

Pentru implementarea sistemului de automatizare a procesului de turnare a microfiredelor integral, *Figura 4.19*, a fost adăugată unitatea de comandă și control a interfeței de comunicare descrise în *AHDL, Anexa 2.6*, care oferă posibilitatea de configurare inițială a arhitecturii rețelei

neuronale generice prin înscrierea matricei ponderilor sinaptice și păstrarea acestora în blocul de memorie *WeightsRAM*. După etapa de configurare inițială a arhitecturii rețelei neuronale generice care constă în stocarea a 200 ponderi a câte 32 biți fiecare, ceea ce în sumă reprezintă 800 baiți de date de configurare, arhitectura neuronală generică poate fi utilizată în regimul de cerere – răspuns. În acest regim arhitectura recepționează secvențial câte o valoare de intrare și stochează în blocul de memorie *InputsRAM* vectorul de intrare, apoi după expirarea unui număr anumit de tacte, unitatea de comandă și control generează semnalul *Ready* care semnifică faptul că procesul de inferență a luat sfârșit, iar vectorul de ieșire este deja format în zona de memorie special destinată *OutputsRAM* și este gata de a fi transmis.

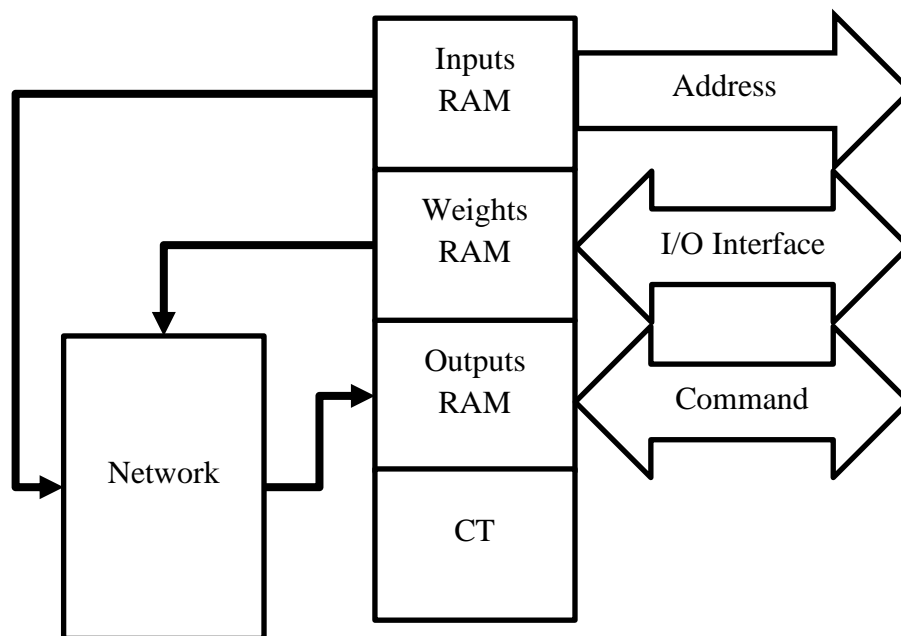


Figura 4.19 Schema de structură a interfeței rețelei neuronale artificiale generice.

Atât zona de memorie *InputsRAM* cât și *OutputsRAM* reprezintă spații de memorie a câte 10 cuvinte fiecare organizate pe 32 biți fapt ce impune rezervarea a 80 baiți de memorie din resursele disponibile circuitului utilizat la sinteza arhitecturii respective pentru stocarea intrărilor și ieșirilor sistemului decizional.

La contorizarea numărului de tacte necesare pentru stabilizarea ieșirilor arhitecturii rețelei neuronale generice în sistem a fost inclus un numărător modulo 20 care asigură stabilitatea procesului de inferență a arhitecturii de tip *Hopfield*. Rezultatul corect al procesului de inferență a rețelei neuronale generice va deveni valid după un interval maxim de 20 tacte. Inițierea unui nou proces de inferență începe cu scrierea un vectorului nou de intrare. În dependență de arhitectura specifică a rețelei neuronale simulate în baza arhitecturii neuronale generice, scrierea intrărilor și citirea ieșirilor poate fi optimizată prin accesarea punctată doar a intrărilor sau ieșirilor necesare.

Pentru implementarea sistemului inteligent de comandă și control în procesul de turnare a microfiredelor în baza arhitecturii rețelei neuronale generice proiectate, a fost realizată colectarea inițială a datelor, care ulterior au fost preprocesate utilizând funcțiile și instrumentele disponibile din pachetul de programe *Matlab*. Mai apoi datele colectate au fost utilizate în toolbox-ul *NNTools* pentru antrenarea și evaluarea performanțelor rețelei neuronale artificiale. După exportarea rețelei neuronale antrenate în mediul de simulare *Simulink*, a fost realizat modelul unui sistem de testare utilizând metoda de simulare *Arduino IO* pentru a verifica, în condiții reale, comportamentul sistemului de automatizare a procesului de turnare a microfiredelor.

Pentru extragerea matricei de ponderi sinaptice din obiectul de tip *network* obținut în urma exportării modelului rețelei, a fost elaborat un script, *Anexa 3*, care oferă posibilitatea de a converti orice arhitectură a rețelei neuronale artificiale proiectate în pachetul de programe *Matlab* într-o arhitectură de tip *Hopfield* echivalentă, descrisă printr-o matrice de adiacență [120]. Utilizând această metodă a fost exportată matricea de ponderi sinaptice cu dimensiunea 20×10 .

Pentru simplificarea operațiilor în virgulă mobilă poate fi aplicată conversia numerelor în virgula fixă sau în numere întregi cum a fost procedat și în cazul de față. Pentru aceasta poate fi aplicată funcția *fixdt(1, 32, 0)* [114, 115]. O altă metodă de optimizare a calculelor pe arhitectura hardware ar consta în înmulțirea variabilelor de sistem cu un anumit coeficient 10^y , unde y reprezintă precizia dorită, totodată ținându-se cont de faptul că ieșirile generate de arhitectură vor necesita ajustare corespunzătoare. În cazul în care procedura respectivă este aplicată doar asupra matricei de ponderi sinaptice sau asupra intrărilor, ajustarea va constitui 10^y , iar dacă se înmulțesc cu coeficienți diferiți și intrările și ponderie sinaptice, coeficientul de ajustare a ieșirilor va reprezenta $10^{-(x+y)}$, unde x reprezintă precizia intrărilor, iar y reprezintă precizia ponderilor sinaptice (4.3).

$$10^x \times I \times 10^y \times W = 10^{x+y} \times O \quad (4.3)$$

Odată ce a fost obținută matricea de adiacență (matricea de ponderi sinaptice) aceasta poate fi utilizată pentru configurarea arhitecturii rețelei neuronale artificiale generice prin interfața de comunicare.

Cu scopul asigurării adaptabilității sistemului la diferite condiții ale problemelor, spre soluționarea cărora acesta este orientat, a fost luată decizia de a încadra în structura acestuia un nucleu de microprocesor care ar oferi posibilitate utilizatorului de a dezvolta programe noi de configurare a arhitecturii hardware sau diferiți algoritmi de antrenare a rețelei neuronale generice. Cu acest scop în *Platform Designer* a fost construit un sistem care ar fi în stare să implementeze funcționalitățile extinse menționate anterior, *Figura 4.20*. Pentru interacțiunea mai comodă a dezvoltatorului cu sistemul decizional neuronal prin intermediul IDE-ului *Eclipse*, utilizând

limbajului de programare C++, a fost adăugat un nucleu de microprocesor pe 32 biți *Nios II Processor* notat *Cpu*, *Figura 4.20*.

Pentru stocarea programului de comunicare, a matricei ponderilor sinaptice și a algoritmului de antrenare a fost rezervată o zonă de memorie *RAM* cu o capacitate de 4096 celule pe 32 biți din resursele circuitului *FPGA*, utilizând componenta *On-Chip Memory* notată ca *OnChipRAM*, *Figura 4.20*. Capacitatea acestui bloc de memorie poate varia în dependență de complexitatea programului sau a resurselor utilizate pentru sinteza arhitecturii neuronale generice cu scopul optimizării procesului de distribuție a resurselor interne a circuitului *FPGA* în baza căruia va fi implementată arhitectura.

Pentru testarea sistemului integral a fost adăugată interfața *JtagUART*, *Figura 4.20*, utilizând componenta *Jtag UART Intel FPGA IP* care asigură conexiunea între placa de dezvoltare și consola de sistem – *System Console* a pachetului de programe *Platform Designer* pentru testarea și depanarea programului de bază.

Adresarea memoriei din cadrul arhitecturii neuronale generice este realizată prin interfațarea acesteia cu microprocesorul printr-o interfață de ieșire paralelă *Address* pe 5 biți, *Figura 4.20*. Magistrala comună de date intrare/ieșire *DataInOut* a fost implementată pentru asigurarea transferului paralel de date pe 32 biți dintre microprocesor și arhitectura neuronală generică, *Figura 4.20*. Această magistrală este utilizată atât pentru configurarea inițială a arhitecturii neuronale generice, pentru asigurarea acesteia cu date de intrare și pentru accesarea rezultatelor procesului de inferență. Pentru controlul arhitecturii neuronale generice au fost prevăzute și o serie de semnale de comandă cum ar fi: *WR* – semnal de ieșire pentru microprocesor în dependență de care acesta poate comuta regimul de funcționare al arhitecturii neuronale generice (*WR='1'* regim de scriere, *WR='0'* regim de citire) și *Ready* – semnal de intrare pentru microprocesor generat de către arhitectura neuronală generică cu ajutorul căruia aceasta indică microprocesorului faptul că procesul de inferență a luat sfârșit și rezultatele acestuia sunt valide, *Figura 4.20*. Toate aceste interfețe au fost implementate cu ajutorul componentei *PIO (Paralel I/O) Intel FPGA IP*.

Pentru sincronizarea tuturor componentelor sistemului integral este utilizat blocul de sincronizare notat *Clk* care generează un semnal de tact de 50MHz.

Connections	Name	Description	Export	Clock	Base	End	IRQ	
	Clk	Clock Source	clk	exported				
	clk_in	Clock Input	clk					
	clk_in_reset	Reset Input	reset					
	clk	Clock Output	clk					
	clk_reset	Reset Output	reset					
	Cpu	Nios II Processor						
	clk	Clock Input	clk					
	reset	Reset Input	reset					
	data_master	Avalon Memory Mapped Master						
	instruction_master	Avalon Memory Mapped Master						
	irq	Interrupt Receiver					IRQ 0	
	debug_reset_request	Reset Output						
	debug_mem_slave	Avalon Memory Mapped Slave						
	custom_instruction_m...	Custom Instruction Master				0x0002_0800	0x0002_0fff	IRQ 31
	OnChipRAM	On-Chip Memory (RAM or ROM) Intel ...						
clk1	Clock Input	clk1						
s1	Avalon Memory Mapped Slave							
reset1	Reset Input	reset1			0x0001_0000	0x0001_3fff		
JtagUART	JTAG UART Intel FPGA IP							
clk	Clock Input	clk						
reset	Reset Input	reset						
avalon_jtag_slave	Avalon Memory Mapped Slave							
irq	Interrupt Sender							
Address	PIO (Parallel I/O) Intel FPGA IP							
clk	Clock Input	clk						
reset	Reset Input	reset						
s1	Avalon Memory Mapped Slave							
external_connection	Conduit		address					
DataInOut	PIO (Parallel I/O) Intel FPGA IP							
clk	Clock Input	clk						
reset	Reset Input	reset						
s1	Avalon Memory Mapped Slave							
external_connection	Conduit		datainout					
WR	PIO (Parallel I/O) Intel FPGA IP							
clk	Clock Input	clk						
reset	Reset Input	reset						
s1	Avalon Memory Mapped Slave							
external_connection	Conduit		wr					
Ready	PIO (Parallel I/O) Intel FPGA IP							
clk	Clock Input	clk						
reset	Reset Input	reset						
s1	Avalon Memory Mapped Slave							
external_connection	Conduit		ready					

Figura 4.20 Structura sistemului neuronal cu autoorganizare elaborat în Platform Designer.

Pentru interconectarea sistemului cu microprocesor și a arhitecturii rețelei neuronale generice în Platform Designer au fost exportate toate interfețele necesare, Figura 4.20, apoi a fost generată componenta grafică ANNSystem, Figura 4.21. În baza acestor componente a fost realizată schema integrală a sistemului neuronal cu autoorganizare prezentată în Figura 4.21.

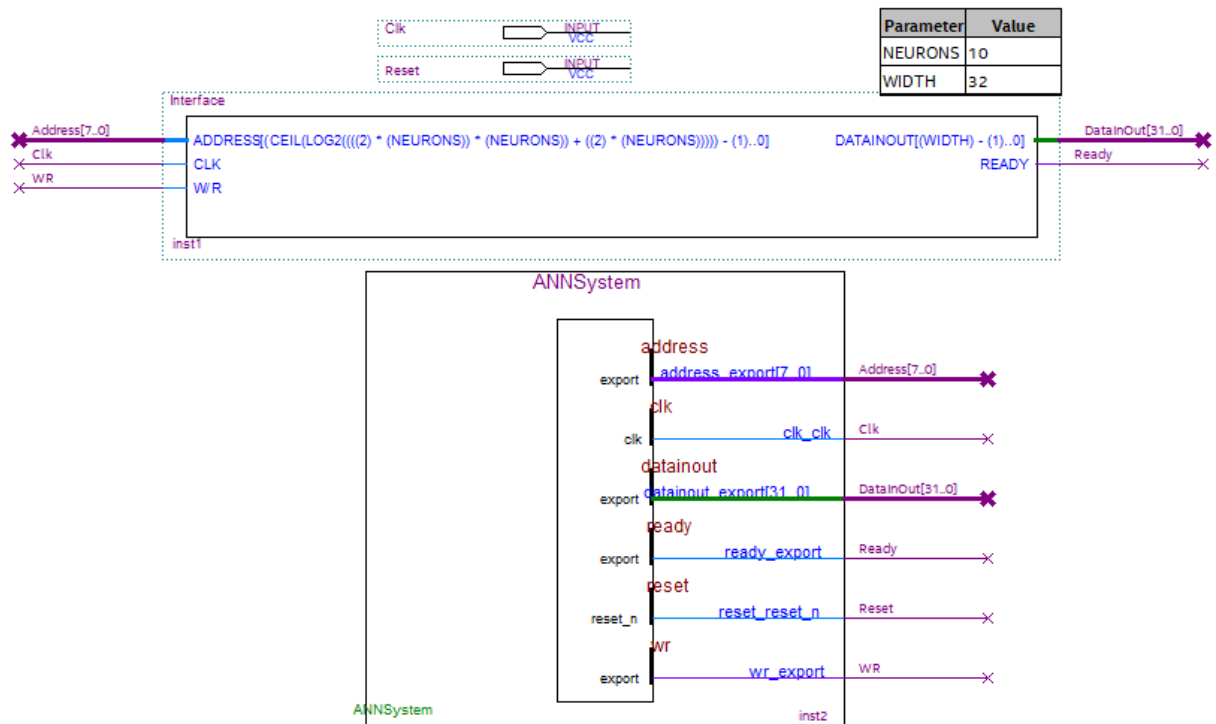


Figura 4.21 Schema de interconectare a sistemului neuronal cu autoorganizare.

În rezultatul analizei și sintezei schemei sistemului neuronal cu autoorganizare a fost obținută arhitectura acestuia prezentată în *Figura 4.22* cu ajutorul căreia poate fi analizată vizual organizarea internă a sistemului dezvoltat pentru a depista și corecta erorile logice, erorile de structură sau reorile de conexiune între componentele acestuia.

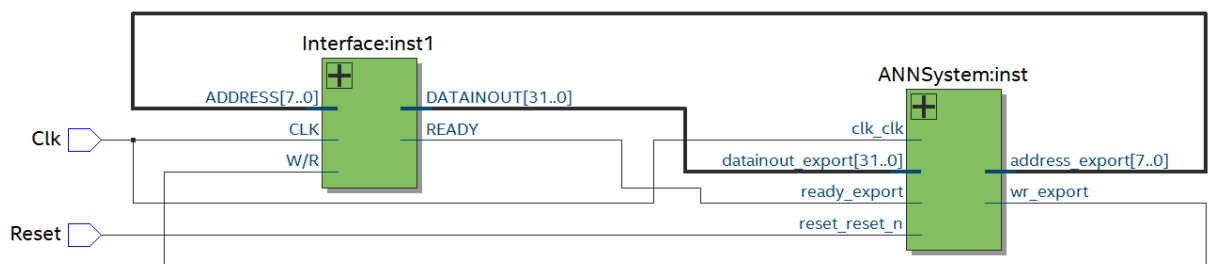


Figura 4.22 Arhitectura internă a sistemului neuronal cu autoorganizare.

După compilarea proiectului și maparea arhitecturii hardware obținute pe diferite familii de circuite *FPGA* cu ajutorul pachetului de programe *Quartus Prime 18.1*, pe cale experimentală a fost identificat circuitul din familia *Cyclone 10 LP* care dispune de resurse necesare pentru implementarea acestei arhitecturi – *10CL120YF484I7G* unde *10C* indică familia circuitului, *L* – variația familiei, *120* indică numărul de elemente logice *120K*, *Y* – tensiunea de alimentare a nucleului *0.9V*, *F* – tipul capsulei *FBGA*, *484* – specifică numărul de pini ai circuitului, *I* –

temperatura de lucru de la -40 până la +100 °C industrial, C – clasa circuitului, 7 – viteza de lucru a circuitului, G – RoHS. Prețul la zi a circuitului constituie în jur de 130 €.

În rezultatul compilării proiectului și mapării pe circuit a arhitecturii obținute au fost utilizate 50% din elementele logice, 8074 registre, 2 pini, 44032 biți de memorie și 576 din 576 de multiplicatoare, celelalte fiind sintetizate în baza elementelor logice, *Figura 4.23*.

Flow Status	Successful - Wed Jan 22 12:42:35 2020
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	ANN
Top-level Entity Name	ANN
Family	Cyclone 10 LP
Total logic elements	59,135 / 119,088 (50 %)
Total registers	8074
Total pins	2 / 278 (< 1 %)
Total virtual pins	0
Total memory bits	44,032 / 3,981,312 (1 %)
Embedded Multiplier 9-bit elements	576 / 576 (100 %)
Total PLLs	0 / 4 (0 %)
Device	10CL120YF484I7G
Timing Models	Final

Figura 4.23 Rezultatul compilării arhitecturii rețelei neuronale artificiale generice pe circuitul FPGA din familia *Cyclone 10*.

În baza informației obținute din rezultatele compilării sistemului integral a fost cercetată piața dispozitivelor de dezvoltare în baza circuitelor FPGA și a fost determinat ca cel mai potrivit pentru cerințele specificate de sistem *Intel® Cyclone® 10 GX FPGA Development Kit* prețul căruia constituie 1200 \$. La baza kit-ului respectiv se află circuitul din familia *Cyclone 10* cu seria *10CX220YF80E5G*, circuit care asigură totalmente cerințele față de resursele hardware ale sistemului dezvoltat. Kitul de dezvoltare respectiv este dotat pe lângă circuitul FPGA de bază, adăugător cu câteva circuite CPLD din familia *MAX 10*, care la rândul său, dispun de interfețe de intrare analogice, fapt care elimină problema utilizării circuitelor FPGA pentru procesarea semnalelor analogice.

În rezultatul implementării sistemului neuronal cu autoorganizare a fost obținută arhitectura hardware a acestuia. Reieșind din datele de simulare a fost identificat circuitul potrivit care ar asigura toate necesitățile specifice ale sistemului proiectat. Procesul de selectare a circuitului a fost realizat ținându-se cont inclusiv de îmbunătățirile potențiale ale arhitecturii date cum ar fi dezvoltarea întregului sistem doar în baza unui singur circuit, posibilitatea utilizării

autonome a sistemului în versiunea de implementare finală care poate fi obținută prin stocarea matricei ponderilor sinaptice într-un bloc de memorie RAM/ROM.

4.3 Concluzii la capitolul 4

În acest capitol au fost prezentate implementări hardware ale sistemelor de suport decizional fuzzy și neuronal dezvoltate pentru controlul automat al procesului de turnare a microfiredelor. Soluțiile prezentate au scopul de a simplifica semnificativ procesul de cercetare și implementare în hardware a sistemelor de suport decizional bazate pe logica fuzzy și pe rețelele neuronale artificiale. Cu acest scop a fost realizată o tentativă de argumentare a eficienței utilizării arhitecturilor generice reconfigurabile la proiectarea sistemelor inteligente. Pentru aceasta au fost prezentate principiile de bază și etapele de proiectare ale sistemelor fuzzy și neuronale.

CONCLUZII GENERALE

Lucrarea conține abordări originale în preluarea experienței operatorului uman și soluții ingineresti moderne pentru aplicarea cunoștințelor extrase în automatizarea proceselor de producție. Aplicarea rezultatelor obținute oferă posibilități extinse în procesul de cercetare a sistemelor de suport decizional.

În rezultatul sintezei rezultatelor obținute, pot fi formulate următoarele concluzii generale:

1. A fost dezvoltată o metodă specifică de extragere a experienței operatorului uman [79, 83, 87, 88] care permite identificarea funcțiilor de apartenență din datele statistice colectate.

2. Au fost propuse metode de dezvoltare ale sistemelor de suport decizional fuzzy și neuronal în baza experienței preluate și a parametrilor specifici extrași de la operatorii umani [8, 63, 95, 118]. Metodele respective permit dezvoltarea unor sisteme de comandă și control inteligent în procesele de producție.

3. Au fost dezvoltate modele Simulink pentru estimarea eficienței modelelor propuse în procese decizionale, care reprezintă soluții independente pentru comanda și controlul proceselor tehnologice [95, 113, 118].

4. Au fost obținute relațiile teoretice care permit estimarea cantității și tipului de resurse necesare pentru implementarea arhitecturilor adaptive în dependență de tipul și caracteristicile sistemului [113].

5. Au fost proiectate și implementate componente parametrizate și arhitecturi hardware adaptive care oferă posibilitatea de dezvoltare a sistemelor fuzzy și neuronale [105, 112].

6. Au fost proiectate și implementate sisteme fuzzy și neuronale cu proprietăți de autoorganizare [113, 117], care oferă suportul decizional al operatorului în procesul de control al calității microfiredor.

În urma analizei rezultatelor obținute poate fi estimată măsura în care au fost atinse obiectivele cercetării:

1. Au fost analizate aspectele generale ale sistemelor neuro-fuzzy și metodele de extragere a cunoștințelor. Au fost identificate posibilitățile și avantajele utilizării acestor metode pentru automatizarea proceselor de luare a deciziilor (Capitolul 1).

2. Au fost propuse, elaborate și dezvoltate metode și algoritmi de extragere a cunoștințelor de la operatorului uman (Capitolul 2), care oferă posibilitatea preluării experienței operatorilor umani și utilizarea acestora pentru automatizarea proceselor de producție industriale.

3. Au fost proiectate arhitecturi hardware reconfigurabile pentru sisteme neuronale și fuzzy (Capitolul 3), care pot servi ca set de instrumente generice pentru dezvoltarea sistemelor hardware cu inteligență artificială.

4. Cu ajutorul arhitecturilor neuronale și fuzzy reconfigurabile au fost proiectate sisteme hibride de luare a deciziilor. În baza simulărilor a fost realizată analiza comparativă a sistemelor de luare a deciziilor proiectate (Capitolul 4). Rezultatele analizei realizate justifică aplicabilitatea arhitecturilor proiectate, iar utilizarea arhitecturilor reconfigurabile la dezvoltarea sistemelor inteligente simplifică semnificativ etapele de proiectare și implementare.

5. A fost confirmată posibilitatea implementării practice a structurilor cu autoorganizare în baza arhitecturilor reconfigurabile (Capitolul 4).

RECOMANDĂRI

Ca direcții de cercetare de viitor putem menționa:

- Automatizarea algoritmului de preprocesare și de identificare a funcțiilor de apartenență din datele statistice colectate de la operatorul uman.
- Dezvoltarea algoritmilor de autoorganizare a arhitecturilor adaptive.
- Transcrierea arhitecturilor hardware din *AHDL* în *VHDL* cu scopul implementării acestora pe circuitele reconfigurabile ale diferitor producători.
- Implementarea diferitor tipuri de interfețe de comunicare cu sistemul de suport decizional autonom pentru interfațarea acestuia cu diferite sisteme hardware moderne.

Toate aceste îmbunătățiri simplifică esențial procesul de proiectare și implementare a sistemelor inteligente în baza rețelelor neuronale artificiale și a celor fuzzy tot odată oferind mai multe oportunități comunității cercetătorilor și dezvoltatorilor în domeniul sistemelor inteligente.

BIBLIOGRAFIE

1. MAIER, P., SACHENBACHER, M., RÜHR, TH., KUHN, L. *Integrated Plan Tracking and Prognosis for Autonomous Production Processes*. In: *Proc. of 14th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA-2009)*. IEEE, Sept. 22-26, 2009. Palma de Mallorca, Spain, 2009, pp. 310-316. ISBN 978-1-4244-2727-7.
2. BEETZ, M., BUSS, M., WOLLHERR, D. *Cognitive Technical Systems — What Is the Role of Artificial Intelligence?* In: Hertzberg J., Beetz M., Englert R., eds. *Advances in Artificial Intelligence (KI 2007)*. Lecture Notes in Computer Science, vol. 4667. Springer, Berlin, 2007, pp. 19–42. ISBN 978-3-540-74565-5.
3. MAIER, P., SACHENBACHER, M., RÜHR, TH., KUHN, L. *Integrated Diagnosis and Plan Assessment for Autonomous Production Processes*. In: *On-line Proc. of The IJCAI-09 Workshop on Self- and Autonomous Systems: reasoning and integration challenges (SAS-09)*. July 13, Pasadena, California, USA, 2009, pp. 51-58.
4. LEHA, A., PANGERCIC, D., RÜHR TH., BEETZ M. *Optimization of Simulated Production Process Performance using Machine Learning*. In: *Proc. of 14th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA-2009)*. IEEE, Sept. 22-26, Palma de Mallorca, Spain, 2009, pp. 1637-1641. ISBN 978-1-4244-2727-7.
5. OMRI, M. N. *Fuzzy Ontology-Based Querying User' Requests Under Uncertain Environment*. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 14(3), 2020, pp. 41-59. [citată 19.06.2020]. DOI:10.4018/IJCINI.2020070103.
6. NIST Project: *Knowledge Extraction and Application for Manufacturing Operations* [online]. [citată 05.07.2020]. Disponibil: <https://www.nist.gov/programs-projects/knowledge-extraction-and-application-manufacturing-operations>
7. ТАММ, Б. Г., ПУУСЕПП, М. Э., ТАВАСТ Р. Р. *Анализ и моделирование производственных систем*. Под общ. ред. Б. Г. Тамма. – М.: Финансы и статистика, 1987. 191 с.
8. CARBUNE, V., CALMICOV, I., GISCA, V. *Arhitectură pentru suport argumentativ al deciziilor luate în procesele industriale*. In: *The 8th International Conference on Microelectronics and Computer Science (ICMCS-2014): proceedings*, Chişinău, Moldova, October 22-25, 2014. Ch.: Tehnica-UTM, 2014, pp. 483-486. ISBN 978-9975-45-5-329-5.
9. TEODORESCU, H. N., KANDEL, A. *Hardware Implementation of Intelligent Systems*. Physica-Verlag Heidelberg, 2001. 282 pp. ISBN 978-3-7908-2491-9.

10. AGGARWAL, C. *Neural Networks and Deep Learning*. Springer International Publishing, 2018, 497 p. ISBN 978-3-319-94462-3.
11. MOGA, Ș.-M. *Rețele neuronale artificiale* [online]. București, [citată 02.10.2019]. Disponibil: https://www.mihaimoga.com/freeware/MogaStefanMihai_1100_ReferatStiintific.pdf
12. *Rețele neuronale artificiale* [online]. [citată 02.10.2019]. Disponibil: http://www.mrm.ugal.ro/balan_site/e-books/miassm-pdf/cap.1.pdf
13. *What are Neuro-Fuzzy Systems?* [online]. 1997 [citată 03.10.2019]. Disponibil: <http://fuzzy.cs.ovgu.de/nfdef.html>
14. NAUCK, D., KLAWONN, F., KRUSE, R. *Foundations of Neuro-Fuzzy Systems*. New York: John Wiley&Sons, 1997. 316 p. ISBN 978-0-471-97151-1
15. VIEIRA, J., DIAS, F. M., MOTA, A. *Neuro-Fuzzy Systems: A Survey*. In: *WSEAS Transactions on Systems* [online]. 2004, vol. 3, issue 2, pp. 414-419 [citată 12.10.2019]. ISSN 1109-2777. Disponibil: https://www.researchgate.net/publication/242073375_Neuro-Fuzzy_Systems_A_Survey
16. GUȚULEAC, E., CALMÎCOV, I., ZAPOROJAN, S., GIRLEANU, I. *Modelarea și analiza securității rețelelor de calculatoare prin rețele Petri markoviene fuzzy intuiționiste cu jocuri stocastice*. In: *Meridian Ingineresc*. 2017, nr. 1, pp. 16-25. ISSN 1683-853X.
17. WEINSHENKER, C. A. *Characterization of Neural Network Backpropagation on Chiptlet-based GPU Architectures* [online]. Williamsburg, 2017. 41 p. [citată 22.12.2019]. Disponibil: <https://scholarworks.wm.edu/honorstheses/1068/>
18. CHOKSHI, N., MCFARLANE, D. *A distributed architecture for reconfigurable control of continuous process operations*. In: *Journal of Intelligent Manufacturing*. 2008, vol. 19, pp. 215-232. eISSN 1572-8145.
19. РУТКОВСКАЯ, Д., ПИЛИНСКИЙ, М., РУТКОВСКИЙ, Л. *Нейронные сети, генетические алгоритмы и нечеткие системы*. Пер. с польского: И. Д. Рудинский. М.: Горячая линия-Телеком, 2006. 452 с. ISBN 5-93517-103-1.
20. *Caracteristici fundamentale ale rețelelor neurale artificiale* [online]. [citată 03.10.2019]. Disponibil: http://scs.etc.tuiasi.ro/iciocoiu/courses/DSP/course5/capitol2_NN.pdf
21. *Ce este organizarea de sine? Procese, principii și teorie a autoorganizării* [online]. [citată 03.10.2019]. Disponibil: <https://olnafu.ru/formare/95865-ce-este-organizarea-de-sine-procese-principii-%C8%99i.html>

22. *Auto-organizarea sistemelor*. In: PEAT, D. *De la certitudine la incertitudine* [online]. 2012, cap. 54 [citată 03.10.2019]. Disponibil: <https://www.scientia.ro/fizica/de-la-certitudine-la-incertitudine-david-peat/4251-auto-organizarea-sistemelor.html>
23. *Autoorganizarea și emergența în sistemele adaptive complexe* [online]. [citată 03.10.2019]. Disponibil: <http://www.asecib.ase.ro/BCIB/CAPITOLUL-5.pdf>
24. GUȚULEAC, E. *Descriptive Self-Reconfigurable Generalized Stochastic Petri Nets for Performance Modelling of Computer Systems*. In: *Buletinul Institutului Politehnic din Iași: Automatica și Calculatoare*. 2005, t. 51 (55), fasc. 1-4, pp. 121-136. ISSN 1220-2169.
25. DAVIES, E. R. *Machine vision: Theory, algorithms, practicalities*. 3rd ed. San Francisco: Morgan Kaufmann, 2004. 934 p. ISBN 978-0-12-206093-9.
26. NILSSON, N. J. *Artificial intelligence: A new synthesis*. San Francisco: Morgan Kaufmann, 1998. 513 p. ISBN 978-1-55860-467-4.
27. RUSSELL, S., NORVIG, P. *Artificial Intelligence: A modern approach*. 2nd ed. New Jersey: Prentice Hall, 2003. 1132 p. ISBN 0137903952.
28. ALTY, J. L., COOMBS, M. J. *Expert systems: Concepts and examples*. New York: John Wiley and Sons, 1984.
29. HERNÁNDEZ-GUZMÁN, V.M., SILVA-ORTIGOZA, R. *Automatic Control with Experiments*. Springer International Publishing, 2019, 992 p.
30. YAGER, R. R., ed. *Fuzzy set and possibility theory: Recent developments*. New York: Pergamon Press, 1982.
31. DUMITRACHE, I., coord. *Automatica*. București: Ed. Academiei Române, 2009. Vol. 1. ISBN 978-973-1883-4.
32. ASTRÖM, K. J., HAGGLUND, T. *Advanced PID control*. USA: The Instrumentation, Systems and Automation Society, 2006. 460 p. ISBN 978-1-55617-942-6.
33. PEPTEA, T. C. *Metode și tehnici de achiziționare, reprezentare și utilizare a cunoștințelor SLT* [online]. Sibiu [citată 22.12.2019]. Disponibil: <https://ro.scribd.com/doc/82666463/referat-Metode-%C8%99i-tehnici-de-achizi%C8%9Bionare-reprezentare-%C8%99i-utilizare-a-cuno%C8%99tin%C8%9Belor>
34. БЕРШТЕЙН, Л. С., БОЖЕНЮК, А. В. *Нечеткие модели принятия решений: дедукция, индукция, аналогия*. Таганрог: Изд-во ТРТУ, 2001. 110 с.
35. KIENDL, H., KRAUSE, P., SCHAUTEN, D., SLAWINSKI, T. *Data-Based Fuzzy Modeling for Complex Applications*. In: SCHWEFEL, H. P., WEGENER, I., WEINERT, K., eds. *Advances in Computational Intelligence*. Springer, Berlin, 2003, pp 46-77.

36. LAZAROVICI, GH., MICLE, D. *Introducere în arheologia informatizată* [online]. [citată 03.10.2019]. Disponibil: <http://arheologie.ulbsibiu.ro/publicatii/carti/ai/1e.htm>
37. SAEED ALI AMER, M. *Baze de cunoștințe. Interfețe de interogare* [online]: rez. tz. de doct. Pitești, 2018. 47 p. [citată 03.10.2019]. Disponibil: <https://www.upit.ro/document/29530/rezumat-sept2018-ali.pdf>
38. *Metode pentru reprezentarea cunoștințelor: Caracteristici ale cunoștințelor* [online]. [citată 03.10.2019]. Disponibil: <http://www.scrigroup.com/calculatoare/algorithmi/Metode-pentru-reprezentarea-cu84715.php>
39. ZULEAN, M., ALIGICA, D. P. *Deciziile strategice în condiții de incertitudine*. In: ZULEAN, M., ALIGICA, D. P., coord. *Politica de securitate națională: Concepte, instituții, procese* [online]. Iași: Polirom, 2007, pt. 2, cap. 8, pp. 131-145 [citată 22.12.2019]. ISBN 978-973-46-0624-5. Disponibil: https://www.researchgate.net/publication/286602130_Politica_de_securitate_nationala_Concepte_institutii_procese
40. ZUGRAV, I. *Procesul decizional în condiții de incertitudine cu aplicații în activitatea de antreprenariat*. In: *Dezvoltarea inovativă, colaborativă, incluzivă a cooperativelor: teorie, practică, perspective: lucrările conf. șt.-practice intern.*, 13-14 septembrie 2018. Chișinău: Univ. Cooperatist-Comercială din Moldova, 2018, vol. 2, pp. 140-146. ISBN 978-99753272-7-5.
41. IANCU, E., VANCEA, R. *Considerații generale asupra sistemelor inteligente*. In: *Analele Universității din Oradea: Științe economice* [online]. 2006, t. 15, vol. 2, pp. 750-755 [citată 03.10.2019]. ISSN 1582-5450. Disponibil: <http://steconomiceuoradea.ro/anale/volume/2006/statistica-si-informatica-economica/23.pdf>
42. PRAKASH, A., DESHMUKH, S. G. *Modelling of FMS control policy: AIS-based fuzzy expert system*. In: *International Journal of Industrial and Systems Engineering*. 2011, vol. 8, no. 1, pp. 38-60. ISSN 1748-5037.
43. GHIONEA, F., POPA, E. *Rezolvare deterministă versus rezolvare fuzzy*. In: *Buletinul AGIR* [online]. 2016, anul 21, nr. 2, pp. 68-75 [citată 22.12.2019]. eISSN 2247-3548. https://www.buletinulagir.agir.ro/numar_revista.php?id=130
44. FIAC, L., MIRAMOND, B., UPEGUI, A., VANNEL, F. *Dynamic parallel configuration for self-adaptive hardware architectures*. In: *NASA/ESA Conference on Adaptive Hardware and Systems* [online], 14-17 July 2014, Leicester, UK [citată 02.09.2019]. Disponibil: <https://www.semanticscholar.org/paper/Dynamic-parallel-reconfiguration-for-self-adaptive-Fiack-Miramond/3bdf136f31baeb057512a4aef884ac9d74473a11>

45. SOTO, J., MORENO, J. M., CABESTANY, J. *A self-adaptive hardware architecture with fault tolerance capabilities*. In: *Neurocomputing* [online]. 2013, vol. 121, pp. 25-31 [citat 02.09.2019]. ISSN 0925-2312. Disponibil: <https://www.sciencedirect.com/science/article/abs/pii/S0925231213004293?via%3Dihub>
46. HENNESSY, J., PATTERSON, D. *Computer Architecture: A Quantitative Approach*. 5th ed. San Francisco: Morgan Kaufmann, 2012. 857 p. ISBN 978-0-12-383872-8.
47. HENNESSY, J., PATTERSON, D. *Computer Architecture: A Quantitative Approach*. 3rd ed. San Francisco: Morgan Kaufmann, 2003. 1136 p. ISBN 978-1558605961.
48. MUELLER, S., PAUL, W. J. *Computer Architecture: Complexity and Correctness*. Berlin: Springer, 2000. 553 p. ISBN 3-540-67481-0.
49. HWANG, K., JOTWANI, N. *Advanced computer architecture: Parallelism, Scalability, Programmability*. 2nd ed. New Delhi: McGraw-Hill Education, 2010. 810 p. ISBN 978-0070702103.
50. STAUNSTRUP, J., WOLF, W., eds. *Hardware/Software Co-Design: Principles and Practice*. Boston: Springer, 1997. 396 p. ISBN 978-1-4419-5018-5.
51. *Intel® FPGAs and Programmable Devices*. © Intel Corporation [citat 22.09.2019]. Disponibil: <https://www.intel.com/content/www/us/en/products/programmable.html>
52. UPTON, E. et al. *Learning Computer Architecture with Raspberry Pi* [online]. Indianapolis: John Wiley and Sons, 2016. 528 p. [citat 22.12.2019]. ISBN 978-1-119-18393-8. Disponibil: [http://163.178.104.150/ci1210/practicas/RAPSBERRY/Learning%20Computer%20Architecture%20with%20Raspberry%20Pi%20by%20Eben%20Upton,%20Jeffrey%20Duntemann%202016%20\(1st%20Edition\).pdf](http://163.178.104.150/ci1210/practicas/RAPSBERRY/Learning%20Computer%20Architecture%20with%20Raspberry%20Pi%20by%20Eben%20Upton,%20Jeffrey%20Duntemann%202016%20(1st%20Edition).pdf)
53. JERRAYA, A., YOO, S., VERKEST, D., WEHN, N., eds. *Embedded Software for SoC* [online]. New York [etc.]: Springer, 2004. 530 p. [citat 22.12.2019]. ISBN 1-4020-7528-6. Disponibil: https://books.google.md/books?id=otQLCAAQAQBAJ&pg=PR5&lpg=PR5&dq=ISBN:+1-4020-7528-6&source=bl&ots=kUftkLvuv&sig=ACfU3U0aBkqTWxzFh6EoKM85J24r43zcAQ&hl=ro&sa=X&ved=2ahUKEwixzfl51fPnAhXhsYsKHRO1A_gQ6AEwAHoECA8QAQ#v=onepage&q=ISBN%3A%201-4020-7528-6&f=false
54. ZAPOROJAN, S., PLOTNIC, C., CALMICOV, I., LARIN, V. *A knowledge-based approach for microwire casting plant control*. In: JOZEF CZYK, J. AND ORSKI, D., eds.

- Knowledge-Based Intelligent System Advancements: Systemic and Cybernetic Approaches*. IGI Global, Hershey, PA, 2010, pp. 419-437.
55. CALMÎCOV, I. *Soluții încorporate pentru sisteme de măsurare și control în producerea și aplicarea microfiredor: tz. de doct. în tehnică*. Chișinău, 2018, 119 p.
 56. MUSIL, M. R. *Combining Algorithms for More General AI* [online]. Portland, 2018 [citată 22.12.2019]. Disponibil: https://pdxscholar.library.pdx.edu/mcecs_mentoring/20/
 57. DRIANKOV, D., HELLENDORN, H., REINFRANK, M. *An introduction to fuzzy control*. New York: Springer, 1993.
 58. PASSINO, K. M., YURKOVICH, S. *Fuzzy control*. Boston: Addison-Wesley Longman, 1998.
 59. LUCERO, Y. C., NAVA, P. A. *A Method for Membership Function Generation from Training Samples* [online]. [citată 12.12.2019]. Disponibil: <http://wwwold.ece.utep.edu/research/webfuzzy/docs/electro99/electro-99.html>
 60. KOSKO, B. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs: Prentice Hall, 1992. 449 p. ISBN 0-13-612334.
 61. HONG, T.-P., LEE, C.-Y. *Induction of fuzzy rules and membership functions from training examples*. In: *Fuzzy Sets and Systems* [online]. 1996, vol. 84, issue 1, pp. 33-47 [citată 12.12.2019]. ISSN 0165-0114. Disponibil: <https://www.sciencedirect.com/science/article/abs/pii/0165011495003053?via%3Dihub>
 62. COUSO, I., BORGELT, C., HULLERMEIER, E., KRUSE, R. *Fuzzy Sets in Data Analysis: From Statistical Foundations to Machine Learning*. In: *Computational Intelligence Magazine*, vol. 14(1), 2019, pp. 31-44.
 63. ZAPOROJAN, S., CARBUNE, V., CALMICOV, I. *Fuzzy logic control based on reconfigurable membership tables*. In: GRZECH, A., SWIATEC, P., DRAPALA, J., eds. *Advances in Systems Science*. Warsaw: Academic Publishing House EXIT, 2010, pp. 385-390. ISBN 978-83-60434-77-2.
 64. SILER, W., BUCKLEY, J. J. *Fuzzy expert systems and fuzzy reasoning* [online]. Hoboken: John Wiley and Sons, 2005. 405 p. [citată 12.09.2019]. ISBN 0-471-38859-9. Disponibil: http://s1.nonlinear.ir/epublish/book/Fuzzy_Expert_Systems_and_Fuzzy_Reasoning_0471388599.pdf
 65. BABUSKA, R., MAMDANI, E. *Fuzzy control*. In: *Scholarpedia the peer-reviewed open-access encyclopedia* [online]. Last mod.: 21 October 2011, 04:09 [citată 22.12.2019]. Disponibil: http://www.scholarpedia.org/article/Fuzzy_control

66. BANKS, W., HAYWARD G. *Fuzzy logic in embedded microcomputers and control systems* [online]. Waterloo: Byte Craft Limited, 2002. 70 p. [citat 12.09.2019]. Disponibil: <http://phaedsys.co.uk/principals/bytecraft/bytecraftdata/bcfuzlogic.pdf>
67. BOICULESE, L., TEODORESCU, H., DIMITRIU, G. *Optimizaton of fuzzy controllers by exhaustive, trial & error and genetic methods. Medical applications*. In: *Computer Science Journal of Moldova*. 1996, vol. 4, no. 1, pp. 69-87. ISSN 1561-4042.
68. CHEN C.-L., CHEN, Y.-M. *Self-organizing fuzzy logic controller design*. In: *Computers in Industry* [online]. 1993, vol. 22, issue 3, pp. 249-261 [citat 22.12.2019]. Disponibil: <https://www.sciencedirect.com/science/article/abs/pii/016636159390092F>
69. MASTACAN, L., ALAH, I., ANITA, L. *Fuzzy control for thermal plant*. In: *Proceedings of the 4th International Conference on Development and applications systems*, 21-22 May 1998. Suceava: University Press, 1998, pp. 107-112.
70. MENDEL, J. M., MOUZOURIS, G. C. *Designing fuzzy logic systems*. In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* [online]. 1997, vol. 44, no. 11, 1997, pp. 885-895 [citat 12.12.2019]. ISSN 1057-7130. Disponibil: <http://sipi.usc.edu/~mendel/publications/Mouzouris%26Mendel%20CAS%2011-97.pdf>
71. VON ALTROCK, C. *Practical fuzzy logic design* [online]. In: *Proceedings of the International Design and Testing Conference - IDTC'96*. Paris, 1996 [citat 20.11.2019]. Disponibil: http://www.fuzzytech.com/e/e_a_pfd.html
72. KETATA, R., DE GEEST, D., TITLI A. *Fuzzy controller: design, evaluation, parallel and hierarchical combination with a PID controller*. In: *Fuzzy Sets and Systems* [online]. 1995, vol. 71, issue 1, pp. 113-129 [citat 12.12.2019]. ISSN 0165-0114. Disponibil: <https://www.sciencedirect.com/science/article/abs/pii/016501149400189E>
73. WAN, E. A. *Control Systems: Classical, Neural and Fuzzy* [online]. Oregon: [s. n.], 1998. 283 p. [citat 22.12.2019]. Disponibil: http://www.ifko.ktu.lt/~raimund/SWOT_FCM/LECTURE%20NOTES.pdf
74. ROMAN-IFTENE, A.-M. *Contribuții la modelarea sistemelor de conducere automată utilizând algoritmi neuro-fuzzy* [online]: tz. de doct. București, 2014. 167 p. [citat 15.10.2019]. Disponibil: http://instal.utcb.ro/Documente_Website/teze/Teza_Andreea_Iftene.pdf
75. ALÁŠKOVÁ, K., Kliestik, T., Mišanková, M. *The Role of Fuzzy Logic in Decision Making Process*. In: *2nd International Conference on Management Innovation and Business Innovation (ICMIBI 2014)*, Bangkok, THAILAND, 2014.

76. GAN, G., LI, B., LI, X., WANG, S. *Advanced Data Mining and Applications*. In: *The Proceedings of 14th International Conference, ADMA 2018*. November 16–18, Nanjing, China, 2018. Lecture Notes in Artificial Intelligence, vol. 11323. Springer, 2018, 532 p.
77. d'ACIERNO, A., ESPOSITO, M., De PIETRO, G. *An extensible six-step methodology to automatically generate fuzzy DSSs for diagnostic applications* [online]. BMC Bioinformatics, 14 jan, 2013, [accesat 01.03.2020]. Disponibil: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3548688/>
78. SADOLLAH, A. *Introductory Chapter: Which Membership Function is Appropriate in Fuzzy System?*, In: *Fuzzy Logic Based in Optimization Methods and Control Systems and Its Applications* [online], IntechOpen, October 31, 2018, [accesat 01.03.2020]. Disponibil: <https://www.intechopen.com/books/fuzzy-logic-based-in-optimization-methods-and-control-systems-and-its-applications/introductory-chapter-which-membership-function-is-appropriate-in-fuzzy-system->
79. CARBUNE, V. *Fuzzy functions of expert knowledge encapsulated within statistical workflow data*. In: *Journal of Engineering Science*, vol. XXVII, no. 3, 2020, pp. 146-155. ISSN 2587-3474.
80. CESA-BIANCHI, N., LUGOSI, G. *Prediction, Learning, and Games*. New York: Cambridge University Press, 2006. 408 p. ISBN 0521841089.
81. GAINDRIC, C. *Abordări sistemice în luarea deciziilor: suport de curs*. Chișinău: Univ. Acad. de Științe a Moldovei, 2017. 156 p.
82. CANO, J.C., NAVA, P. *A fuzzy method for automatic generation of membership function using fuzzy relations from training examples*. In: *Proceedings of Fuzzy Information Processing Society*. NAFIPS. Annual Meeting of the North American, 2002, pp.158 – 162.
83. ZAPOROJAN, S., CĂRBUNE, V., CALMÎCOV, I. *Data-Based Technique for Extracting Knowledge from Data Generated in Experiments*. In: *Pre-Proceedings of the IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP 2020)*. September 3-5, 2020, Cluj-Napoca, Romania, pp. 13-19, ISBN 978-1-7281-9079-2.
84. YOSHIKAWA A. *Improvement of Membership Function Identification Method in Usability and Precision* [online]. In: *Advances in Soft Computing*. Roy R., Furuhashi T., Chawdhry P.K., eds. Springer, London, 1999, [accesat 01.03.2020]. Disponibil: https://link.springer.com/chapter/10.1007/978-1-4471-0819-1_18
85. NIERADKA G., BUTKIEWICZ B. *A Method for Automatic Membership Function Estimation Based on Fuzzy Measures* [online]. In: *Foundations of Fuzzy Logic and Soft Computing*. Melin P., Castillo O., Aguilar L.T., Kacprzyk J., Pedrycz W. eds IFSA 2007.

- Lecture Notes in Computer Science, vol 4529. Springer, Berlin, Heidelberg, 2007, [accesat 01.03.2020]. Disponibil: https://link.springer.com/chapter/10.1007/978-3-540-72950-1_45
86. PAZHOUMAND-DAR, H., LAM, C., MASEK, M. *Automatic Generation of Fuzzy Membership Functions using Adaptive Mean-shift and Robust Statistics*. ICAART, 2016, pp.160-171.
 87. ZAPOROJAN, S., CALMICOV, I., PAVEL, V., LARIN, V., CĂRBUNE, V. *Sistem de măsurare a parametrilor firelor magnetice bistabile asistat de calculator*. In: *The 16th International Exhibition of Research, Innovation and Technological Transfer INVENTICA-2012*. June 13-15, 2012. Iasi: [s. n.], 2012, pp. 632-633. ISSN 1844-7880.
 88. ZAPOROJAN, S., CALMÎCOV, I., PLOTNIC, C., CĂRBUNE, V. *Monitorizarea procesului de fabricație a microfivelor*. In: *Conferința Națională de Interacțiune Om-Calculator*. 2-3 septembrie 2013, Cluj-Napoca. București: MATRIX ROM, 2013, pp. 173-176. ISSN 2344-1690.
 89. ЗАБОРОВСКИЙ, В. и др. *Способ управления процессом литья микропровода*. Авторское свидетельство СССР № 1088076, МПК H01B 13/06, № 2811754. Заяв. 27.09.1979. Оpubл.: 23.04.1984. В: Бюл. № 15, с. 201.
 90. РОЙЗЕН, З. Е., ЙОЙШЕР, А. М. *Коммутирующее устройство*. Авторское свидетельство СССР № 1365168, МПК H01H 37/68, № 4047833. Заяв. 02.04.1986. Оpubл.: 23.04.1988. В: Бюл. № 1, с. 201.
 91. ЗАБОРОВСКИЙ, В. И. и др. *Устройство для автоматического регулирования погонного сопротивления микропровода в установке для его литья*. Авторское свидетельство СССР № 862242, МПК H01B 13/06, № 2852501. Заяв. 14.12.1979. Оpubл.: 07.09.1981. В: Бюл. № 33, с. 266.
 92. ЗАБОРОВСКИЙ, В. И. и др. *Устройство для автоматического регулирования погонного сопротивления микропровода в установке для его литья*. Авторское свидетельство СССР № 860143, МПК H01B 13/06, № 2864778. Заяв. 13.12.1979. Оpubл.: 30.08.1981. В: Бюл. № 32, с. 246.
 93. ЛЕОНЕНКОВ, А. В. *Нечеткое моделирование в среде MATLAB и fuzzyTECH*. СПб: БХВ-Петербург, 2005. 736 с. ISBN 5-94157-087-2.
 94. *Simularea sistemelor cu logică fuzzy în Matlab: prezentarea Toolbox-ului Fuzzy Logic* [online]. [citat 22.12.2019]. Disponibil: http://www.bel.utcluj.ro/rom/dce/goltean/tice/lab/2%20SimulareaSLF_Matlab.pdf
 95. CĂRBUNE, V., CALMÎCOV, I. *Modele neuronale și fuzzy în preluarea experienței operatorului uman*. In: *Telecommunications, Electronics and Informatics (ICTEI-2018)*:

- proceeding of the 6th intern. conf., May 24-27, 2018. Chişinău: Tehnica-UTM, 2018, pp. 263-266. ISBN 978-9975-45-540-4.
96. DIOSAN, L., DUMITRESCU, D. *Evolutionary coalition formation in full connected and scale free networks*. In: *International Journal of Computers, Communications & Control (IJCCC)* – special issue ICCCC, 3, 2008, pp. 259-265.
 97. KUMAR, K. *Knowledge Extraction From Trained Neural Networks*. In: *International Journal of Information and Network Security*. 2012, vol. 1, no. 4, pp. 282-293. ISSN 2089-3299.
 98. *Design a simple Neural Network On MATLAB using nntoolbox* [online]. [citat 18.07.2019].
Disponibil: <https://www.youtube.com/watch?v=hi2TgggG87U>
 99. *Engineering Applications of Neural Networks (EANN 2013)* [online]: proceedings of the 14th intern. conf., September 13-16, 2013, Halkidiki, Greece. Springer, 2013. Vol. 1. [citat 22.12.2019]. ISBN 978-3-642-41013-0. Disponibil: https://books.google.md/books?id=Swq7BQAAQBAJ&pg=PA150&lpg=PA150&dq=traini ng+state+plot+matlab&source=bl&ots=c -RaP628D&sig=iFcupiSDhRCjzmCmZpGF5bg- qgc&hl=ro&sa=X&redir_esc=y#v=onepage&q=training%20state%20plot%20matlab&f=false
 100. SAMADANI, M., KITIO KWUIMY, C. A., NATARAJ, C. *Diagnostics of a nonlinear pendulum using computational intelligence*. In: *Proceedings of the ASME 2013 International Dynamic Systems and Control Conference*. October 21-23, 2013, Palo Alto, California, USA. 2013, pp. 1-8 [citat 22.12.2019]. Disponibil: https://www.researchgate.net/publication/258567472_Diagnostics_of_a_Nonlinear_Pendulum_Using_Computational_Intelligence
 101. YADAV, A. K., MALIK, H., CHANDEL, S. S. *Selection of most relevant input parameters using WEKA for artificial neural network based solar radiation prediction models*. In: *Renewable and Sustainable Energy Reviews* [online]. 2014, vol. 31, pp. 509-519 [citat 11.12.2019]. ISSN 1364-0321. Disponibil: https://www.researchgate.net/publication/260015601_Selection_of_most_relevant_input_parameters_using_WEKA_for_artificial_neural_network_based_solar_radiation_prediction_models
 102. *Difference between regression and performance plot of Artificial neural network in MATLAB* [online]. [citat 22.12.2019]. Disponibil: <https://stackoverflow.com/questions/30001458/difference-between-regression-and-performance-plot-of-artificial-neural-network>

103. HUSSEIN, F. *HexArray: A Novel Self-Reconfigurable Hardware System* [online]: diss. of. doct. of philosophy in electrical and computer eng. Boise, 2017. 237 p. [citat 23.07.2019]. Disponibil: <https://scholarworks.boisestate.edu/td/1261/>
104. ECKHART, P. J. *Tiled DANNA: Dynamic Adaptive Neural Network Array Scaled Across Multiple Chips* [online]: master's thesis. Knoxville, 2017. 76 p. [citat 23.07.2019]. Disponibil: https://trace.tennessee.edu/utk_gradthes/4870/
105. CĂRBUNE, V., CHIRILĂ, S., PODUBNÎI, M. *Componente Hardware Parametrizate pentru Rețele Neuronale Artificiale*. In: *Microelectronics and Computer Science (ICMCS-2011)*: proceedings of the 7th intern. conf., September 22-24, 2011. Chișinău: UTM, 2011, vol. 1, pp. 300-303. ISBN 978-9975-45-174-1.
106. ZADEH, L. A. *Fuzzy sets*. In: *Information and Control* [online]. 1965, vol. 8, issue 3, pp. 338-353 [citat 22.12.2019]. ISSN 0019-9958. Disponibil: <https://www.sciencedirect.com/science/article/pii/S001999586590241X>
107. CARDARILLI, G. K., LOJACONO, R., RE, M. *High Performance Fuzzy Processors*. In: TEODORESCU, H.-N., JAIN, L. C., KANDEL, A., eds. *Hardware Implementation of Intelligent Systems*. Heidelberg: Springer, 2001, pt. 2, chap. 4, pp. 121-146. *Studies in Fuzziness and Soft Computing*, vol. 74. ISBN 978-3-7908-2491-9.
108. FALCHIERI, D., GABRIELLI, A., GANDOLFI, E. *A Digital Fuzzy Processor for Fuzzy-Rule-Based Systems*. In: TEODORESCU, H.-N., JAIN, L. C., KANDEL, A., eds. *Hardware Implementation of Intelligent Systems*. Heidelberg: Springer, 2001, pt. 2, chap. 5, pp. 147-164. *Studies in Fuzziness and Soft Computing*, vol. 74. ISBN 978-3-7908-2491-9.
109. MURPHY, K. P. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012, 1067 p. ISBN 978-0-262-01802-9.
110. HOPFIELD, J.J. *Neural Network and Physical Systems with Emergent Collective Computational Abilities*. In: *Proceedings of the National Academy of Science of the United States of America*. April, 1982.
111. ALI, H. K., MOHAMMED, E. Z. *Design Artificial Neural Network Using FPGA*. In: *International Journal of Computer Science and Network Security* [online]. 2010, vol. 10, no. 8, pp. 88-92 [citat 22.12.2019]. ISSN 1738-7906. Disponibil: https://www.researchgate.net/publication/267377905_Design_Artificial_Neural_Network_Using_FPGA
112. CARBUNE, V., ZAPOROJAN, S. *Analysis and Design of a Specialized Pipeline for Numerical Algorithms Implementation*. In: *Microelectronics and Computer Science*

- (ICMCS-2009): proceedings of the 6th intern. conf., October 1-3, 2009. Chişinău: UTM, 2009, vol. 1, pp. 241-244. ISBN 978-9975-45-045-4.
113. CĂRBUNE, V., CALMÎCOV, I., ZAPOROJAN, S., MORARU, V. *Maparea reţelelor neuronale artificiale dezvoltate în Matlab pe structuri hardware reconfigurabile*. In: *Culegerea de lucrări a conferinţei a XIVa "Zilele Academiei de Stiinte Tehnice din Romania"*, 17-18 Octombrie 2019, Chisinau, Republica Moldova. AGIR, pp. 01-06, 2020, ISSN: 2066-6586 (in press).
114. ROCHER, R., MENARD, D. HERVE, N., SENTIEYS, O. *Fixed-Point Configurable Hardware Components*. In: *Journal on Embedded Systems* [online]. 2006, no. 1, 13 p. [citât 22.12.2019]. ISSN 1741-1068. Disponibil: https://www.researchgate.net/publication/26487352_Fixed-Point_Configurable_Hardware_Components
115. ROY, S., BANERJEE, P. *An Algorithm for Converting Floating-Point Computations to Fixed-Point in MATLAB based FPGA design*. In: *Proceedings of the 41st annual Design Automation Conference* [online]. June 7-11, 2004, San Diego, USA. New York: ACM, 2004, pp. 484-487 [citât 22.12.2019]. ISBN 978-1-58113-828-8. Disponibil: https://www.researchgate.net/publication/221061073_An_algorithm_for_converting_floating-point_computations_to_fixed-point_in_MATLAB_based_FPGA_design
116. KARAGIANNIS, S. et al. *Prediction of Surface Texture Characteristics in Turning of FRPs Using ANN*. In: *Engineering Applications of Neural Networks* [online]: September 13-16, Halkidiki, Greece. Berlin: Springer, 2013, vol. 383, pt. 1, pp. 144-153 [citât 22.12.2019]. ISBN 978-3-642-41013-0. Disponibil: https://link.springer.com/chapter/10.1007/978-3-642-41013-0_15
117. GUȚULEAC, E., ZAPOROJAN, S., GIRLEANU, I., CĂRBUNE, V. *Hybrid Stochastic Petri Nets With Matrix Attributes for Modeling of Discrete-Continuous Process*. In: *Meridian Ingineresc*. 2016, nr. 2, pp. 34-40. ISSN 1683-853X.
118. CĂRBUNE, V. *Model neuronal decizional în baza experienţei operatorului uman*. In: *Microelectronics and Computer Science (ICMCS-2017): proceedings of the 9th intern. conf.*, October 19-21, 2017. Chişinău: Tehnica-UTM, 2017, pp. 202-205.
119. MARINESCU, S.-I. *Cercetări privind Aplicarea Reţelelor Neuronale în Managementul Organizaţional* [online]: rez. tz. de doct. Sibiu, 2017. 51 p. [citât 03.10.2019]. Disponibil: <http://doctorate.ulbsibiu.ro/wp-content/uploads/Rezumatdoctorat-Marinescu.pdf>
120. WIBOWO, F. *An Analysis of FPGA Hardware Platform Based Artificial Neural Network*. In: *Journal of Physics: Conference Series*. 2019, vol. 1201, pp. 1-9. ISSN 1742-6588.

ANEXE

Anexa 1. Deschierarea în AHDL a motorului de inferență generic.

A1.1 Descrierea în AHDL a funcției parametrice FUZZYAND.

PARAMETERS

(

Width = 0

);

SUBDESIGN FUZZYAND

(X1[Width-1..0], X2[Width-1..0] : INPUT ;

Y[Width-1..0] : OUTPUT;

)

BEGIN

IF X1[Width-1..0]<=X2[Width-1..0] THEN

Y[Width-1..0]= X1[Width-1..0];

ELSIF X2[Width-1..0]<=X1[Width-1..0]THEN

Y[Width-1..0]= X2[Width-1..0];

END IF;

END;

A1.2 Descrierea în AHDL a funcției parametrice FUZZYOR.

PARAMETERS

```
(  
  Width = 0  
);
```

SUBDESIGN FUZZYOR

```
(  X1[Width-1..0], X2[Width-1..0]  : INPUT ;  
  Y[Width-1..0]                    : OUTPUT;  
)
```

BEGIN

```
IF X1[Width-1..0]>=X2[Width-1..0] THEN  
Y[Width-1..0]= X1[Width-1..0];  
ELSIF X2[Width-1..0]>=X1[Width-1..0]THEN  
Y[Width-1..0]= X2[Width-1..0];  
END IF;  
END;
```

A1.3 Descrierea în AHDL a componentei parametrice FUZZYCEL.

PARAMETERS

```
(  
    Width = 1  
);
```

SUBDESIGN FUZZYCEL

```
(  
    A[Width-1..0]      : INPUT;  
    B[Width-1..0]      : INPUT;  
    OP[1..0]           : INPUT;  
    C[Width-1..0]      : OUTPUT;  
)  
BEGIN  
IF OP[0]==B"00" THEN  
C[] = A[];  
END IF;  
IF OP[1..0]==B"01" THEN  
C[] = B[];  
END IF;  
IF A[] <= B[] AND OP[0]==B"10" THEN  
    C[] = A[];  
ELSIF B[] <= A[] AND OP[0]==B"10" THEN  
    C[] = B[];  
END IF;  
IF A[] >= B[] AND OP[0]==B"11" THEN  
    C[] = A[];  
ELSIF B[] >= A[] AND OP[0]==B"11" THEN  
    C[] = B[];  
END IF;  
END;
```

Anexa 2. Deschiera în AHDL a arhitecturilor neuronale generice.

A2.1 Descrierea în AHDL a componentei parametrice de generare a produselor.

PARAMETERS

```
(  
    Inputs = 1,  
    Width = 1  
);
```

SUBDESIGN P

```
(  
    Xi[Inputs-1..0][Width-1..0]: INPUT ;  
    Wi[Inputs-1..0][Width-1..0]: INPUT ;  
    Pi[Inputs-1..0][Width-1..0]: OUTPUT;  
)
```

BEGIN

FOR i IN 0 TO Inputs-1 GENERATE

```
    Pi[i][]=Xi[i][]*Wi[i][];
```

END GENERATE;

END;

A2.2. Descrierea în AHDL a componentei parametrice de sumare paralelă.

```
PARAMETERS
(
    Width = 0,
    Inputs = 0
);
CONSTANT N = 2^(ceil(LOG2(Inputs)));
CONSTANT E = 2*N-1;
SUBDESIGN S
(
    Pi[0..Inputs-1][0..Width-1]: INPUT ;
    So[0..Width-1] : OUTPUT;
)
VARIABLE Si[0..E-1][0..Width-1]: NODE;
BEGIN
Si[0..Inputs-1][]=Pi[0..Inputs-1][];
Si[Inputs..N-1][]=GND;
FOR i IN 0 TO E-3
    GENERATE
        IF i==0 GENERATE
            Si[N][]=Si[i][]+Si[i+1][];
        ELSE GENERATE
            IF i MOD 2==0 GENERATE
                Si[N+i DIV 2][]=Si[i][]+Si[i+1][];
            END GENERATE;
        END GENERATE;
        --i=i+1;
    END GENERATE;
So[]=Si[E-1][];
END;
```

A2.3. Descrierea în AHDL a componentei parametrice a funcției de activare.

PARAMETERS

```
(  
    Width = 1  
);
```

SUBDESIGN T

```
(  
    Si[Width-1..0]: INPUT ;  
    Ti[Width-1..0]: OUTPUT;  
)
```

BEGIN

```
    Ti[Width-1..0]=Si[Width-1..0];  
END;
```

A2.4 Descrierea în AHDL a modelului parametric a neuronului.

```
INCLUDE "P.inc";
INCLUDE "S.inc";
INCLUDE "T.inc";
PARAMETERS
(
    NeuronInputs = 1,
    NeuronWidth = 1
);
SUBDESIGN Neuron
(
    Xi[NeuronInputs-1..0][NeuronWidth-1..0]: INPUT ;
    Wi[NeuronInputs-1..0][NeuronWidth-1..0]: INPUT ;
    CLK                                     : INPUT ;
    Oi[NeuronWidth-1..0]                   : OUTPUT;
)
VARIABLE Reg[NeuronWidth-1..0]: DFF;
    PP    : P WITH (Width = NeuronWidth, Inputs = NeuronInputs);
    SS    : S WITH (Width = NeuronWidth, Inputs = NeuronInputs);
    TT    : T WITH (Width = NeuronWidth);
BEGIN
    PP.Xi[][] = Xi[][];
    PP.Wi[][] = Wi[][];
    SS.Pi[][] = PP.Pi[][];
    TT.Si[] = SS.Si[];
    Reg[] = TT.Ti[];
    Reg[].CLK=CLK;
    Oi[] = Reg[];
END;
```

A2.5 Descrierea în AHDL a modelului parametric a rețelei neuronale.

```
INCLUDE "Neuron.inc";
PARAMETERS
(
    NetNeurons = 1,
    NetWidth = 1
);
SUBDESIGN Network
(
    I[NetNeurons-1..0][NetWidth-1..0]          : INPUT ;
    W[2*NetNeurons*NetNeurons-1..0][NetWidth-1..0]: INPUT ;
    CLK                                          : INPUT ;
    O[NetNeurons-1..0][NetWidth-1..0]          : OUTPUT;
)
VARIABLE Neurons[NetNeurons-1..0] : Neuron WITH (NeuronWidth = NetWidth,
NeuronInputs = 2*NetNeurons*NetNeurons);
BEGIN
Neurons[].Xi[2*NetNeurons*NetNeurons-1..2*NetNeurons*NetNeurons-NetNeurons][] = I[][];
    Neurons[].Xi[2*NetNeurons*NetNeurons-NetNeurons-1..0][] = O[][];
    Neurons[].Wi[][] = W[][];
    Neurons[].CLK = CLK;
    O[][] = Neurons[].Oi[];
END;
```

A2.6 Descrierea în AHDL a interfeței parametrice de comunicare cu rețeaua neuronală.

```
INCLUDE "lpm_decode.inc";
INCLUDE "lpm_mux.inc";
INCLUDE "lpm_counter.inc";
INCLUDE "Network.inc";
INCLUDE "lpm_bustri.inc";
PARAMETERS
(
    NEURONS = 10,
    WIDTH = 64
);
SUBDESIGN Interface
(
    ADDRESS[CEIL(LOG2(2*NEURONS*NEURONS+2*NEURONS))-1..0] : INPUT;
    DATAINOUT[WIDTH-1..0] : BIDIR;
    CLK : INPUT;
    W/R : INPUT;
    READY : OUTPUT;
)
VARIABLE RAM[0..2*NEURONS*NEURONS + NEURONS - 1][WIDTH-1..0] : DFFE;
DEC:lpm_decode WITH (LPM_WIDTH =
CEIL(LOG2(2*NEURONS*NEURONS+2*NEURONS)), LPM_DECODES =
2*NEURONS*NEURONS + NEURONS);
MUX:lpm_mux WITH (LPM_WIDTH = WIDTH, LPM_SIZE =
2^CEIL(LOG2(2*NEURONS*NEURONS+2*NEURONS)), LPM_WIDTHS =
CEIL(LOG2(2*NEURONS*NEURONS+2*NEURONS)), LPM_PIPELINE = 0);
CNT:lpm_counter WITH (LPM_WIDTH = CEIL(LOG2(3*NEURONS)),
LPM_DIRECTION="UP", LPM_MODULUS = 3*NEURONS);
NET:Network WITH (NetWidth = WIDTH, NetNeurons = NEURONS);
BusTri:lpm_bustri WITH (LPM_WIDTH = WIDTH);
EQ:NODE;
BEGIN
```



```

RAM[0][0].CLK = CLK;
FOR i IN 0 TO 2*NEURONS*NEURONS + NEURONS - 1 GENERATE
    RAM[i][0].ena = DEC.eq[i] AND W/R;
END GENERATE;
RAM[0..2*NEURONS*NEURONS + NEURONS - 1][0] = BusTri.result[0];
DEC.data[0] = ADDRESS[0];
MUX.sel[0] = ADDRESS[0];
MUX.data[0..2*NEURONS*NEURONS + NEURONS - 1][0] =
RAM[0..2*NEURONS*NEURONS + NEURONS - 1][0];
MUX.data[2*NEURONS*NEURONS + NEURONS..2*NEURONS*NEURONS +
2*NEURONS - 1][0] = NET.O[0..NEURONS-1][0];
IF CNT.q[0]==3*NEURONS - 1 THEN
    EQ = GND;
ELSE
    EQ = VCC;
END IF;
CNT.clock = CLK;
CNT.clk_en = NOT (EQ);
CNT.aclr = W/R;
NET.CLK = CLK;
NET.W[0][0] = RAM[0..2*NEURONS*NEURONS - 1][0];
NET.I[0][0] = RAM[2*NEURONS*NEURONS..2*NEURONS*NEURONS+NEURONS -
1][0];
READY = EQ;
BusTri.data[0] = MUX.result[0];
DATAINOUT[0] = BusTri.tridata[0];
BusTri.enabledt = W/R;
BusTri.enabletr = NOT(W/R);
END;

```

Anexa 3. Listingul pogramului de exportare a matricei ponderilor sinaptice.

```
close all;
clear all;
clc;
format compact;
neurons = 0;
for i = 1:net.numLayers
    neurons = neurons + net.layers{i}.size;
end

n = max(net.inputs{1}.size + 1, neurons);
W = zeros(2*n,n);

ln = 0;
for i = 1:net.numLayers
    W(net.inputs{1}.size + 1,ln + 1:ln + `net.layers{i}.size) = net.b{i}.';
    ln = ln + net.layers{i}.size;
end

W(1:net.inputs{1}.size, 1:net.layers{1}.size) = net.IW{1}.';

ln = 0;
for i = 1:net.numLayers - 1
    W(n + ln + 1 :n + ln + net.layers{i}.size, ln + net.layers{i}.size + 1:ln + net.layers{i}.size
    + 1 + net.layers{i + 1}.size - 1) = net.LW{i + 1, i}.';
    ln = ln + net.layers{i}.size ;
end
```

Anexa 4. Act de implementare.



Aprob:

Director

“Microfir Tehnologii Industriale” S.R.L.

Lilian Chicu

” 20 01 2020 ”



Aprob:

Prorector

pentru cercetare și doctorat UTM

dr. hab. prof. univ.

Mircea Bemic

” 23 01 2020 ”



ACT

de implementare în procesul
de cercetare-dezvoltare a microfiredor
a elaborării tehnico-științifice

”Sistem informațional de măsurare a parametrilor microfiredor”

Prin prezentul confirmăm implementarea în procesul tehnologic de cercetare-dezvoltare a microfiredor cu izolație din sticlă a Sistemului informațional de măsurare a parametrilor microfiredor elaborat în colaborare în cadrul Proiectului instituțional *”Modele, metode și interfețe pentru conducerea și optimizarea sistemelor de fabricație inteligente”* (cifrul proiectului **15.817.02.28A**).

Sistemul informațional de măsurare a parametrilor microfiredor se utilizează pentru efectuarea caracterizărilor microfiredor magnetice cu izolație din sticlă în cadrul companiei **Microfir Tehnologii Industriale**, Chișinău, Republica Moldova.

Director-adjunct pentru cercetare,
Compania “Microfir Tehnologii Industriale” S.rl
Vladimir Larin

Colaborator științific
Compania “Microfir Tehnologii Industriale” S.rl
Victor Pavel

Coordonator proiect,
dr., conf. univ.
Sergiu Zaporozjan

Lector universitar UTM
cercetător științific
Viorel Cărbune

Declarația privind asumarea răspunderii

Subsemnatul, declar pe răspundere personală că materialele prezentate în teza de doctorat sunt rezultatul propriilor cercetări și realizări științifice. Conștientizez că, în caz contrar, urmează să suport consecințele în conformitate cu legislația în vigoare.


Cărbune, Viorel

Semnătura



Data

16.11.2020

CURRICULUM VITAE		
<p>Numele: Prenumele: Data și locul nașterii:</p>	<p>Cărbune Viorel 8 martie 1985, or. Glodeni, Republica Moldova</p>	
<p>STUDII: Superioare: Doctorat:</p>	<p>Universitatea Tehnică a Moldovei, 2003-2007, Facultatea Calculatoare Informatică și Microelectronică, specialitatea Calculatoare; 2007-2008 - studiile de masterat, specialitatea Calculatoare. Doctorand, studii la zi, Universitatea Tehnică a Moldovei, 2011-2014, specialitatea 232.01 – “Sisteme de conducere, calculatoare și rețele informaționale”</p>	
<p>Domenii de interes științific:</p>	<p>Sisteme Încorporate; Inteligența Artificială; Proiectarea cu Dispozitive FPGA.</p>	
<p>Activitatea profesională: 2007-2014 2014-2020</p>	<p>Inginer programator la catedra Calculatoare, Universitatea Tehnică a Moldovei. Lector la catedra Calculatoare, Universitatea Tehnică a Moldovei.</p>	
<p>Participări în proiecte științifice naționale și internaționale:</p>	<p>Participat în proiecte 4 proiecte de transfer tehnologic, instituționale și internaționale.</p>	
<p>Participări la foruri științifice:</p>	<p>Participări la 12 foruri științifice din Republica Moldova și străinătate.</p>	
<p>Publicații:</p>	<p>Au fost publicate 14 de lucrări științifice în domeniul sistemelor fuzzy, sistemelor neuronala, sistemelor de producție a microfidelor, arhitecturilor specializate.</p>	
<p>Premii, mențiuni, Distincții:</p>	<p>Diplome de excelență – 6; Medalii de aur – 2, de argint – 2.</p>	
<p>Cunoașterea limbilor:</p>	<p>Română, rusă, engleză.</p>	
<p>Date de contact:</p>	<p>str. Studenților 7/1, MD 2040, Chișinău, Republica Moldova tel.: +37368282207 email: viorel.carbune@calc.utm.md</p>	